

EACL 2023

**The 17th Conference of the European Chapter of the
Association for Computational Linguistics**

Findings of EACL 2023

May 2-6, 2023

The EACL organizers gratefully acknowledge the support from the following sponsors.

Diamond and Welcome Event



Diamond



Platinum and D&I Ally



Platinum



Silver



Bronze



©2023 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-959429-47-0

Message from the General Chair

Welcome to the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL). This is the flagship European conference dedicated to European and international researchers, covering a broad spectrum of research areas of Computational Linguistics and Natural Language Processing.

Organizing a scientific conference of the prestige and size of EACL is always a great honor associated with several challenges. Our team had to tackle unusual complexities: this conference was one of the first scheduled to be in person after the long period of online conferences forced by COVID pandemic. The bidding process for a location, which typically takes place several years before the actual start of the conference, is mainly driven by the aim of expanding and involving the science community of all European countries: EACL selected Kyiv, Ukraine, as the physical location. As you all know, in February 2022, an unpredictable and dramatic event happened, the war between Russian and Ukraine, which made the organization in Kyiv impossible.

Considering the importance of physical interaction among researchers, especially after the restrictions imposed by the COVID pandemic, we worked hard with the EACL and ACL boards to find an alternative location, able to delight our attendees. Our team achieved this seemingly impossible goal of organizing a conference in a new location a few months before its start: we selected Dubrovnik, Croatia, while preserving the original aim of strengthening the connection with the Ukrainian community. In this respect, the Ukraine local committee will feature a dedicated panel session, “Low-resource languages in NLP products”, and a workshop to highlight work on Ukrainian language technologies. Following the latest conference, EACL 2023 will be “hybrid,” serving both virtual and in-person participants. As our official local chairs are not from the physical location, we needed a local team from Croatia for helping with the logistics. As a result, the main unexpected novelty of EACL 2023 is to have two local organizing committees from two different European countries.

In the remainder of this preface, I would like to thank EACL contributors chronologically with respect to my work timeline for EACL: Roberto Basili and Shuly Wintner, the new and former Presidents of ACL, along with the EACL board – thanks for having trusted me to manage the organization of the conference in rather complicated times. I started to be confident that we would have done a good job after Isabelle Augenstein and Andreas Vlachos accepted the role of PC Chairs. They have performed amazing work, creating an outstanding program, and also helping me in recruiting our fantastic organization team. A special thank is due to Preslav Nakov (EACL officer) for his support: thanks to his action, the proactiveness of David Yarowsky, and the fairless effort of Jennifer Rachford (our new secretary of the ACL business office), we successfully implemented the apparently unrealistic idea of switching from the already planned online conference to a hybrid setting with a physical location in Dubrovnik. Regarding the online side of our hybrid conference, we partnered with Underline (Sol Rosenberg, Damira Mrcic and Luka Simic), who also gave us support for managing the entire conference. While finalizing the location, we started to activate the different sections of the conference, for which my acknowledgements are again in chronological order:

- Ukraine Local Committee, Viktoriia Kolomiets, Mariana Romanyshyn, Oleksii Molchanovskiy, Oles Dobosevych, was instrumental in preserving our initial goal of connecting the Ukraine research community, organizing a panel and a workshop.
- The website chairs, Pepa Atanasova and Julius Cheng, started immediately to design our website, even when almost no information was available.
- The workshop chairs, Zeerak Talat and Antonio Toral, selected our conferences and led the selection of workshops for the joint ACL call.

- The tutorial chairs, Sameer Pradhan and Fabio Massimo Zanzotto, together with the ACL chairs, took care of the tutorial selection for the ACL related conferences.
- The demonstration chairs, Danilo Croce and Luca Soldaini, created a parallel conference program to select exciting demos.
- The Publicity Chairs, Laura Biester, Leshem Choshen and Joel Tetrault, have been our interface with the science community through social media platforms.
- The Publication Chairs, Carolina Scarton and Ryan Cotterell, produced high-quality proceedings, thanks to their competence and experience.
- The diversity and inclusion chairs, Sara Tonelli, Elena Cabrio, Verena Rieser, Spandana Gella, took care of DI and performed an amazing job, also working on hundreds applications.
- The Local Organising Committee of Croatia, Marko Tadić, Krešimir Šojat, and Daša Farkaš, gave essential help for the logistics, Visa, and student volunteers.
- Student Research Workshop Chairs, Matthias Lindemann, Alban Petit, and Elisa Bassignana, along with their faculty advisors Valerio Basile and Natalie Schluter, helped in setting the bases for forming great NLP researchers of the future.
- Our entire program committee, Senior Area Chairs, Area Chairs, reviewers, and best paper committee, was essential for obtaining our high-quality scientific program.
- The ACL's sponsorship director Chris Callison-Burch took care of our sponsorships.
- The student volunteers, as usual, are essential for a successful conference execution.
- Priscilla Rasmussen, our former ACL business office secretary, continued to provide us with useful advice.

Finally, I would like to thank our sponsors for helping us to fund scholarships and DI initiatives.

Alessandro Moschitti
 Amazon Alexa AI, Los Angeles, USA
 EACL 2023 General Chair

ACL Statement on the Ukraine situation

March 11, 2022

The Association for Computational Linguistics (ACL) condemns in the strongest possible terms the actions of the Russian Federation government in invading the sovereign state of Ukraine and engaging in war against the Ukrainian people. We stand together with Ukrainian NLP colleagues, the Ukrainian people, Russian NLP colleagues and Russian people who condemn the actions of the Russian Federation government, and all those around the world who have been impacted by the invasion.

As a small token of our solidarity with the Ukrainian people, the ACL has decided to temporarily sever its ties with Russia-based organizations, while at the same time allowing Russian scientists to remain part of the ACL community. In practice, this means that the ACL will refrain from accepting any sponsorship or allowing any exhibits from Russian-headquartered entities at ACL-run events. Russian scholars are still welcome to participate in ACL events and publish at ACL venues.

The ACL is committed to peace and condemns any form of violence and harassment. We are also committed to peaceful co-operation, mutual understanding, and tolerance across borders. NLP scholars from both Ukraine and Russia are welcome to get in touch with the ACL with any concerns.

Tim Baldwin, on behalf of the ACL Executive

Message from the Program Chairs

Welcome to the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL). After the last edition in 2021 having been held fully online due to the COVID pandemic, EACL 2023 is being held in “hybrid” mode this year, serving both virtual and in-person participants in Dubrovnik, Croatia. While the original plan was to hold the conference in Kyiv (which was the plan originally for EACL 2021), the ongoing war made the organisation in Ukraine impossible. In order to ensure that the original aim of strengthening the connections with the Ukrainian community is still served, our program features a dedicated session and a workshop to highlight work on Ukrainian language technologies.

Submission and Acceptance

EACL 2023 accepted direct submissions, as well as submissions via ARR. For direct submissions, abstracts were needed to be registered one week prior to the submission date.

In total, EACL 2023 received 1550 submissions, the largest number to date, with the 2021 edition having received 1400 submissions. Out of those, 1045 were long and 505 were short paper submissions. 81 were ARR papers that were committed to EACL. 249 submissions were withdrawn throughout the reviewing process, including before the full paper submission deadline. 55 papers were desk rejected for various reasons (missing the limitations section, anonymity policy, multiple submission policy, plagiarism or formatting violations).

By the time we as the programme chairs made acceptance decisions, 1166 submissions were still active in the system. We kept the acceptance rate in line with previous *ACL conferences, resulting in 281 papers accepted to the main conference (24.1%), and 201 papers accepted to the Findings of EACL (17.2%), with the remaining 58.7% being rejected. One paper accepted to the main conference and four papers accepted to Findings were subsequently withdrawn. Out of the final set of accepted main conference papers, we invited 178 to be presented orally, and all 281 papers accepted to the main conference to be presented during in-person sessions, as well as a plenary virtual poster session. The EACL 2023 program also features six papers from the Transactions of the Association for Computational Linguistics (TACL) journal, and one from the Computational Linguistics (CL) journal.

Limitations Section

Following EMNLP 2022, we required that each submitted paper must include an explicitly named Limitations section, discussing the limitations of the work. This was to counterbalance the practice of over-hyping the take-away messages of papers, and to encourage more rigorous and honest scientific practice. This discussion did not count towards the page limit, and we asked reviewers to not use the mentioned limitations as reasons to reject the paper, unless there was a really good reason to.

Areas

To ensure a smooth process, the submissions to EACL 2023 were divided into 21 areas. The areas mostly followed these of previous EACL, and more broadly *ACL conferences, reflecting the typical divisions in the field. We also had a special area for papers for which both SACs had a conflict of interest. Those papers were reviewed by the reviewers and ACs in their original areas, but the paper recommendations were made by a dedicated SAC, who was a senior member of the NLP community. The most popular areas with over 100 submissions were “Generation and Summarization”, “Language Resources and Evaluation”, and “Machine Learning in NLP”.

Best Paper Awards

From the papers submitted to EACL 2023, we selected 25 papers accepted to the main conference as candidates for a Best Paper award, based on nominations by the reviewers. These papers were assessed by the Best Paper Award Committee, who also determined the types of paper awards, following the ACL Conference Awards Policy. The selected best papers and runner-ups will be announced in a dedicated plenary session for Best Paper Awards on 4 May 2023.

Programme Committee Structure and Reviewing

Similar to prior NLP conferences, we adopted the hierarchical program committee structure, where for each area we invited 1-2 Senior Area Chairs (SACs), who worked with a team of Area Chairs (ACs), and a larger team of reviewers. We relied on statistics from prior years to estimate how many SACs, ACs and reviewers would be needed and ended up with 43 SACs, 118 ACs and 1634 reviewers. For identifying ACs and reviewers, we used the reviewer lists from prior *ACL conferences, and also encouraged all EACL 2023 authors to serve as reviewers, using a mandatory form requesting further information on their ability to serve as ACs, reviewers or emergency reviewers, which authors had to fill in on Softconf when registering their abstracts. We passed this information on to SACs, who were responsible for recruiting ACs and reviewers.

Rather than making assignments using a matching algorithm, we asked ACs and reviewers to bid on registered abstracts within their areas, to achieve a better fit. We went with this solution as the number of papers per area was relatively small, and we wanted to avoid poor reviewing assignments as much as possible. We then made an initial paper assignment, in which we ensured that each paper would be reviewed by at least one reviewer who bid “yes” for the submission, and by no reviewers who bid “no” for the submission.

Afterwards, we asked the SACs to fine-tune the allocations, and ensure each paper had one AC and three reviewers assigned to it.

To ensure the review quality, we provided detailed guidelines about what reviewers should and shouldn’t do in a review, based on the EMNLP 2022 guidelines. We also asked reviewers to flag papers for potential ethical concerns.

For pre-reviewed ARR papers, we asked SACs to not rely mainly on the reviewer scores, but to make their recommendations based on the text of the reviews, meta-reviews and the papers themselves. For making acceptance decisions, we mostly followed SAC recommendations, though also taking into account the overall quality of papers submitted to the conference. Where recommendations seemed overly harsh or lenient given the reviewers’ scores, reviews, author responses, or discussions amongst reviewers, we engaged in a dialogue with the respective SACs to make the final decision about the papers in question.

Ethics Committee

We also formed an Ethics Committee (EC) dedicated to ethical issues. The ethics committee considered 21 papers that were flagged by the technical reviewing committee for ethical concerns. Out of these, 10 were conditionally accepted, meaning the ethics issues had to be addressed in the camera-ready version, to be verified by the EC prior to final acceptance, and the other 11 were accepted as is. The authors of all conditionally accepted papers submitted the camera-ready version and a short response that explained how they had made the changes requested by the EC. The EC double-checked these revised submissions and responses, and confirmed that the ethical concerns had been addressed. As a result, all conditionally accepted papers were accepted to the main conference or Findings.

ACL Rolling Review

ACL Rolling Review (ARR) is an initiative of the Association for Computational Linguistics, where the reviewing and acceptance of papers to publication venues are done in a two-step process: (1) centralized rolling review and (2) the ability to commit the reviewed papers to be considered for publication by a publication venue. For EACL 2023, we decided to follow EMNLP 2022’s example and run a process which is separate from ARR, but also allows for ARR submissions. Specifically, authors could either submit papers to EACL 2023 directly, or commit ARR reviewed papers by a certain date. We coordinated with the ARR team to extract the submission, review and meta-review from the OpenReview system, according to a submission link that the author provided when committing their ARR submission to EACL. The ARR commitment deadline was set one month after the direct submission deadline since the ARR submissions already have their reviews and meta-recommendation. These ARR papers were then ranked by the SACs together with the direct submissions in the track, and based on the reviews and meta-reviews from ARR. Overall, EACL had 81 papers committed from ARR, of these 24 were accepted to the main conference and 20 were accepted to Findings of EACL.

Presentation Mode

We made the decision on which papers would be invited for oral poster presentations based on several factors: the relative rank of the paper according to SAC recommendation, whether the paper had been recommended for a best paper award by at least one reviewer, and for TACL and CL papers, the authors’ preference of presentation mode.

Keynotes and Panel

Another highlight of our program are the plenary sessions, for which we scheduled three talks, as well a panel:

- a keynote talk by Joyce Chai (University of Michigan) on “Language Use in Embodied AI!”
- a keynote talk by Edward Greffentette (Cohere AI and University College London) on “Going beyond the benefits of scale by reasoning about data”
- a keynote talk by Kevin Munger (Penn State University) on Chatbots for Good and Evil”
- a panel on “low-resource languages in NLP products” led by Mariana Romanyshyn with Viktoria Kolomiets (Grammarly), Mariana Romanyshyn (Grammarly), Oleksii Molchanovskyi (Ukrainian Catholic University) and Oles Doboşevych (Ukrainian Catholic University)

Thank You

EACL 2023 is the result of a collaborative effort and a supportive community, and we want to acknowledge the efforts of so many people with whom we worked directly and made significant efforts in putting together the programme for EACL 2023!

- Our General Chair, Alessandro Moschitti, who led the whole organising team, and helped with many of the decision processes;
- Our 43 Senior Area Chairs, who were instrumental in every aspect of the review process, from recruiting Area Chairs, correcting reviewer assignments, to making paper acceptances;
- Our 118 Area Chairs, who had the role of interacting with the reviewers, leading paper review discussions, and writing meta-reviews;

- The 1634 reviewers, who provided valuable feedback to the authors; The emergency reviewers, who provided their support at the last minute to ensure a timely reviewing process;
- Our Best Paper Selection Committee, who selected the best papers and the outstanding papers: Jonathan Kummerfeld (chair), Joakim Nivre, Bonnie Webber, Tamar Solorio and Hanna Hajishirzi;
- Our Ethics Committee, chaired by Zeerak Talat, for their hard work to ensure that all the accepted papers addressed the ethical issues appropriately, under a very tight schedule;
- Our amazing Publication Chairs, Carolina Scarton and Ryan Cotterell for compiling the proceedings in good time for the conference;
- Our Publicity Chairs, Laura Biester, Leshem Choshen and Joel Tetrault, for their work on managing the communications on social media platforms;
- Our website chairs, Pepa Atanasova and Julius Cheng for putting together the website for the conference and keeping it up to date;
- Damira Mrsic from Underline, for her support in developing the virtual conference platform;
- Jennifer Rachford, who has worked tirelessly online and on-site to ensure that EACL 2023 is a success.

We're looking forward to a great EACL 2023!

Isabelle Augenstein (University of Copenhagen, Denmark)
Andreas Vlachos (University of Cambridge, UK)
EACL 2023 Programme Committee Co-Chairs

Organizing Committee

General Chair

Alessandro Moschitti, Amazon Alexa

Program Chairs

Isabelle Augenstein, University of Copenhagen
Andreas Vlachos, University of Cambridge

Publications Chairs

Ryan Cotterell, ETH Zürich
Carolina Scarton, University of Sheffield

Workshop Chairs

Zeeraq Talat, Simon Fraser University
Antonio Toral, University of Groningen

Tutorials Chairs

Sameer Pradhan, University of Pennsylvania
Fabio Massimo Zanzotto, University of Rome, “Tor Vergata”

Demonstrations Chairs

Danilo Croce, University of Rome, “Tor Vergata”
Luca Soldaini, Allen Institute for AI

Publicity Chairs

Joel Tetreault, Dataminr
Leshem Choshen, IBM AI research; Hebrew University of Jerusalem
Laura Biester, University of Michigan

Website Chairs

Pepa Atanasova, University of Copenhagen
Julius Cheng, University of Cambridge

Sponsorship Director

Chris Callison-Burch, University of Pennsylvania

Diversity and Inclusion Chairs

Elena Cabrio, Université Côte d’Azur, Inria, CNRS, I3S

Sara Tonelli, Fondazione Bruno Kessler
Verena Rieser, Heriot-Watt University
Spandana Gella, Amazon Alexa

Student Research Workshop Chairs

Matthias Lindemann, University of Edinburgh
Alban Petit, Université Paris-Saclay
Elisa Bassignana, IT University of Copenhagen

Student Research Workshop Faculty Advisors

Valerio Basile, University of Turin
Natalie Schluter, IT University of Copenhagen; Apple

Local Organising Committee

Marko Tadić, University of Zagreb
Krešimir Šojat, University of Zagreb
Daša Farkaš, University of Zagreb

Ukraine Local Committee

Viktoria Kolomiets, Grammarly
Mariana Romanyshyn, Grammarly
Oleksii Molchanovskyi, Ukrainian Catholic University
Oles Dobosevych, Ukrainian Catholic University

Program Committee

Anaphora, Discourse and Pragmatics

Bonnie Webber, University of Edinburgh
Michael Strube, Heidelberg Institute for Theoretical Studies

Computational Social Science and Social Media

Maria Liakata, Queen Mary University of London
Kalina Bontcheva, University of Sheffield

Conflicts of Interests

Joakim Nivre, Research Institutes of Sweden

Dialogue and Interactive Systems

Diarmuid Ó Séaghdha, Apple
Matthew Purver, Queen Mary University of London

Document analysis, Text Categorization and Topic Models

Nikolaos Aletras, University of Sheffield
Ekaterina Shutova, University of Amsterdam

Ethical and Sustainable NLP

Nafise Sadat Moosavi, Department of Computer Science, The University of Sheffield
Yonatan Belinkov, Technion

Ethics Review

Zeerak Talat, Simon Fraser University

Generation and Summarization

Ondrej Dusek, Charles University
Chenghua Lin, Department of Computer Science, University of Sheffield

Information Extraction

Roberto Navigli, Sapienza University of Rome
Mrinmaya Sachan, ETH Zurich

Information Retrieval and Search

Bruno Martins, IST and INESC-ID
Fabrizio Silvestri, Sapienza, University of Rome

Interpretability and Model Analysis

Dong Nguyen, Utrecht University
Roi Reichart, Technion - Israel Institute of Technology

Language Grounding and Multi-Modality

Grzegorz Chrupała, Tilburg University
Desmond Elliott, University of Copenhagen

Language Resources and Evaluation

Roman Klinger, University of Stuttgart
Omri Abend, The Hebrew University of Jerusalem

Linguistic Theories, Cognitive Modeling and Psycholinguistics

Barry Devereux, Queen's University, Belfast
Natalie Schluter, IT University of Copenhagen

Machine Learning for NLP

James Henderson, Idiap Research Institute
Vlad Niculae, University of Amsterdam

Machine Translation

Wilker Aziz, University of Amsterdam
Rico Sennrich, University of Zurich

Multidisciplinary and other NLP Applications

Annie Priyadarshini Louis, Google Research UK
Yulan He, King's College London

Multilinguality

Ivan Vulić, University of Cambridge
Alexander Fraser, Ludwig-Maximilians-Universität München

Phonology, Morphology, and Word Segmentation

Thierry Poibeau, LATTICE (CNRS and ENS/PSL)
François Yvon, ISIR CNRS and Sorbonne Université

Question Answering

Jonathan Berant, Tel Aviv University and AI2
Pontus Stenetorp, University College London

Semantics: lexical

Chris Biemann, Universität Hamburg
Mark Stevenson, University of Sheffield

Semantics: sentence level and other areas

Aliaksei Severyn, Google
Douwe Kiela, Hugging Face

Sentiment Analysis and Argument Mining

Veronique Hoste, LT3, Ghent University
Ivan Habernal, Technical University of Darmstadt

Tagging, Chunking, Syntax and Parsing

Marco Kuhlmann, Linköping University
Shay B. Cohen, University of Edinburgh

Area Chairs

Khalid Al Khatib, Malihe Alikhani, Mikel Artetxe, Akari Asai, Duygu Ataman, Niranjana Balasubramanian, Jeremy Barnes, Max Bartolo, Valerio Basile, Laurent Besacier, Iacer Calixto, Kris Cao, Tanmoy Chakraborty, Bharathi Raja Chakravarthi, Guanyi Chen, Wenhua Chen, Eleanor Chodroff, Caio Corro, Çağrı Çöltekin, Orphee De Clercq, Miryam De Lhoneux, Tejaswini Deoskar, Rotem Dror, Yanai Elazar, Arash Eshghi, Amir Feder, Yang Feng, Radu Florian, Anette Frank, Markus Freitag, André Freitas, Annemarie Friedrich, Jie Fu, Wei Gao, Mor Geva, Dan Goldwasser, Yulia Grishina, Lin Gui, Sara Hooker, Shujian Huang, Patrick Huber, Dieuwke Hupkes, Peter Jansen, Kristen Johnson, David Jurgens, Simon Keizer, Casey Kennington, Daniel Khashabi, Tushar Khot, Germán Kruszewski, Lun-wei Ku, Matthieu Labeau, Gerasimos Lampouras, Mirella Lapata, Gabriella Lapesa, Anne Lauscher, Piji Li, Bin Liang, Bang Liu, Craig Macdonald, Pranava Madhyastha, Andrea Madotto, Saad Mahamood, Jonathan Mallinson, Zita Marinho, Eugenio Martínez Cámara, Florian Metze, Sabrina Mielke, Pushkar Mishra, Saif Mohammad, Preslav Nakov, Shashi Narayan, Debora Nozza, Alexander Panchenko, Alexandros Papangelis, Nikolaos Pappas, Panupong Pasupat, Gabriele Pergola, Jonas Pfeiffer, Yuval Pinter, Edoardo Maria Ponti, Christopher Potts, Daniel Preotiuc-pietro, Emily Prud'hommeaux, Simon Razniewski, Siva Reddy, Sara Rosenthal, Alla Rozovskaya, Keisuke Sakaguchi, Tanja Samardzic, Minjoon Seo, Ehsan Shareghi, Ravi Shekhar, Carina Silberer, Miikka Silfverberg, Felix Stahlberg, Svetlana Stoyanchev, Saku Sugawara, Hiroya Takamura, Niket Tandon, Sara Tonelli, Nicola Tonellotto, Lonneke Van Der Plas, David Vandyke, David Vilares, Elena Voita, Svitlana Volkova, Shuai Wang, Derry Tanti Wijaya, Adina Williams, Wei Wu, Fei Xia, Deyi Xiong, Ikuya Yamada, Marcos Zampieri, Sina Zarrieß, Chrysoula Zerva, Arkaitz Zubiaga

Reviewers

Sadaf Abdul Rauf, Muhammad Abdul-mageed, Ibrahim Abu Farha, Lasha Abzianidze, Angus Addlesee, David Adelani, Stergos Afantenos, Severine Affeldt, Rodrigo Aggeri, Piush Aggarwal, Željko Agić, Ameeta Agrawal, Roei Aharoni, Wasi Uddin Ahmad, Sina Ahmadi, Natalie Ahn, Xi Ai, Laura Aina, Akiko Aizawa, Md. Shad Akhtar, Hend Al-khalifa, Rami Al-rfou, Ahmed Alajrami, David Alfter, Bashar Alhafni, Hamed Alhoori, Hafsa Ali, Afra Alishahi, Miguel A. Alon-

so, Sultan Alrowili, Bharat Ram Ambati, Silvio Amir, Samuel Amouyal, Reinald Kim Amplayo, Jisun An, Vishal Anand, Raviteja Anantha, Antonios Anastasopoulos, Tim Anderson, Melanie Andresen, Anelia Angelova, Alan Ansell, Francesco Antici, Diego Antognini, Maria Antoniak, Dimosthenis Antypas, Reut Apel, Emilia Apostolova, Jun Araki, Oscar Araque, Arturo Argueta, Akhil Arora, Ekaterina Artemova, Elliott Ash, Md.sadek Hossain Asif, Arian Askari, Zhenisbek Assylbekov, Aitziber Atutxa Salazar, Eleftherios Avramidis, Cem Rifki Aydin, Mahmoud Azab

Bharathi B, Jinheon Baek, Selene Baez Santamaria, Parsa Bagherzadeh, Vikas Bahirwani, Fan Bai, Jinyeong Bak, Timothy Baldwin, Miguel Ballesteros, Forrest Sheng Bao, Edoardo Barba, Francesco Barbieri, Ander Barrena, Pierpaolo Basile, Roberto Basili, Ali Basirat, Riza Batistana-
navarro, Timo Baumann, Rachel Bawden, Christos Baziotis, Ian Beaver, Nadia Bebeschina, Frederic Bechet, Tilman Beck, Beata Beigman Klebanov, Tadesse Destaw Belay, Meriem Beloucif, Farah Benamara, Luca Benedetto, Joshua Bensemman, Gábor Berend, Thales Bertaglia, Michele Bevilacqua, Rasika Bhalerao, Rohan Bhambhoria, Rishabh Bhardwaj, Sumit Bhatia, Arnab Bhat-
tacharya, Rajarshi Bhowmik, Zhen Bi, Iman Munire Bilal, Alexandra Birch, Debmalya Biswas, Eduardo Blanco, Nate Blaylock, Su Lin Blodgett, Jelke Bloem, William Boag, Ben Bogin, Francis Bond, Georgeta Bordea, Logan Born, Emanuela Boros, Elizabeth Boschee, Cristina Bosco, Zied Bouraoui, Tom Bourgeade, Laurestine Bradford, Stephanie Brandl, Ana Brassard, Jonathan Brophy, Caroline Brun, Christian Buck, Sven Buechel, Paul Buitelaar, Razvan Bunescu, Laurie Burchell, Miriam Butt, Jan Buys, Lisa Bylinina, Bill Byrne

Laura Cabello Piqueras, Elena Cabrio, Samuel Cahyawijaya, Agostina Calabrese, Nitay Calderon, Eduardo Calò, Jose Camacho-collados, Ricardo Campos, Marie Candito, Shuyang Cao, Ziqiang Cao, Fabio Carrella, Xavier Carreras, Jorge Carrillo-de-albornoz, Lucien Carroll, Fabio Casati, Tommaso Caselli, Pierluigi Cassotti, Francesco Cazzaro, Amanda Cercas Curry, Dumitru-
clementin Cercel, Christophe Cerisara, Alessandra Cervone, Rahma Chaabouni, Haixia Chai, Tuhin Chakrabarty, Yllias Chali, Ilias Chalkidis, Hou Pong Chan, Zhangming Chan, Anshuma Chandak, Senthil Chandramohan, Buru Chang, Ernie Chang, Yung-chun Chang, Guan-lin Chao, Emile Chapuis, Shubham Chatterjee, Rochana Chaturvedi, Kushal Chawla, Ciprian Chelba, Canyu Chen, Chacha Chen, Chung-chi Chen, Derek Chen, Fuxiang Chen, Hsin-hsi Chen, Jie Chen, Lei Chen, Lei Chen, Meng Chen, Mingda Chen, Qian Chen, Qianglong Chen, Qibin Chen, Qingcai Chen, Sanxing Chen, Shizhe Chen, Tongfei Chen, Xiaoli Chen, Xiuying Chen, Yan-ying Chen, Yi-pei Chen, Yunmo Chen, Zhiyu Chen, Fei Cheng, Shanbo Cheng, Emmanuele Chersoni, Ethan A. Chi, Jenny Chim, Hyundong Cho, Key-sun Choi, Alexandra Chronopoulou, George Chryso-
stomou, Alessandra Teresa Cignarella, Philipp Cimiano, Elizabeth Clark, Chloé Clavel, Simon Clematide, Ann Clifton, Miruna Clinciu, Oana Cocarascu, Davide Colla, Andrei Coman, Simone Conia, John Conroy, Paul Cook, Gonçalo Correia, Israel Cuevas, Peng Cui, Shaobo Cui, Tonya Custis, Arthur Câmara

Thenmozhi D., Jeff Da, Giovanni Da San Martino, Raj Dabre, Gautier Dagan, Deborah Dahl, Wenliang Dai, Xiang Dai, Rumen Dangovski, Falavigna Daniele, Verna Dankers, Aswarth Abhi-
lash Dara, Franck Dary, Mithun Das Gupta, Saurabh Dash, Brian Davis, Heidar Davoudi, Michiel De Jong, Loic De Langhe, Budhaditya Deb, Alok Debnath, Thierry Declerck, Mathieu Dehouck, Luciano Del Corro, Sebastien Delecraz, Vera Demberg, David Demeter, Steve Deneefe, Yuntian Deng, Pascal Denis, Nina Dethlefs, Daniel Deutsch, Murthy Devarakonda, Hannah Devinney, Prajit Dhar, Shehzaad Dhuliawala, Luigi Di Caro, Mona Diab, Shizhe Diao, Gaël Dias, Caiwen Ding, Chenchen Ding, Liang Ding, Nemanja Djuric, Giovanna Maria Dora Dore, Bonaventure F. P. Dossou, Jad Doughman, Doug Downey, Gabriel Doyle, Mauro Dragoni, Rotem Dror, Jinhua Du, Yupei Du, Xiangyu Duan, Pablo Duboue, Philipp Dufter, Kevin Duh, Ewan Dunbar, Jonathan Dunn, Gerard Dupont, Nadir Durrani, Ritam Dutt

Oliver Eberle, Sauleh Eetemadi, Steffen Eger, Annerose Eichel, Bryan Eikema, Julian Eisen-schlos, Heba Elfardy, Micha Elsner, Saman Enayati, Aykut Erdem, Akiko Eriguchi, Katrin Erk, Ramy Eskander

Alex Fabbri, Marzieh Fadaee, Fahim Faisal, Neele Falk, Federico Fancellu, Qixiang Fang, Hos-sein Fani, Stefano Faralli, Oladimeji Farri, Nawshad Farruque, Manaal Faruqui, Mehwish Fatima, Adam Faulkner, Pedro Faustini, Marc Feger, Nils Feldhus, Anna Feldman, Ghazi Felhi, Mariano Felice, Weixi Feng, Yue Feng, Manos Fergadiotis, Patrick Fernandes, Daniel Fernández-gonzález, Elisabetta Fersini, George Filandrianos, Elena Filatova, Mark Fishel, Lucie Flek, Michael Flor, Negar Foroutan Eghlidi, Jennifer Foster, Stella Frank, Jesse Freitas, Simona Frenda, Annemarie Friedrich, Lisheng Fu, Fumiyo Fukumoto, Kotaro Funakoshi

David Gaddy, Andrea Galassi, Leilei Gan, Yujian Gan, William Gantt, Junbin Gao, Qiaozhi Gao, Shen Gao, Muskan Garg, Guillermo Garrido, Susan Gauch, Gregor Geigle, Zorik Gekhman, Al-borz Geramifard, Felix Gervits, Mozhdeh Gheini, Reshmi Ghosh, Sucheta Ghosh, Voula Giouli, Dimitris Gkoumas, Serge Gladkoff, Catalina Goanta, Jonas Golde, Seraphina Goldfarb-tarrant, Sujatha Das Gollapalli, Jose Manuel Gomez-perez, Jeff Good, Philip John Gorinski, Koustava Go-swami, Isao Goto, Christan Grant, Thomas Green, Derek Greene, Milan Gritta, Paul Groth, Julian Grove, Adam Grycner, Jiasheng Gu, Jiuxiang Gu, Xiaodong Gu, Yi Guan, Marco Guerini, Nuno M. Guerreiro, Xiaoyu Guo, Yanzhu Guo, Zhihui Guo, Abhinav Gupta, Ankit Gupta, Ankita Gup-ta, Ashim Gupta, Pranjal Gupta, Izzeddin Gur, Suchin Gururangan, Ximena Gutierrez-vasques, Jeremy Gwinnup, Tunga Güngör

Le An Ha, Katharina Haemmerl, Gholamreza Haffari, Joonghyuk Hahn, Michael Hahn, Udo Ha-hn, Eva Hajicova, Dilek Hakkani-tur, Kishaloy Halder, Karina Halevy, Jiuzhou Han, Lifeng Han, Ting Han, Xudong Han, Yo-sub Han, Viktor Hangya, Sanda Harabagiu, Mareike Hartmann, Sadid A. Hasan, Sabit Hassan, Nabil Hathout, Amartya Hatua, Annette Hautli-janisiz, Adi Haviv, Yoshi-hiko Hayashi, Shirley Anugrah Hayati, T. J. Hazen, Rishi Hazra, Han He, Wanwei He, Wei He, Xiaoting He, Xuanli He, Xuehai He, Yun He, Behnam Hedayatnia, Kevin Heffernan, Benjamin Heinzerling, Jindřich Helcl, William Held, Leonhard Hennig, Christian Herold, Jonathan Herzig, Gerhard Heyer, Derrick Higgins, Anthony Hills, Tatsuya Hiraoka, Vinh Thinh Ho, Cuong Hoang, Eben Holderness, Takeshi Homma, Ales Horak, Andrea Horbach, Sho Hoshino, Md Azam Hos-sain, Feng Hou, Yifan Hou, Yufang Hou, Shu-kai Hsieh, I-hung Hsu, Han Hu, Po Hu, Xinyu Hua, Chieh-yang Huang, Fei Huang, Hen-hsen Huang, Jie Huang, Junbo Huang, Kuan-hao Huang, Quz-he Huang, Zhiqi Huang, Vojtěch Hudeček, Pere-Lluís Huguet Cabot, Kai Hui, Chia-chien Hung, Julie Hunter

Nikolai Ilinykh, Dmitry Ilvovsky, Michimasa Inaba, Diana Inkpen, Koji Inoue, Hayate Iso, Ta-kumi Ito, Maor Ivgi, Kenichi Iwatsuki, Vivek Iyer, Peter Izsak

Cassandra L. Jacobs, Sarthak Jain, Masoud Jalili Sabet, Sepehr Janghorbani, Adam Jatowt, Ini-go Jauregi Unanue, Ganesh Jawahar, Harsh Jhamtani, Shaoxiong Ji, Yangfeng Ji, Chengyue Jiang, Junfeng Jiang, Longquan Jiang, Ming Jiang, Yuchen Eleanor Jiang, Ziyang Jiang, Baoyu Jing, Unso Jo, Richard Johansson, Aditya Joshi, Rishabh Joshi, Taehee Jung

Besim Kabashi, Sylvain Kahane, Mihir Kale, Laura Kallmeyer, Ehsan Kamaloo, Hidetaka Ka-migaito, Jaap Kamps, Lis Kanashiro Pereira, Hiroshi Kanayama, Yoshinobu Kano, Diptesh Ka-nojia, Sudipta Kar, Georgi Karadzhev, Elena Karagjosova, Mladen Karan, Sarvnaz Karimi, Börje Karlsson, Sanjeev Kumar Karn, Constantinos Karouzos, Pradeep Karturi, Zdeněk Kasner, Yoshi-hide Kato, Uri Katz, Yoav Katz, Divyansh Kaushik, Pride Kavumba, Daisuke Kawahara, Gary Kazantsev, Ashkan Kazemi, Yova Kementchedjheva, Muhammad Khalifa, Abdul Khan, Sapan

Khosla, Halil Kilicoglu, Gyuwan Kim, Hyunwoo Kim, Jonggu Kim, Joo-kyung Kim, Mi-young Kim, Seungone Kim, Sungdong Kim, Young Jin Kim, Youngbin Kim, David King, Tracy Holloway King, Svetlana Kiritchenko, Jan-christoph Klie, Julien Kloetzer, René Knaebel, Sang-ki Ko, Thomas Kober, Elena Kochkina, Konstantinos Kogkalidis, Mare Koit, Thomas Kollar, Alexander Koller, Mamoru Komachi, Rik Koncel-kedziorski, Grzegorz Kondrak, Sai Koneru, Deguang Kong, Miloslav Konopík, Yannis Korkontzelos, Katerina Korre, Fajri Koto, Alexander Kotov, Mahnaz Koupae, Venelin Kovatchev, Pavel Kral, Lea Krause, Kalpesh Krishna, Mateusz Krubiński, Canasai Kruengkrai, Jaap Kruijt, Ruben Kruiper, Sicong Kuang, Mayank Kulkarni, Deepak Kumar, Sachin Kumar, Shankar Kumar, Olli Kuparinen, Robin Kurtz, Andrey Kutuzov, Haewoon Kwak

Gorka Labaka, Sofie Labat, Faisal Ladhak, Cheng-i Lai, Tuan Lai, Wen Lai, Vasileios Lampos, Gerasimos Lampouras, Lukas Lange, Ekaterina Lapshinova-koltunski, Stefan Larson, Mark Last, Alexandra Lavrentovich, Hoang-quynh Le, Hung Le, Phong Le, Joseph Le Roux, Kevin Leach, Dong-ho Lee, Grandee Lee, Ji-ung Lee, John Lee, Lung-hao Lee, Nayeon Lee, Roy Ka-wei Lee, Els Lefever, Wenqiang Lei, Jochen Leidner, Heather Lent, Ran Levy, Bei Li, Bryan Li, Changmao Li, Cheng Li, Dingcheng Li, Dongfang Li, Jiacheng Li, Jialu Li, Jiazhao Li, Jing Li, Jiyi Li, Juan Li, Lei Li, Liunian Harold Li, Maoxi Li, Miao Li, Peifeng Li, Sheng Li, Shiyang Li, Shuyang Li, Siheng Li, Wei Li, Wei Li, Weikang Li, Wenyan Li, Xiangju Li, Xiaodi Li, Xue Li, Yanran Li, Yanzeng Li, Yaoyiran Li, Yizhi Li, Yongbin Li, Yue Li, Yuncong Li, Zhuang Li, Zichao Li, Chao-chun Liang, Xinnian Liang, Yueqing Liang, Baohao Liao, Jindřich Libovický, Constantine Lignos, Gilbert Lim, Kwan Hui Lim, Tomasz Limisiewicz, Lucy Lin, Weizhe Lin, Zhenxi Lin, Nedim Lipka, Pierre Lison, Shir Lissak, Danni Liu, Fangyu Liu, Fenglin Liu, Hui Liu, Jiangming Liu, Kang Liu, Lei Liu, Nayu Liu, Nelson F. Liu, Tianyu Liu, Tianyu Liu, Ting Liu, Yang Janet Liu, Yiyi Liu, Yonghui Liu, Yongkang Liu, Yue Liu, Zihan Liu, Zitao Liu, Zoey Liu, Nikola Ljubbešić, Sharid Loaiciga, Colin Lockard, Pintu Lohar, Yunfei Long, Oier Lopez De Lacalle, Jaime Lorenzo-trueba, Daniel Loureiro, Junru Lu, Keming Lu, Xing Han Lu, Yanbin Lu, Yao Lu, Yujie Lu, Nurul Lubis, Jiaming Luo, Man Luo, Haoran Lv, Shangwen Lv, Teresa Lynn, Alex Lutu

Meryem M'hamdi, Jie Ma, Jing Ma, Long-long Ma, Mingyu Derek Ma, Xiaofei Ma, Andrew Mackey, Aman Madaan, Avinash Madasu, Mounica Maddela, Manuel Mager, Bernardo Magnini, Adyasha Maharana, Quan Mai, Frederic Mailhot, Jean Maillard, Peter Makarov, Aaron Maladry, Ankur Mali, Anton Malko, Jonathan Mallinson, Eric Malmi, Valentin Malykh, Ramesh Manuvinakurike, Vladislav Maraev, Ana Marasovic, David Mareček, Katerina Margatina, Katja Markert, Edison Marrese-taylor, Federico Martelli, Louis Martin, Héctor Martínez Alonso, Claudia Marzi, Sarah Masud, Sandeep Mathias, Prashant Mathur, Diana Maynard, Sahisnu Mazumder, Alessandro Mazzei, R. Thomas Mccoy, John P. Mccrae, Bridget Mcinnes, Nick Mckenna, Nikhil Mehta, Fanchao Meng, Yan Meng, Zhao Meng, Orfeas Menis Mastromichalakis, Elena Merdjanovska, Eleni Metheniti, Ivan Vladimir Meza Ruiz, Paul Michel, Timothee Mickus, Stuart Middleton, Aristides Milios, Tristan Miller, David Mimno, Erxue Min, Seyedabolghasem Mirroshandel, Paramita Mirza, Abhijit Mishra, Kanishka Misra, Yusuke Miyao, Ashutosh Modi, Alireza Mohammadshahi, Hosein Mohebbi, Afroz Mohiuddin, Diego Molla, Manuel Montes, Mehrad Moradshahi, Roser Morante, Jose G. Moreno, Alejandro Moreo, Marius Mosbach, Pablo Mosteiro, Lili Mou, Diego Moussallem, Maximilian Mozes, Emir Munoz, Dragos Munteanu, Rudra Murthy, Alberto Muñoz-ortiz, Mathias Müller

Dawn Nafus, Masaaki Nagata, Saeed Najafi, Tetsuji Nakagawa, Yuta Nakashima, Diane Napolitano, Jason Naradowsky, Vivi Nastase, Anmol Nayak, Ambreen Nazir, Ani Nenkova, Mariana Neves, Jun-ping Ng, Raymond Ng, Vincent Ng, Axel-cyrille Ngonga Ngomo, Dat Quoc Nguyen, Kiet Nguyen, Nhung Nguyen, Quoc-an Nguyen, Trung Hieu Nguyen, Vincent Nguyen, Xuanfan Ni, Garrett Nicolai, Massimo Nicosia, Feng Nie, Yixin Nie, Jan Niehues, Mitja Nikolaus, Giannis

Nikolentzos, Takashi Ninomiya, Kosuke Nishida, Sergiu Nisioi, Gibson Nkhata, Tadashi Nomoto, Aurélie Névél

Alexander O’connor, Tim Oates, Kemal Oflazer, Shu Okabe, Naoaki Okazaki, Tsuyoshi Okita, Oleg Okun, Eda Okur, Antoni Oliver, Mattia Oppen, Abigail Oppong, Brian Ore, Hadas Orgad, Maite Oronoz, Petya Osenova, Jessica Ouyang

Teresa Paccosi, Ankur Padia, Aishwarya Padmakumar, Shramay Palta, Tuğba Pamay Arslan, Mugdha Pandya, Wei Pang, Pinelopi Papalampidi, Nikos Papasasantopoulos, Sara Papi, Emerson Paraiso, Ashwin Paranjape, Letitia Parcalabescu, Thiago Pardo, Antonio Pareja-lora, Chanjun Park, Jong Park, Sungkyu Park, Alicia Parrish, Tommaso Pasini, Clemente Pasti, Braja Gopal Patra, Viviana Patti, Debjit Paul, Indraneil Paul, Sachin Pawar, Sarah Payne, Pavel Pecina, Jiaxin Pei, Weiping Pei, Stephan Peitz, Baolin Peng, Bo Peng, Hao Peng, Qiyao Peng, Wei Peng, Juan Antonio Perez-ortiz, Charith Peris, Ben Peters, Matthew Peters, Eva Pettersson, Thang Pham, Scott Piao, Maciej Piasecki, Massimo Piccardi, Matúš Pikuliak, Nisha Pillai, Telmo Pires, Flammie Pirinen, Benjamin Piwowarski, Flor Miriam Plaza-del-arco, Brian Plüss, Massimo Poesio, Simone Paolo Ponzetto, Octavian Popescu, Amir Pouran Ben Veysel, Karan Praharaj, Piotr Przybyła, Stephen Pulman, Juan Manuel Pérez

Ehsan Qasemi, Hongjin Qian, Kun Qian, Kechen Qin, Jieli Qiu, Ariadna Quattoni

Ella Rabinovich, Muhammad Rahman, Sunny Rai, Vyas Raina, Sara Rajae, Ori Ram, Taraka Rama, Giulia Rambelli, Abhinav Ramesh Kashyap, Rita Ramos, Alan Ramponi, Leonardo Ranaldi, Tharindu Ranasinghe, Surangika Ranathunga, Priya Rani, Ahmad Rashid, Pushpendre Rastogi, David Rau, Vikas Raunak, Eran Raveh, Shauli Ravfogel, Soumya Ray, Evgeniia Razumovskaia, Hanumant Redkar, Georg Rehm, Ricardo Rei, Machel Reid, Navid Rekabsaz, Ricardo Ribeiro, Giuseppe Riccardi, German Rigau, Matīss Rikters, Tharathorn Rimchala, Laura Rimell, Fabio Rinaldi, Ruty Rinott, Anthony Rios, Lina M. Rojas Barahona, Subendhu Rongali, Michael Rosner, Michael Roth, Guy Rotman, Bryan Routledge, Marco Rovera, Soumyadeep Roy, Yu-ping Ruan, Koustav Rudra, Federico Ruggeri, Irene Russo, Phillip Rust, Max Ryabinin, Maria Ryskina, Egil Rønningstad, Susanna Rücker

Malliga S, Kogilavani S V, Kenji Sagae, Keisuke Sakaguchi, Ander Salaberria, Shailaja Keyur Sampat, David Samuel, Ramon Sanabria, George Sanchez, Hugo Sanjurjo-gonzález, Sonal San-nigrahi, Rodrigo Santos, Naomi Saphra, Ruhi Sarikaya, Anoop Sarkar, Felix Sasaki, Ryohei Sasano, Nishanth Sastry, Danielle Saunders, Thusius Savarimuthu, Beatrice Savoldi, Apoorv Saxena, Federico Scafoglieri, Andreas Scherbakov, Dominik Schlechtweg, Jonathan Schler, Michael Sejr Schlichtkrull, Robin Schmidt, Nathan Schneider, Stephanie Schoch, Annika Marie Schoene, Merel Scholman, Sabine Schulte Im Walde, Philip Schulz, Stefan Schweter, Anastasiia Sedova, Elad Segal, Cory Shain, Guokan Shang, Yutong Shao, Ori Shapira, Matthew Shardlow, Shuaijie She, Artem Shelmanov, Aili Shen, Lingfeng Shen, Xiaoyu Shen, Yuming Shen, Michael Sheng, Qiang Sheng, Tom Sherborne, Freda Shi, Zhan Shi, Zhengxiang Shi, Tomohide Shibata, Yutaro Shigeto, Takahiro Shinozaki, Kumar Shridhar, Akshat Shrivastava, Kai Shu, Raphael Shu, Anna Shvets, Anthony Sicilia, Alejandro Sierra-múnera, João Ricardo Silva, Danilo Silva De Carvalho, Patrick Simianer, Edwin Simpson, Mayank Singh, Pranaydeep Singh, Koustuv Sinha, Sunayana Sitaram, Milena Slavcheva, Kevin Small, Marco Antonio Sobrevilla Cabezudo, Swapna Somasundaran, Kai Song, Linfeng Song, Wei Song, Yan Song, Alexey Sorokin, Xabier Soto, Sajad Sotudeh, Andreas Spitz, Ivan Srba, Makesh Narsimhan Sreedhar, Hiranmai Sri Adibhatla, Balaji Vasan Srinivasan, Miloš Stanojević, Gabriel Stanovsky, Katherine Stasaski, Dario Stojanovski, Alessandro Stolfo, Tomek Strzalkowski, Dan Su, Katsuhito Sudoh, Yoshi Suhara, Alane Suhr, Changzhi Sun, Che-nkai Sun, Jian Sun, Ming Sun, Qingfeng Sun, Zewei Sun, Megha Sundriyal, Hanna Suominen,

Colin Swaelens, Sandesh Swamy, Vinitra Swamy, Piotr Szymański, Danae Sánchez Villegas, Víctor M. Sánchez-cartagena, Felipe Sánchez-martínez

Santosh T.y.s.s, Sho Takase, Zeerak Talat, George Tambouratzis, Fabio Tamburini, Akihiro Tamura, Chenhao Tan, Fei Tan, Xingwei Tan, Liyan Tang, Raphael Tang, Shuai Tang, Xuting Tang, Yuka Tateisi, Marta Tatu, Selma Tekir, Serra Sinem Tekiroğlu, Irina Temnikova, Daniela Teodorescu, Urmish Thakker, Mokanarangan Thayaparan, Anton Thielmann, Brian Thompson, Craig Thomson, Camilo Thorne, Tristan Thrush, Jörg Tiedemann, Refael Tikochinski, Erik Tjong Kim Sang, Evgeniia Tokarchuk, Takenobu Tokunaga, Nadi Tomeh, Marc Tomlinson, Atnafu Lambebo Tonja, Samia Touileb, Marcos Treviso, Chen-tse Tsai, Adam Tsakalidis, Yu-hsiang Tseng, Yuenhsien Tseng, Eleftheria Tsipidi, Don Tuggener, Martin Tutek

Kiyotaka Uchimoto, Dennis Ulmer, Kanimozhi Uma, Prajna Upadhyay, Masao Utiyama

Sowmya Vajjala, Marco Valentino, Antal Van Den Bosch, Daan Van Esch, Carel Van Niekerk, Vincent Vandeghinste, Keith Vanderlinden, Lindsey Vanderlyn, Natalia Vanetik, Rossella Varvara, Shikhar Vashishth, Eva Maria Vecchi, Giulia Venturi, Rakesh Verma, Rohil Verma, Giorgos Vernikos, David Vilar, Serena Villata, Esau Villatoro-tello, Juraj Vladika, Piek Vossen, Thuy Vu, Xuan-son Vu, Ekaterina Vylomova

Tomasz Walkowiak, Yu Wan, Chuan-ju Wang, Fei Wang, Hai Wang, Haoyu Wang, Hong Wang, Jianzong Wang, Jiayi Wang, Jin Wang, Jing Wang, Kaifu Wang, Liang Wang, Lingzhi Wang, Longshaokan Wang, Longyue Wang, Miaosen Wang, Ping Wang, Qingyun Wang, Shun Wang, Wei Wang, Weichao Wang, Xin Wang, Xing Wang, Xinyi Wang, Xu Wang, Yasheng Wang, Yin-ning Wang, Zhaowei Wang, Zhilin Wang, Zhiruo Wang, Prashan Wanigasekara, Moshe Wasserblat, Shinji Watanabe, Lucas Weber, Anna Wegmann, Jerry Wei, Wei Wei, Benjamin Weiss, Gail Weiss, Leonie Weissweiler, Charles Welch, Rongxiang Weng, Aaron White, John Wieting, Gijs Wijnholds, Adina Williams, Miles Williams, Steven Wilson, Genta Winata, Guillaume Wisniewski, Seungpil Won, Ka Ho Wong, Alina Wróblewska, Di Wu, Fangzhao Wu, Minghao Wu, Stephen Wu, Winston Wu, Xianchao Wu, Xiaofeng Wu, Xixin Wu, Yuxiang Wu, Joern Wuebker, Amelie Wührl

Min Xiao, Yuqing Xie, Zhenchang Xing, Chao Xiong, Ying Xiong, Lv Xiucheng, Dongkuan Xu, Fangyuan Xu, Hanzhi Xu, Haotian Xu, Hongfei Xu, Jia Xu, Jinan Xu, Qionghai Xu, Ruifeng Xu, Silei Xu, Xinnuo Xu, Yueshen Xu, Zhen Xu, Huiyin Xue, Linting Xue, Christos Xypolopoulos

Ivan Yamshchikov, An Yan, Ming Yan, Xi Yan, Xifeng Yan, Bohao Yang, Hao Yang, Hsiu-yu Yang, Linyi Yang, Longfei Yang, Shiquan Yang, Tao Yang, Xianjun Yang, Ze Yang, Roman Yangarber, Ken Yano, Tae Yano, Wenlin Yao, Fanghua Ye, Asaf Yehudai, Wen-wai Yim, Seid Muhie Yimam, Congchi Yin, Seunghyun Yoon, Soyoun Yoon, Ori Yoran, Naoki Yoshinaga, Chenyu You, Steve Young, Bei Yu, Juntao Yu, Kai Yu, Pengfei Yu, Shoubin Yu, Tiezheng Yu, Xiaodong Yu, Xinyan Yu, Yanchao Yu, Jianhua Yuan, Shuzhou Yuan, Frances Yung

Olga Zamaraeva, Daoguang Zan, Fabio Massimo Zanzotto, Alessandra Zarccone, Xingshan Zeng, Torsten Zesch, Shuang (sophie) Zhai, Haolan Zhan, Biao Zhang, Bowen Zhang, Ge Zhang, Haodi Zhang, Haopeng Zhang, Jason Zhang, Jianguo Zhang, Lei Zhang, Michael Zhang, Ruiyi Zhang, Sheng Zhang, Shiyue Zhang, Tianchi Zhang, Yanzhe Zhang, Yichi Zhang, Yu Zhang, Yuan Zhang, Yuhui Zhang, Zhirui Zhang, Zhisong Zhang, Hai Zhao, Jinming Zhao, Lin Zhao, Mengjie Zhao, Qinghua Zhao, Tiancheng Zhao, Xiaoyan Zhao, Yilun Zhao, Chujie Zheng, Yinhe Zheng, Alisa Zhila, Yang Zhong, Ben Zhou, Guangyou Zhou, Junpei Zhou, Kaitlyn Zhou, Wangchunshu Zhou, Xiang Zhou, Yichu Zhou, Yue Zhou, Zhengyu Zhou, Su Zhu, Wanrong Zhu, Wanzheng Zhu, Xuan

Outstanding Reviewers

Gavin Abercrombie, Sallam Abualhaija, Yamen Ajjour, Emily Allaway, Milad Alshomary, Talita Anthonio, Lauriane Aufrant, Gorika Azkune, Lisa Beinborn, Valeriia Bolotova-baranova, Michele Cafagna, Deng Cai, Giovanni Cassani, Hanjie Chen, Cheng-han Chiang, Trevor Cohn, Karel D'oosterlinck, Jay Deyoung, Frank Drewes, Markus Dreyer, Tobias Falke, Yimai Fang, Xiaocheng Feng, Olivier Ferret, Antske Fokkens, Saadia Gabriel, Atticus Geiger, Tomas Goldsack, Konstantin Golobokov, Colin Gordon, Liane Guillou, Meiqi Guo, Nitish Gupta, William Havard, Michael Heck, Sophie Henning, Nora Hollenstein, Radu Tudor Ionescu, Tatsuya Ishigaki, Robin Jia, Min-yen Kan, Graham Katz, Christo Kirov, Ioannis Konstas, Michael Kranzlein, Udo Kruschwitz, Roland Kuhn, Yi-an Lai, Young-suk Lee, Yves Lepage, Piyawat Lertvittayakumjorn, Matthias Lindemann, Zhengyuan Liu, Henrique Lopes Cardoso, Brielen Madureira, Yuval Marton, Jonathan May, Kathleen Mckeown, Clara Meister, Zaiqiao Meng, Filip Miletic, Kata Naszadi, Yasumasa Onoe, Juri Opitz, Tiago Pimentel, Barbara Plank, Traian Rebedea, Ehud Reiter, Mathieu Roche, Rudolf Rosa, Candace Ross, Sumegh Roychowdhury, Sebastian Ruder, Elizabeth Salesky, David Schlangen, Hendrik Schuff, Sebastian Schuster, Djamé Seddah, Mattia Setzu, Kyle Shaffer, Vered Shwartz, Olivier Siohan, Matthew Stone, Alessandro Suglia, Benjamin Van Durme, Neeraj Varshney, Jake Vasilakes, Dirk Văth, Henning Wachsmuth, Michael Wiegand, Tomer Wolfson, Hanqi Yan, Eugene Yang, Marceley Zanon Boito, Amir Zeldes

Table of Contents

<i>Using Punctuation as an Adversarial Attack on Deep Learning-Based NLP Systems: An Empirical Study</i>	
Brian Formento, Chuan Sheng Foo, Luu Anh Tuan and See Kiong Ng	1
<i>Self-Supervised Unimodal Label Generation Strategy Using Recalibrated Modality Representations for Multimodal Sentiment Analysis</i>	
Yewon Hwang and Jong-Hwan Kim	35
<i>Fighting FIRE with FIRE: Assessing the Validity of Text-to-Video Retrieval Benchmarks</i>	
Pedro Rodriguez, Mahmoud Azab, Becca Silvert, Renato Sanchez, Linzy Labson, Hardik Shah and Seungwhan Moon	47
<i>Improving Numeracy by Input Reframing and Quantitative Pre-Finetuning Task</i>	
Chung-Chi Chen, Hiroya Takamura, Ichiro Kobayashi and Yusuke Miyao	69
<i>Visualize Before You Write: Imagination-Guided Open-Ended Text Generation</i>	
Wanrong Zhu, An Yan, Yujie Lu, Wenda Xu, Xin Wang, Miguel Eckstein and William Yang Wang	78
<i>ImaginE: An Imagination-Based Automatic Evaluation Metric for Natural Language Generation</i>	
Wanrong Zhu, Xin Wang, An Yan, Miguel Eckstein and William Yang Wang	93
<i>Entity-Aware Dual Co-Attention Network for Fake News Detection</i>	
Sin-han Yang, Chung-chi Chen, Hen-Hsen Huang and Hsin-Hsi Chen	106
<i>CIKQA: Learning Commonsense Inference with a Unified Knowledge-in-the-loop QA Paradigm</i>	
Hongming Zhang, Yintong Huo, Yanai Elazar, Yangqiu Song, Yoav Goldberg and Dan Roth .	114
<i>Data-Efficient Methods For Improving Hate Speech Detection</i>	
Sumegh Roychowdhury and Vikram Gupta	125
<i>Learning the Effects of Physical Actions in a Multi-modal Environment</i>	
Gautier Dagan, Frank Keller and Alex Lascarides	133
<i>FVQA 2.0: Introducing Adversarial Samples into Fact-based Visual Question Answering</i>	
Weizhe Lin, Zhilin Wang and Bill Byrne	149
<i>Revisiting Intermediate Layer Distillation for Compressing Language Models: An Overfitting Perspective</i>	
Jongwoo Ko, Seungjoon Park, Minchan Jeong, Sukjin Hong, Euijai Ahn, Du-Seong Chang and Se-Young Yun	158
<i>Implicit Temporal Reasoning for Evidence-Based Fact-Checking</i>	
Liesbeth Allein, Marlon Saelens, Ruben Cartuyvels and Marie-Francine Moens	176
<i>Active PETs: Active Data Annotation Prioritisation for Few-Shot Claim Verification with Pattern Exploiting Training</i>	
Xia Zeng and Arkaitz Zubiaga	190
<i>Plan-then-Seam: Towards Efficient Table-to-Text Generation</i>	
Liang Li, Ruiying Geng, Chengyang Fang, Bing Li, Can Ma, Binhua Li and Yongbin Li	205
<i>A corpus of metaphors as register markers</i>	
Markus Egg and Valia Kordoni	220

<i>Translate First Reorder Later: Leveraging Monotonicity in Semantic Parsing</i>	
Francesco Cazzaro, Davide Locatelli, Ariadna Quattoni and Xavier Carreras	227
<i>PePe: Personalized Post-editing Model utilizing User-generated Post-edits</i>	
Jihyeon Lee, Taehee Kim, Yunwon Tae, Cheonbok Park and Jaegul Choo	239
<i>Infusing Context and Knowledge Awareness in Multi-turn Dialog Understanding</i>	
Ting-Wei Wu and Biing-Hwang Juang	254
<i>MCoNaLa: A Benchmark for Code Generation from Multiple Natural Languages</i>	
Zhiruo Wang, Grace Cuenca, Shuyan Zhou, Frank F. Xu and Graham Neubig	265
<i>Augmenting pre-trained language models with audio feature embedding for argumentation mining in political debates</i>	
Rafael Mestre, Stuart Middleton, Matt Ryan, Masood Gheasi, Timothy Norman and Jiatong Zhu	274
<i>Improving Retrieval Augmented Neural Machine Translation by Controlling Source and Fuzzy-Match Interactions</i>	
Cuong Hoang, Devendra Sachan, Prashant Mathur, Brian Thompson and Marcello Federico .	289
<i>CALM-Bench: A Multi-task Benchmark for Evaluating Causality-Aware Language Models</i>	
Dhairya Dalal, Paul Buitelaar and Mihael Arcan	296
<i>ezCoref: Towards Unifying Annotation Guidelines for Coreference Resolution</i>	
Ankita Gupta, Marzena Karpinska, Wenlong Zhao, Kalpesh Krishna, Jack Merullo, Luke Yeh, Mohit Iyyer and Brendan O'Connor	312
<i>PREME: Preference-based Meeting Exploration through an Interactive Questionnaire</i>	
Negar Arabzadeh, Ali Ahmadvand, Julia Kiseleva, Yang Liu, Ahmed Hassan Awadallah, Ming Zhong and Milad Shokouhi	331
<i>Sentence Identification with BOS and EOS Label Combinations</i>	
Takuma Udagawa, Hiroshi Kanayama and Issei Yoshida	343
<i>Gauging the Gap Between Human and Machine Text Simplification Through Analytical Evaluation of Simplification Strategies and Errors</i>	
Daichi Yamaguchi, Rei Miyata, Sayuka Shimada and Satoshi Sato	359
<i>Bridging the Gap between Pre-Training and Fine-Tuning for Commonsense Generation</i>	
Haoran Yang, Yan Wang, Piji Li, Wei Bi, Wai Lam and Chen Xu	376
<i>LED: A Dataset for Life Event Extraction from Dialogs</i>	
Yi-Pei Chen, An-Zi Yen, Hen-Hsen Huang, Hideki Nakayama and Hsin-Hsi Chen	384
<i>Reading and Reasoning over Chart Images for Evidence-based Automated Fact-Checking</i>	
Mubashara Akhtar, Oana Cocarascu and Elena Simperl	399
<i>Causal Reasoning of Entities and Events in Procedural Texts</i>	
Li Zhang, Hainiu Xu, Yue Yang, Shuyan Zhou, Weiqiu You, Manni Arora and Chris Callison-Burch	415
<i>Few-Shot Structured Policy Learning for Multi-Domain and Multi-Task Dialogues</i>	
Thibault Cordier, Tanguy Urvoy, Fabrice Lefèvre and Lina M. Rojas Barahona	432

<i>Transfer Knowledge from Natural Language to Electrocardiography: Can We Detect Cardiovascular Disease Through Language Models?</i>	
Jielin Qiu, William Han, Jiacheng Zhu, Mengdi Xu, Michael Rosenberg, Emerson Liu, Douglas Weber and Ding Zhao	442
<i>Practical Takes on Federated Learning with Pretrained Language Models</i>	
Ankur Agarwal, Mehdi Rezagholizadeh and Prasanna Parthasarathi	454
<i>Paper Bullets: Modeling Propaganda with the Help of Metaphor</i>	
Daniel Baleato Rodríguez, Verna Dankers, Preslav Nakov and Ekaterina Shutova	472
<i>Lexical Semantics with Large Language Models: A Case Study of English break"</i>	
Erika Petersen and Christopher Potts	490
<i>SWING: Balancing Coverage and Faithfulness for Dialogue Summarization</i>	
Kung-Hsiang Huang, Siffi Singh, Xiaofei Ma, Wei Xiao, Feng Nan, Nicholas Dingwall, William Yang Wang and Kathleen McKeown	512
<i>Language-Aware Multilingual Machine Translation with Self-Supervised Learning</i>	
Haoran Xu, Jean Maillard and Vedanuj Goswami	526
<i>Cloze Quality Estimation for Language Assessment</i>	
Zizheng Zhang, Masato Mita and Mamoru Komachi	540
<i>Bag of Tricks for In-Distribution Calibration of Pretrained Transformers</i>	
Jaeyoung Kim, Dongbin Na, Sungchul Choi and Sungbin Lim	551
<i>Fine-Tuning Deteriorates General Textual Out-of-Distribution Detection by Distorting Task-Agnostic Features</i>	
Sishuo Chen, Wenkai Yang, Xiaohan Bi and Xu Sun	564
<i>A Question of Style: A Dataset for Analyzing Formality on Different Levels</i>	
Elisabeth Eder, Ulrike Krieg-Holz and Michael Wiegand	580
<i>Task-specific Compression for Multi-task Language Models using Attribution-based Pruning</i>	
Nakyeong Yang, Yunah Jang, Hwanhee Lee, Seohyeong Jeong and Kyomin Jung	594
<i>Zero-shot Transfer of Article-aware Legal Outcome Classification for European Court of Human Rights Cases</i>	
Santosh T.Y.S.S, Oana Ichim and Matthias Grabmair	605
<i>Abstractive Document Summarization with Summary-length Prediction</i>	
Jingun Kwon, Hidetaka Kamigaito and Manabu Okumura	618
<i>Hierarchical Label Generation for Text Classification</i>	
Jingun Kwon, Hidetaka Kamigaito, Young-In Song and Manabu Okumura	625
<i>Active Learning for Multilingual Semantic Parser</i>	
Zhuang Li and Gholamreza Haffari	633
<i>Joint Word and Morpheme Segmentation with Bayesian Non-Parametric Models</i>	
Shu Okabe and François Yvon	640
<i>Cross-Lingual Transfer of Cognitive Processing Complexity</i>	
Charlotte Pouw, Nora Hollenstein and Lisa Beinborn	655
<i>Does Transliteration Help Multilingual Language Modeling?</i>	
Ibraheem Muhammad Moosa, Mahmud Elahi Akhter and Ashfia Habib	670

<i>A Multilingual Dataset of Racial Stereotypes in Social Media Conversational Threads</i>	
Tom Bourgeade, Alessandra Teresa Cignarella, Simona Frenda, Mario Laurent, Wolfgang Schmeisser-Nieto, Farah Benamara, Cristina Bosco, Véronique Moriceau, Viviana Patti and Mariona Taulé . . .	686
<i>Detecting Contextomized Quotes in News Headlines by Contrastive Learning</i>	
Seonyeong Song, Hyeonho Song, Kunwoo Park, Jiyoung Han and Meeyoung Cha	697
<i>Zero-Shot On-the-Fly Event Schema Induction</i>	
Rotem Dror, Haoyu Wang and Dan Roth	705
<i>BanglaNLG and BanglaT5: Benchmarks and Resources for Evaluating Low-Resource Natural Language Generation in Bangla</i>	
Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad and Rifat Shahriyar	726
<i>It's about Time: Rethinking Evaluation on Rumor Detection Benchmarks using Chronological Splits</i>	
Yida Mu, Kalina Bontcheva and Nikolaos Aletras	736
<i>MUTANT: A Multi-sentential Code-mixed Hinglish Dataset</i>	
Rahul Gupta, Vivek Srivastava and Mayank Singh	744
<i>Bridging the Gap between Native Text and Translated Text through Adversarial Learning: A Case Study on Cross-Lingual Event Extraction</i>	
Pengfei Yu, Jonathan May and Heng Ji	754
<i>Scalable Prompt Generation for Semi-supervised Learning with Language Models</i>	
Yuhang Zhou, Suraj Maharjan and Beiye Liu	770
<i>Novel Feature Discovery for Task-Oriented Dialog Systems</i>	
Vinh Thinh Ho, Mohamed Soliman and Abdalghani Abujabal	782
<i>Context Generation Improves Open Domain Question Answering</i>	
Dan Su, Mostofa Patwary, Shrimai Prabhumoye, Peng Xu, Ryan Prenger, Mohammad Shoeybi, Pascale Fung, Anima Anandkumar and Bryan Catanzaro	793
<i>RedHOT: A Corpus of Annotated Medical Questions, Experiences, and Claims on Social Media</i>	
Somin Wadhwa, Vivek Khetan, Silvio Amir and Byron Wallace	809
<i>Paparazzi: A Deep Dive into the Capabilities of Language and Vision Models for Grounding Viewpoint Descriptions</i>	
Henrik Voigt, Jan Hombeck, Monique Meuschke, Kai Lawonn and Sina Zarriß	828
<i>PLACES: Prompting Language Models for Social Conversation Synthesis</i>	
Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Seokhwan Kim, Andy Rosenbaum, Yang Liu, Zhou Yu and Dilek Hakkani-Tur	844
<i>FedPerC: Federated Learning for Language Generation with Personal and Context Preference Embeddings</i>	
Andrew Silva, Pradyumna Tambwekar and Matthew Gombolay	869
<i>A Neural CRF-based Hierarchical Approach for Linear Text Segmentation</i>	
Inderjeet Nair, Aparna Garimella, Balaji Vasani Srinivasan, Natwar Modani, Niyati Chhaya, Srikrishna Karanam and Sumit Shekhar	883
<i>MultiFin: A Dataset for Multilingual Financial NLP</i>	
Rasmus Jørgensen, Oliver Brandt, Mareike Hartmann, Xiang Dai, Christian Igel and Desmond Elliott	894

<i>MLASK: Multimodal Summarization of Video-based News Articles</i> Mateusz Krubiński and Pavel Pecina	910
<i>Going beyond research datasets: Novel intent discovery in the industry setting</i> Aleksandra Chrabrowa, Tsimur Hadeliya, Dariusz Kajtoch, Robert Mroczkowski and Piotr Rybak	925
<i>DATScore: Evaluating Translation with Data Augmented Translations</i> Moussa Kamal Eddine, Guokan Shang and Michalis Vazirgiannis	942
<i>How do decoding algorithms distribute information in dialogue responses?</i> Saranya Venkatraman, He He and David Reitter	953
<i>Benchmarking Long-tail Generalization with Likelihood Splits</i> Ameya Godbole and Robin Jia	963
<i>Exploring Enhanced Code-Switched Noising for Pretraining in Neural Machine Translation</i> Vivek Iyer, Arturo Oncevay and Alexandra Birch	984
<i>XQA-DST: Multi-Domain and Multi-Lingual Dialogue State Tracking</i> Han Zhou, Ignacio Iacobacci and Pasquale Minervini	999
<i>Improving Prediction Backward-Compatibility in NLP Model Upgrade with Gated Fusion</i> Yi-An Lai, Elman Mansimov, Yuqing Xie and Yi Zhang	1010
<i>AmbiCoref: Evaluating Human and Model Sensitivity to Ambiguous Coreference</i> Yuewei Yuan, Chaitanya Malaviya and Mark Yatskar	1023
<i>Improving Unsupervised Out-of-domain detection through Pseudo Labeling and Learning</i> Byoungchan Lee, Jaesik Kim, Junekyu Park and Kyung-Ah Sohn	1031
<i>How Many Data Samples is an Additional Instruction Worth?</i> Ravsehaj Singh Puri, Swaroop Mishra, Mihir Parmar and Chitta Baral	1042
<i>[MASK] Insertion: a robust method for anti-adversarial attacks</i> Xinrong Hu, Ce Xu, Junlong Ma, Zijian Huang, Jie Yang, Yi Guo and Johan Barthelemy ...	1058
<i>ViDeBERTa: A powerful pre-trained language model for Vietnamese</i> Cong Dao Tran, Nhut Huy Pham, Anh Tuan Nguyen, Truong Son Hy and Tu Vu	1071
<i>NapSS: Paragraph-level Medical Text Simplification via Narrative Prompting and Sentence-matching Summarization</i> Junru Lu, Jiazheng Li, Byron Wallace, Yulan He and Gabriele Pergola	1079
<i>Long-tailed Extreme Multi-label Text Classification by the Retrieval of Generated Pseudo Label Descriptions</i> Ruohong Zhang, Yau-Shian Wang, Yiming Yang, Donghan Yu, Tom Vu and Likun Lei	1092
<i>Unsupervised Keyphrase Extraction via Interpretable Neural Networks</i> Rishabh Joshi, Vidhisha Balachandran, Emily Saldanha, Maria Glenski, Svitlana Volkova and Yulia Tsvetkov	1107
<i>Large Language Models are few(1)-shot Table Reasoners</i> Wenhu Chen	1120
<i>Realistic Citation Count Prediction Task for Newly Published Papers</i> Jun Hirako, Ryohei Sasano and Koichi Takeda	1131

<i>“Why do I feel offended?” - Korean Dataset for Offensive Language Identification</i>	
San-Hee Park, Kang-Min Kim, O-Joun Lee, Youjin Kang, Jaewon Lee, Su-Min Lee and SangKeun Lee	1142
<i>Empirical Investigation of Neural Symbolic Reasoning Strategies</i>	
Yoichi Aoki, Keito Kudo, Tatsuki Kuribayashi, Ana Brassard, Masashi Yoshikawa, Keisuke Sakaguchi and Kentaro Inui	1154
<i>Analyzing the Effectiveness of the Underlying Reasoning Tasks in Multi-hop Question Answering</i>	
Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara and Akiko Aizawa	1163
<i>PubMedCLIP: How Much Does CLIP Benefit Visual Question Answering in the Medical Domain?</i>	
Sedigheh Eslami, Christoph Meinel and Gerard de Melo	1181
<i>Multilingual BERT has an accent: Evaluating English influences on fluency in multilingual models</i>	
Isabel Papadimitriou, Kezia Lopez and Dan Jurafsky	1194
<i>Reassessing Evaluation Practices in Visual Question Answering: A Case Study on Out-of-Distribution Generalization</i>	
Aishwarya Agrawal, Ivana Kajic, Emanuele Bugliarello, Elnaz Davoodi, Anita Gergely, Phil Blunsom and Aida Nematzadeh	1201
<i>Our kind of people? Detecting populist references in political debates</i>	
Christopher Klamm, Ines Rehbein and Simone Paolo Ponzetto	1227
<i>SharPT: Shared Latent Space Prompt Tuning</i>	
Bo Pang, Semih Yavuz, Caiming Xiong and Yingbo Zhou	1244
<i>Mini But Mighty: Efficient Multilingual Pretraining with Linguistically-Informed Data Selection</i>	
Tolulope Ogunremi, Dan Jurafsky and Christopher Manning	1251
<i>Long Document Summarization with Top-down and Bottom-up Inference</i>	
Bo Pang, Erik Nijkamp, Wojciech Kryscinski, Silvio Savarese, Yingbo Zhou and Caiming Xiong	1267
<i>Open Information Extraction with Entity Focused Constraints</i>	
Prajna Upadhyay, Oana Balalau and Ioana Manolescu	1285
<i>Hierarchical3D Adapters for Long Video-to-text Summarization</i>	
Pinelopi Papalampidi and Mirella Lapata	1297
<i>An Intra-Class Relation Guided Approach for Code Comment Generation</i>	
Zhenni Wang, Xiaohan Yu, Yansong Feng and Dongyan Zhao	1321
<i>Spelling convention sensitivity in neural language models</i>	
Elizabeth Nielsen, Christo Kirov and Brian Roark	1334
<i>Modelling Language Acquisition through Syntactico-Semantic Pattern Finding</i>	
Jonas Doumen, Katrien Beuls and Paul Van Eecke	1347
<i>Benchmark Data and Evaluation Framework for Intent Discovery Around COVID-19 Vaccine Hesitancy</i>	
Shai Gretz, Assaf Toledo, Roni Friedman, Dan Lahav, Rose Weeks, Naor Bar-Zeev, João Sedoc, Pooja Sangha, Yoav Katz and Noam Slonim	1358
<i>Learning Disentangled Representations for Natural Language Definitions</i>	
Danilo Silva De Carvalho, Giangiacomo Mercatali, Yingji Zhang and André Freitas	1371

<i>Distinguishability Calibration to In-Context Learning</i>	
Hongjing Li, Hanqi Yan, Yanran Li, Li Qian, Yulan He and Lin Gui	1385
<i>Investigating anatomical bias in clinical machine learning algorithms</i>	
Jannik Pedersen, Martin Laursen, Pernille Vinholt, Anne Alnor and Thusius Savarimuthu .	1398
<i>Topic Ontologies for Arguments</i>	
Yamen Ajjour, Johannes Kiesel, Benno Stein and Martin Potthast	1411
<i>Longtonotes: OntoNotes with Longer Coreference Chains</i>	
Kumar Shridhar, Nicholas Monath, Raghuv eer Thirukovalluru, Alessandro Stolfo, Manzil Zaheer, Andrew McCallum and Mrinmaya Sachan	1428
<i>More Robust Schema-Guided Dialogue State Tracking via Tree-Based Paraphrase Ranking</i>	
Alexandru Coca, Bo-Hsiang Tseng, Weizhe Lin and Bill Byrne	1443
<i>Language Model Decoding as Likelihood–Utility Alignment</i>	
Martin Josifoski, Maxime Peyrard, Frano Rajič, Jiheng Wei, Debjit Paul, Valentin Hartmann, Barun Patra, Vishrav Chaudhary, Emre Kiciman and Boi Faltings	1455
<i>Lightweight Spatial Modeling for Combinatorial Information Extraction From Documents</i>	
Yanfei Dong, Lambert Deng, Jiazheng Zhang, Xiaodong Yu, Ting Lin, Francesco Gelli, Soujanya Poria and Wee Sun Lee	1471
<i>On the Generalization Ability of Retrieval-Enhanced Transformers</i>	
Tobias Norlund, Ehsan Doostmohammadi, Richard Johansson and Marco Kuhlmann	1485
<i>Assessing Monotonicity Reasoning in Dutch through Natural Language Inference</i>	
Gijs Wijnholds	1494
<i>Noisy Parallel Data Alignment</i>	
Ruoyu Xie and Antonios Anastasopoulos	1501
<i>Enhancing Dialogue Generation with Conversational Concept Flows</i>	
Siheng Li, Wangjie Jiang, Pengda Si, Cheng Yang, Qiu Yao, Jinchao Zhang, Jie Zhou and Yujiu Yang	1514
<i>SMHD-GER: A Large-Scale Benchmark Dataset for Automatic Mental Health Detection from Social Media in German</i>	
Sourabh Zanwar, Daniel Wiechmann, Yu Qiao and Elma Kerz	1526
<i>Exploring Data Augmentation for Code Generation Tasks</i>	
Pinzhen Chen and Gerasimos Lampouras	1542
<i>Stabilized In-Context Learning with Pre-trained Language Models for Few Shot Dialogue State Tracking</i>	
Derek Chen, Kun Qian and Zhou Yu	1551
<i>Can Demographic Factors Improve Text Classification? Revisiting Demographic Adaptation in the Age of Transformers</i>	
Chia-Chien Hung, Anne Lauscher, Dirk Hovy, Simone Paolo Ponzetto and Goran Glavaš . .	1565
<i>JBLiMP: Japanese Benchmark of Linguistic Minimal Pairs</i>	
Taiga Someya and Yohei Oseki	1581
<i>SMATCH++: Standardized and Extended Evaluation of Semantic Graphs</i>	
Juri Opitz	1595

<i>An Extended Sequence Tagging Vocabulary for Grammatical Error Correction</i>	
Stuart Mesham, Christopher Bryant, Marek Rei and Zheng Yuan	1608
<i>Cheating to Identify Hard Problems for Neural Machine Translation</i>	
Proyag Pal and Kenneth Heafield	1620
<i>Model-Agnostic Bias Measurement in Link Prediction</i>	
Lena Schwertmann, Manoj Prabhakar Kannan Ravi and Gerard de Melo	1632
<i>Divergence-Based Domain Transferability for Zero-Shot Classification</i>	
Alexander Pugantsov and Richard McCreadie	1649
<i>EDU-level Extractive Summarization with Varying Summary Lengths</i>	
Yuping Wu, Ching-Hsun Tseng, Jiayu Shang, Shengzhong Mao, Goran Nenadic and Xiao-Jun Zeng	1655
<i>Chère maison or maison chère"? Transformer-based prediction of adjective placement in French</i>	
Eleni Metheniti, Tim Van de Cruys, Wissam Kerkri, Juliette Thuilier and Nabil Hathout ...	1668
<i>On the Role of Reviewer Expertise in Temporal Review Helpfulness Prediction</i>	
Mir Tafseer Nayeem and Davood Rafiei	1684
<i>Towards a Unified Model for Generating Answers and Explanations in Visual Question Answering</i>	
Chenxi Whitehouse, Tillman Weyde and Pranava Madhyastha	1693
<i>Machine Translation between Spoken Languages and Signed Languages Represented in SignWriting</i>	
Zifan Jiang, Amit Moryossef, Mathias Müller and Sarah Ebling	1706
<i>A Multi-dimensional Evaluation of Tokenizer-free Multilingual Pretrained Models</i>	
Jimin Sun, Patrick Fernandes, Xinyi Wang and Graham Neubig	1725
<i>Neural Ranking with Weak Supervision for Open-Domain Question Answering : A Survey</i>	
Xiaoyu Shen, Svitlana Vakulenko, Marco del Tredici, Gianni Barlacchi, Bill Byrne and Adria de Gispert	1736
<i>Double Retrieval and Ranking for Accurate Question Answering</i>	
Zeyu Zhang, Thuy Vu and Alessandro Moschitti	1751
<i>Evaluating the Diversity, Equity, and Inclusion of NLP Technology: A Case Study for Indian Languages</i>	
Simran Khanuja, Sebastian Ruder and Partha Talukdar	1763
<i>Joint Reasoning on Hybrid-knowledge sources for Task-Oriented Dialog</i>	
Mayank Mishra, Danish Contractor and Dinesh Raghu	1778
<i>Revisiting Offline Compression: Going Beyond Factorization-based Methods for Transformer Language Models</i>	
Mohammadreza Banaei, Klaudia Bałazy, Artur Kasymov, Rémi Lebret, Jacek Tabor and Karl Aberer	1788
<i>PriMeSRL-Eval: A Practical Quality Metric for Semantic Role Labeling Systems Evaluation</i>	
Ishan Jindal, Alexandre Rademaker, Khoi-Nguyen Tran, Huaiyu Zhu, Hiroshi Kanayama, Marina Danilevsky and Yunyao Li	1806
<i>Prompt-based Learning for Text Readability Assessment</i>	
Bruce W. Lee and Jason Lee	1819
<i>Best Practices in the Creation and Use of Emotion Lexicons</i>	
Saif Mohammad	1825

<i>The Role of Semantic Parsing in Understanding Procedural Text</i>	
Hossein Rajaby Faghihi, Parisa Kordjamshidi, Choh Man Teng and James Allen	1837
<i>Named Entity Recognition in a Very Homogenous Domain</i>	
Oshin Agarwal and Ani Nenkova	1850
<i>Crawling The Internal Knowledge-Base of Language Models</i>	
Roi Cohen, Mor Geva, Jonathan Berant and Amir Globerson	1856
<i>Intent Identification and Entity Extraction for Healthcare Queries in Indic Languages</i>	
Ankan Mullick, Ishani Mondal, Sourjyadip Ray, Raghav R, G Chaitanya and Pawan Goyal .	1870
<i>Text-Derived Knowledge Helps Vision: A Simple Cross-modal Distillation for Video-based Action Anticipation</i>	
Sayontan Ghosh, Tanvi Aggarwal, Minh Hoai and Niranjan Balasubramanian	1882
<i>Simple Yet Effective Synthetic Dataset Construction for Unsupervised Opinion Summarization</i>	
Ming Shen, Jie Ma, Shuai Wang, Yogarshi Vyas, Kalpit Dixit, Miguel Ballesteros and Yassine Benajiba	1898
<i>Towards Fine-tuning Pre-trained Language Models with Integer Forward and Backward Propagation</i>	
Mohammadreza Tayaranian Hosseini, Alireza Ghaffari, Marzieh S. Tahaei, Mehdi Rezagholizadeh, Masoud Asgharian and Vahid Partovi Nia	1912
<i>Data Augmentation for Radiology Report Simplification</i>	
Ziyu Yang, Santhosh Cherian and Slobodan Vucetic	1922
<i>Embedding Recycling for Language Models</i>	
Jon Saad-Falcon, Amanpreet Singh, Luca Soldaini, Mike D’Arcy, Arman Cohan and Doug Downey	1933
<i>Trained on 100 million words and still in shape: BERT meets British National Corpus</i>	
David Samuel, Andrey Kutuzov, Lilja Øvrelid and Erik Velldal	1954
<i>Generating Synthetic Speech from SpokenVocab for Speech Translation</i>	
Jinming Zhao, Gholamreza Haffari and Ehsan Shareghi	1975
<i>Bounding the Capabilities of Large Language Models in Open Text Generation with Prompt Constraints</i>	
Albert Lu, Hongxin Zhang, Yanzhe Zhang, Xuezhi Wang and Diyi Yang	1982
<i>Learning to Retrieve Engaging Follow-Up Queries</i>	
Christopher Richardson, Sudipta Kar, Anjishnu Kumar, Anand Ramachandran, Zeynab Raeesy, Omar Khan and Abhinav Sethy	2009
<i>Selective-LAMA: Selective Prediction for Confidence-Aware Evaluation of Language Models</i>	
Hiyori Yoshikawa and Naoaki Okazaki	2017
<i>Multi-View Source Ablation for Faithful Summarization</i>	
Shuyang Cao, Liang Ma, Di Lu, Robert L Logan IV, Joel Tetreault and Alejandro Jaimes . .	2029
<i>Mining Effective Features Using Quantum Entropy for Humor Recognition</i>	
Yang Liu and Yuexian Hou	2048
<i>AdapterSoup: Weight Averaging to Improve Generalization of Pretrained Language Models</i>	
Alexandra Chronopoulou, Matthew Peters, Alexander Fraser and Jesse Dodge	2054
<i>Towards End-to-End Open Conversational Machine Reading</i>	
Sizhe Zhou, Siru Ouyang, Zhuosheng Zhang and Hai Zhao	2064

<i>Generative Knowledge Selection for Knowledge-Grounded Dialogues</i> Weiwei Sun, Pengjie Ren and Zhaochun Ren	2077
<i>Evaluating the Tradeoff Between Abtractiveness and Factuality in Abtractive Summarization</i> Markus Dreyer, Mengwen Liu, Feng Nan, Sandeep Atluri and Sujith Ravi	2089
<i>Fairness in Language Models Beyond English: Gaps and Challenges</i> Krithika Ramesh, Sunayana Sitaram and Monojit Choudhury	2106
<i>Global-Local Modeling with Prompt-Based Knowledge Enhancement for Emotion Inference in Conversation</i> Renxi Wang and Shi Feng	2120
<i>Headline Token-based Discriminative Learning for Subheading Generation in News Article</i> Joonwon Jang and Misuk Kim	2128
<i>Decipherment as Regression: Solving Historical Substitution Ciphers by Learning Symbol Recurrence Relations</i> Nishant Kambhatla, Logan Born and Anoop Sarkar	2136
<i>A Survey on Recent Advances in Keyphrase Extraction from Pre-trained Language Models</i> Mingyang Song, Yi Feng and Liping Jing	2153
<i>Prompting for explanations improves Adversarial NLI. Is this true? {Yes} it is {true} because {it weakens superficial cues}</i> Pride Kavumba, Ana Brassard, Benjamin Heinzerling and Kentaro Inui	2165
<i>JobXMLC: EXtreme Multi-Label Classification of Job Skills with Graph Neural Networks</i> Nidhi Goyal, Jushaan Kalra, Charu Sharma, Raghava Mutharaju, Niharika Sachdeva and Ponnurangam Kumaraguru	2181
<i>ViLPAct: A Benchmark for Compositional Generalization on Multimodal Human Activities</i> Terry Yue Zhuo, Yaqing Liao, Yuecheng Lei, Lizhen Qu, Gerard de Melo, Xiaojun Chang, Yazhou Ren and Zenglin Xu	2192
<i>Grammatical Error Correction through Round-Trip Machine Translation</i> Yova Kementchedjhieva and Anders Søgaard	2208
<i>Does Masked Language Model Pre-training with Artificial Data Improve Low-resource Neural Machine Translation?</i> Hiroto Tamura, Toshio Hirasawa, Hwicheon Kim and Mamoru Komachi	2216
<i>Performance and Risk Trade-offs for Multi-word Text Prediction at Scale</i> Aniket Vashishtha, S Sai Prasad, Payal Bajaj, Vishrav Chaudhary, Kate Cook, Sandipan Dandapat, Sunayana Sitaram and Monojit Choudhury	2226
<i>Searching for Better Database Queries in the Outputs of Semantic Parsers</i> Anton Osokin, Irina Saporina and Ramil Yarullin	2243
<i>Style-Aware Contrastive Learning for Multi-Style Image Captioning</i> Yucheng Zhou and Guodong Long	2257
<i>Strategize Before Teaching: A Conversational Tutoring System with Pedagogy Self-Distillation</i> Lingzhi Wang, Mrinmaya Sachan, Xingshan Zeng and Kam-Fai Wong	2268

<i>ICA-Proto: Iterative Cross Alignment Prototypical Network for Incremental Few-Shot Relation Classification</i>	
Wangjie Jiang, Zhihao Ye, Bang Liu, Ruihui Zhao, Jianguang Zheng, Mengyao Li, Zhiyong Li, Yujiu Yang and Yefeng Zheng	2275
<i>A Large-Scale Multilingual Study of Visual Constraints on Linguistic Selection of Descriptions</i>	
Uri Berger, Lea Frermann, Gabriel Stanovsky and Omri Abend	2285
<i>How Much Syntactic Supervision is Good Enough"?</i>	
Hiroshi Noji and Yohei Oseki	2300
<i>Are the Best Multilingual Document Embeddings simply Based on Sentence Embeddings?</i>	
Sonal Sannigrahi, Josef van Genabith and Cristina España-Bonet	2306
<i>Improving User Controlled Table-To-Text Generation Robustness</i>	
Hanxu Hu, Yunqing Liu, Zhongyi Yu and Laura Perez-Beltrachini	2317
<i>Better Pre-Training by Reducing Representation Confusion</i>	
Haojie Zhang, Mingfei Liang, Ruobing Xie, Zhenlong Sun, Bo Zhang and Leyu Lin	2325
<i>MAFiD: Moving Average Equipped Fusion-in-Decoder for Question Answering over Tabular and Textual Data</i>	
Sung-Min Lee, Eunhwan Park, Daeryong Seo, Donghyeon Jeon, Inho Kang and Seung-Hoon Na	2337
<i>Transformer-based Models for Long-Form Document Matching: Challenges and Empirical Analysis</i>	
Akshita Jha, Adithya Samavedhi, Vineeth Rakesh, Jaideep Chandrashekar and Chandan Reddy	2345
<i>Simple and Effective Multi-Token Completion from Masked Language Models</i>	
Oren Kalinsky, Guy Kushilevitz, Alexander Libov and Yoav Goldberg	2356
<i>A Survey on Dynamic Neural Networks for Natural Language Processing</i>	
Canwen Xu and Julian McAuley	2370
<i>Transformers with Learnable Activation Functions</i>	
Haishuo Fang, Ji-Ung Lee, Nafise Sadat Moosavi and Iryna Gurevych	2382
<i>The Solvability of Interpretability Evaluation Metrics</i>	
Yilun Zhou and Julie Shah	2399
<i>Reliable Gradient-free and Likelihood-free Prompt Tuning</i>	
Maohao Shen, Soumya Ghosh, Prasanna Sattigeri, Subhro Das, Yuheng Bu and Gregory Wornell	2416
<i>Combining Psychological Theory with Language Models for Suicide Risk Detection</i>	
Daniel Izmaylov, Avi Segal, Kobi Gal, Meytal Grimland and Yossi Levi-Belz	2430
<i>Cross-Lingual Question Answering over Knowledge Base as Reading Comprehension</i>	
Chen Zhang, Yuxuan Lai, Yansong Feng, Xingyu Shen, Haowei Du and Dongyan Zhao	2439
<i>Delving Deeper into Cross-lingual Visual Question Answering</i>	
Chen Liu, Jonas Pfeiffer, Anna Korhonen, Ivan Vulić and Iryna Gurevych	2453
<i>Bridging Argument Quality and Deliberative Quality Annotations with Adapters</i>	
Neele Falk and Gabriella Lapesa	2469

<i>Interventional Probing in High Dimensions: An NLI Case Study</i> Julia Rozanova, Marco Valentino, Lucas Cordeiro and André Freitas	2489
<i>Program Synthesis for Complex QA on Charts via Probabilistic Grammar Based Filtered Iterative Back-Translation</i> Shabbirhussain Bhaisaheb, Shubham Paliwal, Rajaswa Patil, Manasi Patwardhan, Lovekesh Vig and Gautam Shroff	2501
<i>Exploiting Language Characteristics for Legal Domain-Specific Language Model Pretraining</i> Inderjeet Nair and Natwar Modani	2516
<i>Global Constraints with Prompting for Zero-Shot Event Argument Classification</i> Zizheng Lin, Hongming Zhang and Yangqiu Song	2527
<i>Distillation of encoder-decoder transformers for sequence labelling</i> Marco Farina, Duccio Pappadopulo, Anant Gupta, Leslie Huang, Ozan Irsoy and Thamar Solorio	2539
<i>Predicting Desirable Revisions of Evidence and Reasoning in Argumentative Writing</i> Tazin Afrin and Diane Litman	2550
<i>Discourse Structure Extraction from Pre-Trained and Fine-Tuned Language Models in Dialogues</i> Chuyuan Li, Patrick Huber, Wen Xiao, Maxime Amblard, Chloe Braud and Giuseppe Carenini	2562
<i>Relation Extraction with Weighted Contrastive Pre-training on Distant Supervision</i> Zhen Wan, Fei Cheng, Qianying Liu, Zhuoyuan Mao, Haiyue Song and Sadao Kurohashi . .	2580
<i>CK-Transformer: Commonsense Knowledge Enhanced Transformers for Referring Expression Comprehension</i> Zhi Zhang, Helen Yannakoudakis, Xiantong Zhen and Ekaterina Shutova	2586
<i>Curricular Next Conversation Prediction Pretraining for Transcript Segmentation</i> Anvesh Rao Vijjini, Hanieh Deilamsalehy, Franck Deroncourt and Snigdha Chaturvedi . . .	2597

Using Punctuation as an Adversarial Attack on Deep Learning Based NLP Systems: An Empirical Study

Brian Formento^{1,2}, Chuan Sheng Foo^{2,3}, Luu Anh Tuan⁴ and See Kiong Ng¹

¹Institute of Data Science, National University of Singapore

²Institute for Infocomm Research, A*STAR

³Centre for Frontier AI Research, A*STAR

⁴Nanyang Technological University

brian.formento@u.nus.edu, foo_chuan_sheng@i2r.astar.edu.sg

Abstract

This work empirically investigates punctuation insertions as adversarial attacks on NLP systems. Data from experiments on three tasks, five datasets, and six models with four attacks show that punctuation insertions, when limited to a few symbols (apostrophes and hyphens), are a superior attack vector compared to character insertions due to 1) a lower after-attack accuracy ($A_{aft-atk}$) than alphabetical character insertions; 2) higher semantic similarity between the resulting and original texts; and 3) a resulting text that is easier and faster to read as assessed with the Test of Word Reading Efficiency (TOWRE)). The tests also indicate that 4) grammar checking does not mitigate punctuation insertions and 5) punctuation insertions outperform word-level attacks in settings with a limited number of word synonyms and queries to the victim’s model. Our findings indicate that inserting a few punctuation types that result in easy-to-read samples is a general attack mechanism. In light of this threat, we assess the impact of punctuation insertions, potential mitigations, the mitigation’s tradeoffs, punctuation insertion’s worst-case scenarios and summarize our findings in a qualitative casual map, so that developers can design safer, more secure systems.

1 Introduction

The goal of an attack is to disrupt a natural language processing (NLP) model’s classification accuracy. The motivation behind researching these adversarial attacks is to create a toolbox of methods to attack systems while also pointing out flaws to improve the models’ robustness. Previous work on adversarial research showed that deep learning-based NLP models are sensitive to slight changes in the input (Ebrahimi et al., 2018) such as character perturbations or word substitutions. However, these attack vectors have three major flaws: 1) letter perturbations can be detected by grammar checkers; 2) these attacks may change the meaning or, worse,

the human label of the sentence (e.g., ‘she’ to ‘he’ with a character deletion for a gender classification system (Zang et al., 2020)); 3) they can make a sample unreadable. Although word-level attacks that change words to a perturbing synonym make these perturbations almost invisible to humans, the cumulative effect of multiple synonym substitutions in a sentence can make the sample harder to understand. Furthermore, the attack must find perturbing word synonyms when attacking samples in a specific domain, such as biology or law, which may be challenging if the algorithm uses general word embeddings with no domain knowledge.

Punctuation insertions, on the other hand, may be a feasible attack vector that is unaffected by the limitations of character perturbations/word substitutions, since it is hard for grammar checkers to detect punctuation (Section 5.5) while also not drastically changing the meaning of the sentence (Sections 5.7, 5.8). Removing punctuation causes deep learning models to perform worse (Ek et al., 2020), as punctuation contains critical information that models require to function correctly (Jones, 1994). Furthermore, punctuation can hold adversarial downstream information (Formento et al., 2021) that may be exploited by malicious users. Punctuation attacks remain an understudied area: Previous works on the topic (Hosseini et al., 2017; Eger and Benz, 2020; Formento et al., 2021) only casually explored punctuation and ignored whether it can generalize or show which punctuation symbols are best suited for intrusion attacks.

Contributions: Through extensive empirical studies, we have determined that punctuation insertions can outperform, in terms of $A_{aft-atk}$, alphabetical character insertions (as shown in Section 5.1) and, under certain conditions, word substitution (5.2), allowing for a user-controllable tradeoff between after-attack accuracy ($A_{aft-atk}$), sample quality, and attack time efficiency when used together in a multi-level attack (5.3). Specifically,

hyphen (Hy) and apostrophe (Ap) insertions are the most effective at avoiding straightforward defense mechanisms (as shown in Sections 5.4, 5.5) while preserving the original meaning, as evidenced by achieving 100% semantic similarity in our tests (as shown in Section 5.7). Additionally, by using the TOWRE test, we have demonstrated that inserting only one punctuation type significantly increases attack readability by increasing reading speeds by 800% compared to character insertions, and 96.8% compared to using multiple types of punctuation (as shown in Section 5.8) without compromising the attack performance (Section 5.9). To aid in the understanding of our findings, we have also introduced a casual map in Figure 5.

2 Related Work

Adversarial attacks on NLP systems can be categorized in terms of the level of granularity of the perturbation. **Character-level** attacks (Ebrahimi et al., 2018; Eger and Benz, 2020; Eger et al., 2019; Belinkov and Bisk, 2018; Sun et al., 2020; Boucher et al., 2021) modify individual characters in words to force the tokenizer to process multiple unrelated embeddings instead of the original, resulting in decreased performance. **Word-level** attacks (Jin et al., 2020; Li et al., 2020; Maheshwary et al., 2020) employ a search algorithm to locate useful perturbing embeddings (Jin et al., 2020; Li et al., 2020; Maheshwary et al., 2020) or operations (Tan et al., 2020; Li et al., 2021) that are clustered close to the candidate attack word’s embedding given a similarity constraint (such as the Universal Sentence Encoder (Cer et al., 2018)). **Multi-level** attacks combine multiple types of perturbations, making the attack cumulative. Textbugger (Li et al., 2019), which uses both character-level and word-level attacks, is an example of a multi-level attack.

Although previous research has investigated character and word-level attacks, few have studied the use of punctuation attacks. To the best of our knowledge, only Zéroe (Eger and Benz, 2020), Perspective Atk (Hosseini et al., 2017) and SSTA (Formento et al., 2021) have researched punctuation as an attack vector. While the former two randomly insert symbols within a word, the third revealed that symbols contain adversarial information and can be inserted as padding with little further optimization. Zéroe, in particular, is a benchmark of ten different character attacks. Out of these ten, Zéroe Intrude is the only one focusing on punctua-

tion and is thus used as one of the gold standards in this paper.

Our work builds on these previous works by further exploring Zéroe Intrude and the concept, introduced initially in SSTA, that model-specific symbols can attack binary classifiers when used as padding. Our work contributes to the discussion on punctuation symbols being a general mechanism to attack deep learning models while also improving readability through the novel use of the TOWRE metric, which tracks how quickly someone can read the adversarial text.

3 Methodology

3.1 Overview

Suppose we have a sequence classifier $f : \mathcal{X} \mapsto \mathcal{Y}$, that takes an input sequence of words $x = (\tau_1, \dots, \tau_n) \in \mathcal{X}$ with ground truth label y and outputs a prediction $\hat{y} = f(x)$. An adversarial attack on input x and classifier f would perturb τ , for example, using character manipulations or word substitutions, to produce a new adversarial sample \hat{x} that is misclassified by f such that $f(\hat{x}) \neq y$.

We investigate punctuation and multi-level attacks in gray-box and black-box settings. Specifically, we explore the effects of inserting punctuation when the victim’s model classification logit is leaked (**gray-box**) and when it is not (**black-box**). We use a variation of DeepWordBug (DWB) and the original Zéroe Intrude (ZI) attack in these settings. In addition, we combine punctuation together with word substitutions in a gray-box setting (**multi-level**) to evaluate if punctuation can augment word-level attacks. We provide a more detailed description of the respective attacks used in the following sections.

3.2 Attack foundations and baselines

We build upon and compare our results to the following four attack baselines: 1) *Zéroe Intrude (ZI)*, a simple black-box attack (see Section 3.5 (Eger and Benz, 2020)); 2) *DeepWordBug (DWB)*, which uses four-character level perturbations including delete, swap, insert, and nearby character swap (Gao et al., 2018); 3) *TextFooler*, a popular baseline that uses word synonyms from counterfeited embeddings to perturb the sample perturbation (Jin et al., 2020); and 4) *SememePSO*, a recent method that uses a seme (e.g., a morpheme) to create a word substitution together with PSO (Zang et al., 2020).

3.3 Gray-box punctuation attack

As a representative gray-box punctuation attack, we implement a variant of DWB through the TextAttack framework that performs only punctuation insertions instead of alphabetical insertions, swaps, deletions, and substitutions. We denote this punctuation variant as DeepWordBugPunc (DWBP). DWBP has three main steps:

- Step 1: Determine the essential words with set $\tau_R = \{\tau_1 \dots \tau_k\}$ for an NLP model f using a word delete schema, ranking them from highest to lowest in terms of output logit change. A delete schema, popularized by BERT-Attack (Li et al., 2020), analyzes the logit change when a word is removed from a sample.
- Step 2: Use user-defined set γ (e.g. $\gamma = \{-'\}$) and the RPos (Random Position) and RPunc (Random Punctuation) flags to return a set of transformations $\{\tau_k\}$ from highest-ranking word τ_k from Step 1.
- Step 3: Search over the attack space by querying the victim’s model with samples modified with the transformations from Step 2. Keep the best transformation with regard to the logit and semantic similarity score. The next word from τ_R is then perturbed. This is repeated until either $f(x) \neq f(\hat{x})$ or the algorithm iterates through τ_R . This process is called Greedy Search with Word Replacement (GSRW).

In summary, for a sample x , the algorithm identifies the top words in τ_R . It gradually modifies them by inserting one punctuation symbol and making calls to the victim’s model through the GSRW Algorithm 1. Optimizing over τ_R results in GSRW being a time-efficient query alternative to the greedy search algorithm. It gradually replaces τ_R in x with transformations from Step 2 by calling Algorithm 2.

Algorithm 2 takes a word and decides the location and punctuation type to insert with the **RPos** and **RPunc** flags. These two flags, when set to false, allow the algorithm to explore the entire attack space. This in turn creates many transformation variations with γ , therefore allowing GSRW to check the adversarial performance of each symbol in γ at each position within the word τ_k . GSRW keeps the transformation if the change creates a successful reduction in logit score. After an adversarial candidate \hat{x} is found, the semantic similarity

between x and \hat{x} with $S' = Sim(x, \hat{x})$ is calculated with a deep learning model (Cer et al., 2018). GSRW will reject all perturbations that miss a semantic similarity threshold, set at 0.8, which ensures a good tradeoff between sample quality and adversarial strength (Li et al., 2019). The algorithm repeats this procedure until the end condition.

The difference between DWBP and DWB is that DWB transforms a word with a composition of transformations (letter substitution, deletion, swap, or insertion), and all the transformed words are added to $\{\hat{\tau}_k\}$. Appendix D.1 gives an extended description for the three steps. We tested all variants of RPos and RPunc when applicable.

Algorithm 1 τ_k Transform Function with GSRW

Input: Word ranking τ_R , Sample x , Symbols γ

Output: Adversarial sentence \hat{x}

```

1: Initialize  $\hat{x} = x$ 
2: for each  $\tau_k$  in  $\tau_R$  do
3:   if  $\text{len}(\tau_k) < 2$  or  $\tau_k = \text{Stop-Word}$  then
4:     skip
5:   else
6:     Transformations Set  $\{\hat{\tau}_k\} = TF\gamma(x(\tau_k), \gamma)$ 
7:     for  $\hat{\tau}_k$  in Transformations Set  $\{\hat{\tau}_k\}$  do
8:        $\hat{x} \leftarrow \hat{\tau}_k$ 
9:        $\hat{x}^{Adv}, \hat{x}^{Score} = f(\hat{x})$ 
10:      if Perturbation successful then
11:        Keep best  $\hat{\tau}_k$ 
12:      else
13:        Don't keep change  $\rightarrow$  next word
14: return  $\hat{x}$ 

```

Algorithm 2 Step2: τ_k Transform Function TF

Input: Word τ_k , Symbols γ , Bool: RPos/ RPunc

Output: Adversarial word $\hat{\tau}_k$

```

1: Transformations =  $\emptyset$ 
2: if RPos then
3:   if RPunc then
4:      $i = \text{RandInt}(\text{Start}_{Idx}, \text{End}_{Idx})$ 
5:     Transformations  $\leftarrow \hat{\tau}_k = \tau_k[:i] + \gamma_{random} + \tau_k[i:]$ 
6:   else
7:      $i = \text{RandInt}(\text{Start}_{Idx}, \text{End}_{Idx})$ 
8:     for  $j$  in  $\gamma$  do
9:       Transformations  $\leftarrow \hat{\tau}_k = \tau_k[:i] + \gamma_j + \tau_k[i:]$ 
10:  else
11:    if RPunc then
12:      for  $i$  in  $|\text{Start}_{Idx} - \text{End}_{Idx}|$  do
13:        Transformations  $\leftarrow \hat{\tau}_k = \tau_k[:i] + \gamma_{random} + \tau_k[i:]$ 
14:    else
15:      for  $i$  in  $|\text{Start}_{Idx} - \text{End}_{Idx}|$  do
16:        for  $j$  in  $\gamma$  do
17:          Transformations  $\leftarrow \hat{\tau}_k = \tau_k[:i] + \gamma_j + \tau_k[i:]$ 
18: Return Transformations

```

3.4 Gray-box multi-level attack

We also evaluated the performance of punctuation insertions when used in conjunction with word-level attacks. To conduct this assessment, we employed two baselines TextFooler and SememePSO.

- TextFooler/DWBP: This variant uses the same word scoring function and the GSWR search algorithm. However, $\hat{\tau}_k$ will be a mix of word synonym and punctuation insertion transformations of τ_k .
- SememePSO/DWBP: This variant uses the same word scoring function but with particle swarm optimization (PSO) as a search technique. PSO uses a population-based evolutionary algorithm that exploits the interactions between individuals in a population to find a solution in a search space. $\hat{\tau}_k$ will be a mix of sememes (a type of word substitution) and punctuation insertion transformations of τ_k .

See Section D.5 in the Appendix for details on TextFooler/DWBP and SememePSO/DWBP.

3.5 Black-box punctuation attack

As a representative black-box attack, we implement a variant of the ZI algorithm instead of DWB, as the latter requires access to logits that are absent in this setting. ZI is a simple black-box attack that randomly perturbs a word in a sample with probability p . It then adds a random symbol from this list—!@#\$%^&'()*+,-./:;<=>?@[\\]^_{}|—between two letters with the same probability p , which we define as baseline ZI. In our variant, ZI perturbs a word with probability p (defined as ZIP) but uses the same predefined symbol.

4 Experimental Setup

4.1 Backbone models and tasks

We evaluated the *BERT* (Devlin et al., 2019), *RoBERTa* (Liu et al., 2019), *XLNet* (Yang et al., 2019), *DistilBERT* (Sanh et al., 2020) models on classification (*MR*), entailment (*MNLI*, *SNLI*), and question answering (*QNLI*, *QQP*) tasks. We also used a *CNN* and *LSTM* for *MR* (details are provided in Appendix C).

4.2 Evaluation metrics

We use the evaluation framework previously proposed in (Morris et al., 2020), where an evaluation set is perturbed and out of the Total At-

tacked Samples (*TAS*) set the Number of Successful Attacks ($N_{succ-atk}$), Number of Failed Attacks ($N_{fail-atk}$) and Number of Skipped Attacks ($N_{skp-atk}$) are recorded. *After attack accuracy* ($A_{aft-atk} = \frac{N_{fail-atk}}{TAS}$), the most important metric, represents how well the attacker can fool the model across a dataset. Lower values of $A_{aft-atk}$ indicate that the attacker can fool the model better. *After success rate* ($A_{succ-rte} = \frac{N_{succ-atk}}{TAS - N_{skp-atk}}$), is similar to $A_{aft-atk}$ but ignores previously misclassified samples. *Percentage of perturbed words* refers to the percentage of words the algorithm perturbs out of the number of words in the sample. This metric should be as low as possible, as perturbing more words makes the sample’s perturbation more detectable. *Semantic similarity* (Jin et al., 2020; Maheshwary et al., 2020) is an automatic similarity index that describes the visual difference between two samples using a deep learning model. In this case, the Universal Sentence Encoder (Cer et al., 2018) is used, along with a cosine similarity measure between the output embeddings. A value of 1 indicates that the two inputs are semantically equivalent, while 0 represents no similarity. *Average number of queries* represents the number of times the algorithm must invoke the model to perform inference. This metric should be kept low to avoid detection.

4.3 Human evaluation

To evaluate the quality of adversarial samples, we conducted four human studies. The first three are the same tests used in TextFooler (Jin et al., 2020), and Hard-Label (Maheshwary et al., 2020). These tests analyzed the adversarial sample for 1) *grammatical correctness*, where reviewers rate the grammatical correctness of the original and adversarial samples on a scale from 1–5, where 1: many grammatical mistakes and 5: no grammatical mistakes; 2) *reviewer classification accuracy*, where reviewers predict the label of each sample; and 3) *similarity*, where reviewers rate if the two samples are similar (1), dissimilar (0), or ambiguous (0.5); 4) *readability*, where the novel application of TOWRE (Tarar et al., 2015) was used to analyze the quality of adversarial words in character-level black-box attacks. TOWRE is a widely used test that measures an individual’s reading accuracy and speed. We adapted TOWRE to record the quality of adversarial examples. Specifically, the reviewer pronounces a list of words, where each word was modified with

one out of four different perturbation types introduced with the ZI algorithm. We record the words per minute (WPM) and error rates. All tests had two reviewers who reviewed 100 samples in the first three tests and 36 in the fourth. Agreement between the reviewers was assessed with Krippendorff’s alpha, where a score of 1 indicates complete agreement and -1 indicates complete disagreement. Further implementation details on the human testing method and details on Krippendorff’s alpha are in Appendix F.

4.4 Defense Baselines

We evaluate fine-tuning and adversarial training as baseline defenses. In detail, it is possible to remove all punctuation during training, fine-tune the model for further epochs on this new punctuationless dataset, and at inference, always strip all punctuation (Table 18). We also experimented with adversarial training (Table 15 in Section 5.6) by using a standard technique (Morris et al., 2020) that is further described in Appendix E.3.

5 Experiments

We use the methodology in Section 3 and experimental setup to explore how punctuation insertions compare to character manipulations (Section 5.1). In Section 5.4, 5.5 and 5.6 we demonstrate how straightforward defence techniques fail and succeed and Sections 5.7, 5.8, and 5.9 highlight the advantages of punctuation insertions where no defence technique is present. The γ choices for each test are summarized and justified in Appendix E.2.

5.1 Punctuation vs character manipulations

How does an attack change when using punctuation insertions instead of letter manipulations? Punctuation insertions can degrade NLP model performance while preserving semantic similarity. The system’s $A_{aft-atk}$ is overall reduced (see Figure 1) while semantic similarity remains at 0.96–1.00 when using punctuation insertions (DWBP) compared to 0.87–0.90 when using DWB. Each DWBP box represents the $A_{aft-atk}$ for a dataset across all models with RPos = False. The lower the $A_{aft-atk}$ the more perturbing the attack.

Hyphen, apostrophe, full stop, or comma insertions lower $A_{aft-atk}$ more than any other letter in the alphabet (Figure 2). Values in Figure 2 reflect the after-attack difference [%] between using a letter or punctuation type in an intrusion attack.

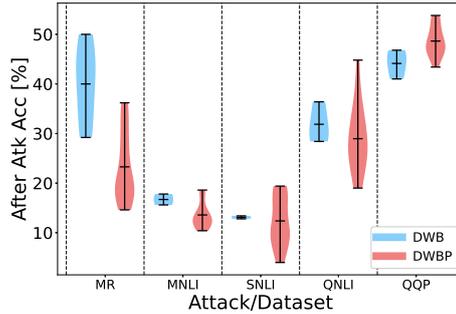


Figure 1: Gray box attack performance

Green/positive values represent an improvement and purple/negative values a decrease between the punctuation symbol on the x-axis and the letters on the y-axis when using DWBP. Each attack in Figure 2 has a constant number of queries, [%] of perturbed words, and query time. The extended results are in Appendix L.

Observation This experiment clarifies that if **any** internal punctuation is present, the system **is vulnerable** and that it is more susceptible to such insertions than other character manipulations and alphabet insertions. We limit our reporting to BERT on MR because other model results are consistent. Full tabular results for other Models and datasets for Figure 1 are in Appendix I in Tables 6 and 7 (“Without Grammar”). While Figure 2 has the other model’s results in Appendix L.

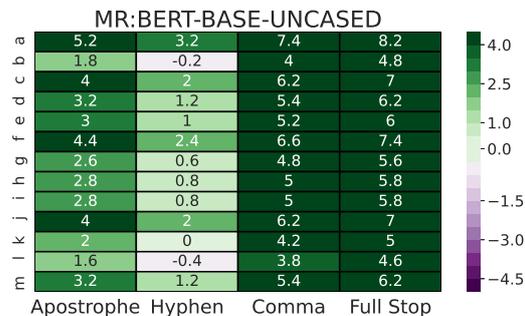


Figure 2: Punctuation vs character insertions. Green indicates positive values; purple indicates negative values

5.2 Punctuation vs word-level attacks

Are there advantages in using punctuation insertions instead of word substitutions? DWBP can also be compared to TextFooler since, with DWBP, a punctuation symbol is mapped to an embedding when using a word piece tokenizer. Figures 3 and 4 each show increasing numbers of unique punctuation symbols from γ (DWBP) or synonyms per word (TextFooler), ranging from 1 to 10. For

DWBP, γ is set to . for $N = 1$, . for $N = 2$, up to . for $N = 10$. In TextFooler, N represents the number of synonyms per word. Figure 3 displays the relationship between N (represented by the points and the x-axis) and improvement in $A_{succ-rte}$ (y-axis). Figure 4 displays the relationship between N (represented by the points), number of queries (x-axis), and the effect on $A_{aft-atk}$ (y-axis). Both experiments used all variations of RPunc/RPos on BERT-MR.

Observation The effectiveness of punctuation insertions is demonstrated DWBP when constrained on N and queries, as seen by the higher $A_{succ-rte}$ with low N in Figure 3 and the low $A_{aft-atk}$ with few queries in Figure 4. Similar results for MNLI can be found in Appendix H.

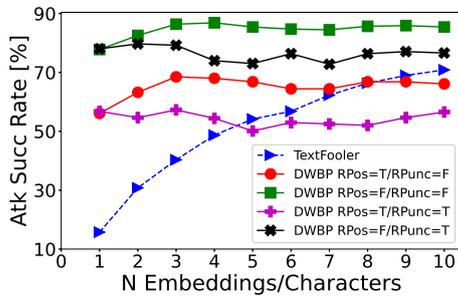


Figure 3: Punctuation embedding efficiency.

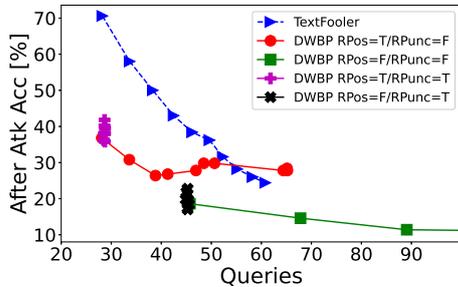


Figure 4: Punctuation query efficiency.

5.3 Punctuation as a multi-level attack

We investigate a composite experiment where τ_k is composed of word substitutions and punctuation insertions. The methodology is introduced in Section 3.4 and the details are given in Appendix D.5. We set RPunc = False, RPos = False, and $\gamma = -$.

Observation: The results in Table 1 indicate that incorporating punctuation insertions into the optimization process enhances TextFooler and SememePSO on BERT trained on MR. The additional findings in Section G.1 of the appendix present re-

sults for all tasks and models, and provide further observations.

Dataset	Model (Orig Acc)	Method	After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Avg Time Taken [s]	Avg Number Queries
MR	BERT (83.8)	DWBP	17.4	18.32	1	0.721	74.7
		TextFooler	9.4	17.54	0.82	1.3072	118.5
		TextFooler/DWBP	7.6	18.31	0.89	1.122	105.35
		SememePSO	7	16.52	0.81	16.1811	4950.71
		SememePSO/DWBP	6	9.99	0.89	7.3252	988.44

Table 1: Multi-level DWBP. Full results in Appendix J

5.4 Removing punctuation as a defense

How does removing punctuation perform as a defense? In this section, we sought to evaluate the effectiveness of simple defenses in countering punctuation attacks by examining the impact of various forms of punctuation removal on attack performance. To aid in this assessment, we employed the use of a casual map in Figure 5, which allows for tracking of the defender’s behavior in response to the attacker’s changing strategy.

The casual map, presented in the blue quadrant, begins with the "Base Model" on the right-hand side, representing the unchanged finetuned model from Hugging Face, in this instance, specifically BERT finetuned on MR. Adjacent to this model is a large red table, which represents the significant performance drop when utilizing punctuation insertions. For the sake of simplicity, in this map, we limited ourselves to the use of full stops (FS), commas (Co), which are common external punctuation types, and apostrophes (Ap) and hyphens (Hy), which are common internal punctuation types. Given this threat, we identified and explored three options for the defender to take. Beneath the "Base Model," the first option is to remove all punctuation ("All"), which secures the system but leads to an original performance drop of -2.6%. The second option, just beneath "All" is to remove all punctuation found inside of words. While this approach solves the problem, it becomes challenging to identify if a punctuation was inserted by mistake by a user or to prevent the attacker from inserting a whitespace before or after the punctuation insertion. If the attacker adds a whitespace, the attack defaults to the large red table. Furthermore, removing all internal punctuation has a noticeable original performance drop of -1.2%. An alternative to this is to remove all internal punctuation but make an exception for Hy and Ap, reducing the original performance drop to 0%, however, the system remains vulnerable to Ap and Hy. Given the persistent vulnerability to Ap and Hy, the defender may employ a grammar checker to reject all

samples that do not meet a certain grammatical correctness level. When implemented, the robustness of the model increases dramatically, resulting in a semi-secure model.

The blue quadrant also shows "Finetune with punctuation." This base model was further trained and then compared to further training the model when all the punctuation is removed (See "All + Finetune"). As previously highlighted, removing all punctuation can secure the system, the reduced performance drop of -0.6% now indicates that this approach has less of a trade-off between securing the system and original accuracy drop.

In addition, we also explored adversarial training. We discuss the findings of this quadrant in Section 5.6 and it's experimental setup in Section E.3 in the Appendix.

Observation Using a grammar checker increases the robustness of this task. However, as pointed out in the next section in Figure 6, the red candlesticks representing DWBP have a large attack variance depending on the dataset, symbol used, and model. Hence, for a task that results in a semi-secure model, another task may result in a semi-broken model. This reasoning also applies to black box punctuation attacks with ZIP, as pointed out by the large variance in the red candlesticks in Figure 7. Another aspect to consider is the original accuracy drop in performance experienced in the yellow boxes. Depending on the application, this may be acceptable/negligible or unacceptable/too high.

5.5 Grammar checkers as a defense

If a grammar checker preprocesses an input, how does the attack performance change? Another common idea is that character-level attacks are easy to defend against using a grammar checker (Zang et al., 2020). Although adding a grammar checker before processing the input lowers the effectiveness of the attack, punctuation is nonetheless a successful insertion technique with RPos = False, particularly when compared to DWB (Figure 6). Punctuation insertions are also effective in black-box settings (ZIP) and are **as competitive** as alphabetical character manipulations in gray-box settings (DWB) (Figure 7). The high variance of ZIP means that inserting some symbols can lower performance comparably, if not more than any character manipulation technique introduced in DWB. For example, ZIP Ap achieves a 7.8% lower $A_{aft-atk}$ than DWB.

The full results can be found in Appendix I (column "With Grammar" in Tables 6 and 7, and the performance of ZIP in Table 10).

Observation DWBP is more successful with the attack, except on the [%] of perturbed words. These results show a curious property of punctuation attacks by highlighting that the [%] of perturbed words is not necessarily aligned with semantic similarity. Therefore, it is possible to have a highly perturbed sample (in terms of [%] of perturbed words) that is nonetheless readable and potentially preserves the original information.

5.6 Adversarial training as a defense

How does adversarial training benefit learning? In this section, we aimed to robustify the model by experimenting with adversarial training on the MR dataset. To test this, we employed the use of the DWBP with hyphens and apostrophes (Hy and Ap). Our findings suggest that adversarial training for language models improves $A_{aft-atk}$. Specifically, $A_{aft-atk}$ increased by 7.4% with Hy and 6.4% with Ap on BERT, as shown in Figure 5. This is demonstrated in the "Adv Training" quadrant, where this model was further finetuned for 4 epochs on the base dataset, while the models beneath it were trained for 4 epochs where the base dataset was extended by 20% with adversarial samples containing either apostrophes or hyphens. The effects of adversarial training were minimal, but did result in an improvement to the model not undergoing any adversarial training. This can be observed by comparing the values in the Broken model to the left of "Adv Training" and to the Broken models that have been adversarially trained beneath "Adv Training". On LSTM, $A_{aft-atk}$ increased by 2.4% with Hy and 1.6% with Ap, with negligible drops/increases in original accuracy, as shown in Table 15 in the appendix.

Observation Our findings are in agreement with previous works, which highlight that adversarial training on large language models, such as BERT or LSTMs, can improve both original and adversarial accuracy (Zhu et al., 2020; Miyato et al., 2017; Cheng et al., 2019; Yoo and Qi, 2021). However, other studies suggest that robustness and generalization may be at odds with one another (Li et al., 2021; Eger and Benz, 2020; Meng and Wattenhofer, 2020). Our experiments also indicate that although adversarial training improves the $A_{aft-atk}$, there is still a large drop in performance.

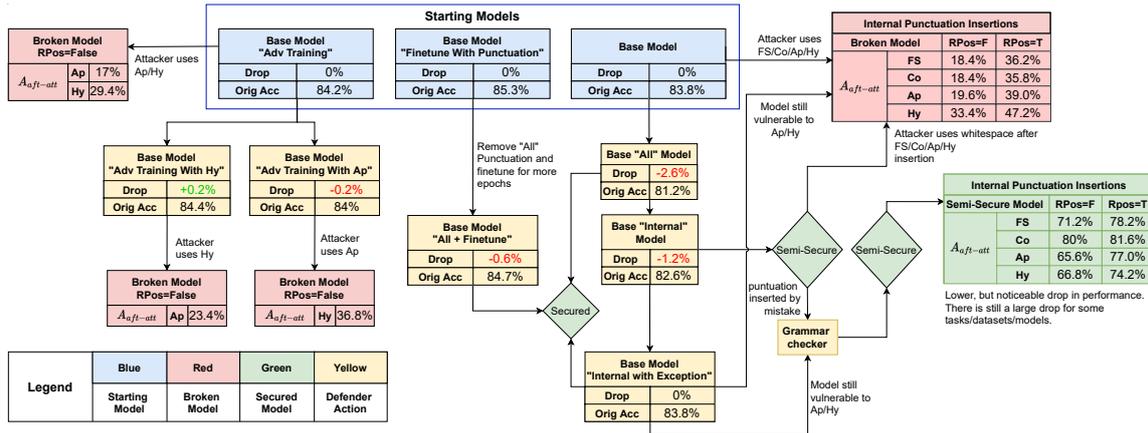


Figure 5: Qualitative casual map for defender/attacker strategy (Section 5.4), values represent BERT-MR.

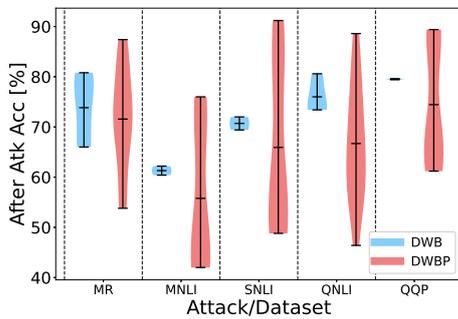


Figure 6: Gray-box attacks against grammar checker

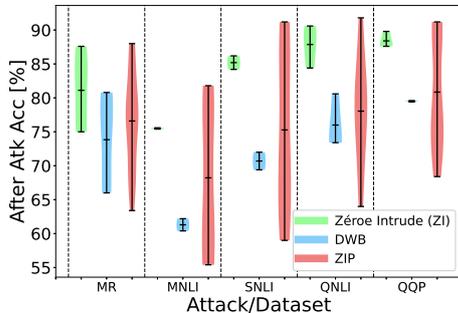


Figure 7: Black-box attacks against grammar checker

5.7 Semantic similarity of punctuation attacks

How similar are samples that have been perturbed with punctuation to the originals? Earlier tests concluded that removing punctuation is an impractical defense technique. We now evaluate the perturbation quality. Apostrophe and hyphen insertions attained a perfect score of 1 for similarity across all samples (in both human and automatic evaluations), a 98% on reviewer classification accuracy (nominal Krippendorff’s alpha = 0.960), and a grammatical correctness score difference of 1.29 between the original samples (3.14/5) and adversarial sam-

ples (4.43/5), with ordinal Krippendorff’s alphas of 0.459 and -0.004 for the original and adversarial samples, respectively. We provide qualitative examples in Table 2 to highlight how the sample changes with punctuation insertions.

MR	A dark comedy that goes for sick and demented (Negative Sentiment)
TextFooler	A dark comedy that goes for psychopathic and coot humor honestly to do so . the film is without object .
DWBP	A dark comedy that goes for sick and demented humor simply to do so . the movie is withou’t intent .

Table 2: Qualitative examples of DWBP and TextFooler. **Bold** words represent a perturbed word

Observation The grammar test is widely used (Jin et al., 2020; Maheshwary et al., 2020). However, the low Krippendorff’s alphas for grammatical correctness suggest the low reliability of the test in indicating grammatical correctness. Analyzing the visual effect of inserting punctuation makes it possible to observe that the semantics remained unchanged. However, such changes are very noticeable to a human (Table 2).

5.8 Limiting punctuation and readability

Does limiting the punctuation types improve readability? Our tests suggest that focusing on a few types of punctuation facilitate meaning preservation (Section 5.7). Another reason to limit punctuation insertions is to improve readability. To test readability, we used TOWRE, where a reviewer pronounces a list of words with four different perturbations in the test using the Zéro algorithm with $p = 0.8$ (high perturbation strength). The four types

are: 1) no perturbation (original); 2) ZIP with apostrophe (ZIP Ap), 3) ZI; and 4) character insertions. ZI uses all punctuation symbols from Section 3.5 and character insertions uses all alphabetic characters. Our ZIP Ap method has the fastest reading speeds. Specifically, Table 4 shows an improvement in WPM from 7 WPM (character insertion) to 32 WPM (ZI) to 63 WPM (ZIP Ap), with a ratio Krippendorff’s alpha of 0.977 and a consistent error rate reduction from character insertions (71.43) to ZI (6.17) to ZIP Ap insertions (1.43). In terms of reading speeds, ZIP Ap is an improvement over character insertions by 800% and by 96% over ZI.

Observation Compared to character insertions and ZI, apostrophe insertions by ZIP Ap are easier and faster to read, as seen with the perfect semantic similarity, WPM improvement, and error reduction. We also show, for the first time, that an attacker cannot use alphabetical character insertions in a high perturbation black-box setting as the samples become too scrambled (Table 3).

MNLI		
Original	Premise	Not only that but they don't pay the money either
	Hypothesis	They also do not contribute financially.
Character Insertions	Hypothesis	Th zezy also do not contribute fdiunlavnyckiawulvlywv .
Zéro	Hypothesis	Th jely also do not contribute
Intrude	Hypothesis	fj i^ n>a l n(c-i a) *l{y}' .
Insertions (Full Stop)	Hypothesis	Th .e.y also do not contribute f.i.n.a.n.c.i.a.l.l.y.

Table 3: Qualitative examples of FS Insertions vs ZI vs Ch. **Bold** words represent a perturbed word

TOWRE	Method			
	Original	ZIP Ap	Zéro Intrude	Character Insertions
Time [s]	24.54	33.60	60.00	60.00
WPM	86.64±9.6	63.35±7.3	32.50±1.5	7.00±0
Errors	0.00±0	0.50±0.5	2.00±0	5.00±1
Error Rate [%]	0.00±0	1.43±1.4	6.17±0.2	71.43±14.3
Self-Corrections	1.00±0	0.50±0	1.50±0	0.50±0.5
Self-Correction Rate [%]	2.86±0	1.43±1.4	4.55±1.3	7.14±7.1

Table 4: Reading efficiency for the four perturbations

5.9 Limiting punctuation types

Is the attack still effective when using a limited punctuation set? Despite limiting the types of punctuation, ZIP performs similarly to ZI and better than character insertions (Figure 8). The test in Figure 8 explores the ability of ZIP with Ap (apostrophe), Hy (hyphen), Co (comma), and FS (full stop) insertions to generalize to black-box attacks. We compare these ZIP intrusions to ZI with all punctuation types and character insertions (Ch) with all alphabet letters using the ZI algorithm on MR-BERT.

Figure 8 shows the delta change in $A_{aft-atk}$ for each attack technique against the others for $p = 0.8$. Each square represents the $A_{aft-atk}$ from the x-axis attack method minus the $A_{aft-atk}$ from the y-axis attack method. Table 14 in the Appendix displays the $A_{aft-atk}$ [%] and semantic similarity (S) values for Figure 8. Limiting punctuation with ZIP also avoids grammar checking better than using all punctuation types with ZI (Figure 7) as ZIP can focus on one highly perturbing symbol.

Observation ZIP Ap achieved comparable results to that of Ch and ZI (Figure 8) where the difference is even smaller when comparing with other models. Using character insertions can thus be deemed counterproductive and should be avoided. Attacks should instead focus on only one punctuation type, such as Ap, since, compared to Ch and ZI as Section 5.8 highlighted, readability is preserved.

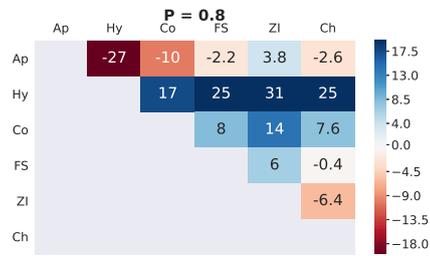


Figure 8: X-Axis $A_{aft-atk}$ minus Y-Axis $A_{aft-atk}$. ZIP (Ap, Hy, Co, FS) vs ZI, Ch

6 Conclusion

Researching adversarial attacks aims to create a toolbox to identify flaws and improve model robustness. Results show that punctuation insertions as an attack are more effective than character manipulations (Figure 1) and alphabetical character insertions (Figure 2) and better evade grammar checkers (Figures 6, 7). Punctuation insertions preserve more information and are faster to read (Section 5.7, 5.9). Simple defenses and adversarial training are not necessarily effective (Section 5.4, 5.6). The information-preserving characteristic of this attack could potentially evade censorship. Conversely, this highlights that a system deployed to combat fake news and offensive language propagation can potentially be compromised by this use of punctuation. Our defense findings are summarized in Figure 5. We hope this inspires further research in the under-explored area of punctuation and how to process it. The code is available¹.

¹Provided at [EmpiricalPunctuationInsertionAttacks](#)

7 Limitations

This work considers only classification tasks, which raises questions on whether such punctuation types can generalize to research tasks such as fake news, offensive content detection, and seq-to-seq tasks such as translation. From our experiments, we can conclude that punctuation insertion attacks (DWBP) with one symbol (apostrophe or hyphen), given our evaluation metrics work better in terms of after-attack accuracy, readability, and defense avoidance than alphabetical character insertions. However, We’ve found some limitations and cases where punctuation insertions with apostrophes or hyphens don’t work better than the alternative. For example, ZI with all punctuation symbols can achieve on some datasets and models a lower after-attack accuracy, therefore, a better attack success rate Figure 8 than using ZIP with an apostrophe of 3.8%. This increase in performance, however, has a cost since the sample will be harder to read. We only tested on English language, punctuation insertions on other languages are mostly unexplored.

References

- Yonatan Belinkov and Yonatan Bisk. 2018. [Synthetic and natural noise both break neural machine translation](#). In *Sixth International Conference on Learning Representations*.
- Nicholas P. Boucher, Iliia Shumailov, Ross Anderson, and Nicolas Papernot. 2021. [Bad characters: Imperceptible nlp attacks](#). *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. [Universal sentence encoder for English](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.
- Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. [Robust neural machine translation with doubly adversarial inputs](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. [HotFlip: White-box adversarial examples for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.
- Steffen Eger and Yannik Benz. 2020. [From hero to z’eroe: A benchmark of low-level adversarial attacks](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 786–803, Suzhou, China. Association for Computational Linguistics.
- Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. [Text processing like humans do: Visually attacking and shielding NLP systems](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.
- Adam Ek, Jean-Philippe Bernardy, and Stergios Chatzikyriakidis. 2020. [How does punctuation affect neural models in natural language inference](#). In *Proceedings of the Probability and Meaning Conference (PaM 2020)*, Gothenburg. Association for Computational Linguistics.
- Brian Formento, See-Kiong Ng, and Chuan Sheng Foo. 2021. [Special symbol attacks on nlp](#). *International (Joint) Conference on Neural Networks*.
- Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. [Black-box generation of adversarial text sequences to evade deep learning classifiers](#). In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.
- Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. [Deceiving google’s perspective api built for detecting toxic comments](#). corr abs/1702.08138 (2017). *arXiv preprint arxiv:1702.08138*.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. [Is bert really robust? a strong baseline for natural language attack on text classification and entailment](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025.
- Bernard E. M. Jones. 1994. [Exploring the role of punctuation in parsing natural text](#). In *COLING 1994*

Volume 1: The 15th International Conference on Computational Linguistics, Kyoto, Japan.

- Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. [Contextualized perturbation for textual adversarial attack](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.
- Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. [Textbugger: Generating adversarial text against real-world applications](#). *Proceedings 2019 Network and Distributed System Security Symposium*.
- Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. [BERT-ATTACK: Adversarial attack against BERT using BERT](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2020. [Generating natural language attacks in a hard label black box setting](#). In *AAAI Conference on Artificial Intelligence*.
- Zhao Meng and Roger Wattenhofer. 2020. [A geometry-inspired attack for generating natural language adversarial examples](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6679–6689, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. 2017. [Adversarial training methods for semi-supervised text classification](#). In *International Conference on Learning Representations*.
- John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. [TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. [Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter](#).
- Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. [Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert](#).
- Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020. [It’s morphin’ time! combating linguistic discrimination with inflectional perturbations](#). *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Jessica M. Tarar, Elizabeth B. Meisinger, and Rachel H. Dickens. 2015. Test review: Test of word reading efficiency—second edition (towre-2) by torgesen, j. k., wagner, r. k., & rashotte, c. a. *Canadian Journal of School Psychology*, 30(4):320–326.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*.
- Jin Yong Yoo and Yanjun Qi. 2021. [Towards improving adversarial training of NLP models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. [Word-level textual adversarial attacking as combinatorial optimization](#). *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. [FreeLb: Enhanced adversarial training for natural language understanding](#). In *International Conference on Learning Representations*.

A Future Work

How punctuation attacks can augment an effective adversarial learning schema is still an open question. Our punctuation insertion serve as a foundation for future punctuation manipulation. Hyphens, for example, can be used to s-p-e-l-l out words, syl-la-bi-fi-ca-tion, or to indicate s-stammering or so-so-sobbing in a sentence. There is no research exploring whether stammering or sobbing punctuation perturbations could generate a high-quality adversarial attack on NLP without compromising the meaning.

Exploring whether the identified punctuation types and attacks generalize to more complex prediction tasks like fake news, offensive content detection, and seq-to-seq tasks such as translation is an interesting topic for future work.

B Appendix: Ethics Statement

This research was conducted in accordance with the ACM Code of Ethics.

C Appendix: Task and Datasets

MR: The Rotten Tomatoes movie review dataset holds a sentiment classification task with positive/negative reviews. **QNLI:** Is a question-answering dataset where an annotator extracts the answer from a reference text. The task is to allow the model to predict whether the context sentence holds the answer to the question. **QQP:** Is a duplicate question detection task, where the model is required to detect if the two questions are asking the same thing. **SNLI:** Is composed of human-written sentence pairs where each annotator generates an entailing for each given premise. **MNLI:** It is similar to SNLI but covers multiple genres.

Task	Dataset	Train	Test	Avg Len (Test)	Classes
Sentiment Classification	MR	8.5k	1k	18.7	2
Entailment	MNLI	392k	9.8k	29.2	3
	SNLI	550k	10k	21.4	3
Question Answering	QNLI	105k	5.4k	37.6	2
Duplicate Question	QQP	363k	40k	22.2	2

Table 5: Overview of datasets used in experiments

D Appendix: Methodology Details

Each attack composition has three components, a word scoring function, a set of transformation func-

tions, and a search algorithm.

D.1 Gray-box

D.2 Step 1: Word scoring function

The original DeepWordBug paper introduces four word scoring functions: Replace-1 Score, Temporal Head Score, Temporal Tail Score, and Combination Score. All of which are now outdated. Therefore, for the Gray-Box tests, we use the same schema as that of TextFooler, popularized by BERT-Attack (Li et al., 2020). BERT-Attack records the original sentence’s inference logit. Then for each word in the input, the word is deleted. BERT-Attack then extracts a new logit with the remainder of the sample, tracking the difference in value between the original logit and the new logit for each word. It then regards the words with the most significant output change as the most important to f . The original input sentence x with τ_n words is turned to z_L tokens through a tokenizer function F_t in $z \in (z_1 \dots z_L)$ tokens. To find the set of most important words, which we call $\tau_R = \{\tau_1 \dots \tau_k\}$, that need to be perturbed to attack f , the delete schema associates a rank value R_k for each x_{τ_k} , or sample x without top word τ_k . R_k . Calculate τ_k with:

$$R_i = f(F_t(x))_{x_{Score}} - f(F_t(x_{x \cap x_{\setminus \tau_k}}))_{x_{Score}} \quad (1)$$

where $x_{\setminus \tau_k} = (\tau_1, \dots, \tau_{k-1}, \tau_{k+1}, \dots, \tau_k)$. Thereafter $\forall R_k \in R$ each word τ_i are ranked highest to lowest, resulting in τ_R , x_{Score} represents the output logit from model f .

D.3 Step 2: Transformation

For every top word in step 1, the second step finds ‘transformation’ candidates.

DeepWordBug returns four total candidates. The first candidate has a random letter character inserted in a random position. The second has a random letter deleted. The third has a random letter substituted with another, and the fourth changes the position of two adjacent letters.

DeepWordBugPunc adds punctuation symbols in the sentence to create candidates. The number of candidates depends on γ , RPos, and RPunc. γ is user-specific and is the punctuation types that can be inserted. An example is $\gamma = \{ - ' \}$. With RPos and RPunc It is possible to choose whether to insert a γ_{random} punctuation symbol at a random location, or return candidates for all possible punctuation insertions and position of such insertions.

D.4 Step 3: Optimization

For every word, Algorithm 2 returns a set of transformations. To choose which transformation is best and whether to keep it, we explore Greedy Search with Word Replacement (GSRW). GSRW is a time-efficient query modification applied to the greedy search algorithm. It replaces with a transformation only words strongly correlated with a change in the output when removed from the input. GSRW keeps the transformation if the change creates a successful perturbation. After an adversarial candidate is found the semantic similarity is calculated, with a deep learning model (Cer et al., 2018), between x and \hat{x} with $S' = Sim(x, \hat{x})$. GSRW will reject all perturbations that do not meet a semantic similarity threshold (set at 0.8).

If $|\tau_k| = 1$ or the word is part of a predefined set of stop words, the algorithm does not do the operation. As the algorithm perturbs top words τ_k , it checks for: if the i perturbation was successful at reducing the logit score, if so, the algorithm keeps the perturbation with γ_i , we define this new sample as \hat{x} . This is repeated until either $f(x) \neq f(\hat{x})$ or the algorithm runs out of τ_k .

D.5 Multi-level extension

We also evaluated the performance of punctuation insertions when used in conjunction with word-level attacks. To conduct this assessment, we employed two baselines: 1) TextFooler, a popular method that utilizes word synonyms from counterfeited embeddings to perturb the sample (Jin et al., 2020); and 2) SememePSO, a recent approach that employs a sememe (e.g., a morpheme) to create a word substitution, in conjunction with the use of PSO (Zang et al., 2020).

D.5.1 Gray-box multi-level attack

We explored two multi-level attacks based on TextFooler and SememePSO respectively:

- TextFooler/DWBP: This variant uses the same word scoring function and the GSRW search algorithm. However, $\hat{\tau}_k$ will be a mix of word synonym and punctuation insertion transformations of τ_k .
- SememePSO/DWBP: This variant uses the same word scoring function but with particle swarm optimization (PSO) as a search technique. PSO uses a population-based evolutionary algorithm that exploits the interactions

between individuals in a population to find a solution in a search space. $\hat{\tau}_k$ will be a mix of sememes (a type of word substitution) and punctuation insertion transformations of τ_k .

We performed multi-level attacks to explore their effect on deep learning models. The TextFooler/DWBP and SememePSO/DWBP methods result in $\{\hat{\tau}_k\}$ having both word substitutions and punctuation insertion candidates. For TextFooler/DWBP, TextFooler returns 20-word substitutions, and since RPunc and RPos are both false, DWBP returns K transformations. K is proportionate to the number of letters in the word and the length of γ . In our tests, $\gamma = \{-'\}$. Appendix D.5 gives an extended description for the two multi-level attacks and the TextFooler/SememePSO baselines.

To be clear, although we change SememePSO in the SememePSO/DWBP test and TextFooler in the TextFooler/DWBP, we compare SememePSO/DWBP and TextFooler/DWBP to their **unaltered baselines**.

D.5.2 TextFooler

Where Line 6 returns only TextFooler’s word synonym substitutions. For τ_k , the algorithm will return 50-word substitutions. This baseline uses GSRW.

D.5.3 TextFooler/DWBP

Line 6 in Algorithm 2 is changed to both call TextFooler’s word substitution and DeepWordBugPunc’s punctuation insertion functionality and concatenating the resulting transformations in Transformation Set $\hat{\tau}_k$. For TextFooler/DWBP, TextFooler returns 20-word substitutions, and since RPunc and RPos are both False, DWBP returns N number of transformations. N is proportionate to the number of letters in the word and the length of γ . In our Tests $\gamma = \{ ' - \}$. This baseline uses GSRW. Hyperparameter-wise, we reduce TextFooler/DeepWordBugPunc word embeddings for TextFooler from 50 to 20 on all tasks.

D.5.4 SememePSO

uses word substitutions based on sememes together with a different search algorithm based on particle swarm optimization (PSO). We use an existing implementation of SememePSO from the TextAttack library. PSO exploits a swarm composed of individual samples called particles that interact within a space to find a solution iteratively. Every particle,

which in the case of SememePSO is a sample with a sememe word substitution, has a position in the search space and a velocity. Multiple samples with a sememe word substitution form a swarm. Each particle in the swarm is initialized with a random velocity and position. PSO, after that, records for each particle its own best position in the search space and a global best position. This best position is calculated using an optimization score, which is the victim’s output logit for a classification task. If one of the samples achieves the desired optimization score, the algorithm is terminated since this sample can attack the model. Otherwise, each particle has its position and velocity updated with values from the individual best position, global best position, the inertia weight, two acceleration coefficients, and two random coefficients. The PSO components would replace lines 10-13 in Algorithm 2.

D.5.5 SememePSO/DWBP

uses both sememe word substitutions and punctuation insertions to construct $\{\tau_k\}$ and uses PSO to find the best substitution out of this set. Hyperparameter-wise for SememePSO/DWBP, the attack is changed by reducing the SememePSO population size from 60 to 5 (MR, QNLI) to 2 (MNLI, SNLI, and QQP) and reducing the number of iterations from 20 to 2 for all tasks.

E Appendix: Implementation Details

E.1 Attack detail

All tests were carried out with the TextAttack (Morris et al., 2020) framework to ensure repeatability, standardization, and ease of future integration. The DeepWordBug baseline, for fairness comparison, has a cosine semantic similarity constraint set to 0.8 with (Cer et al., 2018) to ensure the perturbed sample does not differ too much from the original sample and is comparable to other baselines. For TextFooler, SememePSO we keep the default implementation from TextAttack when comparing them with DWBP in Table 1, Tables in Appendix J and Figure 4,3.

For each sample, we keep the After attack accuracy, the number of queries, semantic similarity, and [%] of perturbed words. These metrics are then averaged across 500 samples to complete each test. All data was sourced from the test set of their respective dataset and sampled under a common/standard seed $\in 755$, which is the standard

seed used in the TextAttack framework. For DeepWordBugPunc the tests in Section 5.2 have been conducted on one punctuation symbol and with RPos = False while $\gamma = \{ ' - \}$, RPos = True and RPunc = True for tests in section 5.3.

We used BERT, XLNET, and RoBERTa with 110 million parameters and DistilBERT with 66 million parameters. Every test has been run on a 32GB NVIDIA Tesla V100. The TextFooler and DeepWordBugPunc tests took approximately between 30 min and 1 hour to run, while PSO took between 5 and 10 hours. Regarding the human studies, the participants were not paid and were sourced from a lab at a university. All the participants were made aware verbally of how the data would be used. All scientific artifacts from this paper will be made available on GitHub under an MIT license.

It is possible to keep the perturbed words %, semantic similarity, the average time taken, and the average number of queries to concentrate on changes in $A_{aft-atk}$ by adding a word limit constraint on the % of words perturbed in the input. We use this strategy to construct Figure 2.

E.2 Choice of symbols

The experiments in 5 narrow down a choice for γ . We focus on the most frequent punctuation for each dataset (Table 16 in the Appendix) and find that the distribution of common punctuation is similar across datasets. We therefore use all punctuation for Section 5.1 and 5.5 and the ten most popular symbols for Section 5.2, while the other tests focus on apostrophes, hyphens, commas, and full stops (the two most common internal and non-internal symbols; Figure 2, Section 5.9; Figure 8). We use the results from Sections 5.1 and 5.2 to justify multi-level attacks with apostrophes and hyphens. $\gamma = \{ - ' \}$ is a good choice since they are internal punctuation and create added problems to the defender (see Sections 5.4 and 5.5). The human studies in Sections 5.7 and 5.8 tested $\gamma = \{ - ' \}$ and $\gamma = \{ ' \}$; we did not do human tests on other punctuation insertions as they are visually similar. Nonetheless, we believe the results will be similar regardless of the punctuation type inserted (full stop, comma, apostrophe, or hyphen).

E.3 Adversarial training details

The standard adversarial technique in (Morris et al., 2020; Yoo and Qi, 2021) works by, at each epoch, finding the adversarial sample for each datapoint

(if it exists). It then extends the base dataset by 20% using the adversarial data. For MR, we do fine-tuning and adversarial training for 4 epochs with a batch size of 16 and a learning rate of $2e^{-5}$. We compare this by fine-tuning the same model using the same hyperparameters but on the base dataset.

F Appendix: Human Evaluation Details

F.1 Appendix: Gray-box and multi-level human evaluation

We follow the evaluation strategy used in TextFooler (Jin et al., 2020) and Hard-Label (Maheshwary et al., 2020). Therefore evaluate the quality of the generated samples across three metrics; Grammatical Correctness: Measures in the Likert scale, between 1 and 5. The reviewer compares the adversarial sentence to the grammar of the original as a reference. Classification: Asks the reviewer to classify the sample. We then check if the human classification matches the true label, Similarity: The user inputs a number representing one of three choices where dissimilar is 0, ambiguous 0.5, and 1 similar. The three tests were conducted with two native English-speaking students from India and the UK who have a tertiary university education. They were trained using 3 test samples. We sampled 100 samples at random from MR targeting BERT for this test.

F.2 Appendix: Black-box human evaluation

Finally, we introduce a novel application of TOWRE (Tarar et al., 2015). To generate the word list, we extract all words from the NLTK python package and pick six words randomly for each word length between 4 and 9. The reviewer pronounces 36 words as accurately and fast as possible. The test reports the number of words correctly pronounced, the number of errors, self-corrections, and the time to pronounce the 36 words or the number correctly pronounced in 1 minute. The WPM (Words Per Minute) metric extrapolates from the time or the correct number of words. The reviewers conducting TOWRE are from Singapore and Brazil. Both hold tertiary education. All the tests were conducted in one sitting and took 15 minutes each. To ensure no duplicates existed in the word list, we manually checked the 145 words across the 4 tests and found no duplicates. TOWRE was initially introduced to measure sight word reading fluency. It is widely used in clinical practices to diagnose

dyslexia or reading difficulties in children.

F.3 Krippendorff’s alpha

We use the Krippendorff’s Alpha reliability metric to detect whether a test has statistical significance. Krippendorff’s Alpha extracts a value between -1 and 1 after highlighting the agreement between multiple reviewers in a trial. This metric can calculate statistical reliability for nominal (classification, semantic similarity), ordinal (grammar test), and ratio (WPM) data types. A value close to -1 represents complete disagreement between reviewers normalizing by chance, 0 represents neither statistical agreement nor disagreement, and 1 is perfect agreement.

G Extra Findings

G.1 Punctuation as a multi-level attack

Extra Observation We find an interesting trade-off between $A_{aft-atk}$, sample quality, and attack time efficiency depending on the influence of punctuation insertions over the word-level attacks. Hyperparameter-wise, the changes in Section D.5.1 increase the attack effectiveness of punctuation insertions by decreasing classification accuracy after the attack ($A_{aft-atk}$) while increasing the quality/meaning/readability of the text. These changes are also more efficient compared to other hyperparameters because the number of queries and amount of time taken to optimize the sample are decreased.

Other hyperparameters can achieve lower $A_{aft-atk}$ but at the cost of time, queries, and sample quality. We hypothesize that this interesting behavior derives from punctuation insertions being unconstrained by a similarity constraint. These attacks can inject information from different parts of the embedding space by inserting punctuation and avoiding word substitutions. Analyzing the visual effect of inserting punctuation makes it possible to observe that the semantics remained unchanged. However, such changes are more noticeable than word substitutions (Table 2).

H Appendix: Extended Budget Study

Figure 9 illustrates how the $A_{succ-rte}$ improves as each word in the sample can be either replaced with N synonyms (TextFooler) or have one of N punctuation characters inserted in the word (DWBP) in four different ways according to how the flags RPos/RPunc are set. The behaviour of RPos and

RPunc changes DWBP, as previously explained in Section 3.3.

Increasing the number of word synonyms in TextFooler or potential punctuation symbols in DWBP results in more transformations ($\hat{\tau}_k$) that the GSWR algorithm needs to evaluate by performing queries to the victim model, therefore searching for the optimal transformation. The query response is shown in Figure 10. Both tests suggest that DWBP performs better with limited word synonyms and limited queries from the attacker. On the other hand, TextFooler performs better when the algorithm has many synonym candidates to chose from for each word.

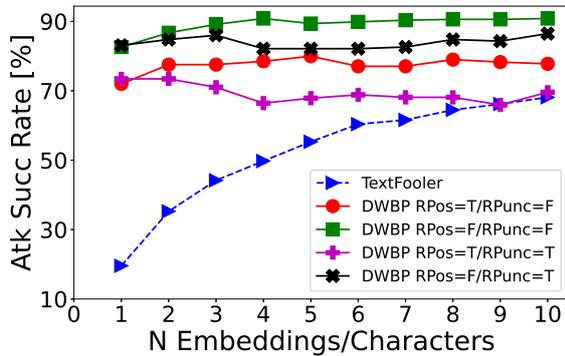


Figure 9: After Success Rate (higher is better) as the number of characters in γ is increased for DWBP vs the number N of synonyms is increased for TextFooler

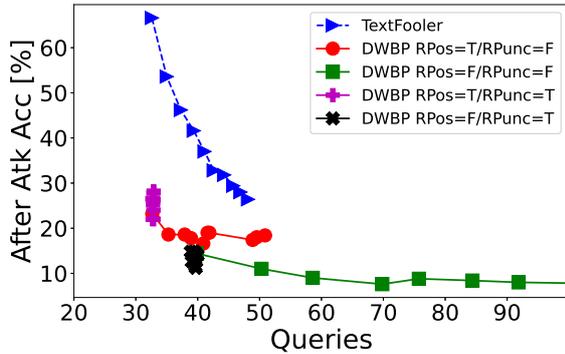


Figure 10: After Attack Accuracy (Lower is better) vs the number of queries required to find an adversarial solution. Each point represents the number of unique punctuation symbols (for DWBP) or synonyms (for TextFooler) from 1 to 10

I Appendix: Extended Non-Grammar/Grammar Checker Attack Results

The extended results of DWBP when a model employs a grammar checker (Language Tool) as a

defense technique are in Table 6. The table is with RPos = False. With RPos = True (Table 8), although it requires less queries the attack is not as effective, especially when there is a grammar checker (Figure 11 and 12). We also report the results for the most frequent non internal punctuation with RPos = False (Table 7) and with RPos = True (Table 9). Limiting punctuation is also effective against a grammar checker. The findings in fact generalize to a black box attack (Table 10). This table shows that Zeroe with all characters is ineffective and limiting punctuation is competitive with a gray-box character attack technique.

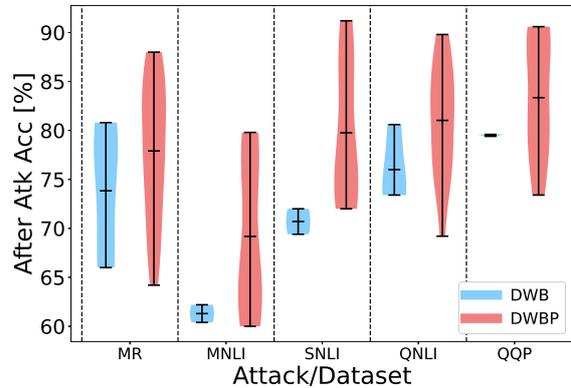


Figure 11: Summary of 'With Grammar Checker' (Table 8 and 9) $A_{aft-atk}$ across datasets with RPos = True

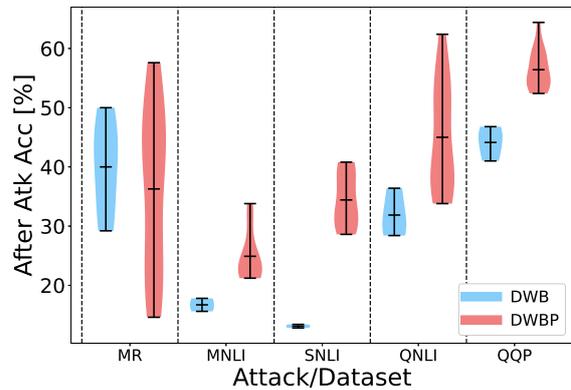


Figure 12: Summary of 'Without Grammar Checker' (Table 8 and 9) $A_{aft-atk}$ across datasets with RPos = True

J Appendix: Extended Multi-level Attack Results

The results for multi-level DWBP and DWBP on MR, MNLI, SNLI, QQP and QNLI across all models is shown in Table 11,12,13

Dataset	Model (Orig Acc)	Method	Without Grammar Checker				With Grammar Checker			
			After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Avg Number Queries	After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Avg Number Queries
MR	CNN (76.6)	DWB	34.2	9.86	0.87	32.3	66.4	7.41	0.88	26.23
		DWBP -	26.6	14.24	1	44.09	54.6	9.61	1	29.9
		DWBP '	14.8	16.86	1	43.23	53.8	10.3	1	28.22
	LSTM (77)	DWB	29.2	10.16	0.87	32.05	66	7.5	0.88	26.32
		DWBP -	26.4	13.71	1	43.1	53.8	10.8	1	30.02
		DWBP '	19.2	14.91	1	42.2	55.6	10.33	1	28.44
	BERT (83.8)	DWB	43.2	10.79	0.87	35.5	77.8	8.4	0.89	26.95
		DWBP -	33.4	14.59	1	46.51	66.8	11.75	1	31.58
		DWBP '	19.6	17.89	1	47.02	65.6	12.54	1	29.52
	RoBERTa (88)	DWB	50	11.58	0.87	35.17	80.8	8.64	0.87	26.73
		DWBP -	36.2	16.44	1	45.88	71.2	12.66	1	31.25
		DWBP '	18.8	19.64	1	47.53	72.6	13.14	1	30.02
XLNet (87)	DWB	43.4	11.06	0.86	34.92	78.2	7.77	0.88	26.89	
	DWBP -	35.8	15.77	1	46.29	70	12.69	1	31.58	
	DWBP '	20.2	19.63	1	47.86	71	12.02	1	30.11	
MNLI	BERT (82.8)	DWB	15.6	7.67	0.9	38.98	62.2	5.94	0.91	31.89
		DWBP -	18.4	7.61	1	39.07	46	7.42	1	33.82
		DWBP '	12.2	8.62	1	40.36	42.6	8.62	1	33.85
	DistilBERT (80.6)	DWB	17.8	7.24	0.9	38.84	60.4	6.15	0.9	31.82
		DWBP -	18.6	7.54	1	38.93	42.4	7.25	1	33.68
		DWBP '	12	8.22	1	40.19	42	8.39	1	33.7
SNLI	BERT (91.2)	DWB	13.4	8.45	0.88	29.58	69.4	6.28	0.89	23.83
		DWBP -	18.8	7.44	1	29.71	51.2	7.41	1	25.41
		DWBP '	10.2	8.28	1	30.23	52.4	7.7	1	25.12
	DistilBERT (86.6)	DWB	12.8	8.51	0.89	29.92	72	6.13	0.89	24.25
		DWBP -	19.4	7.48	1	29.73	52.6	7.14	1	25.65
		DWBP '	7.4	8.79	1	30.56	48.8	8.4	1	25.45
QNLI	BERT (91.2)	DWB	30.8	8.98	0.9	65.04	74	6.37	0.92	47.69
		DWBP -	38	7.7	1	70.61	59.8	7.4	1	51.46
		DWBP '	27.6	9.54	1	75.12	55.4	8.56	1	51.86
	RoBERTa (92)	DWB	36.4	9.79	0.9	65.86	80.6	5.99	0.93	47.94
		DWBP -	44.8	9.39	1	76.46	71.6	7.08	1	52.79
		DWBP '	32	11.36	1	80.47	66.6	8.86	1	53.72
DistilBERT (86.2)	DWB	28.4	9.23	0.91	63.68	73.4	6.23	0.92	47.96	
	DWBP -	35.4	7.87	1	71.42	58.6	6.55	1	51.5	
	DWBP '	26.2	9.41	1	74.53	55.4	7.56	1	51.94	
QQP	BERT (90.4)	DWB	46.8	8.42	0.9	39.42	79.6	7.48	0.9	25.79
		DWBP -	50.6	7.4	1	39.24	61.2	8.07	1	28.58
		DWBP '	47	8.44	1	41.98	62.8	8.95	1	28.68
	DistilBERT (90.8)	DWB	41	9.77	0.89	37.87	79.4	7.01	0.91	25.97
		DWBP -	53.2	7.35	1	38.67	62.8	8.11	1	28.53
		DWBP '	45	9.16	1	41.62	61.4	9.37	1	28.7
XLNet (91.2)	DWB	44.6	9.58	0.89	38.93	79.6	7.44	0.9	25.95	
	DWBP -	53.2	8.85	1	38.91	68.8	9.65	1	28.6	
	DWBP '	47.6	10.18	1	42.8	70.2	10.46	1	29.05	

Table 6: Results without (Original) and when using the LanguageTool grammar checker with RPos=False and internal punctuation

K Appendix: Black-Box Heatmaps

The extended results for the performance difference between ZIP (apostrophe (Ap), hyphen (Hy), comma (Co), full stop (FS) and for QQP question mark (Qu)), character insertions, Zéroé on MR, MNLI, SNLI, QQP and QNLI are in Figure 14 for LSTM on MR, Figure 13 for BERT on MR, Figure 16 for DistilBERT on MNLI, Figure 15 for BERT on MNLI, 18 for DistilBERT on SNLI, Figure 17 for BERT on SNLI, 20 for DistilBERT on QNLI, Figure 19 for BERT on QNLI, 22 for DistilBERT on QQP, Figure 21 for BERT on QQP. For BERT on MR we present the values to construct figure 13 in Table 14 as an example.

L Appendix: Punctuation vs Characters

The extended results for the performance increase in terms of after attack accuracy between inserting letters and punctuation (apostrophe, hyphen, comma, full stop), character insertions, Zéroé on MR, MNLI, SNLI, QQP and QNLI are in Figure 23 for LSTM on MR, Figure 24 for BERT on MR, Figure 25 for DistilBERT on MNLI, Figure 26 for BERT on MNLI, 27 for DistilBERT on SNLI, Figure 28 for BERT on SNLI, 29 for DistilBERT on QNLI, Figure 30 for BERT on QNLI, 31 for DistilBERT on QQP, Figure 32 for BERT on QQP.

The results show a constant improvement across all tasks except for QQP when inserting punctuation. Interestingly the strongest punctuation inser-

Dataset	Model (Orig Acc)	Method	Without Grammar Checker				With Grammar Checker			
			After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Avg Number Queries	After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Avg Number Queries
MR	CNN (76.6)	DWBP .	15.4	16.61	0.97	44.1	62.2	9.43	0.99	27.42
		DWBP ,	14.6	16.84	1	43.2	71	6.14	1	24.27
		DWBP "	27.2	14.12	1	44.2	75.8	4.47	1	22.11
	LSTM (77)	DWBP .	19.4	14.86	0.97	42.93	63.4	8.73	0.99	27.31
		DWBP ,	19.2	14.91	1	42.2	72.6	5.76	1	24.28
		DWBP "	26.6	13.62	1	43.16	76.2	4.7	1	22.01
	BERT (83.8)	DWBP .	18.4	16.72	0.97	45.57	71.2	10.1	0.98	27.81
		DWBP ,	18.4	16.64	1	45.78	80	9.1	1	24.57
		DWBP "	29.4	14.43	1	44.94	83.2	6.15	1	22.32
	RoBERTa (88)	DWBP .	19.4	19.35	0.96	48.81	79.2	9.96	0.98	28.22
		DWBP ,	18.6	19.53	1	47.12	85	5.06	1	24.6
		DWBP "	34.6	16.5	1	46.49	87.4	6.24	1	22.34
XLNet (87)	DWBP .	17.8	19.85	0.97	48.34	75.6	10.44	0.99	28.31	
	DWBP ,	18	19.68	1	47.46	84.4	5.85	1	24.7	
	DWBP "	34	15.89	1	45.69	87	0	0	22.41	
MNLI	BERT (82.8)	DWBP .	14	7.94	1	39.98	47.8	7.46	1	32.62
		DWBP ,	12.6	7.77	1	39.59	76	4.1	1	30.09
		DWBP)	11.6	8.08	1	40.04	71.2	5.38	1	30.4
	DistilBERT (80.6)	DWBP .	12.8	7.36	1	39.47	45.4	7.28	1	32.37
		DWBP ,	13.2	7.41	1	39.28	74	5.28	1	30.07
		DWBP)	10.4	7.95	1	39.49	70.4	6.22	1	30.4
SNLI	BERT (91.2)	DWBP .	10	7.88	1	29.98	57	6.92	1	24.2
		DWBP ,	10.6	8.31	1	30.08	85.6	5.92	1	22.54
		DWBP "	17	7.75	1	29.79	91.2	0	0	22.13
	DistilBERT (86.6)	DWBP .	9.6	8.14	1	30.26	54	7.55	1	24.6
		DWBP ,	4	8.84	1	30.19	80.6	5.91	1	23.01
		DWBP "	16.8	7.36	1	29.58	85.8	5.81	1	22.58
QNLI	BERT (91.2)	DWBP ,	25	9.36	1	73.9	83.6	5.12	1	42.49
		DWBP ?	26.8	9	1	74.66	66.6	7.27	1	48.93
		DWBP ?	25	9.92	1	74.08	58.2	8.33	1	50.53
	RoBERTa (92)	DWBP ,	28.6	12.23	1	79.46	88.6	4.62	1	42.47
		DWBP .	32.2	11.6	1	80.69	76.4	7.69	1	49.69
		DWBP ?	31.4	11.84	1	81.15	71	8.39	1	52.09
DistilBERT (86.2)	DWBP ,	19	10.09	1	70.73	79.2	4.93	1	42.5	
	DWBP .	19.2	9.66	1	70.2	63	7.11	1	48.87	
	DWBP ?	23.2	7.66	1	71.08	46.4	7.78	1	49.13	
QQP	BERT (90.4)	DWBP ?	46.2	8.41	1	41.96	64.4	8.33	1	28.03
		DWBP ,	48.6	8.29	1	42.31	86.8	6.28	1	23.5
		DWBP "	49.4	7.78	1	39.01	88.6	8.86	1	23.28
	DistilBERT (90.8)	DWBP ?	43.4	9.39	1	41.27	64.8	9.39	1	28.04
		DWBP ,	46.2	8.98	1	41.69	87.2	6.27	1	23.54
		DWBP "	50.4	7.82	1	37.95	88.2	6.98	1	23.26
XLNet (91.2)	DWBP ?	47.4	10.19	1	42.83	70.8	10.65	1	28.33	
	DWBP ,	47.6	10.34	1	42.64	89.2	7.12	1	23.6	
	DWBP "	53.8	9.45	1	38.69	89.4	7.64	1	23.32	

Table 7: Results without (Original) and when using the LanguageTool grammar checker with RPos=False and most frequent non internal punctuation from Table 16

tion appears to vary between tasks. For example, the comma is the strongest in MNLI for BERT, while the full stop is strongest for SNLI on BERT. Moreover, whether there are character insertions or punctuation insertions in the QQP task seems to have little to no difference; at times, character insertions are better, for example, when inserting a hyphen in QQP when trained on BERT. We speculate that QQP is hard to attack, whether using character or punctuation insertions. It could be hard to attack because a model is sensitive to samples with similar question pairs. Hence, it is easy to perturb them to become unsimilar by adding character or punctuation symbols. However, to perturb a nonsimilar question pair to become similar is harder, and neither character nor punctuation sym-

bols can do this. Future research to prove this can investigate this phenomenon by plotting the ROC and Precision/Recall graphs. However, the high $A_{aft-atk}$ in 13 and Table 6 in the Appendix, especially compared to other tasks, is a good indication of this theory being correct. Exploring the reasons behind these phenomena, and introducing a novel attack that can further decrease the $A_{aft-atk}$ of QQP, could be an interesting entry point for future research.

M Appendix: Adversarial training results

See Table 15 for the results of adversarial training using DWBP with hyphen insertions.

Dataset	Model (Orig Acc)	Method	Without Grammar Checker				With Grammar Checker			
			After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Avg Number Queries	After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Avg Number Queries
MR	CNN (76.6)	DWB	34.2	9.86	0.87	32.3	66.4	7.41	0.88	26.23
		DWBP -	26.8	14.18	1	26.54	67	9.42	1	24.39
		DWBP ’	14.8	16.85	1	26.37	68.4	7.11	1	23.99
	LSTM (77)	DWB	29.2	10.16	0.87	32.05	66	7.5	0.88	26.32
		DWBP -	26.6	13.57	1	26.23	64.2	9.43	1	24.29
		DWBP ’	19.2	14.91	1	26.05	67.2	7.87	1	23.87
	BERT (83.8)	DWB	43.2	10.79	0.87	35.5	77.8	8.4	0.89	26.95
		DWBP -	47.2	15.49	1	28.67	74.2	11.05	1	24.86
		DWBP ’	39	18.21	1	28.94	77.4	9.61	1	24.37
	RoBERTa (88)	DWB	50	11.58	0.87	35.17	80.8	8.64	0.87	26.73
		DWBP -	55.2	16.24	1	28.58	81.2	11.28	1	24.85
		DWBP ’	43.6	19.12	1	29.43	83.8	9.48	1	24.45
XLNet (87)	DWB	43.4	11.06	0.86	34.92	78.2	7.77	0.88	26.89	
	DWBP -	57.6	14.6	1	28.79	79.2	9.35	1	25.02	
	DWBP ’	47.4	19.22	1	29.72	82.4	9.04	1	24.55	
MNLI	BERT (82.8)	DWB	15.6	7.67	0.9	38.98	62.2	5.94	0.91	31.89
		DWBP -	33.8	7.99	1	32.58	63.4	6.84	1	31.42
		DWBP ’	26	9.54	1	33.04	68	6.56	1	31.21
	DistilBERT (80.6)	DWB	17.8	7.24	0.9	38.84	60.4	6.15	0.9	31.82
		DWBP -	31.4	8.25	1	32.49	60	6.9	1	31.29
		DWBP ’	24.4	9.5	1	32.88	62.8	6.89	1	31.24
SNLI	BERT (91.2)	DWB	13.4	8.45	0.88	29.58	69.4	6.28	0.89	23.83
		DWBP -	39.8	8.43	1	24.63	72.6	7.02	1	23.43
		DWBP ’	33.8	10.12	1	24.98	76.4	6.9	1	23.14
	DistilBERT (86.6)	DWB	12.8	8.51	0.89	29.92	72	6.13	0.89	24.25
		DWBP -	40.8	7.8	1	24.89	72.8	6.45	1	23.74
		DWBP ’	28.6	9.7	1	25.2	72	6.43	1	23.52
QNLI	BERT (91.2)	DWB	30.8	8.98	0.9	65.04	74	6.37	0.92	47.69
		DWBP -	48.4	7.95	1	47.62	76.4	6.07	1	43.75
		DWBP ’	39.6	9.55	1	48.67	77.8	6.34	1	43.71
	RoBERTa (92)	DWB	36.4	9.79	0.9	65.86	80.6	5.99	0.93	47.94
		DWBP -	62.4	8.56	1	49.13	83.4	5.86	1	43.96
		DWBP ’	52.6	10.83	1	50.83	84.4	6.18	1	43.75
DistilBERT (86.2)	DWB	28.4	9.23	0.91	63.68	73.4	6.23	0.92	47.96	
	DWBP -	48.6	7.85	1	47.87	75.2	6.15	1	43.89	
	DWBP ’	39.8	9.57	1	48.88	75	6.43	1	43.88	
QQP	BERT (90.4)	DWB	46.8	8.42	0.9	39.42	79.6	7.48	0.9	25.79
		DWBP -	54.2	8.53	1	27.46	73.4	7.95	1	25.1
		DWBP ’	52.4	9.22	1	28.01	77.6	7.72	1	25.05
	DistilBERT (90.8)	DWB	41	9.77	0.89	37.87	79.4	7.01	0.91	25.97
		DWBP -	57.4	8.1	1	27.44	73.4	7.67	1	25.12
		DWBP ’	52.4	10.39	1	28.1	78.8	7.68	1	25.15
XLNet (91.2)	DWB	44.6	9.58	0.89	38.93	79.6	7.44	0.9	25.95	
	DWBP -	59.8	9.91	1	27.53	81.4	8.07	1	25.26	
	DWBP ’	60.2	11.18	1	28.46	84.4	8.75	1	25.32	

Table 8: Results without (Original) and when using the LanguageTool grammar checker with RPos=True and internal punctuation

N Appendix: Analysis

N.1 Empirical punctuation counts across datasets

The variance of symbols within each dataset is high. Table 16 shows the number of punctuation symbols and their proportion as a percentage of other characters for each dataset. The table is subdivided into ‘Total Punctuation’ and ‘Internal punctuation’ or the punctuation only appearing within words, such as apostrophes and hyphens. This distinction is essential, as Section 5 empirically motivates why punctuation can be used as an attack vector and cannot be easily defended.

N.2 Removing Punctuation

Table 17 shows the impact on all models across all datasets of removing either all punctuation, only internal punctuation, or just removing internal punctuation except the apostrophe and hyphen, which the two punctuation characters over-represented within words, as seen from Table 16. On the other hand, Table 18 shows how the original accuracy changes if the models are finetuned on data with no punctuation.

N.3 Most frequent punctuation in dataset attack

Table 19 highlights the drop in performance by the type of punctuation symbol used in the attack. The attack uses the most frequent symbols in a sample

Dataset	Model (Orig Acc)	Method	Without Grammar Checker				With Grammar Checker			
			After Attack	Perturbed	Semantic	Avg Number	After Attack	Perturbed	Semantic	Avg Number
			Acc [%]	Words [%]	Sim	Queries	Acc [%]	Words [%]	Sim	Queries
MR	CNN (76.6)	DWBP .	16.2	16.41	0.97	26.34	70.2	7.21	0.99	23.51
		DWBP ,	14.6	16.83	1	26.35	74	4.55	1	22.47
		DWBP "	27	14.15	1	26.56	76.2	5	1	21.89
	LSTM (77)	DWBP .	20	14.68	0.98	26.03	70.2	7.21	0.99	23.34
		DWBP ,	19.2	14.91	1	26.05	74.8	6.37	1	22.34
		DWBP "	26.6	13.58	1	26.23	76.6	4.67	1	21.77
	BERT (83.8)	DWBP .	36.2	16.26	0.97	28.47	78	7.74	0.98	23.73
		DWBP ,	35.8	16.19	1	28.5	81.6	8.65	1	22.64
		DWBP "	44.4	13.91	1	28.33	83.2	6.15	1	22.05
	RoBERTa (88)	DWBP .	46	18.44	0.98	29.42	86	8.91	1	23.85
		DWBP ,	42.8	18.9	1	29.25	87.4	5.94	1	22.66
		DWBP "	55.2	14.86	1	28.55	88	0	0	22.09
XLNet (87)	DWBP .	43.6	18.52	0.97	29.4	84.4	7.22	0.99	23.95	
	DWBP ,	45.6	18.53	1	29.53	85.2	6.39	1	22.79	
	DWBP "	56.4	15.3	1	28.73	87	0	0	22.16	
MNLI	BERT (82.8)	DWBP .	22	9.63	1	32.86	63.8	5.92	1	30.76
		DWBP ,	23.2	8.76	1	32.79	79.8	3.76	1	29.71
		DWBP)	22.8	9.64	1	32.9	77.2	5.11	1	29.95
	DistilBERT (80.6)	DWBP .	21.8	8.77	1	32.65	62	5.76	1	30.75
		DWBP ,	22.4	9.01	1	32.67	77.6	5.09	1	29.67
		DWBP)	21.2	9.15	1	32.67	77.2	4.52	1	29.93
SNLI	BERT (91.2)	DWBP .	33.4	9.17	1	24.87	76.4	6.76	1	22.85
		DWBP ,	31	9.68	1	24.85	89	6.53	1	22.21
		DWBP "	39.8	9.23	1	24.69	91.2	0	0	22.09
	DistilBERT (86.6)	DWBP .	29.4	9.68	1	25.21	75.8	5.86	1	23.25
		DWBP ,	30.4	9.33	1	25.15	85.2	4.71	1	22.64
		DWBP "	37.2	8.34	1	24.87	86.2	5.16	1	22.52
QNLI	BERT (91.2)	DWBP .	37.6	9.79	1	48.5	88.6	4.99	1	40.14
		DWBP ,	40.6	8.98	1	48.59	81.6	5.01	1	42.52
		DWBP ?	36.2	9.99	1	48.36	81.4	5.05	1	43.1
	RoBERTa (92)	DWBP .	52.4	11.43	1	50.71	89.8	3.76	1	39.97
		DWBP ,	56.4	9.99	1	51	87	4.62	1	42.45
		DWBP ?	54.8	10.6	1	51.06	85.8	5.22	1	43.05
DistilBERT (86.2)	DWBP .	36.8	10.06	1	48.7	82.8	4.97	1	40.1	
	DWBP ,	33.8	9.62	1	47.88	77	5.09	1	42.46	
	DWBP ?	35	8.13	1	47.85	69.2	5.27	1	42.67	
QQP	BERT (90.4)	DWBP ?	54.8	9.45	1	28.08	78.4	8.18	1	24.79
		DWBP ,	53.6	9	1	28.01	88.8	8.2	1	23.08
		DWBP "	55.4	8.41	1	27.45	90.2	5	1	23
	DistilBERT (90.8)	DWBP ?	53.6	9.86	1	28.07	80.4	7.6	1	24.81
		DWBP ,	54.4	10.1	1	28.17	89	6.4	1	23.11
		DWBP "	56.4	8.82	1	27.38	89.8	7.36	1	23.03
	XLNet (91.2)	DWBP ?	58.8	11.63	1	28.51	83.6	7.68	1	24.88
		DWBP ,	58.4	11.57	1	28.46	90.6	7.82	1	23.15
		DWBP "	64.4	10.14	1	27.63	90.4	6.98	1	23.1

Table 9: Results without (Original) and when using the LanguageTool grammar checker with RPos=True and most frequent non internal punctuation from Table 16

for each task in Table 16.

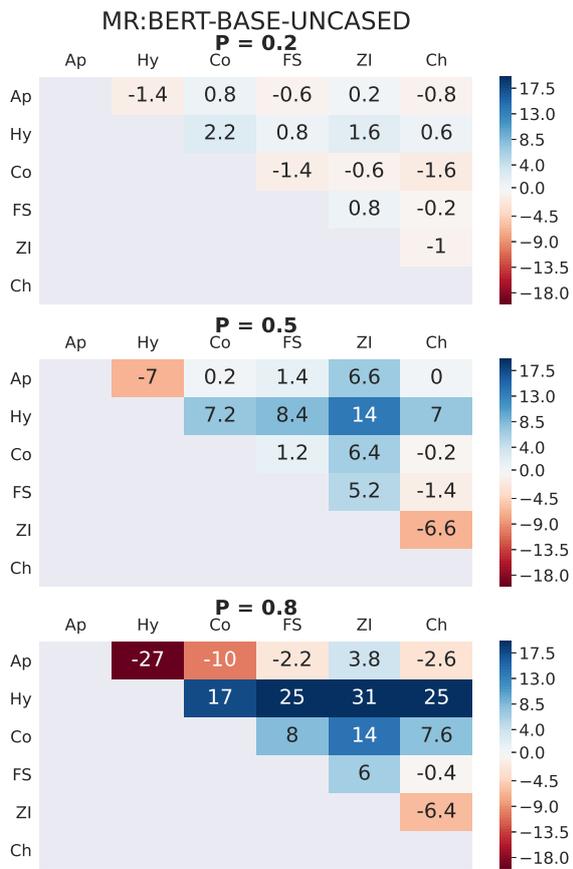


Figure 13

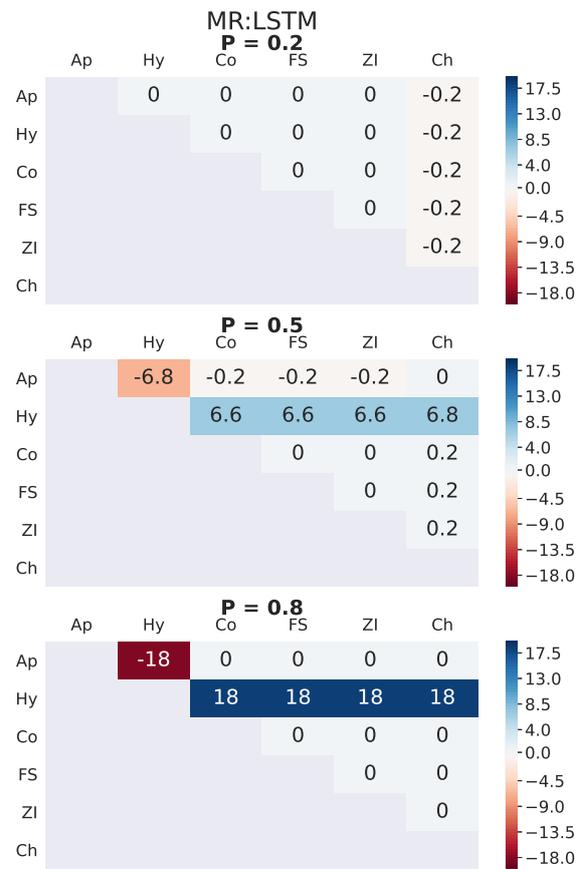


Figure 14

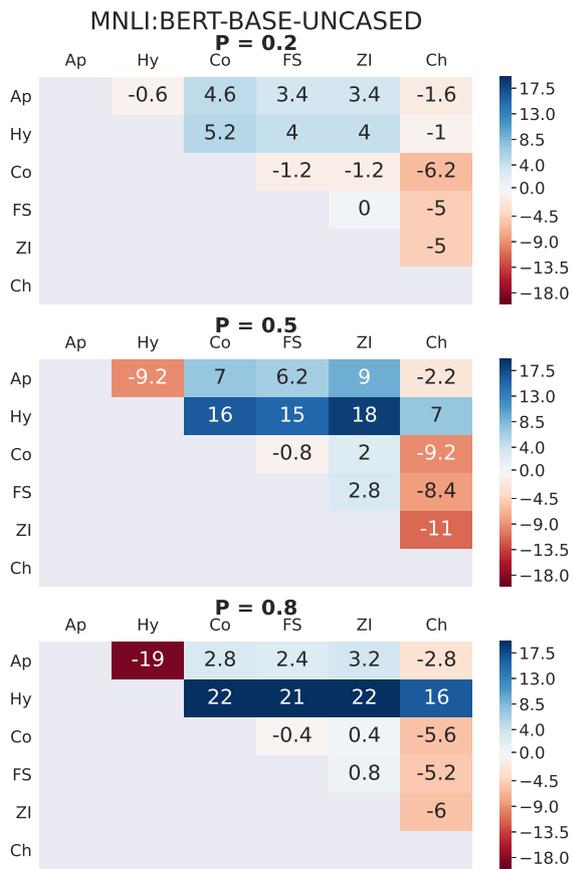


Figure 15

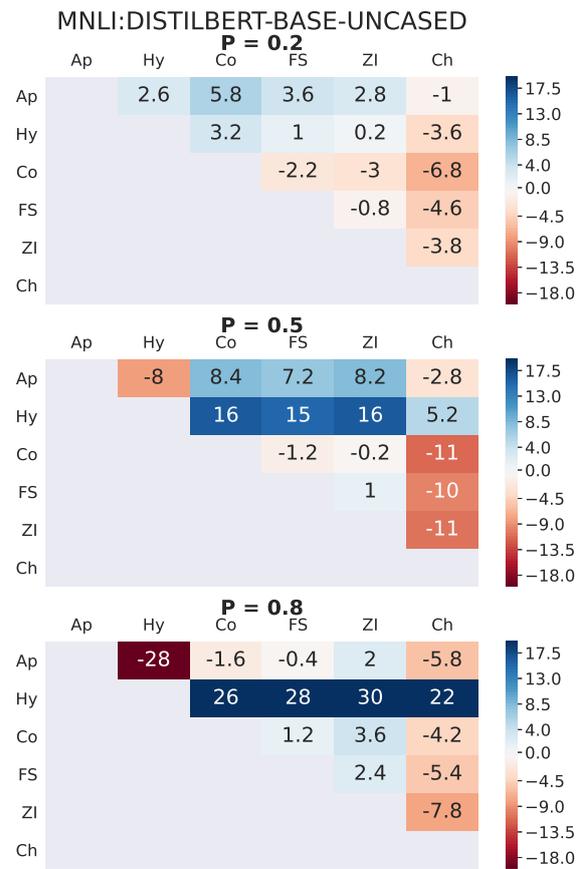


Figure 16

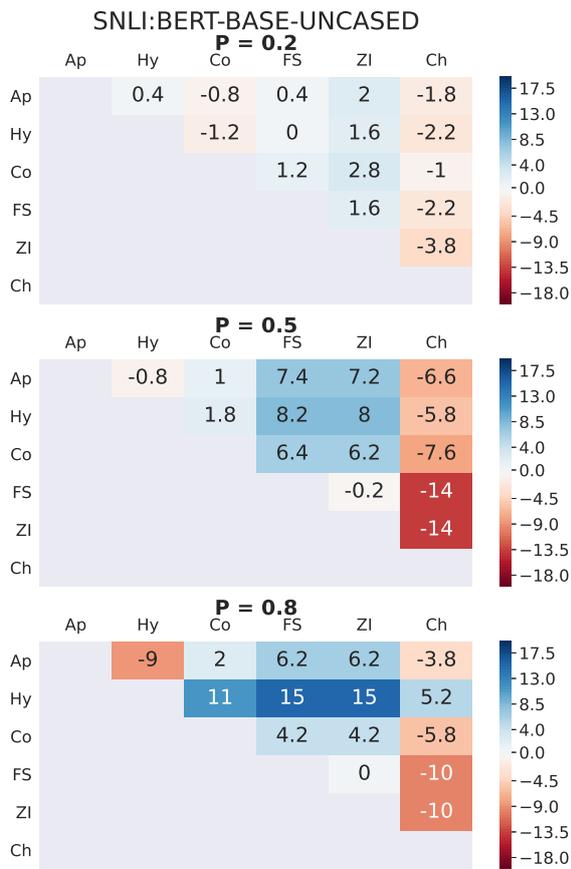


Figure 17

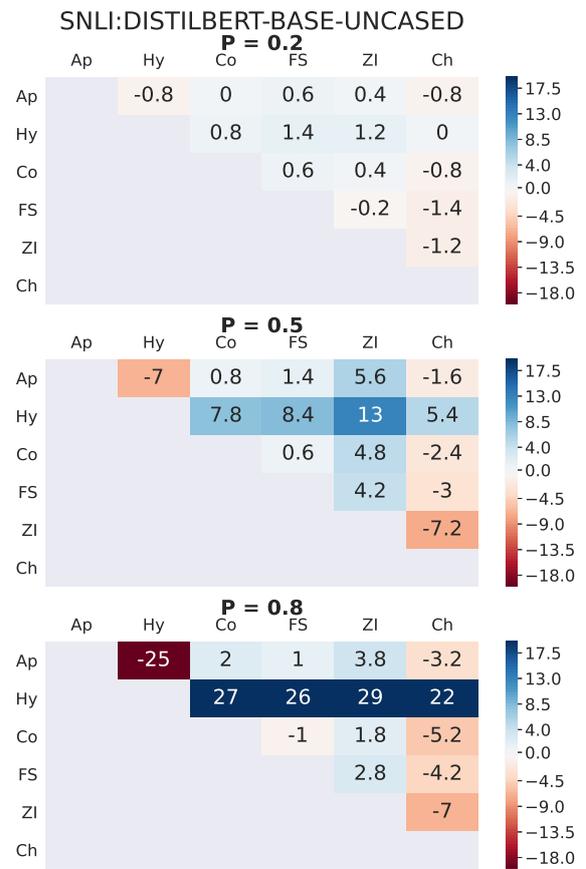


Figure 18

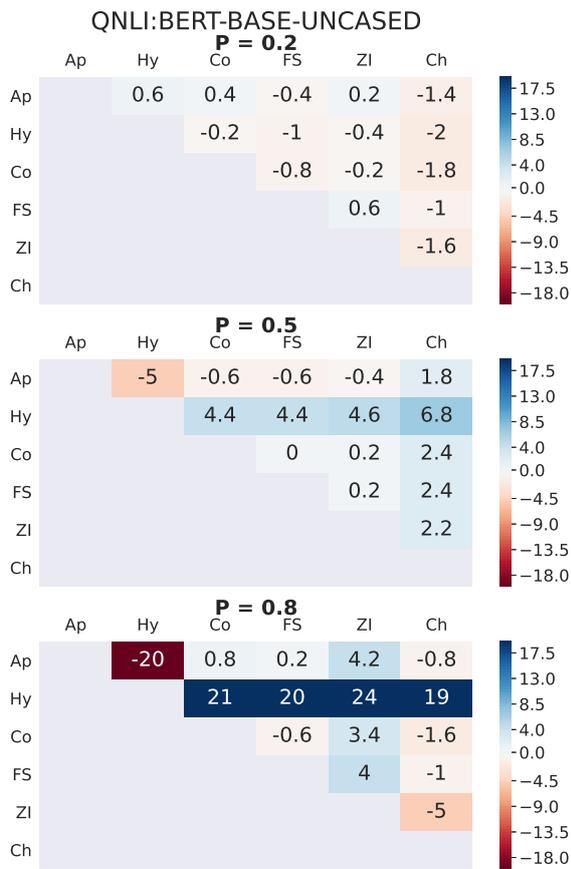


Figure 19

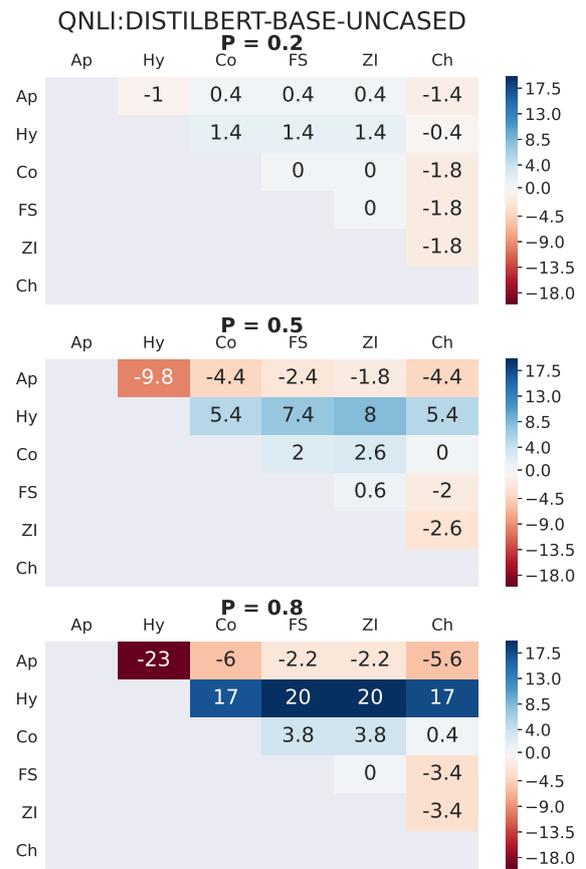


Figure 20

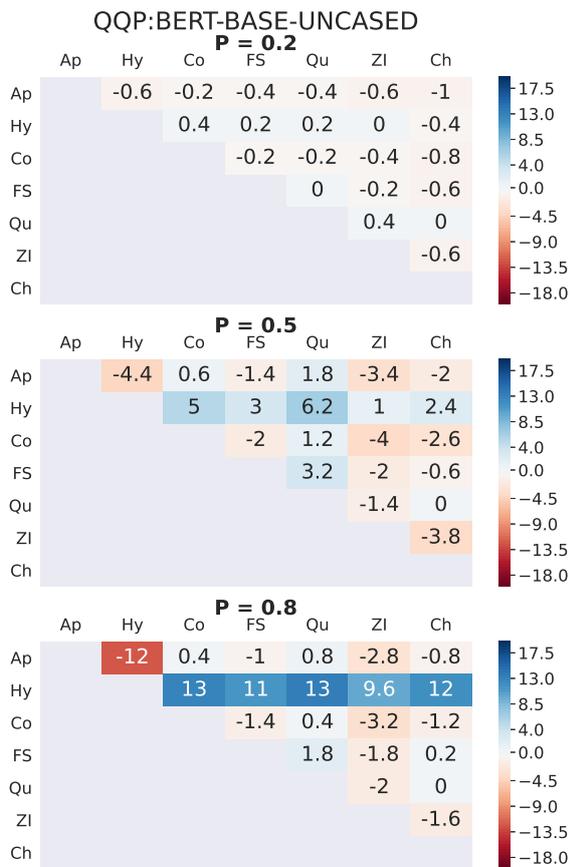


Figure 21

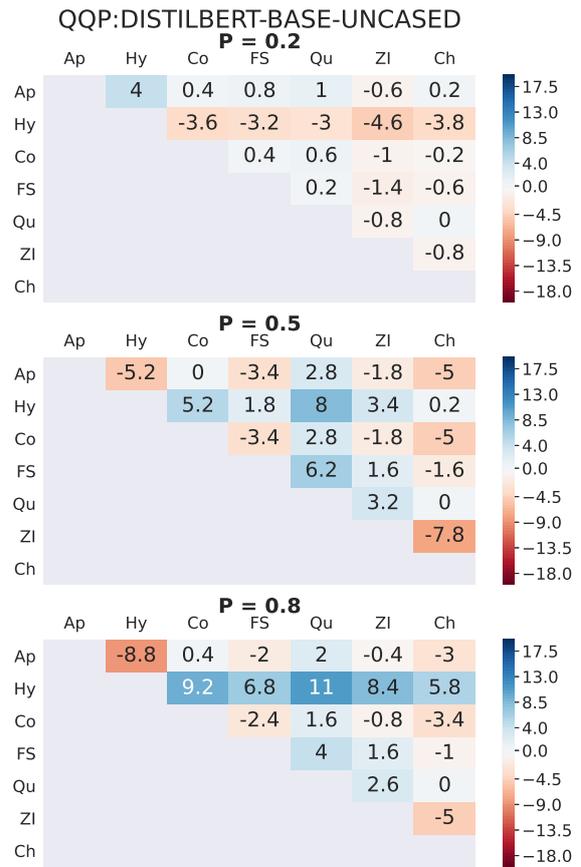


Figure 22

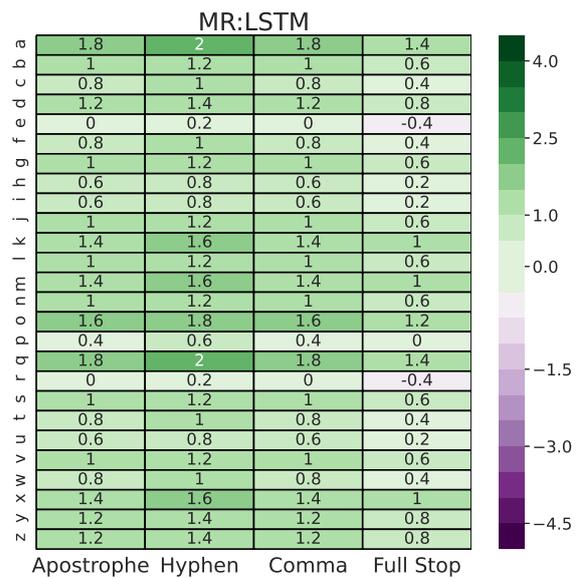


Figure 23

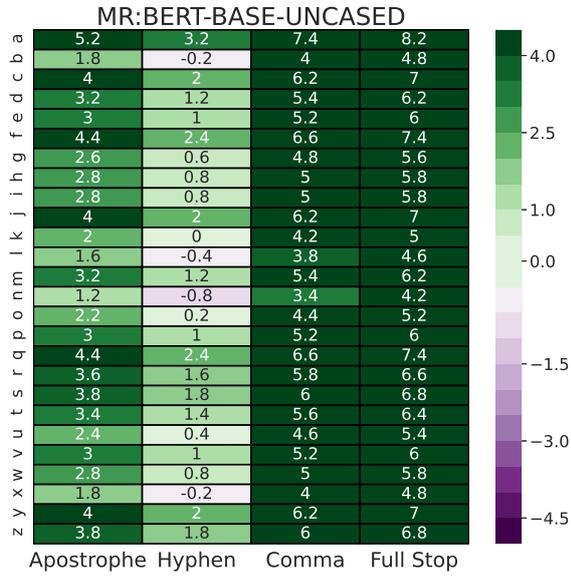


Figure 24

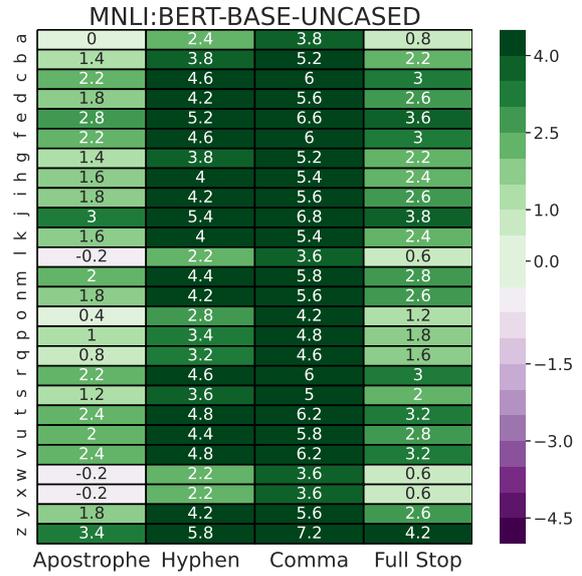


Figure 26

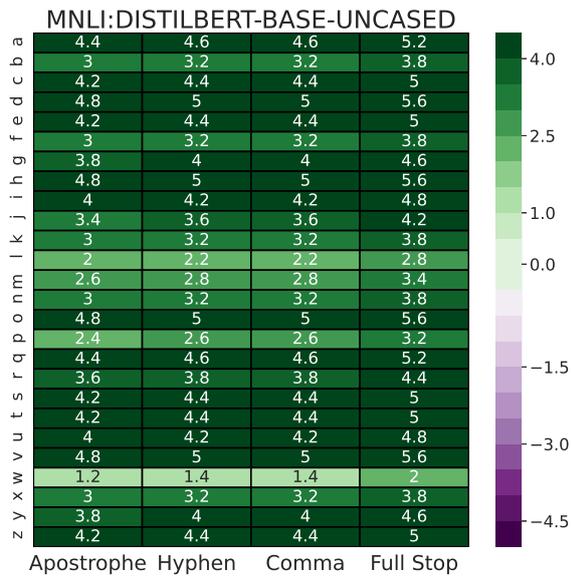


Figure 25

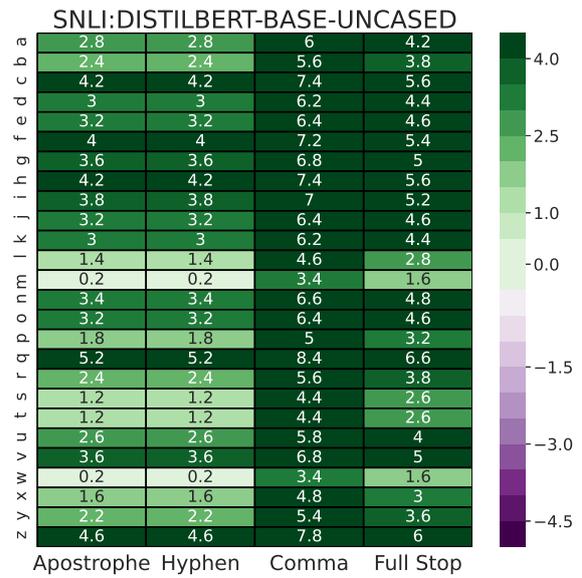


Figure 27

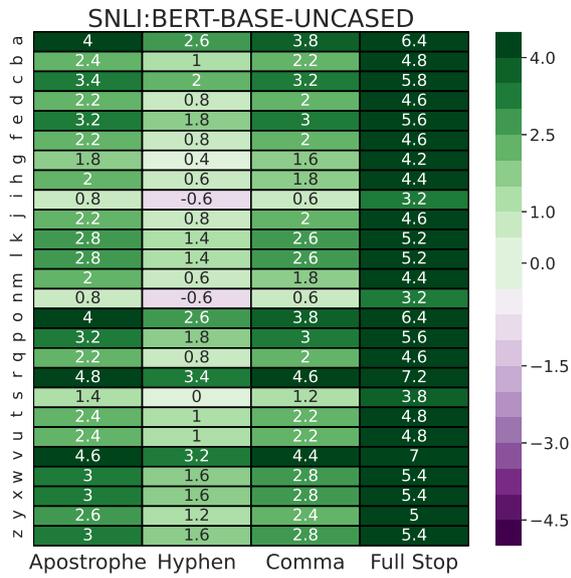


Figure 28

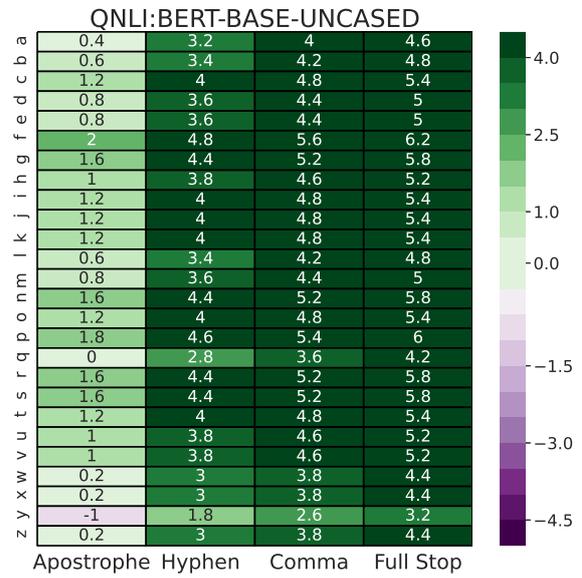


Figure 30

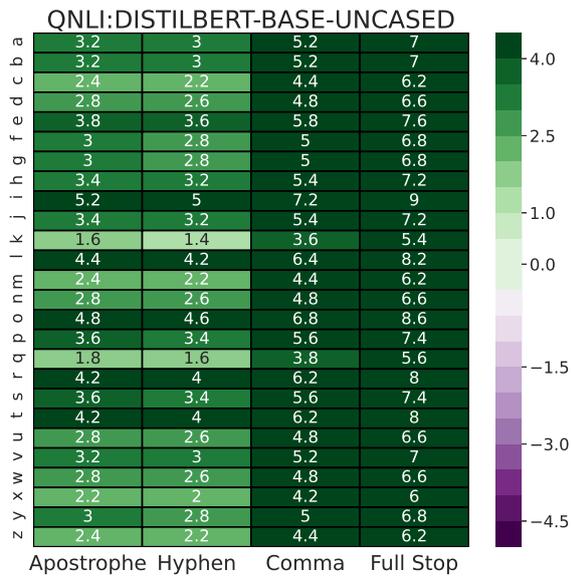
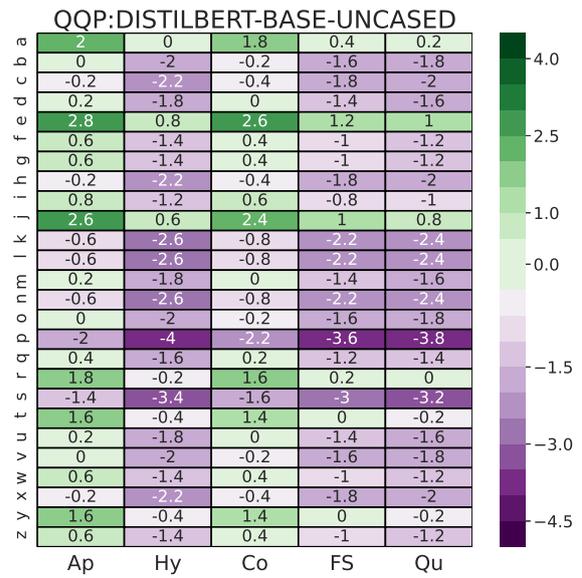


Figure 29



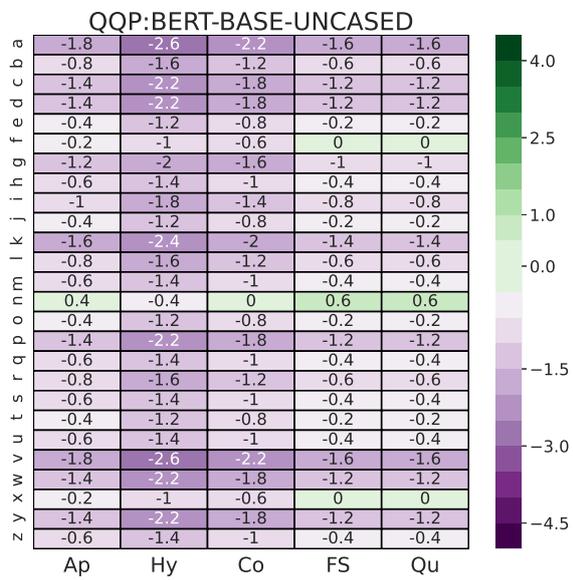


Figure 32

Dataset	Model	Method	After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Average Time Taken [s]	Avg Number Queries	Drop [%]	
MR	CNN (76.6)	Zeroe	75	11.33	0.88	0.4007	0	1.6	
		DWB	66.4	7.41	0.88	0.8542	26.23	10.2	
		ZIP -	69	15	1	0.2901	0	7.6	
		ZIP '	68.8	25.94	1	0.5539	0	7.8	
	LSTM (77)	Zeroe	75.2	8.49	0.92	0.4879	0	1.8	
		DWB	66	7.5	0.88	1.0247	26.32	11	
		ZIP -	66.8	14.02	1	0.3308	0	10.2	
		ZIP '	65.4	24.26	1	0.5739	0	11.6	
	BERT (83.8)	Zeroe	82.2	7.43	0.89	0.3426	0	1.6	
		DWB	77.8	8.4	0.89	1.0962	26.95	6	
		ZIP -	78	13.46	1	0.4014	0	5.8	
		ZIP '	70	25.12	1	0.5532	0	13.8	
	RoBERTa (88)	Zeroe	87.6	12.27	0.95	0.2185	0	0.4	
		DWB	80.8	8.64	0.87	0.9447	26.73	7.2	
		ZIP -	80.6	14.26	1	0.3216	0	7.4	
		ZIP '	78.2	24.09	1	0.4729	0	9.8	
	XLNet (87)	Zeroe	85.6	6.64	0.95	0.7718	0	1.4	
		DWB	78.2	7.77	0.88	1.9823	26.89	8.8	
		ZIP -	79.6	13.6	1	0.5011	0	7.4	
		ZIP '	76.8	24.36	1	0.6806	0	10.2	
	MNLI	BERT (82.8)	Zeroe	75.6	5.14	0.93	0.329	0	7.2
			DWB	62.2	5.94	0.91	0.8279	31.89	20.6
			ZIP -	64.6	6.9	1	0.2379	0	18.2
			ZIP '	57	11.2	1	0.3272	0	25.8
DistilBERT (80.6)		Zeroe	75.4	4.73	0.95	0.3012	0	5.2	
		DWB	60.4	6.15	0.9	0.7645	31.82	20.2	
		ZIP -	69.6	7.49	1	0.1913	0	11	
		ZIP '	57.6	11.99	1	0.3292	0	23	
SNLI	BERT (91.2)	Zeroe	86.2	6.32	0.94	0.2505	0	5	
		DWB	69.4	6.28	0.89	0.5844	23.83	21.8	
		ZIP -	61	6.16	1	0.1865	0	30.2	
		ZIP '	65.8	10.33	1	0.2095	0	25.4	
	DistilBERT (86.6)	Zeroe	84.2	6.61	0.92	0.2391	0	2.4	
		DWB	72	6.13	0.89	0.434	24.25	14.6	
		ZIP -	74.4	6.37	1	0.165	0	12.2	
		ZIP '	65.2	11.37	1	0.2254	0	21.4	
QNLI	BERT (91.2)	Zeroe	88.6	5.94	0.94	1.6294	0	2.6	
		DWB	74	6.37	0.92	1.7717	47.69	17.2	
		ZIP -	81.2	10.69	1	0.5898	0	10	
		ZIP '	73.4	21.69	1	1.0907	0	17.8	
	RoBERTa (92)	Zeroe	90.6	3.19	0.96	0.9367	0	1.4	
		DWB	80.6	5.99	0.93	1.7655	47.94	11.4	
		ZIP -	82	10.59	1	0.7899	0	10	
		ZIP '	77	22.08	1	1.0432	0	15	
DistilBERT (86.2)	Zeroe	84.4	7.36	0.94	0.9437	0	1.8		
	DWB	73.4	6.23	0.92	1.6188	47.96	12.8		
	ZIP -	76	10.14	1	0.5882	0	10.2		
	ZIP '	66.2	20.63	1	1.0017	0	20		
QQP	BERT (90.4)	Zeroe	87.6	6.56	0.92	0.3038	0	2.8	
		DWB	79.6	7.48	0.9	0.6998	25.79	10.8	
		ZIP -	74.4	8.79	1	0.2272	0	16	
		ZIP '	68.4	14.55	1	0.3317	0	22	
	DistilBERT (90.8)	Zeroe	87.8	7.26	0.93	0.3298	0	3	
		DWB	79.4	7.01	0.91	0.6158	25.97	11.4	
		ZIP -	72.8	8.33	1	0.1927	0	18	
		ZIP '	69.8	14.29	1	0.2581	0	21	
XLNet (91.2)	Zeroe	89.8	7.26	0.92	0.8158	0	1.4		
	DWB	79.6	7.44	0.9	1.6554	25.95	11.6		
	ZIP -	78	8.77	1	0.3762	0	13.2		
	ZIP '	75.8	14.34	1	0.455	0	15.4		

Table 10: Results of Zeroe, DWB and ZIP attacks while using the LanguageTool grammar checker

Dataset	Model (Orig Acc)	Method	After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Avg Time Taken [s]	Avg Number Queries
MR	CNN (76.6)	DWBP	14.6	16.99	1	0.0513	69.01
		TextFooler	0.4	11.82	0.85	0.2144	74.79
		TextFooler/DWBP	0.2	13.09	0.89	0.1145	69.75
		SememePSO	2.6	13.73	0.83	0.6824	2711.91
		SememePSO/DWBP	2	10.97	0.86	0.4516	1012.17
	LSTM (77)	DWBP	19.2	15.13	1	0.066	66.9
		TextFooler	0.8	11.43	0.86	0.1943	71.03
		TextFooler/DWBP	0.4	12.87	0.89	0.1289	67.95
		SememePSO	2.8	13.17	0.83	0.7235	2342.27
		SememePSO/DWBP	1.6	10.26	0.86	0.5366	923.21
	BERT (83.8)	DWBP	17.4	18.32	1	0.721	74.7
		TextFooler	9.4	17.54	0.82	1.3072	118.5
		TextFooler/DWBP	7.6	18.31	0.89	1.122	105.35
		SememePSO	7	16.52	0.81	16.1811	4950.71
		SememePSO/DWBP	6	9.99	0.89	7.3252	988.44
	RoBERTa (88)	DWBP	14	19.08	1	0.71	72.42
		TextFooler	5.4	16.21	0.83	1.1566	106.89
		TextFooler/DWBP	5.8	16.92	0.89	0.9861	94.63
		SememePSO	6	17.44	0.8	15.9324	4855.71
		SememePSO/DWBP	5.8	10.96	0.88	9.4678	1225.24
XLNet (87)	DWBP	16.2	19.1	1	2.8193	74.35	
	TextFooler	7.4	15.68	0.83	4.4761	108.5	
	TextFooler/DWBP	5.4	17.19	0.88	3.9146	96.17	
	SememePSO	5.8	16.75	0.81	53.6015	4619.19	
	SememePSO/DWBP	6	10.8	0.88	35.2176	1162.83	

Table 11: Results on classification for multi-level DWBP

Dataset	Model (Orig Acc)	Method	After Attack Acc [%]	Perturbed Words [%]	Semantic Sim	Average Time Taken [s]	Avg Number Queries
MNLi	BERT (82.8)	DWBP	9.6	8.43	1	0.5381	51.26
		TextFooler	12.2	6.99	0.9	0.8749	76.18
		TextFooler/DWBP	4.2	8.02	0.96	0.706	63.68
		SememePSO	20.2	5.9	0.9	2.0034	1200.36
		SememePSO/DWBP	5	6.16	0.94	2.068	208.73
	DistilBERT (80.6)	DWBP	11.4	7.95	1	0.2668	50.55
		TextFooler	12.6	7.54	0.9	0.516	77.88
		TextFooler/DWBP	5.4	7.95	0.96	0.3899	64.85
		SememePSO	21.6	6	0.89	1.0294	1146.93
		SememePSO/DWBP	6.2	6.44	0.94	1.0934	220.54
SNLI	BERT (91.2)	DWBP	7.2	7.99	1	0.4037	38.57
		TextFooler	14	7.46	0.9	0.6992	64.2
		TextFooler/DWBP	3.2	7.72	0.97	0.523	47.96
		SememePSO	16.6	6.9	0.88	2.1876	764.64
		SememePSO/DWBP	2.8	6.63	0.93	1.3637	139.42
	DistilBERT (86.6)	DWBP	6.6	8.36	1	0.2101	38.77
		TextFooler	10	7.75	0.9	0.4072	64.33
		TextFooler/DWBP	1.6	7.79	0.96	0.2848	48.09
		SememePSO	14.4	6.66	0.88	1.1608	689.18
		SememePSO/DWBP	2	6.68	0.93	0.7641	151.06

Table 12: Results on entailment for multi-level DWBP

Dataset	Model	Method	After Attack	Perturbed	Semantic	Average Time	Avg Number
			Acc [%]	Words [%]	Sim	Taken [s]	Queries
QNLI	BERT (91.2)	DWBP	24.4	10.01	1	0.9616	114.79
		TextFooler	22.6	9.46	0.9	1.6601	168.88
		TextFooler/DWBP	18.2	10.32	0.95	1.413	156.15
		SememePSO	37.4	11.13	0.88	44.0359	12838.6
		SememePSO/DWBP	27.2	5.79	0.96	18.2188	2093.68
	RoBERTa (92)	DWBP	26.8	11.9	1	1.0311	120.65
		TextFooler	26.8	9.94	0.9	1.6683	174.98
		TextFooler/DWBP	18.8	11.45	0.94	1.4461	159.57
		SememePSO	41	11.32	0.87	40.4838	14041.7
		SememePSO/DWBP	32.4	6.45	0.95	28.7531	2325.13
	DistilBERT (86.2)	DWBP	23.8	9.28	1	0.4786	110.45
		TextFooler	22	10.04	0.9	0.9577	168.64
		TextFooler/DWBP	13.8	10.66	0.95	0.7953	145.54
		SememePSO	37.2	11.28	0.88	18.1074	13254.8
		SememePSO/DWBP	28.4	6.26	0.96	15.2255	2207.65
	QQP	BERT (90.4)	DWBP	45.2	8.31	1	0.388
TextFooler			42.2	8.44	0.9	0.7218	116.5
TextFooler/DWBP			41.2	8.43	0.97	0.6001	102.06
SememePSO			50.4	7.94	0.88	2.2428	10858
SememePSO/DWBP			42.8	7.36	0.96	3.8072	398.73
DistilBERT (90.8)		DWBP	45.4	8.62	1	0.2059	60.1
		TextFooler	38.6	9.48	0.9	0.4686	114.73
		TextFooler/DWBP	37.6	9.9	0.95	0.3919	100.76
		SememePSO	50	8.59	0.88	2.925	11194.4
		SememePSO/DWBP	40.8	7.84	0.95	1.9978	397.52
XLNet (91.2)		DWBP	44.8	9.88	1	1.7345	61.85
		TextFooler	38.8	9.6	0.89	3.1888	115.37
	TextFooler/DWBP	37.2	9.71	0.94	2.6764	100.81	
	SememePSO	49.4	8.3	0.88	10.0921	10057.7	
	SememePSO/DWBP	41.4	7.91	0.93	10.092	400.73	

Table 13: Results on question answering tasks for multi-level DWBP

P	ZIP Ap		ZIP Hy		ZIP Co	
	$A_{aft-atk}$	S	$A_{aft-atk}$	S	$A_{aft-atk}$	S
0.2	77.8	1.0	79.2	1.0	77.0	1.0
0.5	68.2	1.0	75.2	1.0	68.0	1.0
0.8	47.8	1.0	75.0	1.0	58.0	1.0
P	ZIP FS		ZI Ze		ZI Ch	
	$A_{aft-atk}$	S	$A_{aft-atk}$	S	$A_{aft-atk}$	S
0.2	78.4	0.97	77.6	0.91	78.6	0.83
0.5	66.8	0.94	61.6	0.79	68.2	0.72
0.8	50.0	0.91	44.0	0.69	50.4	0.62

Table 14: Results of black-box insertions on MR/BERT

Dataset	Model	Baseline Orig Acc [%]	Baseline After Attack Acc [%]	Robust Orig Acc [%]	Robust After Attack Acc [%]
MR Hy	LSTM	78.2	29.8	77.6	32.2
	BERT	84.2	29.4	84.4	36.8
MR Ap	LSTM	78.2	19.8	78.8	21.4
	BERT	84.2	17.0	84.0	23.4

Table 15: Adversarial training

Dataset	Punctuation	Counts	Percentage
MR	Total Punctuation Count		
	.	2596	1.37E+00%
	,	1934	1.02E+00%
	'	1073	5.68E-01%
	-	1007	5.33E-01%
	"	146	7.72E-02%
	[58	3.07E-02%
]	58	3.07E-02%
	Internal Punctuation Count		
	'	922	5.02E-01%
	-	718	3.91E-01%
	/	17	9.26E-03%
]	4	2.18E-03%
	[2	1.09E-03%
	MNLI	Total Punctuation Count	
.		3499	1.22E+00%
,		2041	7.13E-01%
'		1460	5.10E-01%
-		527	1.84E-01%
)		156	5.45E-02%
(150	5.24E-02%
?		125	4.37E-02%
Internal Punctuation Count			
'		1347	4.81E-01%
-		496	1.77E-01%
.		68	2.43E-02%
,		49	1.75E-02%
?		15	5.36E-03%
SNLI		Total Punctuation Count	
	.	3523	1.99E+00%
	,	598	3.38E-01%
	-	113	6.39E-02%
	'	54	3.06E-02%
	"	28	1.58E-02%
	&	3	1.70E-03%
	/	1	5.66E-04%
	Internal Punctuation Count		
	-	113	6.55E-02%
	'	50	2.90E-02%
	.	5	2.90E-03%
	/	1	5.79E-04%
	,	1	5.79E-04%
	QNLI	Total Punctuation Count	
,		3755	9.98E-01%
.		2328	6.18E-01%
?		1983	5.27E-01%
-		734	1.95E-01%
'		715	1.90E-01%
"		672	1.79E-01%
(562	1.49E-01%
Internal Punctuation Count			
-		708	1.93E-01%
'		611	1.67E-01%
.		202	5.52E-02%
,		155	4.23E-02%
-		55	1.50E-02%
QQP		Total Punctuation Count	
	?	4220	2.10E+00%
	,	521	2.60E-01%
	"	470	2.34E-01%
	'	460	2.29E-01%
	.	349	1.74E-01%
	-	162	8.08E-02%
	(138	6.88E-02%
	Internal Punctuation Count		
	'	397	2.04E-01%
	-	146	7.50E-02%
	.	93	4.78E-02%
	/	89	4.57E-02%
	(30	1.54E-02%

Table 16: Frequency of total punctuation in samples and frequency of punctuation only found within words

Dataset	Model	Method	New Orig Acc [%]	Drop [%]
MR	CNN (76.6)	All	72.2	4.4
		Internal	74.4	2.2
		Internal With Exception	76.6	0
	LSTM (77)	All	72.8	4.2
		Internal	74.2	2.8
		Internal With Exception	77	0
	BERT (83.8)	All	81.2	2.6
		Internal	82.6	1.2
		Internal With Exception	83.8	0
	RoBERTa (88)	All	86.2	1.8
		Internal	87.8	0.2
		Internal With Exception	88	0
XLNet (87)	All	84.6	2.4	
	Internal	86.4	0.6	
	Internal With Exception	87	0	
MNLI	BERT (82.8)	All	80.8	2
		Internal	82.4	0.4
		Internal With Exception	82.4	0.4
	DistilBERT (80.6)	All	78.4	2.2
		Internal	80	0.6
		Internal With Exception	80.4	0.2
SNLI	BERT (91.2)	All	90.8	0.4
		Internal	91.2	0
		Internal With Exception	91.2	0
	DistilBERT (87)	All	86.6	0.4
		Internal	87	0
		Internal With Exception	87	0
QNLI	BERT (91.2)	All	87.2	4
		Internal	90.6	0.6
		Internal With Exception	90.8	0.4
	RoBERTa (92)	All	91.2	0.8
		Internal	92	0
		Internal With Exception	92	0
DistilBERT (86.2)	All	84.2	2	
	Internal	85.8	0.4	
	Internal With Exception	85.8	0.4	
QQP	BERT (90.4)	All	88.6	1.8
		Internal	90	0.4
		Internal With Exception	90	0.4
	DistilBERT (90.8)	All	88.6	2.2
		Internal	90.6	0.2
		Internal With Exception	90.8	0
	XLNet (91.2)	All	89.8	1.4
		Internal	91.2	0
		Internal With Exception	91.2	0

Table 17: Results when punctuation is removed

Dataset	Model (Orig Acc)	Method	After Attack Acc [%]	Average Time Taken [s]	Drop [%]
MR	CNN (76.6)	DWBP .	15.4	0.0412	61.2
		DWBP ,	14.6	0.0407	62
		DWBP "	27.2	0.0391	49.4
	LSTM (77)	DWBP .	19.4	0.0574	57.6
		DWBP ,	19.2	0.0564	57.8
		DWBP "	26.6	0.0544	50.4
	BERT (83.8)	DWBP .	18.4	0.4748	65.4
		DWBP ,	18.4	0.462	65.4
		DWBP "	29.4	0.4428	54.4
	RoBERTa (88)	DWBP .	19.4	0.499	68.6
		DWBP ,	18.6	0.4812	69.4
		DWBP "	34.6	0.4459	53.4
XLNet (87)	DWBP .	17.8	1.8583	69.2	
	DWBP ,	18	1.855	69	
	DWBP "	34	1.7026	53	
MNLI	BERT (82.8)	DWBP .	14	0.4317	68.8
		DWBP ,	12.6	0.4317	70.2
		DWBP)	11.6	0.4423	71.2
	DistilBERT (80.6)	DWBP .	12.8	0.2232	67.8
		DWBP ,	13.2	0.2195	67.4
		DWBP)	10.4	0.2231	70.2
BERT (91.2)	DWBP .	10	0.323	81.2	
	DWBP ,	10.6	0.3258	80.6	
	DWBP "	17	0.3175	74.2	
SNLI	DistilBERT (86.6)	DWBP .	9.6	0.1677	77
		DWBP ,	4	0.175	82.6
		DWBP "	16.8	0.168	69.8
QNLI	BERT (91.2)	DWBP .	25	0.6877	66.2
		DWBP ,	26.8	0.6731	64.4
		DWBP ?	25	0.7068	66.2
	RoBERTa (92)	DWBP .	28.6	0.756	63.4
		DWBP ,	32.2	0.7564	59.8
		DWBP ?	31.4	0.7678	60.6
	DistilBERT (86.2)	DWBP .	19	0.377	67.2
		DWBP ,	19.2	0.3675	67
		DWBP ?	23.2	0.3559	63
QQP	BERT (90.4)	DWBP ?	46.2	0.311	44.2
		DWBP .	48.6	0.3075	41.8
		DWBP "	49.4	0.305	41
	DistilBERT (90.8)	DWBP ?	43.4	0.169	47.4
		DWBP .	46.2	0.1698	44.6
		DWBP "	50.4	0.1617	40.4
	XLNet (91.2)	DWBP ?	47.4	1.3345	43.8
		DWBP .	47.6	1.3518	43.6
		DWBP "	53.8	1.3201	37.4

Table 19: Results when only one punctuation symbol type is used in the attack

Dataset	Model	Baseline	Finetune with no punctuation	Drop [%]
		Finetune with punctuation Eval Acc [%]	Eval Acc [%]	
MR	LSTM	79.8±0.5	78.9±0.4	0.9
	BERT	85.3±0.8	84.7±0.6	0.6
MNLI	BERT	84.9	83.5	1.4
SNLI	BERT	89.9	88.6	1.3

Table 18: Finetuning on no punctuation

MR (Negative)	A dark comedy that goes for sick and demented humor simply to do so . the movie is without intent .
TextFooler (Positive)	A dark comedy that goes for psychopathic and coot humor honestly to do so . the film is without object .
DWBP (Positive)	A dark comedy that goes for sick and demented humor simply to do so . the movie is withou't intent .
MNLI (Entailment)	Premise: Sit down, will you?" Tuppence sat down on the chair facing him. Hypothesis: He asked Tuppence to sit on a red chair.
TextFooler (Neutral)	He asked Tuppence to assisi on a flushed chair.
DWBP (Neutral)	He asked Tuppence to sit on a r'ed c'hair .

Table 20: Qualitative examples of DWBP vs TextFooler. **Bold** words represent a perturbed word

Self-Supervised Unimodal Label Generation Strategy Using Recalibrated Modality Representations for Multimodal Sentiment Analysis

Yewon Hwang and Jong-Hwan Kim
School of Electrical Engineering, KAIST
Daejeon, Republic of Korea
{ywhwang, johkim}@rit.kaist.ac.kr

Abstract

While multimodal sentiment analysis (MSA) has gained much attention over the last few years, the main focus of most work on MSA has been limited to constructing multimodal representations that capture interactions between different modalities in a single task. This was largely due to a lack of unimodal annotations in MSA benchmark datasets. However, training a model using only multimodal representations can lead to suboptimal performance due to insufficient learning of each uni-modal representation. In this work, to fully optimize learning representations from multimodal data, we propose SUGRM which jointly trains multimodal and unimodal tasks using recalibrated features. The features are recalibrated such that the model learns to weight the features differently based on the features of other modalities. Further, to leverage unimodal tasks, we auto-generate unimodal annotations via a unimodal label generation module (ULGM). The experiment results on two benchmark datasets demonstrate the efficacy of our framework.¹

1 Introduction

These days, we can easily spot AI systems in our society that serve or assist humans. Understanding human emotions has become a critical factor for these AI systems to seamlessly integrate into human’s life (Castillo et al., 2018; De Graaf and Allouch, 2013). However, understanding humans’ emotions is not a trivial task. This is because humans tend to express their feelings through multiple cues in a complex form. Emotions can be expressed simply through language, but they can also be manifested through facial expression, behaviors or even tone of voice (Morency et al., 2011). Moreover, sometimes these cues signal a compatible emotion, while other times they signal conflicting emotions,

e.g., positive language with a condescending tone of voice indicates sarcasm (Robins et al., 2009).

Taking this nature into account, multimodal sentiment analysis (MSA) has become an active field of research which aims to understand the affective state of humans through visual, acoustic, and textual features. In general, when working with multimodal data like in MSA, each modality contains both supplementary and complementary information to each other, providing richer information about the data. This leads to improved performance over using only one modality (Vaezi Joze et al., 2020). However, capturing information in each modality as well as modeling the interactions between different modalities still remain challenging tasks to unravel (Hazarika et al., 2020).

Most of the existing works on MSA revolve around learning a joint representation which encompasses information from all modalities through sophisticated fusion methods varying from tensor-based (Zadeh et al., 2017) to attention-based methods (Tsai et al., 2019; Rahman et al., 2020), where the learning process happens in a single task. Single task learning was a dominant learning framework in MSA particularly due to the nature of the benchmark datasets: CMU-MOSI (Zadeh et al., 2016) and CMU-MOSEI (Bagher Zadeh et al., 2018). Considering all modalities, only one comprehensive sentiment intensity value (i.e., multimodal label, y_m) is annotated in both datasets due to the laborious labeling process. Meaning, unimodal labels (y_t, y_a, y_v) are omitted in the datasets. However, a recent study (Yu et al., 2021) argued the absence of unimodal annotations hinders capturing modality-specific information and proposed a module that auto-generates unimodal annotations from the multimodal labels.

In this work, we propose a novel framework, SUGRM, which leverages a self-supervised unimodal label generation strategy using recalibrated modality representations for MSA. First, we recal-

¹Our code is available at: <https://github.com/skystarhyw/SUGRM>

brate modality representations using Modality Recalibration Module (MRM). This allows the model to dynamically adjust features based on the features of other modalities. Further, motivated by (Yu et al., 2021), we propose a new unimodal label generation module (ULGM), which generates unimodal annotations (y_t, y_a, y_v) based on the multimodal annotation (y_m) in a self-supervised manner.

Different from (Yu et al., 2021), which preserves feature space of each modality, we project features of each modality into a common semantic feature space. Thus, our ULGM hypothesizes the distance between two features in a common semantic feature space is proportional to the distance between the corresponding labels in a label space. This not only allows simpler calculation of the offset (see section 3.3), but also avoids the problem in (Yu et al., 2021); that is, when two distances from a multimodal feature 1) to the center of negative multimodal features and 2) to the center of positive multimodal features are approximately equal, the generated unimodal label diverges. This could lead to unstable learning, potentially causing the model to fall into a local minima.

Our experiment results not only empirically validate our hypothesis, but also prove that using recalibrated modality representation as well as our ULGM lead to enhanced performance. The main contributions of our work can be summarized as follows:

- We introduce Modality Recalibration Module (MRM) for MSA which recalibrates modality features based on features of other modalities.
- We design a novel unimodal label generation module (ULGM) to expand MSA to multi-task learning and jointly train unimodal and multimodal tasks.
- Not only does our method outperform the previous SOTA results, but the experiment results validate the effectiveness of our framework.

2 Related Work

Prior works of MSA mainly focused on improving fusion between multi-modalities as well as learning joint representations. In earlier works, early fusion (Pérez-Rosas et al., 2013; Poria et al., 2016) and late fusion (Zadeh et al., 2016) were popular fusion methods to combine the multiple modalities. Later, more sophisticated methods of

fusion were proposed using a multi-dimensional tensor (Zadeh et al., 2017), attention mechanism (Zadeh et al., 2018a,b), multi-stage fusion (Liang et al., 2018) and low rank tensors to improve efficiency of fusion (Liu et al., 2018). In (Wang et al., 2019), the authors dynamically adjusted a word representation by calculating a shift caused by accompanying nonverbal information. More recent works have focused on applying Transformer architecture to better capture interactions between modalities and learn feature representations. For instance, (Rahman et al., 2020) was directly built upon (Wang et al., 2019), but used pretrained Transformer based language models to improve the performance. (Tsai et al., 2019) proposed cross-modal attention to latently adapt a target modality from source modalities. (Cheng et al., 2021) reduced the computational burden in (Tsai et al., 2019), by generating sparse attention matrices and compressing a long sequence to a short sequence. Further, a multi-task learning approach has been applied in recent MSA (Akhtar et al., 2019; Yu et al., 2021) to increase data efficiency.

Taking inspiration from the previous work (Yu et al., 2021), we expand a learning framework of MSA to multi-task learning. The benefits of multi-task learning is that each task helps a learning process of other tasks. This allows the model to learn better generalized representations that are shared across the tasks. Further, we recalibrate features of each modality and efficiently model inter-, intra-modality relationships by adopting the work of (Hu et al., 2018; Vaezi Joze et al., 2020; Cheng et al., 2021).

3 Methodology

3.1 Problem Definition

We define the input to the model as $I_{s \in \{t,a,v\}}$ which is composed of three types of modalities-text, audio, and video. The goal of our model is to take I_s as input and predict a sentiment intensity $\hat{y} \in \mathbb{R}$. To aid the learning process, our model generates labels for each modality $y_s \in \mathbb{R}$ during training.

3.2 Overall Architecture

Our framework consists of multimodal and unimodal tasks where they share modality representations as shown in Figure 1.

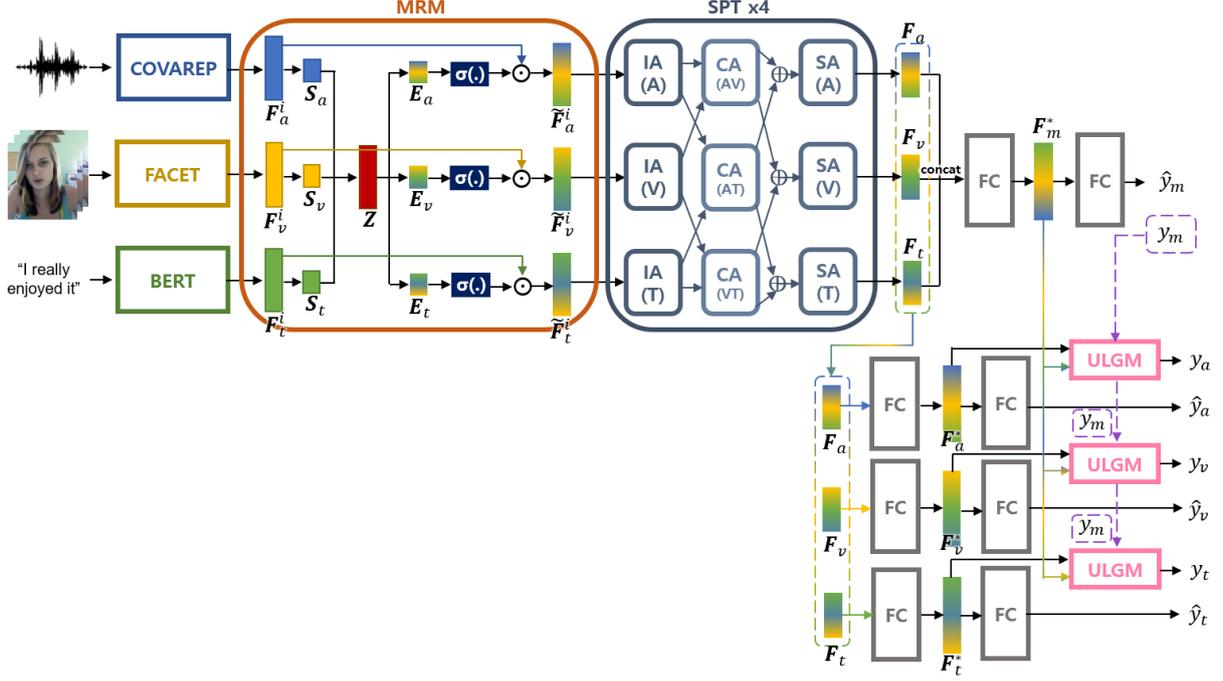


Figure 1: The overall architecture of SUGRM. The $y_a, y_v,$ and y_t are the unimodal annotations generated from our ULGM based on the human-annotated multimodal label y_m to enable supervised learning of the unimodal tasks. The $\hat{y}_a, \hat{y}_v, \hat{y}_t,$ and \hat{y}_m are the predicted sentiment values from the unimodal and multimodal tasks.

3.2.1 Multimodal Task

In the multimodal task, modality features ($F_{s \in \{t, a, v\}}^i$) are initially extracted from pretrained BERT (Devlin et al., 2019), COVAREP (Degottex et al., 2014), and FACET (iMotions, 2013) for textual, acoustic, and visual information, respectively. Then these features are passed through Modality Recalibration Module (MRM) for feature recalibration. After the features are recalibrated, the final feature representation of each modality is captured using Sparse Phased Transformer (SPT).

Modality Recalibration Module. MRM recalibrates modality features using squeeze and excitation (SE) technique (Hu et al., 2018). This particular idea was studied in the case of CNN in (Vaezi Joze et al., 2020). Here, we show how SE can be expanded to the MSA application. MRM receives $F_s^i \in \mathbb{R}^{l_s \times d_s}$ as input, where l_s is the sequence length and d_s is the feature dimension of s -modality, and squeeze the input along the sequence length using global average pooling:

$$S_s(d) = \frac{1}{l_s} \sum_{l=1}^{l_s} F_s^i(l, d),$$

where $s \in \{t, a, v\}$ and $d = 1, \dots, d_s$. Then the excitation process is performed to apply different weight calibrations for each modality. First,

squeezed features are concatenated and fed into a series of a fully connected network and ReLU to learn a global multimodal embedding Z :

$$Z = ReLU(W_z[S_t; S_a; S_v] + b_z).$$

Here, the fully connected network reduces feature dimension. Then we compute excitation signals using another fully connected network as follows:

$$E_s = W_s Z + b_s.$$

The second fully connected network restores the original feature dimension, adopting bottleneck architecture. The reason for this is to reduce the number of computations and improve generalization (Hu et al., 2018). Finally, the input features are recalibrated through a following gating mechanism:

$$\tilde{F}_s^i = 2 \times \sigma(E_s) \odot F_s^i,$$

where $\sigma(\cdot)$ is the sigmoid function and \odot is the element-wise product along the feature dimension. Since the numbers returned by sigmoid function (between 0 and 1) are multiplied by the original features, each feature is rescaled based on its importance. Finally, the textual, acoustic, and visual features after MRM can be described as follows:

$$\tilde{F}_s^i = MRM(F_s^i; \theta^{mrm}) \in \mathbb{R}^{l_s \times d_s},$$

where θ^{mrm} are the parameters of MRM.

Sparse Phased Transformer. SPT (Cheng et al., 2021) extracts the final feature representation of each modality using the recalibrated features. The motivation behind SPT is twofold: to extract more informative features by modeling intra- and inter-modalities (preferred over LSTM²) and to build a more efficient and lighter model (preferred over (Cheng et al., 2021)²). SPT alleviates the computational burden of the self-attention mechanism in the vanilla Transformer. Instead of generating a full attention matrix, SPT generates a sparse attention matrix to reduce computational complexity.³ Multimodal SPT is composed of input attention, cross attention, and self attention. Input attention (IA) compresses input sequence into hidden states. Then the hidden states of two different modalities are interacted through cross attention (CA). Finally, self attention (SA) refines the feature representations of each modality. For the technical details of SPT, refer to (Cheng et al., 2021) on which our implementation of SPT is based.

We denote the final feature representation for each modality as follows:

$$\mathbf{F}_s = SPT(\tilde{\mathbf{F}}_s^i; \theta^{spt}) \in \mathbb{R}^{d_s},$$

where SPT is the process of [IA→CA→SA] repeated 4 times and θ^{spt} are the parameters of SPT. Finally, the last element of the sequence is selected as a sequence representation.

To obtain a fusion representation, we concatenate each modality representation and project into a lower-dimensional feature space \mathbb{R}^{d_c} as follows:

$$\mathbf{F}_m^* = ReLU(\mathbf{W}_1^m [\mathbf{F}_t; \mathbf{F}_a; \mathbf{F}_v] + \mathbf{b}_1^m).$$

Lastly, the multimodal sentiment is predicted as follows:

$$\hat{y}_m = \mathbf{W}_2^m \mathbf{F}_m^* + b_2^m.$$

3.2.2 Unimodal Task

For the unimodal task, we use the feature representation of each modality obtained from the multimodal task ($\mathbf{F}_{s \in \{t, a, v\}}$). Then we map each feature representation into the same feature space as \mathbb{R}^{d_c} (i.e., a common semantic feature space) as follows:

$$\mathbf{F}_s^* = ReLU(\mathbf{W}_1^s \mathbf{F}_s + \mathbf{b}_1^s).$$

²Three options were considered as a final feature extractor: LSTM, multimodal Transformer (Cheng et al., 2021), and SPT (See Table 4).

³The authors of SPT (Cheng et al., 2021) claim that the number of parameters is reduced to 10% of (Tsai et al., 2019) which utilizes the vanilla Transformer encoder.

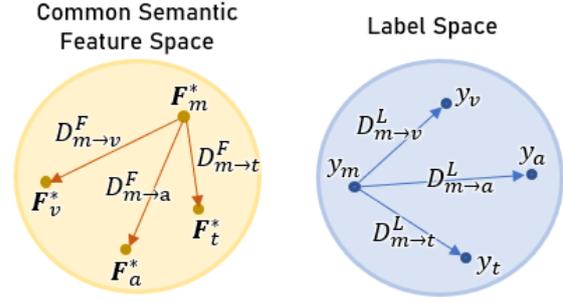


Figure 2: The distance from multimodal feature (\mathbf{F}_m^*) to s -modal feature (\mathbf{F}_s^*) in a common semantic feature space: $D_{m \rightarrow s}^F$, and the distance from multimodal label (y_m) to s -modal label (y_s) in a label space: $D_{m \rightarrow s}^L$.

Then the final sentiment prediction from each modality is obtained through an independent fully-connected layer:

$$\hat{y}_s = \mathbf{W}_2^s \mathbf{F}_s^* + b_2^s.$$

The unimodal tasks are trained using supervised learning, where labels for each modality are obtained via non-parametric Unimodal Label Generation Module (ULGM):

$$y_s = ULGM(y_m, \mathbf{F}_m^*, \mathbf{F}_s^*).$$

Finally, the multimodal task and three unimodal tasks are jointly trained.

3.3 ULGM

The goal of ULGM is to generate labels for each unimodality based on multimodal labels and modality representations. As shown in Figure 2, our ULGM is designed based on the notion that the distance between two features in a common semantic feature space is proportional to the distance between the corresponding labels in a label space:

$$D_{m \rightarrow s}^F \propto D_{m \rightarrow s}^L,$$

where $s \in \{t, a, v\}$. Our ULGM computes the offset of unimodal label y_s with respect to the multimodal label y_m based on the distance from the multimodal feature to each unimodal feature. We consider two factors when computing the offset: the magnitude and the direction.

Magnitude of offset. To calculate the offset, we argue that the maximum distance within the common semantic feature space is proportional to the maximum distance within the label space. In CMU-MOSI and -MOSEI datasets, the multimodal

labels range from -3 to +3, meaning the distance between multimodal features with labels -3 (\mathbf{F}_m^{*-3}) and +3 (\mathbf{F}_m^{*+3}) must correspond to the maximum distance within the common semantic feature space. Therefore, any $D_{m \rightarrow s}^F$ greater than the maximum distance is clipped to $D_{max}^F = \|\overline{\mathbf{F}_m^{*+3}} - \overline{\mathbf{F}_m^{*-3}}\|$:

$$D_{m \rightarrow s}^F = \begin{cases} \|\mathbf{F}_m^* - \mathbf{F}_s^*\|, & \text{if } D_{m \rightarrow s}^F \leq D_{max}^F, \\ D_{max}^F, & \text{otherwise,} \end{cases}$$

where $\overline{\mathbf{F}_m^{*+3}}$ and $\overline{\mathbf{F}_m^{*-3}}$ are the mean of \mathbf{F}_m^{*+3} and \mathbf{F}_m^{*-3} , respectively, and $\|\cdot\|$ is L2 normalization.

Based on our notion and the above argument, we can consider the following relationship from which we can obtain the magnitude of the offset from a multimodal label to an unimodal label:

$$D_{m \rightarrow s}^F / D_{max}^F = D_{m \rightarrow s}^L / D_{-3 \rightarrow +3}^L,$$

$$D_{m \rightarrow s}^L = \frac{D_{m \rightarrow s}^F}{D_{max}^F} D_{-3 \rightarrow +3}^L.$$

Direction of offset. In order to determine the direction of the offset, we identify the position of the s -modal feature with respect to the multimodal feature. To do that, we first take the average of the multimodal features with positive annotations ($\overline{\mathbf{F}_m^{*+}}$) and negative annotations ($\overline{\mathbf{F}_m^{*-}}$). Then we locate the multimodal and the s -modal features within this realm of feature space as shown in Figure 3. Using the distance from modality representations ($\mathbf{F}_{x \in \{m, t, a, v\}}^*$) to $\overline{\mathbf{F}_m^{*+}}$ and $\overline{\mathbf{F}_m^{*-}}$, we can determine the direction of the offset as follows:

$$Direction = \begin{cases} +, & \text{if } \frac{D_s^p}{D_s^n} < \frac{D_m^p}{D_m^n}, \\ -, & \text{if } \frac{D_s^p}{D_s^n} > \frac{D_m^p}{D_m^n}, \\ 0, & \text{if } \frac{D_s^p}{D_s^n} = \frac{D_m^p}{D_m^n}, \end{cases}$$

where $D_s^p = \|\mathbf{F}_s^* - \overline{\mathbf{F}_m^{*+}}\|$, $D_s^n = \|\mathbf{F}_s^* - \overline{\mathbf{F}_m^{*-}}\|$, $D_m^p = \|\mathbf{F}_m^* - \overline{\mathbf{F}_m^{*+}}\|$, and $D_m^n = \|\mathbf{F}_m^* - \overline{\mathbf{F}_m^{*-}}\|$. Finally, we obtain the unimodal label y_s as follows:

$$y_s = \begin{cases} y_m + D_{m \rightarrow s}^L, & \text{if direction is +,} \\ y_m - D_{m \rightarrow s}^L, & \text{if direction is -,} \\ y_m, & \text{if direction is 0.} \end{cases}$$

Unimodal Label Update Scheme. We update the generated unimodal labels using a momentum-based update policy (Yu et al., 2021) as follows:

$$y_s^e = \begin{cases} y_m & \text{for } e = 1, \\ \frac{e-1}{e+1} y_s^{(e-1)} + \frac{2}{e+1} y_s^e & \text{for } e > 1, \end{cases}$$

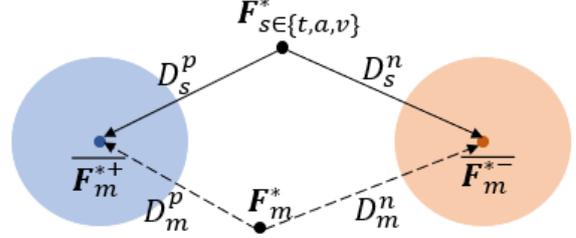


Figure 3: An illustration of positions of modality representations with respect to the mean of multimodal representations with positive labels ($\overline{\mathbf{F}_m^{*+}}$) and negative labels ($\overline{\mathbf{F}_m^{*-}}$) in the common semantic feature space.

where $s \in \{t, a, v\}$ and e is epoch. This scheme is used to mitigate the instability of labels that are generated at the beginning of epochs in which the learning of the modality features is trivial. This update scheme allows the labels generated in later epochs to have greater impact than the ones generated in earlier epochs. After a sufficient number of iterations, unimodal labels become stabilized, resulting in a stable training process of unimodal tasks. As can be seen in Figure 4, the labels stabilize within 15 epochs.

3.4 Objective Function for Training

For the objective function, we investigated three loss functions that are widely used in regression tasks: L1 loss, L2 loss, and Huber loss. Based on our loss ablation study (see Table 8 in Appendix), we use L1 loss as the objective function for both multimodal and unimodal tasks. We minimize the sum of the two loss functions over N training samples to optimize the entire model as follows:

$$L = \frac{1}{N} \sum_i (|\hat{y}_m^i - y_m^i| + \sum_s^{\{t, a, v\}} w_s^i * |\hat{y}_s^i - y_s^i|),$$

where the first term corresponds to the multimodal task, and the second term corresponds to the unimodal tasks optimization. Note the loss functions for the unimodal tasks are weighted by w_s^i , where $w_s^i = \tanh(|y_s^i - y_m^i|)$ (Yu et al., 2021) such that the model can target the samples with larger difference between the multimodal label and the generated unimodal label more rigorously during training.

4 Experimental Settings

4.1 Datasets

We use the two most popular English benchmark datasets for MSA: CMU-MOSI (Zadeh et al., 2016) and CMU-MOSEI (Bagher Zadeh et al., 2018). CMU-MOSI dataset consists of 2,199 labeled video clips taken from 93 videos by 89 speakers. The videos were crawled from YouTube and encompass opinions on movies, books, and products. Each video is annotated with sentiment on a [-3,3] range. CMU-MOSEI dataset is the most comprehensive dataset for sentiment analysis and emotion recognition which comprises more than 65 hours worth of 23,453 annotated video segments from 1,000 speakers addressing 250 different topics. Each video is annotated with sentiment on a [-3,3] range as well as six discrete emotions: happy, sadness, anger, disgust, surprise, and fear. We only utilize sentiment values from CMU-MOSEI in this task. See Table 6 in Appendix for the dataset split.

4.2 Baselines

We compare the performance of our model with previous state-of-the-art MSA models. The superscript A indicates the proposed method only works on the aligned settings, while UA indicates the proposed method works on both unaligned and aligned settings.⁴

EF-LSTM.^A Early Fusion LSTM concatenates the multimodal features at the input level.

LF-LSTM.^{UA} Late Fusion LSTM combines modality-wise decisions using a voting mechanism.

TFN.^A The Tensor Fusion Network (Zadeh et al., 2017) models intra- and inter-modality dynamics through multi-dimensional tensors.

RAVEN.^A The Recurrent Attended Variation Embedding Network (Wang et al., 2019) models nonverbal sequences and dynamically shifts word representations based on nonverbal cues.

MCTN.^A The Multimodal Cyclic Translation Network (Pham et al., 2019) learns robust joint representations via multimodal cyclic translations using a cycle consistency loss.

⁴Multimodal data in CMU-MOSI and MOSEI are loaded from different sources which come at different frequencies, making the multimodal data “unaligned” in terms of sequence length. (The lengths of text, audio, video segments are 50, 375, 500, respectively for the unaligned dataset.) These unaligned data have been preprocessed through CMU-Multimodal SDK (<https://github.com/A2Zadeh/CMU-MultimodalSDK>) to align different modalities such that they have the same sequence length of 50. Note, our method works on both aligned and unaligned settings.

MULT.^{UA} The Multimodal Transformer (Tsai et al., 2019) uses cross-modal attention to model interactions between asynchronous modalities and latently adapt one modality to another.

MAG-BERT.^A The Multimodal Adaptation Gate for BERT (Rahman et al., 2020) is an improvement of RAVEN which applies multimodal adaptation gate at the first layer of the BERT model.

SPT.^{UA} The multimodal Sparse Phased Transformer (Cheng et al., 2021) is an improvement of MulT in terms of efficiency by using a sampling function to generate a sparse attention matrix.

Self-MM.^{UA} The Self-Supervised Multi-task Multimodal sentiment analysis network (Yu et al., 2021) generates a unimodal label for each modality and jointly trains multimodal and unimodal tasks.

4.3 Implementation Details

We trained our framework using NVIDIA TITAN Xp and Intel i7-9700K. We use the batch size of 32 and Adam as the optimizer for both datasets. For more implementation details such as hyperparameters for each dataset, see Table 7 in Appendix.

4.4 Evaluation Metrics

We evaluate our model using four metrics: weighted binary F1 score (F1-Score), binary classification accuracy (Acc₂), Mean Absolute Error (MAE), and Pearson correlation (Corr). For F1-Score and Acc₂, we report the model performance in two ways: negative/non-negative (Zadeh et al., 2017) and negative/positive (Tsai et al., 2019).

5 Results and Analysis

5.1 Quantitative Results

Tables 1 and 2 show the experiment results on the aligned and unaligned MOSI and MOSEI datasets, respectively. Our model outperformed all of the previous SOTA baseline models on all metrics for the MOSI dataset, and achieved either SOTA or comparable-to-SOTA results on the MOSEI dataset for both the aligned and unaligned datasets. Note, CTC (Graves et al., 2006) was introduced to allow some models (Wang et al., 2019; Pham et al., 2019) that originally only work on the aligned dataset to work on the unaligned dataset in Table 2. Unlike the previous observation (Tsai et al., 2019), our model shows greater strength in the unaligned dataset than the aligned dataset. This is beneficial in that it allows omission of extra data alignment

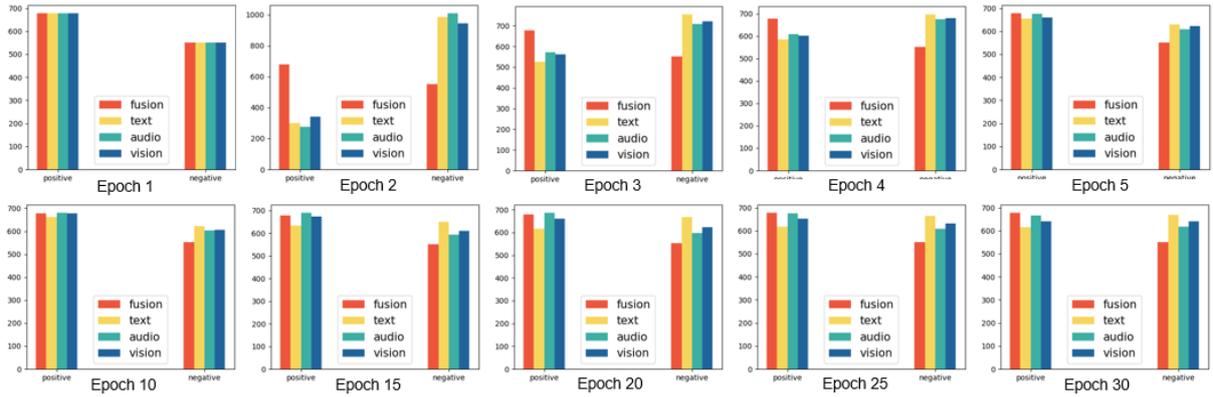


Figure 4: Visualization of the generated unimodal labels update process throughout epochs on CMU-MOSI dataset

step and data to have its inherent trait of unalignment, which could further facilitate real-time sentiment analysis.

5.2 Ablation Study

To explore the contributions of the unimodal tasks in our model, we conducted experiments using combinations of different unimodal tasks as shown in Table 3. The general trend of the results shows that incorporating the unimodal tasks leads to improvement in the model performance, which proves the effectiveness of our model. Particularly, using all three unimodal tasks along with the multimodal task resulted in substantial performance gain on all metrics compared to using the multimodal task alone on the MOSI dataset. An interesting trend on the MOSI dataset is that the performance rather decreased when only one of the unimodal tasks was added. However, we can observe that the addition of more than one unimodal task helps the model to achieve better results. On the other hand, introducing all the unimodal tasks (M,T,A,V) on the MOSEI dataset did not show as apparent performance gain as the MOSI dataset. However, we can easily observe a generally increasing trend in performance with the addition of unimodal tasks on the MOSEI dataset.

To compare our ULGM as well as the effectiveness of our architecture against that of Self-MM (Yu et al., 2021), we conducted an ablation study as shown in Table 4. Our model surpassed the performance of Self-MM via the combination of MRM, SPT, and ULGM_{ours} modules. To study the effectiveness of each module, we added MRM to Self-MM, replaced LSTM in Self-MM with SPT for learning sequence representation, and replaced ULGM_{Self-MM} with ULGM_{ours}. The addition of

MRM and the replacement of SPT on the MOSI dataset certainly led to improved performance but on a limited range of metrics. However, the replacement of ULGM_{ours} significantly increased the performance on all metrics. Results on the MOSEI dataset show a notable performance boost in all tasks across a wide range of metrics. Particularly, the replacement of SPT, which showed trivial results on the MOSI dataset, played an important role in improving the performance on the MOSEI dataset.

Similarly, we removed or replaced MRM, SPT, and ULGM_{ours} to evaluate their contribution to our model. First, we removed MRM, replaced SPT with the vanilla Transformer encoder (TE) (Tsai et al., 2019) and LSTM, and replaced ULGM_{ours} with ULGM_{Self-MM}. The results in Table 4 predominantly show that the inclusion of all modules results in the best performance. Replacing SPT with the vanilla Transformer encoder and ULGM_{ours} with ULGM_{Self-MM} led to an increase in certain metrics. However, not only the improvement is minuscule for both replacements, but the opportunity cost for exchanging computational efficiency with such minuscule improvement is rather counterproductive particularly for the SPT → TE replacement.

5.3 Qualitative Results

To evaluate the quality of the generated labels of each modality, we display four samples from the CMU-MOSI dataset in Table 5. We observe that the generated unimodal annotations are generally in line with the descriptions from the text, acoustic, and visual information. This further confirms the efficacy of our ULGM.

Table 1: Results on the aligned CMU-MOSI and CMU-MOSEI datasets. In Acc₂ and F1-Score, the left side of the “/” is the “negative/non-negative” method and the right side is the “negative/positive” method.

Model	MOSI				MOSEI			
	F1-Score	Acc ₂	MAE	Corr	F1-Score	Acc ₂	MAE	Corr
EF-LSTM	-/75.6	-/75.8	1.053	0.613	-/78.8	-/79.1	0.665	0.621
LF-LSTM	-/75.4	-/76.4	1.037	0.620	-/80.0	-/79.4	0.625	0.655
TFN	74.1/75.2	74.8/76.0	0.955	0.649	-	-	-	-
RAVEN	-/76.6	-/78.0	0.915	0.691	-/79.5	-/79.1	0.614	0.662
MCTN	-/79.1	-/79.3	0.909	0.676	-/80.6	-/79.8	0.609	0.670
MuT	-/82.8	-/83.0	0.871	0.698	-/82.3	-/82.5	0.580	0.703
SPT	-/82.9	-/82.8	-	-	-/82.8	-/82.6	-	-
MAG-BERT	82.4/84.0	82.5/84.0	0.778	0.766	81.7/84.7	81.3/84.8	0.567	0.742
Self-MM	82.3/84.4	82.4/ 84.5	0.736	0.786	83.2/85.0	82.9/84.8	0.533	0.766
Ours	82.8/84.5	82.8/84.5	0.723	0.798	83.9/85.1	83.9/85.0	0.541	0.758

Table 2: Results on the unaligned CMU-MOSI and CMU-MOSEI datasets. Note that CTC method (Graves et al., 2006) was employed to EF-LSTM, RAVEN, and MCTN to apply these models on the unaligned setting.

Model	MOSI				MOSEI			
	F1-Score	Acc ₂	MAE	Corr	F1-Score	Acc ₂	MAE	Corr
EF-LSTM+CTC	-/74.5	-/73.6	1.078	0.542	-/75.9	-/76.1	0.680	0.585
LF-LSTM	-/77.8	-/77.6	0.988	0.624	-/78.2	-/77.5	0.624	0.656
RAVEN+CTC	-/73.1	-/72.7	1.076	0.544	-/75.7	-/75.4	0.664	0.599
MCTN+CTC	-/76.4	-/75.9	0.991	0.613	-/79.7	-/79.3	0.631	0.645
MuT	-/81.0	-/81.1	0.889	0.686	-/81.6	-/81.6	0.591	0.694
SPT	-/81.3	-/81.2	-	-	-/82.7	-/82.4	-	-
Self-MM	82.8/84.6	82.9/84.6	0.733	0.780	82.0/ 84.6	81.7/ 84.7	0.530	0.765
Ours	84.3/86.3	84.4/86.3	0.703	0.800	83.6/84.0	83.7/84.4	0.544	0.748

Table 3: An ablation study on the benefits of the unimodal tasks using the unaligned datasets. The bold numbers indicate the best performance, and the underlined numbers indicate enhanced performance from introducing the unimodal tasks to the multimodal task.

Model	MOSI				MOSEI			
	F1-Score	Acc ₂	MAE	Corr	F1-Score	Acc ₂	MAE	Corr
M	82.5/84.1	82.5/84.0	0.755	0.779	81.5/84.7	80.9/84.7	0.539	0.759
M,V	81.1/82.1	81.1/82.0	0.774	0.757	79.5/83.7	78.9/83.6	0.543	0.752
M,A	81.9/83.6	81.9/83.5	0.764	0.770	<u>82.7/85.2</u>	<u>82.4/85.3</u>	<u>0.532</u>	0.763
M,T	81.0/81.5	80.9/81.4	0.773	0.779	80.8/83.7	80.4/83.8	0.530	0.763
M,A,V	83.6/85.0	83.5/84.9	0.731	0.782	81.6/84.4	83.3/84.6	0.533	0.757
M,A,T	<u>82.7/84.2</u>	<u>82.7/84.2</u>	0.804	0.762	82.9/84.5	<u>82.8/84.8</u>	<u>0.535</u>	0.752
M,V,T	<u>83.6/84.7</u>	<u>83.5/84.6</u>	<u>0.748</u>	0.778	82.9/82.7	83.4/83.4	0.540	0.748
M,T,A,V	84.3/86.3	84.4/86.3	0.703	0.800	83.6/84.0	83.7/84.4	0.544	0.748

Table 4: An ablation study on the contribution of MRM, SPT, and our ULGM using the unaligned datasets. The bold numbers indicate the best performance, and the underlined numbers indicate enhanced performance compared to the baseline model. Superscript A, RP, and RM indicate added, replaced, and removed module, respectively.

Baseline	Added/Removed/ Replaced Module	MOSI				MOSEI			
		F1-Score	Acc ₂	MAE	Corr	F1-Score	Acc ₂	MAE	Corr
Self-MM	-	82.8/84.6	82.9/84.6	0.733	0.780	82.0/84.6	81.7/84.7	0.530	0.765
	MRM ^A	82.4/84.1	82.5/84.2	<u>0.718</u>	0.791	83.5/85.0	83.3/85.1	0.542	0.756
	SPT ^{RP}	82.8/84.3	82.8/84.3	0.735	<u>0.785</u>	<u>82.7/85.6</u>	<u>82.3/85.7</u>	0.534	0.771
	ULGM _{ours} ^{RP}	83.5/85.6	83.7/85.7	0.710	<u>0.790</u>	<u>83.0/85.3</u>	<u>82.7/85.3</u>	0.538	0.757
Ours	-	84.3/86.3	84.4/86.3	0.703	0.800	83.6/84.0	83.7/ 84.4	0.544	0.748
	MRM ^{RM}	81.5/82.9	81.5/82.8	0.761	0.767	79.2/83.4	78.5/83.4	0.541	0.746
	TE ^{RP}	84.2/85.6	84.1/85.5	0.720	0.802	82.1/81.9	83.8/82.8	0.553	0.750
	LSTM ^{RP}	79.2/81.7	79.5/81.9	0.801	0.740	77.3/82.1	76.5/82.0	0.556	0.744
	ULGM _{Self-MM} ^{RP}	82.1/83.3	82.1/83.2	0.726	0.797	0.79.5/ 84.1	78.8/84.0	0.541	0.756

Table 5: Four samples from the CMU-MOSI dataset. It shows the predictions from each modality as well as the generated unimodal annotations (S_G , where $S \in \{T, A, V\}$) during training.

Text	Acoustic	Visual	Prediction	Annotation
"Everytime that was like a jump everyone jumped,"	Fast paced slightly thrilled	slightly smiling	M: 0.1, T: 0.1 A: 0.5, V: 0.7	M: 0.8, T _G : 0.6 A _G : 0.9, V _G : 0.7
"I was really hoping that this one be just as good."	Monotonic and emphasis on "really"	Slightly frowning	M: -0.1, T: -0.2 A: 0.5, V: 0.2	M: -0.8, T _G : -0.3 A _G : 0.0, V _G : -0.7
"Looks exactly the same as this character in Defiance."	Relaxed and firm	Squinting eye and raising eyebrows	M: 0.2, T: -0.1 A: 0.5, V: 0.3	M: 0.2, T _G : 0.1 A _G : 0.7, V _G : 0.1
"I don't know what they are complaining about it."	High pitched and emphasis on "what"	smiling and head roll on "what"	M: 1.1, T: 0.3 A: 1.7, V: 1.6	M: 1.8, T _G : 0.9 A _G : 1.5, V _G : 1.5

6 Conclusion and Future Work

In this paper, we proposed SUGRM, a novel framework for multimodal sentiment analysis (MSA) which incorporates unimodal subtasks to aid the learning process of the multimodal task. To enable this, we first designed Modality Recalibration Module (MRM) so that features of each modality are recalibrated based on the features of other modalities. Then, we designed a unimodal label generation module (ULGM) based on the notion that the distance between two features in a common semantic feature space is proportional to the distance between the corresponding labels in a label space. From this, ULGM was able to generate unimodal annotations from the multimodal label in a self-supervised manner, which saved a tremendous amount of human labor. The experiment results validated our notion as well as the reliability of the unimodal labels generated from our ULGM.

For future work, expanding the framework to jointly train sentiment and emotion tasks could be worthwhile. Recently (Akhtar et al., 2019) proposed that MSA and Multimodal Emotion Recognition are closely correlated; therefore their tasks can be carried out jointly. Applying contrastive learning for different emotion classes and exploiting correlation between sentiment and emotion could help achieve better results in both tasks.

Limitations

A limitation of our work is that the initial features for audio and video are extracted using off-the-shelf frameworks: COVAREP and FACET. Therefore these features are fixed and cannot be further fine-tuned unlike the text features which are fine-tuned during training. Working with fixed features, compared to dynamic features which can be adjusted via learning, inevitably results in subpar

performance. We expect this limitation can be alleviated by making our framework completely end-to-end by using raw audio and video data and introducing learning-based audio and video feature extraction modules. However, using raw data can exponentially increase memory usage which is another challenge that needs to be considered. Further, by introducing additional MRM and SPT modules, our method took approximately twice the time as the (Yu et al., 2021) during inference using the unaligned MOSI dataset.⁵ Double in inference time hinders the community's strive to build faster and more compact models.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback. This work was supported by the Institute of Information & communications Technology Planning & evaluation(IITP) grant funded by the Korea government(MSIT) (No.2020-0-00842, Development of Cloud Robot Intelligence for Continual Adaptation to User Reactions in Real Service Environments).

References

Md Shad Akhtar, Dushyant Chauhan, Deepanway Ghosal, Soujanya Poria, Asif Ekbal, and Pushpak Bhattacharyya. 2019. [Multi-task learning for multimodal emotion recognition and sentiment analysis](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 370–379, Minneapolis, Minnesota. Association for Computational Linguistics.

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria,

⁵After 10 runs, the average inference time for our method was approximately 0.775 seconds, while (Yu et al., 2021) was 0.378 seconds.

- Erik Cambria, and Louis-Philippe Morency. 2018. [Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246, Melbourne, Australia. Association for Computational Linguistics.
- José Carlos Castillo, Álvaro Castro-González, Fernando Alonso-Martín, Antonio Fernández-Caballero, and Miguel Ángel Salichs. 2018. Emotion detection and regulation from personal assistant robot in smart environment. In *Personal assistants: Emerging computational technologies*, pages 179–195. Springer.
- Junyan Cheng, Iordanis Fostirooulos, Barry Boehm, and Mohammad Soleymani. 2021. [Multimodal phased transformer for sentiment analysis](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2447–2458, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Maartje MA De Graaf and Somaya Ben Allouch. 2013. Exploring influencing variables for the acceptance of social robots. *Robotics and autonomous systems*, 61(12):1476–1486.
- Gilles Degottex, John Kane, Thomas Drugman, Tuomo Raitio, and Stefan Scherer. 2014. Covarep—a collaborative voice analysis repository for speech technologies. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 960–964. IEEE.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.
- Devamanyu Hazarika, Roger Zimmermann, and Soujanya Poria. 2020. Misa: Modality-invariant and-specific representations for multimodal sentiment analysis. In *Proceedings of the 28th ACM international conference on multimedia*, pages 1122–1131.
- Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141.
- iMotions. 2013. [Facet imotions biometric research platform](#).
- Paul Pu Liang, Ziyin Liu, AmirAli Bagher Zadeh, and Louis-Philippe Morency. 2018. [Multimodal language analysis with recurrent multistage fusion](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 150–161, Brussels, Belgium. Association for Computational Linguistics.
- Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. 2018. [Efficient low-rank multimodal fusion with modality-specific factors](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2247–2256, Melbourne, Australia. Association for Computational Linguistics.
- Louis-Philippe Morency, Rada Mihalcea, and Payal Doshi. 2011. Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 169–176.
- Verónica Pérez-Rosas, Rada Mihalcea, and Louis-Philippe Morency. 2013. [Utterance-level multimodal sentiment analysis](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 973–982, Sofia, Bulgaria. Association for Computational Linguistics.
- Hai Pham, Paul Pu Liang, Thomas Manzini, Louis-Philippe Morency, and Barnabás Póczos. 2019. Found in translation: Learning robust joint representations by cyclic translations between modalities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6892–6899.
- Soujanya Poria, Iti Chaturvedi, Erik Cambria, and Amir Hussain. 2016. Convolutional mkl based multimodal emotion recognition and sentiment analysis. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 439–448. IEEE.
- Wasifur Rahman, Md Kamrul Hasan, Sangwu Lee, AmirAli Bagher Zadeh, Chengfeng Mao, Louis-Philippe Morency, and Ehsan Hoque. 2020. [Integrating multimodal information in large pretrained transformers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2359–2369, Online. Association for Computational Linguistics.
- Diana L Robins, Elinora Hunyadi, and Robert T Schultz. 2009. Superior temporal activation in response to dynamic audio-visual emotional cues. *Brain and cognition*, 69(2):269–278.
- Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. [Multimodal transformer for unaligned multimodal language sequences](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6558–6569, Florence, Italy. Association for Computational Linguistics.

- Hamid Reza Vaezi Joze, Amirreza Shaban, Michael L. Iuzzolino, and Kazuhito Koishida. 2020. Mmtm: Multimodal transfer module for cnn fusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yansen Wang, Ying Shen, Zhun Liu, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. 2019. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7216–7223.
- Wenmeng Yu, Hua Xu, Ziqi Yuan, and Jiele Wu. 2021. Learning modality-specific representations with self-supervised multi-task learning for multimodal sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10790–10797.
- Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. **Tensor fusion network for multimodal sentiment analysis**. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1114, Copenhagen, Denmark. Association for Computational Linguistics.
- Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2018a. Memory fusion network for multi-view sequential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Amir Zadeh, Paul Pu Liang, Soujanya Poria, Prateek Vij, Erik Cambria, and Louis-Philippe Morency. 2018b. Multi-attention recurrent network for human communication comprehension. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*.

A Appendices

A.1 Dataset Split

Table 6: Train, validation, test set split for CMU-MOSI and CMU-MOSEI datasets.

Dataset	# Train	# Valid	# Test	# All
MOSI	1284	229	686	2199
MOSEI	16326	1871	4659	22856

A.2 Hyper-parameter Settings

Table 7: Hyper-parameters used in the two datasets. The second half of the hyper-parameters (bottom row) are for the SPT.

Hyper-parameter	CMU-MOSI	CMU-MOSEI
Batch size	32	32
LR for BERT	$5e-5$	$5e-5$
LR for others	$1e-2$	$1e-3$
output dropout	0.3	0.1
# Encoder layer	4	4
# Head	8	4
Embed size	32	32
Attn dropout	0.3	0.1
ReLU dropout	0.3	0.1
Residual dropout	0.3	0.1
Embed dropout	0.3	0.2

A.3 Loss Function Ablation Study

Table 8: Loss function ablation study on the unaligned MOSI dataset. In Acc_2 and F1-Score, the left side of the “/” is the “negative/non-negative” method and the right side is the “negative/positive” method.

Loss type	F1-Score	Acc_2	MAE	Corr
L1 loss	84.3/86.3	84.4/86.3	0.703	0.800
L2 loss	80.8/81.0	80.8/81.0	0.832	0.737
Huber loss	78.1/79.2	78.2/79.2	0.818	0.744

Fighting FIRE with FIRE: Assessing the Validity of Text-to-Video Retrieval Benchmarks

Pedro Rodriguez,*
Mahmoud Azab, Becka Silvert, Renato Sanchez,
Linzy Labson, Hardik Shah, Seungwhan Moon
Meta AI

Abstract

Searching troves of videos with textual descriptions is a core multimodal retrieval task. Owing to the lack of a purpose-built dataset for text-to-video retrieval, video captioning datasets have been re-purposed to evaluate models by (1) treating captions as positive matches to their respective videos and (2) assuming all other videos to be negatives. However, this methodology leads to a fundamental flaw during evaluation: since captions are marked as relevant *only* to their original video, many alternate videos *also* match the caption, which introduces false-negative caption-video pairs. We show that when these false negatives are corrected, a recent state-of-the-art model gains 25% recall points—a difference that threatens the validity of the benchmark itself. To diagnose and mitigate this issue, we annotate and release 683K additional caption-video pairs. Using these, we recompute effectiveness scores for three models on two standard benchmarks (MSR-VTT and MSVD). We find that (1) the recomputed metrics are up to 25% recall points higher for the best models, (2) these benchmarks are nearing saturation for Recall@10, (3) caption length (generality) is related to the number of positives, and (4) annotation costs can be mitigated through sampling. We recommend retiring these benchmarks in their current form, and we make recommendations for future text-to-video retrieval benchmarks.

1 Introduction

Text-to-video retrieval (TVR) is a challenging multimodal retrieval task (Hu et al., 2011) with practical applications ranging from web search to organizing media collections (Lew et al., 2006). To measure TVR model improvement—despite a dearth of purpose-built TVR benchmarks—researchers created benchmarks by re-purposing video captioning datasets such as MSR-VTT (Xu et al.,

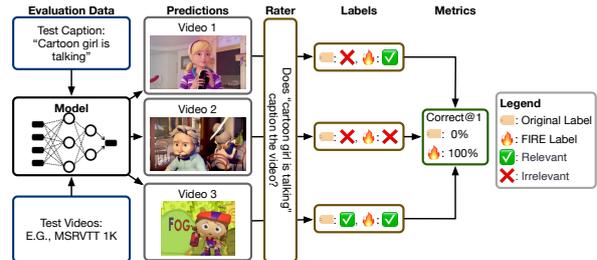


Figure 1: MSR-VTT and MSVD have one positive video per caption (each video’s caption). Captions often match multiple videos, leading to false negatives. When models rank false negatives highly, model quality is understated (full example in Appendix Figure 5). This leads to evaluations where reported metrics do not reflect their true value and are therefore not internally valid (§2.2.1).

2016), MSVD (Chen and Dolan, 2011), and ActivityNet (Heilbron et al., 2015; Krishna et al., 2017). Early work established an evaluation paradigm that treated captions as search queries over the collection of captioned videos (Zhang et al., 2018; Yu et al., 2018; Gabeur et al., 2020); each caption and their corresponding video are positives (relevant) during retrieval, and all other caption-video pairs are negatives (irrelevant).

However, even a cursory inspection of videos and captions reveals many additional positive caption-video pairs (§2). In current benchmarks, *true positives* that are not the video’s original caption are falsely assumed to be *negatives*. Wray et al. (2021) first identified this fundamental, false-negative problem in TVR evaluation; our work builds on this by quantifying the absolute metric differences that false negatives induce (see discussion in §6). Accurate absolute metrics are crucial in industrial settings where deployment criteria are often defined by minimum quality targets. These **False Implicit Relevance** labels introduce measurement error—e.g., CLIP4CLIP’s (Luo et al., 2021) Recall@1 is underestimated by 25% points (§2.2). We estimate measurement error by

*Correspondence to me@pedro.ai

annotating 683K additional caption-video pairs, which we call the FIRE 🔥 dataset (§3).¹

A core measurement principle is that operationalized metrics should strongly correlate to the quantity they intend to measure (Mathison, 2004; Liao et al., 2021). For example, Recall@K operationalizes the intent to measure retrieval quality. Label errors are a common way that measurements are invalidated (Bowman and Dahl, 2021; Northcutt et al., 2021). Our work shows that since TVR metrics are computed with false negative label errors, Recall@K does not accurately reflect retrieval quality, which negates the measurement’s validity. In the remainder of this paper, we posit rationales of why models gain different score boosts (§4.1) and estimate how useful the FIRE dataset is for evaluating future models (§4.2 and §4.3).

To conclude, we review the implications of our findings. Looking to the past, retrieval effectiveness has been understated for some models, which gives an overly pessimistic view of recent advances (Bowman, 2022). Critically, our results also suggest that the MSR-VTT benchmark is nearing saturation and should be retired soon in favor of a purpose-made benchmark. Looking outward, we identify structurally similar benchmarks—such as photo retrieval—that likely also have the same **False Implicit Relevance** problem. A successful benchmark should avoid the pitfalls we identify in this paper, be faithful to the real-world user task it targets (Rowe and Jain, 2005; de Vries et al., 2020), improve reproducibility, and evolve (§7).

2 Text-to-Video Retrieval Evaluation

This section reviews current TVR evaluation practices using two concepts: *internal validity* (Campbell, 1957, §2.2.1) and *construct validity* (Tague-Sutcliffe, 1992, §2.2.2). *Internal validity* refers to whether an evaluation reliably establishes a cause-effect relationship between the measured dependent variable and the independent variable to be estimated (Brewer and Crano, 2014; Liao et al., 2021). In TVR evaluations, false negatives confound model quality and label errors (i.e., is the model wrong or is the label wrong?) which makes *reliably* establishing cause (model quality) and effect (retrieval score) difficult. *Construct validity* “pertains to the degree to which the measure of a construct sufficiently measures the intended concept” (O’Leary-Kelly and J. Vokurka, 1998)—in

¹Data and Code: pedro.ai/multimodal-retrieval-evaluation.

TVR evaluations, an important intended concept is real-world search quality. *Construct validity* asks: can we expect that measuring retrieval quality with the benchmarks at hand generalizes to real-world search quality? This section argues that TVR evaluations are not internally valid or construct valid.

2.1 Model Evaluation

Multimodal retrieval evaluations typically focus on two tasks: text-to-video and video-to-text retrieval. The first task’s goal is—given a text query—to retrieve videos that match; the second task’s goal is—given a video—to retrieve the matching queries. The applications of text-to-video search are straightforward: it is useful for searching the web and personal media.² Since the applications of TVR are clear, and the false-negative problem is present in both tasks, here we focus on TVR.

The MSR-VTT and MSVD Datasets: It is standard for TVR evaluations (Zhang et al., 2018; Yu et al., 2018; Gabeur et al., 2020) to report on MSR-VTT and MSVD, so in the interest of comparability, we use these benchmarks too. Although these datasets were originally meant for evaluating video captioning models, they have been repurposed for TVR (Zhang et al., 2018; Gabeur et al., 2020). In this paper, we focus our investigation on MSR-VTT and MSVD since they are the most prevalent in prior work. MSR-VTT consists of 10K videos, 1K of which are in the test split. Each video has twenty captions, but for evaluation, only one (arbitrarily chosen) caption is used. MSVD contains 1,970 videos, 960 of which are in the test split. Videos have about forty captions; unlike with MSR-VTT, retrieval quality for each caption is evaluated.

Fundamentally, both MSR-VTT and MSVD are video captioning datasets—not retrieval datasets. MSVD addressed the lack of standard benchmarks for paraphrasing (Chen and Dolan, 2011). In the original task, annotators selected short clips from YouTube, watched the clip, and wrote a sentence describing its contents. The process was repeated for each video, with each sentence being written by a new annotator. This conditional independence—given the video—resulted in a diverse set of captions. MSR-VTT captions were collected similarly: independent annotators captioned the same video. Videos were sourced from the output of a commercial video search engine (Xu et al., 2016). In both

²The applications of video-to-text retrieval—that are not simply captioning—are not clear to us.

datasets, video captions are used as search queries and labeled relevant to the original video.

Metrics: Previous TVR work (Zhang et al., 2018; Yu et al., 2018; Gabeur et al., 2020; Luo et al., 2020; Zhu and Yang, 2020; Li et al., 2020; Xu et al., 2021; Park et al., 2022) reports Recall@K (R@K)³ and sometimes supplemental metrics such as median or mean rank of the first correct result. However, R@K in TVR work differs from the textbook information retrieval definition (Manning et al., 2008, p. 155) where

$$\text{R@K} = \frac{\# \text{ retrieved positives in top K}}{\# \text{ total positives in collection}}. \quad (1)$$

In TVR work, query retrieval results are scored one if a relevant video is in the top K and zero otherwise. The traditional definition of Recall@K *only* reduces to this when there is *exactly* one positive in the collection but is not comparable when there are multiple positives per caption—as in this case.

With the difference now salient, we avoid confusion by defining a new quantity Correct@K (C@K) which is 1 if at least one positive is in the top K and 0 otherwise. Correct@K naturally reduces to Recall@K—as defined in prior work—when there is exactly one positive, but handles the additional positives in our work. We recommend reporting Correct@K as well as mean average precision (Su et al., 2015; Mitra and Craswell, 2018, MAP), a metric widely used in Information Retrieval.

The drawback of Correct@K—shared by median (or mean) rank to first positive—is that it does not directly factor in rank order when there are multiple positives in retrieved results, only coarsely factoring in rank via K value. MAP (Mitra and Craswell, 2018, p. 19) is calculated by taking the mean of

$$\text{AvgPrec}_q = \frac{\sum_{\langle i,v \rangle \in R_q} \text{Prec}_{q,i} \times \text{rel}_q(v)}{\sum_{v \in V} \text{rel}_q(v)} \quad (2)$$

for each test query q where i is a video’s position in the ranked list R_q of videos, v is a video in collection V , and $\text{rel}_q(v)$ denotes whether query q is relevant to video v . Intuitively, this translates to calculating the mean of Precision@K for every K where a positive occurs in ranked predictions R_q . In all experiments, we report Correct@K and MAP.

2.2 Questioning the Validity of Evaluations

In this section, we experimentally argue that current TVR evaluations are not *internally valid*. Then

³Typical K values include 1, 5, 10, and 50.

we argue that they are not *construct valid* by considering actual use-cases for video search.

2.2.1 Internal Validity

If an evaluation metric is internally valid (Liao et al., 2021), then model effectiveness (cause) should be *accurately and reliably* reflected in metrics (effect) (Brewer and Crano, 2014). A central hypothesis of this paper is that the prevalence of false negatives invalidates the cause-effect relationship between measured model effectiveness and actual effectiveness—i.e., that correcting false negatives will significantly change metrics.⁴

To test this hypothesis, we build the FIRE dataset, which Fixes Implicit Relevance Errors. We detail the dataset later (§3), but in short, we take strong retrieval models from the past few years and annotate their top ten predictions on both MSR-VTT and MSVD. This process—called system pooling—has been used for decades in information retrieval (Spark-Jones, 1975) and, by construction, eliminates implicit false negatives.⁵ For MSR-VTT, we collect annotations from TeachText (Croitoru et al., 2021), Support-Set Bottlenecks (Patrick et al., 2021, SSB), and CLIP4CLIP (Luo et al., 2021) models; for MSVD, we collect annotations from TeachText and CLIP4CLIP models.^{6,7} Next, we compute model scores using the original positives and compare them to scores calculated with *both* the original positives *and* the new positives in FIRE.

Table 1 clearly demonstrates that FIRE annotations reveal large metric differences in both MSR-VTT and MSVD. For example, the C@1 score of CLIP4CLIP is understated by 25% points, and its C@10 score arguably saturates the benchmark at 95.7%. Even “small” differences such as those for TeachText and SSB are on par with the differences used to claim state-of-the-art results. False negatives directly cause high measurement error, which invalidates the internal validity of the benchmark.

⁴We do not see rank changes in our three models, but score differences suggest that ranks may change with more models.

⁵By implicit, we mean false negative from the lack of labeling and presuming non-positives are (implicitly) negative. There may still be false negatives arising from human error during annotation.

⁶We prioritize models that are (1) publicly available and (2) have sufficient documentation to reproduce.

⁷Annotating MSR-VTT predictions translates to $1,000 * 10 = 10\text{K}$ annotations since only one caption per video is used. This is easy compared to MSVD annotation, which uses tens of captions per video.

Dataset	Metric	TeachText	SSB	CLIP4CLIP
MSR-VTT	C@1	24.1 (23.3 + 0.800)%	27.3 (26.8 + 0.500)%	67.4 (42.4 + 25.0)%
MSR-VTT	C@5	53.2 (50.9 + 2.30)%	55.9 (54.5 + 1.40)%	90.7 (70.4 + 20.3)%
MSR-VTT	C@10	67.0 (64.8 + 2.20)%	68.9 (66.3 + 2.60)%	95.7 (80.2 + 15.5)%
MSR-VTT	AP	36.1 (35.8 + 0.296)%	39.3 (39.2 + 0.0374)%	69.5 (54.9 + 14.7)%
MSVD	C@1	34.7 (19.6 + 15.2)%	Not Annotated	65.3 (46.6 + 18.8)%
MSVD	C@5	64.7 (48.9 + 15.8)%	Not Annotated	89.6 (76.8 + 12.8)%
MSVD	C@10	76.1 (63.9 + 12.2)%	Not Annotated	94.0 (85.4 + 8.61)%
MSVD	AP	44.3 (33.1 + 11.2)%	Not Annotated	71.3 (59.7 + 11.6)%

Table 1: The table shows the impact of FIRE annotations on MSR-VTT and MSVD text-to-video retrieval metrics. “A (B + C)” has metrics computed with FIRE positives (A), only original positives (B), and the delta (C). The deltas emphasize the deleterious effects of false negatives: CLIP4CLIP’s C@1 on MSR-VTT is understated by 25% points.

2.2.2 Construct Validity

In addition to problems with internal validity, we posit that TVR evaluations are also not *construct valid* (Cronbach and Meehl, 1955; O’Leary-Kelly and J. Vokurka, 1998). Construct validity is related to “how closely our evaluations hit the mark in appropriately characterizing the actual anticipated behaviour of the system in the real world or progress on stated motivations and goals for the field” (Raji et al., 2021). What is the real-world use of text-to-video retrieval (or alternatively, the field’s motivations)? Consider the most straightforward answer: that such systems will be used by users to search through video collections, whether on the web or in personal collections. First, search queries issued by real users are very likely not similar to captions written by crowd annotators; this is easily observed by inspecting captions in Table 5 and Appendix Table 6. Second, the video distribution is unlikely to reflect real use-cases as they were selected by annotators or are search results from seed queries. Due to these problems, it seems unlikely that the evaluations are construct valid, and future benchmarks should improve this by building evaluations that match the intended use of models—i.e., be ecologically valid (de Vries et al., 2020).

3 FIRE Dataset Collection and Validation

Next, we describe and analyze the FIRE dataset.

3.1 Annotation Task and Dataset Collection

In the FIRE annotation task, annotators mark whether the displayed caption is relevant to the displayed video. Implicitly, the caption’s video is relevant to it, but how do we judge whether another arbitrary video is relevant? In other words, how should annotators mark whether a caption is relevant to a video? In both datasets (§2.1), the caption

must be completely consistent with the video; otherwise, it would not be an accurate caption. Therefore, we enforce the same condition in our task to preserve the original relevance semantics.⁸

Annotators are instructed to mark a caption as relevant to a video only if every element mentioned in the query could be reasonably considered present. Elements included persons, objects, locations, and activities, as well as quantifiers, qualifiers, and adjectives. Raters are given some leeway to use interpretation and inference but instructed to err in favor of not relevant if the caption is ambiguous or vague. For example, for the caption “a boy playing the violin,” the video must show a boy who is playing the violin, not a video of only violins or a video with only a boy. Screenshots of the annotation interfaces and details of sensitive category handling are in Appendix B. Complete annotation guidelines are included in supplemental materials.

To select caption-video pairs to annotate, we obtain the top ten MSR-VTT and MSVD test set predictions from three models: CLIP4CLIP (Luo et al., 2021), SSB (Patrick et al., 2021), and TeachText (Croitoru et al., 2021). For TeachText, we use model checkpoints available on their webpage. For CLIP4CLIP and SSB, checkpoints are not available, so we train new models and verify that retrieval quality is on par with the literature (see Table 1).

Table 2 summarizes the resulting FIRE dataset. During data collection, 683K labels were collected across a set of 579K unique caption-video pairs. Some duplication was intentional: we obtained a second label for 10% of annotations, and if the labels disagreed, we collected a third label to resolve the disagreement. Elsewhere, duplication was unintentional: for MSVD we did not deduplicate caption-video pairs between two models, so where the pre-

⁸Requiring complete matches makes the annotation task easier by eliminating ambiguous partial match cases.

Dataset	# Pairs	Percent	# Labels
MSR-VTT	24,183	100%	24,507
└ Agreement	24,167	99.9%	-
└ Relevant	2,855	11.8%	-
└ Irrelevant	21,312	88.2%	-
└ Disagreement	16	0.0662%	-
MSVD	555,391	100%	659,126
└ Agreement	553,832	99.7%	-
└ Relevant	39,909	7.21%	-
└ Irrelevant	513,923	92.8%	-
└ Disagreement	1,559	0.281%	-

Table 2: The FIRE dataset is composed of labels for MSR-VTT and MSVD text-video pairs. The positive-to-negative ratio is skewed, reflecting that queries do not match most videos. We multiply annotate a subset to compute annotator agreement rates and Krippendorff’s α . Agreement on MSR-VTT was .931 with $\alpha = .691$ and on MSVD was .958 with $\alpha = .798$. Appendix C disaggregates agreement rates which are consistent.

dictions overlapped, we obtained additional labels. Fortunately, this provided an unexpected opportunity to further validate dataset quality.

3.2 Dataset Quality Validation

Before, throughout, and after the collection, we took steps to collect high-quality data and validate its quality. The annotation task was completed by a team of one hundred raters specifically trained to review caption-video pairs and assess relevance. These annotators completed a 1,000 job training queue, which was reviewed by data quality leads and this paper’s authors. This allowed annotators to learn to annotate according to our guidelines, request clarification to the guidelines, and request tooling improvements. Annotators could also escalate tasks for being too ambiguous or confusing, which occurred less than 0.0001% of the time.

After the dataset was collected, we computed three measures of quality in Table 2: (1) the rate that judgments resolved to a label (Percent), (2) the degree to which examples with multiples labels agreed (Agreement), and (3) the Krippendorff alpha score amongst examples with multiple labels (Krippendorff, 2004). Caption-video pairs resolved to a label 99.9% of the time in MSR-VTT and 99.6% of the time in MSVD. Agreement in both datasets exceeded 90%, and the Krippendorff score suggests reasonable agreement as well. Based on this analysis, we see no evidence of data quality issues. The next section digs deeper into FIRE and suggests explanations for the observed phenomena.

Dataset	Models	Overlap	RBO
MSR-VTT	C4C & SSB	0.0638	0.0568
MSR-VTT	C4C & TT	0.0610	0.0509
MSR-VTT	TT & SSB	0.440	0.231
MSVD	C4C & TT	0.411	0.211

Table 3: Annotated predictions of one model boost the score of another model when predictions overlap. In MSR-VTT, there is little overlap between CLIP4CLIP and other models; there is far more overlap in MSVD.

Model	Data	C@1	C@5	C@10
CLIP4CLIP	All	0.674	0.907	0.957
CLIP4CLIP	New	0.430	0.713	0.812
TeachText	All	0.241	0.532	0.670
TeachText	New	0.239	0.527	0.663
SSB	All	0.273	0.559	0.689
SSB	New	0.271	0.553	0.679

Table 4: We compare C@K of a MSR-VTT model: (1) with all annotations (All) and (2) without the model’s annotated predictions to emulate model development (New). CLIP4CLIP exhibits large differences.

4 Analysis Experiments

The difference FIRE makes on metrics (Table 1) is striking, which begs the question: *why* are there such *large* differences? We suggest explanations for these differences (§4.1) while investigating how these metrics vary under commonplace evaluation settings such as new model development (§4.2).

4.1 Why Are Score Boosts Not Uniform?

FIRE-based metrics are interesting for at least two reasons: (1) the magnitude of difference and (2) the non-uniformity of boosts. Specifically, CLIP4CLIP has a larger boost than TeachText and SSB on MSR-VTT. First, we investigate the degree of prediction overlap between models. When predictions overlap, the models share the boost. Likewise, when they do not overlap, there is an opportunity for differing boosts. Table 3 shows this: on MSR-VTT, CLIP4CLIP and the other two models have little overlap; in contrast, TeachText and SSB have substantial overlap and their boosts are of roughly the same magnitude. Overlap is computed between the top ten predictions of each model using simple overlap and rank-biased overlap (Webber et al., 2010, RBO).⁹ As we might expect based on CLIP4CLIP

⁹If the ordering of predictions amongst the top ten did not matter, the overlap would be acceptable. However, as in most IR settings, we *do* care about the order so use a rank-aware metric like RBO.

and TeachText having large boosts on MSVD, their predictions also overlap. This mechanically explains the difference but fails to explain “why?”

We test the hypothesis that shorter queries have more positives because they are less specific (i.e., general) and speculate that differences in CLIP4CLIP and TeachText pre-training could make CLIP4CLIP fare better on general queries. Intuitively, shorter captions should be less specific and therefore match more videos, so models that handle general captions well should benefit the most. Table 5 and Appendix Table 6 validate this intuition by showing MSVD and MSR-VTT captions. The captions are sampled from the shortest 100 captions, median length captions, and longest 100 captions.

First, we empirically validate that short captions have more positive videos. Figure 2 shows that longer captions have fewer positive videos while shorter captions have more. By construction, since we find only positives if a model predicts them, these are where models make gains.

Figure 3 takes the next step and compares model accuracy as a function of caption length. For each bin of caption lengths (e.g., captions of length zero to twenty characters), we show the proportion of whether both CLIP4CLIP and TeachText are correct, neither are correct, or only one is correct. Empirically, we observe that CLIP4CLIP makes the largest gains from accounting for false negatives with FIRE when queries are short—whether this is due to short queries containing more positives or CLIP4CLIP handling these better is difficult to discern. Although it is difficult to validate, our best, educated guess at a causal reason for CLIP4CLIP finding more positives in MSR-VTT is that its image-text backbone, CLIP (Radford et al., 2021), was trained with text that contains many general captions.

4.2 Does System Pooling Generalize?

Although system pooling eliminates (implicit) false negatives, it comes with the substantial drawback that every new model must have its predictions annotated—otherwise, the results are potentially biased against the new model due to the possibility of false negatives in novel predictions (Yilmaz et al., 2020).¹⁰ System pooling has traditionally been used in synchronized shared tasks where all models are submitted by a deadline and evaluated

¹⁰If a model predicts a video that no prior model does and it is a false negative, then the model’s effectiveness will be underestimated. Yilmaz et al. (2020) study this when comparing traditional and deep learning IR systems.

at the same time, as in the Text REtrieval Conferences (TREC) in IR.¹¹ However, the trend in machine learning and NLP is for continuously running or even dynamic benchmarks (Kiela et al., 2021). Beyond benchmarking, even the development of new models is affected since gains from improved modeling may be understated. The question then is: how large is this bias, and how fast does it decrease with the number of pooled models?

The magnitude of the bias is affected by two factors: (1) the percent of model predictions that do not exist in pooled annotations and (2) the prevalence of false negatives in this subset.¹² While Table 3 captures prediction overlap between pairs of models, it does not capture the setting where some number of models have annotated predictions, and we wish to test a new model. Table 4 calculates (1) model scores when using all annotated predictions versus (2) model scores using only annotated predictions from the other two models. In this small three-model experiment, the bias is unfortunately still significant (24.4% for C@1) for the best model (CLIP4CLIP). Thus, the degree to which the FIRE dataset will mitigate the false negative problem in new model development is dependent on the similarity of new models to current ones. The generalizability also depends on the number of unknown positives, which we indirectly study by plotting the ranks of positive videos (Appendix G).

4.3 Mitigating Annotation Costs by Sampling

A limitation of our method is that until existing annotations include most positives, our method either disadvantages new models or introduces non-trivial annotation costs. Indeed, the costs of exhaustive annotation in our work are substantial, but exhaustive annotation is also excessive if the goal is only to (robustly) estimate model scores. Instead, we propose that future work need only annotate the top 10 predictions from N examples in the evaluation data. But how large should N be so that we can be confident that the difference between model scores is statistically significant? In our next experiment, we use bootstrap sampling to characterize the relationship between N and the effect size corresponding to a statistically significant difference at the 95% confidence level.

In our bootstrap sampling experiment, we treat the 27,763 MSVD test examples as a sample from

¹¹<https://trec.nist.gov>

¹²See Appendix F for analysis of the number of known positive videos per query.

MSVD Short Length Captions	MSVD Median Length Captions
playing panda	a gymnast falls off a balance beam
some work	a person is riding a horse
a man	a girl is riding a bicycle
a baby	two men are pushing an airplane
jumping dachhund	the turtle is playing with the cat
naah	piano is played by an artist
amanplaysaguitar	the girl put stickers on her face
a woman	a boy is reading a card
camp	a little boy is playing golf
plying music	a man is slicing a tomato
MSVD Long Length Captions	
a man holding an open umbrella jumps across a wooden stand in a park and then does a summersault after kicking a wall	
a man in a jail cell motions to a man in another cell who shows the first man his middle finger	
a bowling man picks up a spare in his lane and manages to knock over the one remaining pin in the lane to his right	
a woman is exercising by stepping from right to left and then from left to right while swinging her arms back and forth	
a man wearing a black cape is walking toward a group of people and a man in the group is shooting at him with a pistol	

Table 5: This table shows three sets of MSVD captions sampled from: (1) the 100 shortest captions, (2) median length captions, and (3) the 100 longest captions. As also observed in MSR-VTT captions (Table 6), short captions are general (e.g., “a man”) compared to the longest captions.

a population.¹³ We characterize the population distribution through bootstrap re-sampling of the original sample. Specifically, we estimate the absolute difference in model scores that correspond to a statistically significant effect size (i.e., score difference) at the 95% confidence level. For each sample size $N \in [500, 1000, 3000]$, we (1) re-sample N examples from MSVD evaluation data, (2) calculate scores on the re-sample, (3) repeat this 10,000 times, (4) average the scores then calculate the absolute value of the difference between the average score and score calculated with the full dataset, and finally (5) plot the distribution and score corresponding to the 95% percentile. The experimental results (Figure 4) demonstrate that annotation volumes of 1,000 detect statistically significant differences when C@1 differs by 0.029. The results demonstrate that (1) annotating a subset of test examples detects absolute differences of one absolute point, and (2) the number of annotated test predictions varies based on the metric of interest.

5 Recommendations for Benchmarks

Towards improving TVR evaluations, we make recommendations for both current and future benchmarks. This paper only investigated the effects of false negatives in MSR-VTT and MSVD. However, it is likely that other similarly constructed benchmarks exhibit the same problem, and testing this is important. Second, we show that for MSR-VTT

¹³MSR-VTT is small. To avoid convergence to the sample mean, bootstrap sizes need to stay low.

and MSVD, certain metrics such as Correct@10 are potentially saturated since improvements above CLIP4CLIP’s 95.7% and 94% are plausibly noisy. Consequently, since the remaining gains reside in re-ranking the top K, the community should consider retiring these evaluations. Third, the introduction of multiple positives and use of various K values makes mean average precision attractive since: (1) it factors in preference for correctness at higher ranks and (2) it handles multiple positives.

It is difficult to recommend that model developers exhaustively annotate model predictions. This suggests a future where query or video set size is a trade-off between annotation load and evaluation quality. For example, one might choose to trade-off annotation load with statistical power to differentiate between models (Card et al., 2020). TREC-style, annual shared tasks are one model for this (Voorhees, 2019; Church and Hestness, 2019); instead of building a monolithic benchmark that becomes overfit over time (Blum and Hardt, 2015; Anderson-Cook et al., 2019), stakeholders develop evaluations that evolve with research objectives.

Looking forward, TVR evaluation would benefit from: (1) a purpose-built benchmark that is grounded in an actual use case so as to be ecologically valid (de Vries et al., 2020) and (2) centralized evaluation by submitting runnable models to shared infrastructure such as Dynabench (Kiela et al., 2021). This would improve reproducibility, which was a limiting factor in selecting which model predictions to reproduce in this paper. This

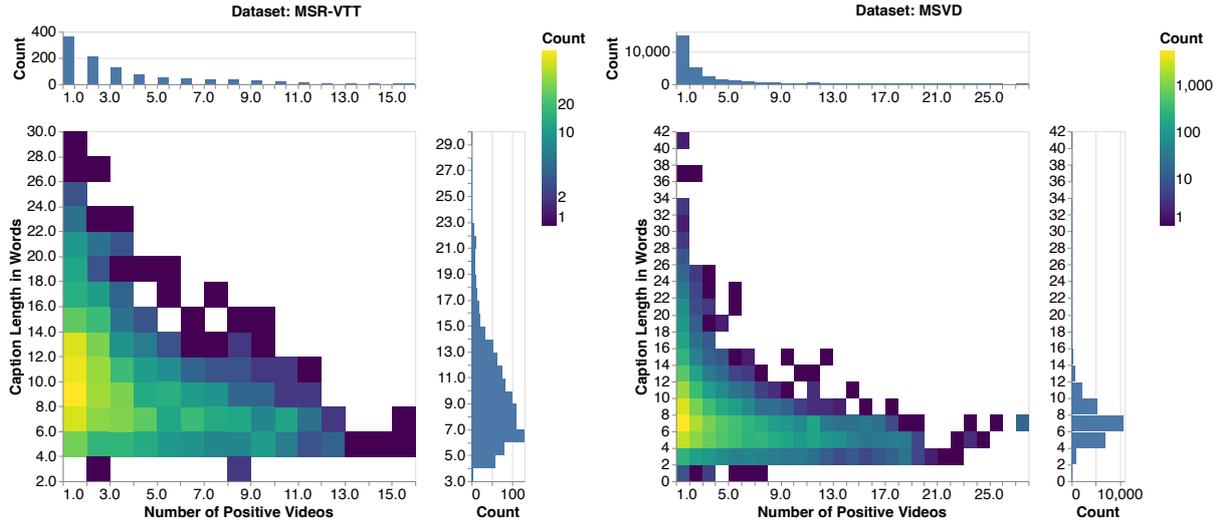


Figure 2: This figure shows the relationship between the number of positive videos and the length of captions in words for MSR-VTT and MSVD. We show a log-scale density heatmap binned by the number of positive videos and caption length; on the margins, are histograms. From this figure, we can infer that: (1) if a caption is long, it is less likely to have many positive videos, and (2) if a caption is short, then the number of positive videos can vary widely.

also makes calculating statistical tests easy (Ethayarajh and Jurafsky, 2020), which are often not reported (Dror et al., 2018; Dodge et al., 2019). TVR modeling has advanced enough to demand better benchmarks for measuring future progress.

6 Related Work

The paper draws on ideas in multimodal retrieval, information retrieval, and evaluation methodology.

Improving Benchmark Quality: Wray et al. (2021) is directly relevant to our work, and we share their motivation: to study the effects of false negatives in TVR evaluations. While we share motivation and our works are complementary, our work differs substantially in methods, contributions, and conclusions. The primary difference is this: our goal is to quantify the difference in absolute metrics that false negatives cause, even if there is no promise the data can be effectively reused in the future; Wray et al. (2021) develop automatically runnable proxy measures that improve the reliability of model rankings, but do not precisely quantify the impact of false negatives on existing metrics since automatic labeling is not equivalent to human annotation. Both these works are valuable: our work conclusively quantifies that false negatives create differences of 25% absolute points and demonstrate that new measures like those by Wray et al. (2021) are necessary for current benchmarks.

Wang et al. (2022) argue that video captioning datasets used in TVR evaluation are noisy due to

low-quality captions but differ by identifying single query tasks as the root problem (as opposed to false negatives) and recommend multi-query evaluation where users make followup refinement queries. While the multi-query problem is important, we do not agree with the assessment that single-query problems should be abandoned for multi-query problems: for example, users often have a low tolerance for voice assistant errors and abandon their query entirely after an error. Both problems are important. Fortunately, the approaches are complementary and should be combined: the multi-query setting still has false negatives, whose effects on measurement can be mitigated with our methods (§4.3). Just as we use predictions to improve datasets, Beyer et al. (2020) improve ImageNet labels by using predictions to reduce the label space which makes the annotation task easier.

Benchmarking: Across machine learning, computer vision, and natural language processing (Eger et al., 2020; Bowman and Dahl, 2021; Rogers, 2021) there is a broad effort to critically examine the benchmarks (Schlangen, 2021), data (Linzen, 2020; Thrush et al., 2022), evaluation methods (Rodriguez et al., 2021), and evaluation paradigms (Rodriguez and Boyd-Graber, 2021; Kiela et al., 2021) used in research studies. This effort goes beyond particular methodologies and extends to identifying the values prized by the community (Sculley et al., 2018; Dotan and Milli, 2020) which are subsequently operationalized in computer vision

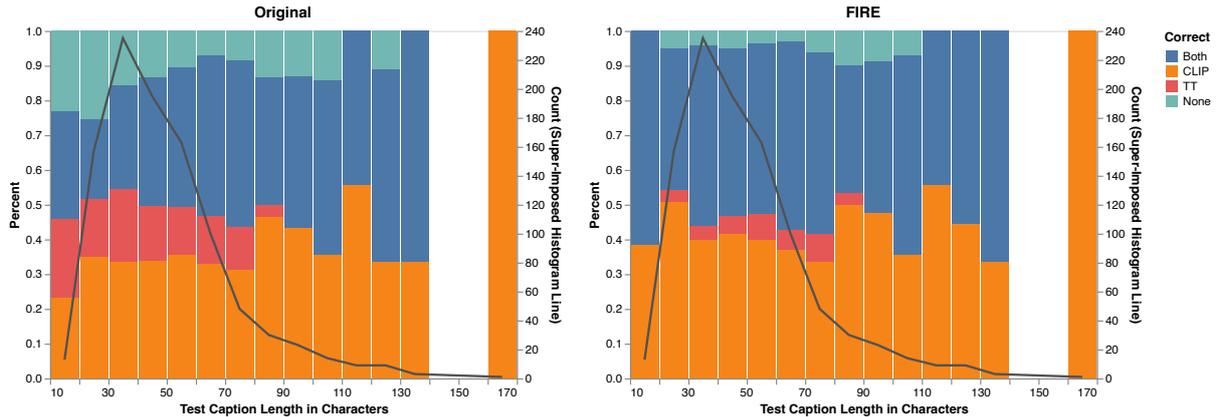


Figure 3: On MSR-VTT, we show relative model effectiveness differences (y-axis and color bars) broken down by test caption length (x-axis); we super-impose the caption length distribution (black line). Short captions tend to be more general, so they should match more videos and produce more false negatives. The gains for both models and especially CLIP4CLIP occur predominantly on this subset (reduction of “None”) as we would expect.

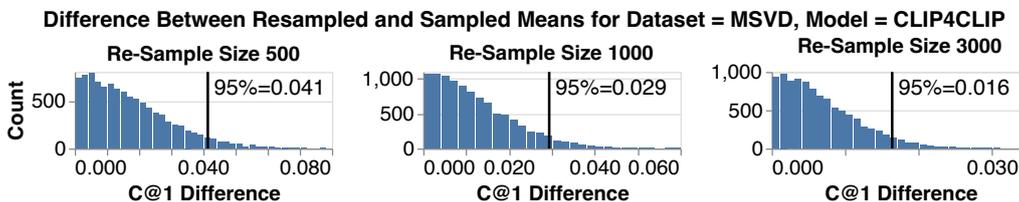


Figure 4: The distribution of absolute differences between bootstrap re-sample estimates of CLIP4CLIP C@1 and the true sample mean, by re-sample size. This estimates the number of annotations to detect an effect size at 95% confidence. Appendix E expands on this experiment by showing results for C@5, C@10, and TeachText.

datasets and benchmarks (Wu et al., 2017; Scheuerman et al., 2021). Our work is in line with this broader initiative and critically examines text-to-video retrieval evaluation methodology.

We examine internal validity (§2.2.1) and find a broken yardstick (Hernandez-Orallo, 2020). By examining construct validity (§2.2.2), we also argue that TVR evaluations should prize usefulness to ecologically valid use cases such as real-world text-to-video search (de Vries et al., 2020). Lastly, our experimental results suggest we may not be far off from retiring MSR-VTT and MSVD for TVR evaluation, something we should not be afraid to do in general (Boyd-Graber and Börschinger, 2020). An alternate approach is smaller, periodic evaluations as in TREC (Smeaton et al., 2002; Voorhees and Tice, 2000; Smeaton et al., 2009). Part of the solution is to create purpose-built datasets with clear goals (Gebru et al., 2021; Bender and Friedman, 2018) as opposed to continually re-using datasets intended for different uses (Koch et al., 2021).

Structurally Similar Tasks: TVR is not the only evaluation with the implicit false negative problem. Our critique is applicable to image retrieval bench-

marks that use caption-media pairs from image captioning datasets (Lin et al., 2014; Plummer et al., 2015) as the only positives (Karpathy and Fei-Fei, 2015; Kim et al., 2021; Singh et al., 2022).

7 Conclusion

In this work, we show that label errors (false negatives) in text-to-video retrieval benchmarks invalidate their *internal validity*—the measured metrics do not accurately reflect reality (§2). Following this, we critique the applicability of benchmarks to real-world use cases (*construct validity*). To estimate the impact of false negatives on benchmark metrics, we collect the FIRE dataset (§3) which contains 683K relevance judgements. Analysis experiments (§4) suggest explanations for why CLIP4CLIP scores higher and estimate system pooling generalization. Based on our findings, we highlight properties that future TVR benchmarks should have and outline approaches to addressing inherent challenges in retrieval evaluation (§5). Finally, we position our work in the broader effort to improve benchmarking by better aligning tasks with the intended use and improving measurement (§6).

8 Limitations

Our work has several notable limitations. First, our experiments use two representative and commonly used TVR datasets (MSR-VTT and MSVD). While we expect that our results will generalize, it is still possible that these results do not generalize. For example, both datasets are based on YouTube videos and annotator-written captions: perhaps videos and captions from alternate sources differ by too much. Similarly, our experiments use three well-known models, so while we expect our results to generalize to similar models, future models may differ substantially in ways that cause the empirical results not to hold. This said, system pooling has long been used in TREC (Voorhees et al., 2005), so we expect this to work for future models as well.

Beyond limitations in generalizability, the in-principle critiques in our work apply only to benchmarks where implicit false positives are likely to be prevalent; it does not apply to benchmarks in general. From the methods perspective, while our computational experiments are coded to be easily reproduced, the scale of our annotations is difficult to reproduce (hence limited reproducibility in this sense), but we do study sampling-based alternatives to mitigate this limitation.

9 Ethics

This section discusses potential ethical issues related to our dataset-centric work. First, we discuss data-related ethics. The FIRE dataset is built on MSR-VTT and MSVD. We distribute the minimal amount of data related to these datasets necessary to reproduce our experiments: triplets of caption identifiers, video identifiers, and annotated labels. Section 3 and Appendix B describe how the data was collected. All annotators were compensated, and the data collection was reviewed before starting. Potential risks due to the use of our dataset are limited by the additional labels we provide for an existing dataset. We thoroughly discuss the risks associated with negatively influencing benchmark reliability (i.e., prediction overlap with future models), and these risks are mitigated by our recommendation that more appropriate datasets be developed.

Our work does not directly have negative societal impacts, but it is feasible that the improved model scores we report could be used to misrepresent the capability of retrieval systems. For example, while we only claim that a model achieves a

particular measure of effectiveness on a particular benchmark, the media often inflates the importance of these metrics (Cuthbertson, 2018). In our work, we intentionally do not connect these higher metrics to more general capability and emphasize the importance of establishing construct validity.

Acknowledgements

We thank Yookoon Park, Prahal Arora, and Bernie Huang for providing code and data that were helpful to kickstart this project. We thank Daniel Haziza for infrastructure support in hosting live demos. We thank Nathan Tokala for their support with annotation infrastructure. For insightful discussion and ideas, we thank Simran Motwani, Patrick Lewis, Thomas Hayes, Joe Barrow, Xilun Chen, Chenglei Si, Ronghang Hu, Max Bain, and Jacob Kahn. For feedback on prior versions of this paper, we thank Rich James, Florian Metze, Peter Rankel, Weijia Xu, Yoo Yeon Sung, Yuandong Tian, John P. Lalor, and Kirmani Ahmed.

References

- Christine M Anderson-Cook, Kary L Myers, Lu Lu, Michael L Fugate, Kevin R Quinlan, and Norma Pawley. 2019. [How to host an effective data competition: Statistical advice for competition design and analysis](#). *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 12(4):271–289.
- Emily M Bender and Batya Friedman. 2018. [Data statements for natural language processing: Toward mitigating system bias and enabling better science](#). *Transactions of the Association for Computational Linguistics*, 6:587–604.
- Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. 2020. [Are we done with ImageNet?](#) *arXiv preprint arXiv:2006.07159*.
- Avrim Blum and Moritz Hardt. 2015. [The ladder: A reliable leaderboard for machine learning competitions](#). In *Proceedings of the International Conference of Machine Learning*. PMLR.
- Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W Rae, Erich Elsen, and Laurent Sifre. 2021. [Improving language models by retrieving from trillions of tokens](#). *arXiv preprint arXiv:2112.04426*.

- Samuel Bowman. 2022. [The dangers of underclaiming: Reasons for caution when reporting how NLP systems fail](#). In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Samuel R Bowman and George E Dahl. 2021. [What will it take to fix benchmarking in natural language understanding?](#) In *Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Jordan Boyd-Graber and Benjamin Börschinger. 2020. [What question answering can learn from trivia nerds](#). In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Marilynn B. Brewer and William D. Crano. 2014. *Research Design and Issues of Validity*, 2 edition, page 11–26. Cambridge University Press.
- D T Campbell. 1957. [Factors relevant to the validity of experiments in social settings](#). *Psychological bulletin*, 54(4):297–312.
- Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. [With little power comes great responsibility](#). In *Proceedings of Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- David Chen and William Dolan. 2011. [Collecting highly parallel data for paraphrase evaluation](#). In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Kenneth Ward Church and Joel Hestness. 2019. [A survey of 25 years of evaluation](#). *Natural Language Engineering*, 25(6):753–767.
- Ioana Croitoru, Simion-Vlad Bogolin, Yang Liu, Samuel Albanie, Marius Lordeanu, Hailin Jin, and Andrew Zisserman. 2021. [TeachText: Crossmodal generalized distillation for text-video retrieval](#). *International Conference on Computer Vision*.
- Lee Joseph Cronbach and Paul E. Meehl. 1955. [Construct validity in psychological tests](#). *Psychological bulletin*, 52 4:281–302.
- Anthony Cuthbertson. 2018. [Robots can now read better than humans, putting millions of jobs at risk](#). *Newsweek*.
- Harm de Vries, Dzmitry Bahdanau, and Christopher Manning. 2020. [Towards ecologically valid research on language user interfaces](#). *arXiv preprint arXiv:2007.14435*.
- Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. [Show your work: Improved reporting of experimental results](#). In *Proceedings of Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Ravit Dotan and Smitha Milli. 2020. [Value-laden disciplinary shifts in machine learning](#). In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*.
- Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. [The hitchhiker’s guide to testing statistical significance in natural language processing](#). In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Steffen Eger, Yang Gao, Maxime Peyrard, Wei Zhao, and Eduard Hovy, editors. 2020. *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*. Association for Computational Linguistics.
- Kawin Ethayarajh and Dan Jurafsky. 2020. [Utility is in the eye of the user: A critique of NLP leaderboards](#). In *Proceedings of Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. [Multi-modal Transformer for Video Retrieval](#). In *European Conference on Computer Vision*.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. [Datasheets for datasets](#). *Commun. ACM*, 64(12):86–92.
- Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. [ActivityNet: A large-scale video benchmark for human activity understanding](#). In *Computer Vision and Pattern Recognition*.
- Jose Hernandez-Orallo. 2020. [AI evaluation: On broken yardsticks and measurement scales](#). In *Workshop on Evaluating Evaluation of Ai Systems at AAAI*.
- Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank. 2011. [A survey on visual Content-Based video indexing and retrieval](#). *IEEE transactions on systems, man and cybernetics. Part C, Applications and reviews: a publication of the IEEE Systems, Man, and Cybernetics Society*, 41(6):797–819.
- Andrej Karpathy and Li Fei-Fei. 2015. [Deep visual-semantic alignments for generating image descriptions](#). In *Computer Vision and Pattern Recognition*, pages 3128–3137.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit

- Bansal, Christopher Potts, and Adina Williams. 2021. [Dynabench: Rethinking benchmarking in NLP](#). In *Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. [Vilt: Vision-and-language transformer without convolution or region supervision](#). In *Proceedings of the International Conference of Machine Learning*.
- Bernard Koch, Emily Denton, Alex Hanna, and Jacob Gates Foster. 2021. [Reduced, reused and recycled: The life of a dataset in machine learning research](#). In *NeurIPS: Datasets and Benchmarks Track*.
- Klaus Krippendorff. 2004. *Content Analysis: an Introduction to its Methodology*. Sage: Thousand Oaks, CA. Chapter 11.
- Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. [Dense-captioning events in videos](#). In *International Conference on Computer Vision*.
- Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. [Deduplicating training data makes language models better](#). In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Michael S Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. 2006. [Content-based multimedia information retrieval: State of the art and challenges](#). *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19.
- Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. [Question and answer Test-Train overlap in Open-Domain question answering datasets](#). In *Proceedings of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. [HERO: Hierarchical encoder for Video+Language omni-representation pre-training](#). In *Proceedings of Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Thomas Liao, Rohan Taori, Inioluwa Deborah Raji, and Ludwig Schmidt. 2021. Are we learning yet? a meta review of evaluation failures across machine learning. In *NeurIPS: Datasets and Benchmarks Track*.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. [Microsoft COCO: Common objects in context](#). In *European Conference on Computer Vision*. Springer International Publishing.
- Tal Linzen. 2020. [How can we accelerate progress towards human-like linguistic generalization?](#) In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. [UniVL: A unified video and language Pre-Training model for multimodal understanding and generation](#).
- Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. [CLIP4Clip: An empirical study of clip for end to end video clip retrieval](#). *arXiv preprint arXiv:2104.08860*.
- Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, England.
- Sandra Mathison. 2004. *Encyclopedia of evaluation*. Sage publications.
- Bhaskar Mitra and Nick Craswell. 2018. An introduction to neural information retrieval. *Foundations and Trends® in Information Retrieval*, 13(1):1–126.
- Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021. [Pervasive label errors in test sets destabilize machine learning benchmarks](#). In *NeurIPS: Datasets and Benchmarks Track*.
- Scott W O’Leary-Kelly and Robert J. Vokurka. 1998. [The empirical assessment of construct validity](#). *Journal of Operations Management*, 16(4):387–405.
- Yookoon Park, Mahmoud Azab, Seungwhan Moon, Bo Xiong, Florian Metze, Gourab Kundu, and Kirmani Ahmed. 2022. [Normalized contrastive learning for text-video retrieval](#). In *Proceedings of Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander G Hauptmann, Joao F. Henriques, and Andrea Vedaldi. 2021. [Support-set bottlenecks for video-text representation learning](#). In *Proceedings of the International Conference on Learning Representations*.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. [Scikit-learn: Machine learning in python](#). *Journal of Machine Learning Research*, 12(85):2825–2830.
- Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. [Flickr30k entities: Collecting Region-to-Phrase correspondences for richer Image-to-Sentence models](#). In *International Conference on Computer Vision*.

- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the International Conference of Machine Learning*. PMLR.
- Inioluwa Deborah Raji, Emily M Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. 2021. [AI and the everything in the whole wide world benchmark](#). In *NeurIPS: Datasets and Benchmarks Track*.
- Pedro Rodriguez, Joe Barrow, Alexander Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. 2021. Evaluation examples are not equally informative: How should that change nlp leaderboards? In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Pedro Rodriguez and Jordan Boyd-Graber. 2021. [Evaluation paradigms in question answering](#). In *Proceedings of Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Anna Rogers. 2021. [Changing the world by changing the data](#). In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Lawrence A Rowe and Ramesh Jain. 2005. [ACM SIGMM retreat report on future directions in multimedia research](#). *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(1):3–13.
- Morgan Klaus Scheuerman, Alex Hanna, and Emily Denton. 2021. [Do datasets have politics? disciplinary values in computer vision dataset development](#). *Proceedings of the ACM on human-computer interaction*, 5(CSCW2):1–37.
- David Schlangen. 2021. [Targeting the benchmark: On methodology in current natural language processing research](#). In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics.
- D Sculley, Jasper Snoek, Alexander B Wiltschko, and A Rahimi. 2018. [Winner’s curse? on pace, progress, and empirical rigor](#). In *Proceedings of the International Conference on Learning Representations*.
- Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. [FLAVA: A foundational language and vision alignment model](#). In *Computer Vision and Pattern Recognition*. IEEE Computer Society.
- Alan F Smeaton, Paul Over, and Wessel Kraaij. 2009. [High-Level feature detection from video in TRECVID: A 5-year retrospective of achievements](#). In Ajay Divakaran, editor, *Multimedia Content Analysis: Theory and Applications*, pages 1–24. Springer US, Boston, MA.
- Alan F Smeaton, Paul Over, and Ramazan Taban. 2002. [The TREC-2001 video track report](#). Technical report, National Institute of Standards and Technology.
- Karen Spark-Jones. 1975. Report on the need for and provision of an ‘ideal’ information retrieval test collection. *Computer Laboratory*.
- Wanhua Su, Yan Yuan, and Mu Zhu. 2015. [A relationship between the average precision and the area under the ROC curve](#). In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*. Association for Computing Machinery.
- Jean Tague-Sutcliffe. 1992. [The pragmatics of information retrieval experimentation, revisited](#). *Information processing & management*, 28(4):467–490.
- Tristan Thrush, Kushal Tirumala, Anmol Gupta, Max Bartolo, Pedro Rodriguez, Tariq Kane, William Gaviria Rojas, Peter Mattson, Adina Williams, and Douwe Kiela. 2022. [Dynatask: A framework for creating dynamic AI benchmark tasks](#). In *Proceedings of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics.
- Ellen M. Voorhees. 2019. [The Evolution of Cranfield](#), pages 45–69. Springer International Publishing, Cham.
- Ellen M Voorhees, Donna K Harman, et al. 2005. *TREC: Experiment and evaluation in information retrieval*, volume 63. MIT press Cambridge, MA.
- Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Zeyu Wang, Yu Wu, Karthik Narasimhan, and Olga Russakovsky. 2022. [Multi-query video retrieval](#). In *European Conference on Computer Vision*.
- William Webber, Alistair Moffat, and Justin Zobel. 2010. [A similarity measure for indefinite rankings](#). *ACM Transactions on Information Systems*, 28(4).
- Michael Wray, Hazel Doughty, and Dima Damen. 2021. [On semantic similarity in video retrieval](#). In *Computer Vision and Pattern Recognition*.
- Zuxuan Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. 2017. [Deep Learning for Video Classification and Captioning](#), pages 3—29. Association for Computing Machinery and Morgan & Claypool.
- Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. 2021. [VLM: Task-agnostic Video-Language model pre-training for video understanding](#). In *Findings of the Association for Computational Linguistics: ACL*. Association for Computational Linguistics.

- Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. [MSR-VTT: A large video description dataset for bridging video and language](#). In *Computer Vision and Pattern Recognition*.
- Emine Yilmaz, Nick Craswell, Bhaskar Mitra, and Daniel Campos. 2020. [On the reliability of test collections for evaluating systems of different types](#). In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA.
- Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. [A joint sequence fusion model for video question answering and retrieval](#). In *European Conference on Computer Vision*.
- Bowen Zhang, Hexiang Hu, and Fei Sha. 2018. [Cross-modal and hierarchical modeling of video and text](#). In *European Conference on Computer Vision*.
- Linchao Zhu and Yi Yang. 2020. [Actbert: Learning global-local video-text representations](#). In *Computer Vision and Pattern Recognition*.

A Model Prediction Comparison

As part of this paper, we develop several web apps to make exploring the data more accessible. For example, Figure 5 compares the predictions of three models along with the labels in the original MSR-VTT dataset compared to augmenting them with FIRE’s labels. The source code repository provides instructions to run these web app demos.

B Annotation Interfaces

The FIRE dataset (§3) was collected using the annotation interface in Figure 6.

In addition to the previously described annotation instruction (§3.1), raters were also instructed on how to handle sensitive categories. The raters were instructed to accept the caption as accurate unless they had compelling, concrete reasons to believe otherwise (e.g., a little baby should be not considered old, and octogenarians with white hair and wrinkled skin should not be considered young); raters should not attempt to make more fine-grained distinctions. In particular, they were instructed not to make any assumptions about gender and accept the gender described by the caption.

C FIRE Data Quality

This section provides additional evidence to validate the quality of the FIRE dataset. Specifically, Figure 7 complements the agreement metrics computed in §3.2 and Table 2 by un-aggregating agreement rates.

D Shorter Captions, Their Generality, and Correlation to Model Behavior

Experiments in §4.1 establish that shorter captions have more positives and longer captions have fewer. We intuitively explain this by stating that shorter captions by nature are less specific, so will, in principle, match more videos. For example, one of the shortest captions in MSVD is “a man” (Table 5) which is less specific than one of the longest captions like “a man holding an open umbrella jumps across a wooden stand in a park and then does a summersault after kicking a wall.” Inspecting these captions also validates our construct validity critique (§2.2.2): they do seem like search queries.

In previous experiments (§4.1), we discussed how caption length is related to which models gain higher boosts. This section breaks down which models gain the most on MSR-VTT by train-test

overlap. We take inspiration from question answering and language modeling, where unintentional textual overlap between train and test sets degrades evaluation and model quality (Lewis et al., 2021; Borgeaud et al., 2021; Lee et al., 2022). Our objective is to measure the degree to which test captions in MSR-VTT are present in the training captions—be it word-for-word or approximate. To measure this, we use Scikit-Learn (Pedregosa et al., 2011) to fit a 5-gram character TF-IDF encoder to the test captions and compute the cosine similarity of each test caption to each train caption. For each test caption, we compute the mean similarity of the top train ten captions and combine this information with Correct@5 scores (Figure 8). The results suggest that TeachText overfits the train set, which may explain its comparatively better scores on the original positives—were it not overfit, train-test overlap should not matter.

Lastly, these factors are not unrelated. Since shorter captions tend to be less specific, these are also the captions that we would expect are more prevalent in the training set, whether in exact form or approximate (e.g., the phrase “a man” is likely in the train set). To test whether these factors are related, we compute the Kendall Tau correlation and Spearman Rank correlation between the train-test textual similarity score and caption length (in both words and characters). As we expect, there is a non-trivial negative correlation between caption length and similarity score (Table 7): the lower the caption length, the higher the train-test overlap score.

E Can Annotation Costs be Mitigated Through Sampling?

In our experiments, we use bootstrap sampling to estimate the number of example annotations needed to detect given effect sizes at the 95% confidence level (§4.3). Figure 4 reports these results for CLIP4CLIP since it was the best model; in practice, it represents the type of model we would test after models like TeachText. Figure 9 extends the results from C@1 to C@5 and C@10. Figure 10 replicates these results, but using the TeachText model.

F Number of Positive Videos per Text Query

The generalization of the FIRE dataset to newer models is reliant on two factors: (1) the number of positive videos per query and (2) whether the

Model Comparison Viewer for MSRVT/MSVD

Original Dataset Label, FIRE Dataset Label

Caption: cartoon girl is talking

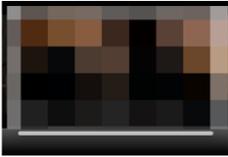
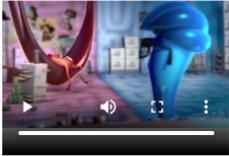
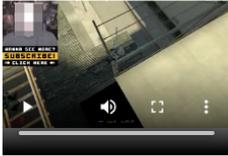
Clip4Clip	Experts	SSB
 <p>Label: <input type="checkbox"/> Irrelevant Label: <input checked="" type="checkbox"/> Relevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>
 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>
 <p>Label: <input type="checkbox"/> Irrelevant Label: <input checked="" type="checkbox"/> Relevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>
 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>
 <p>Label: <input checked="" type="checkbox"/> Relevant Label: <input checked="" type="checkbox"/> Relevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input checked="" type="checkbox"/> Relevant</p>	 <p>Label: <input type="checkbox"/> Irrelevant Label: <input type="checkbox"/> Irrelevant</p>

Figure 5: The web application shows the ranked predictions of three models: CLIP4CLIP, TeachText, and SSB. Qualitatively, CLIP4CLIP predictions better match the query by showing only cartoon videos. This is reflected quantitatively when FIRE labels are incorporated. Lastly, the ranked predictions also show some of the overlap that TeachText and SSB shared.

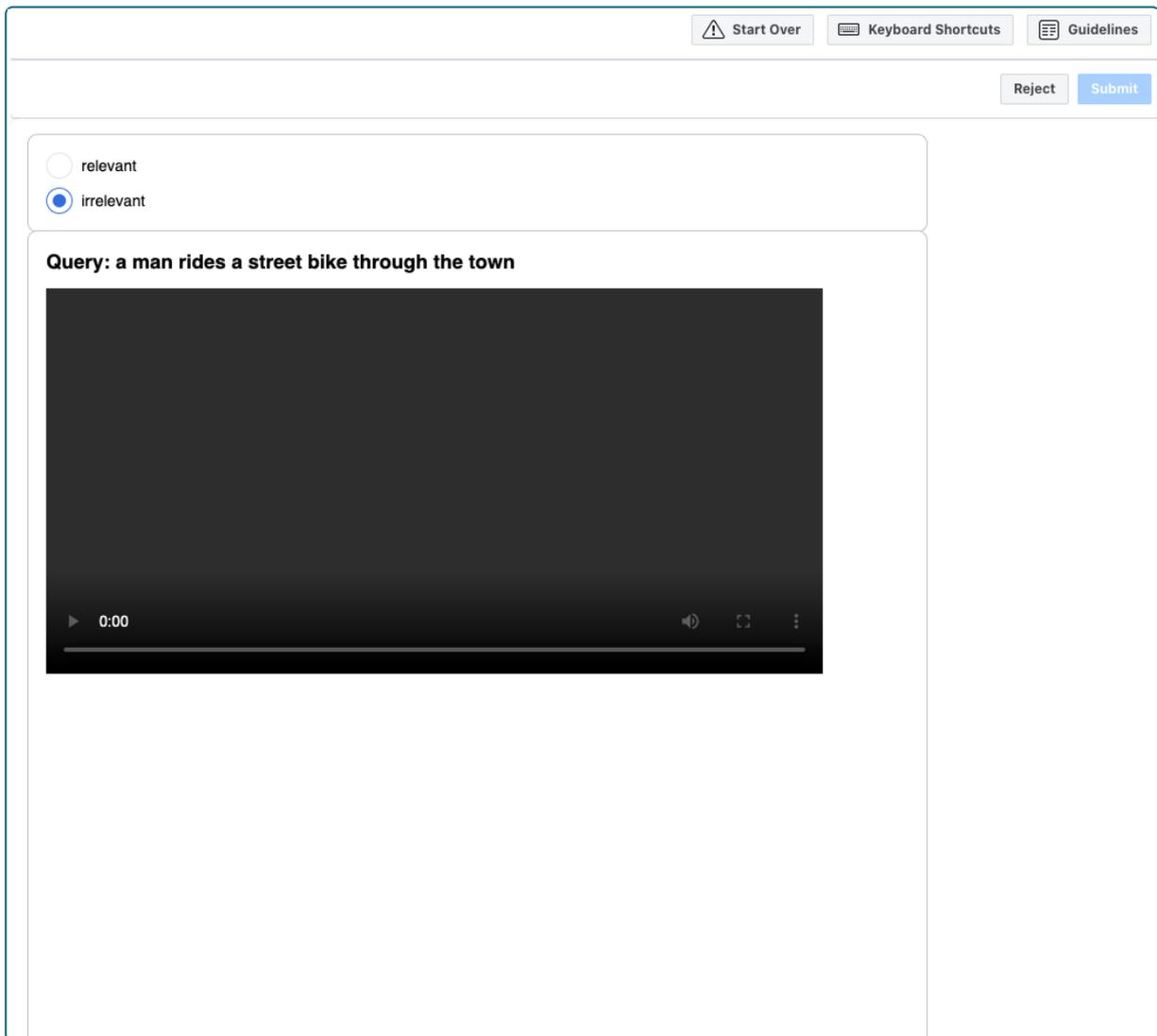


Figure 6: To annotate the FIRE dataset, raters used this annotation interface. The interface shows the candidate query (caption) and video; raters are trained to select “relevant” or “irrelevant” based on whether every component of the query matches the video.

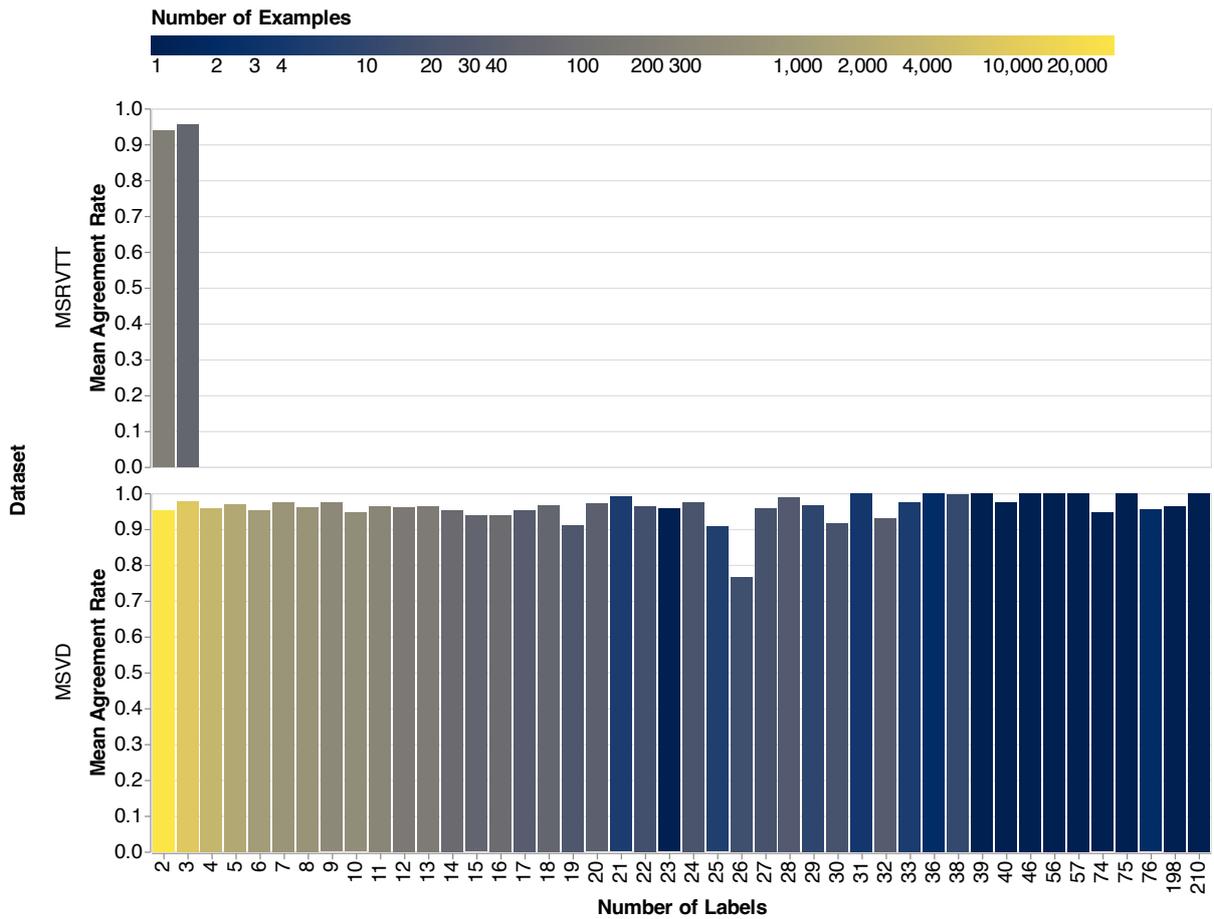


Figure 7: The agreement rate of annotators is broken down by the number of labels. For example, about 10,000 MSVD examples (text-video pairs) were annotated twice; of those, the two labels agreed on about 95% of examples. As we did with the MSR-VTT collection, our intent for the data collection was to de-duplicate text-video pairs and only annotate about 10% of the data multiple times to estimate reliability. However, we accidentally omitted this step for the MSVD collection which resulted in some examples being annotated many times. Fortunately, this provides an unplanned opportunity to further validate inter-annotator agreement.

MSR-VTT Short Length Captions

a man playing video games
anchor talking about a shows
a woman is stirring food
sports are being played
a woman holding a ribbon
a diver goes underwater
baseball player hits ball
cartoon show for kids
two women are embracing
advertisement of seat basket

MSR-VTT Median Length Captions

a man runs into the crowd when trying to catch a basketball
in a music video a man is laying with women while singing
some people video conferencing as they watch a movie
a boy is trying out for a part on the voice kids
basketball players making a shot in the last seven seconds
views of two persons working on the super computer with the head phones on
a character is jumping and floating in the air in a video game
two people playing basketball and the one with a hat makes every shot
batman is beating up bane in a scene from a batman movie
a girl being surprised with a stuffed animal by male friend

MSR-VTT Long Length Captions

a man and a woman are sitting in front of a television and addressing and audience
a woman stirs up some soup sprinkles a spice in and drops a shot of liquid into it
a man is filming as he and a woman watch the news where it shows an area filled with smoke
flight is shaken and the pilots trying to land the flight while they opened the air
the chef adds fish sauce and fish paste to a large stainless steel cooking pot
a girl wearing a dress stands to the side of the screen while lyrics to a song playing in the background appear on the other side
the man is giving an informational speech to a group of people about telling someone something
a girl in blue color dress wearing sitting speaking and television screen with black shirt man beside still image displaying on screen
a man plays a video game where the player has a first person perspective and shoots other characters
a man playing a video game character that is carrying a sword and killing animals with it

Table 6: This table shows three sets of captions from MSR-VTT sampled from: (1) the 100 shortest captions, (2) the median length captions, and (3) the 100 longest captions. As we argued by intuition (§4.1), inspecting these samples validates that the shorter captions are more general (e.g., “sports are being played”) and longer captions are very specific (e.g., “a woman stirs up some soup sprinkles a spice in and drops a shot of liquid into it”).

models we studied in this work predict all the true positives. Estimating the number of true positives per query without exhaustive annotation is difficult at best. However, we can at least characterize how many positives there are when including FIRE annotations. Figure 11 shows a histogram of the number of positive videos per query across MSR-VTT and MSVD. For example, about 350 MSR-VTT queries have only one known positive, which implies that the other 650 have more than one known positive. Unfortunately, even estimating the upper bound would require annotating all the videos for each query in a representative sample (e.g., for a sample of MSR-VTT 200 queries, exhaustive annotation would include $200 * 1,000 = 200,000$ query-video annotations).

G Rank of Positive Videos

While we follow prior work in measuring models based on metrics computed from their top ten predictions, this still leaves open the question: ignoring prior work, is ten predictions the right choice? If ten is the correct choice, then we should see a clear trend that positives are primarily distributed below ten. Figure 12 plots the rank of positive videos in CLIP4CLIP predictions (i.e., 1 is top-ranked) versus their count. As expected, the number of positives drops dramatically before rank 10 (especially for MSVD), although not to zero; the ranks of the original positives suggest there is a long tail of undiscovered positives. Note that the steep dropoff at 10 is due to annotating only the top 10; positives beyond this are either from the original dataset or predicted by other models. From

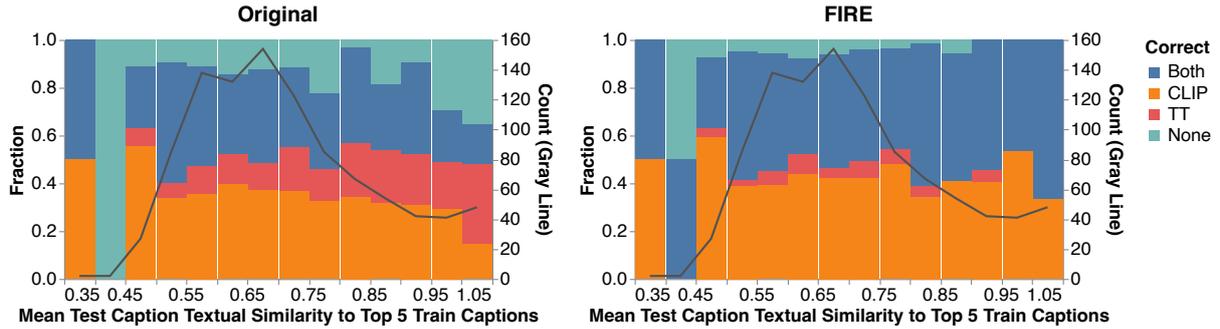


Figure 8: Why are the MSR-VTT score differences between CLIP4CLIP and TeachText when using FIRE large? We test the hypothesis that TeachText (comparatively) overfits textual training data. We compute the textual similarity of each test caption to each train caption with a 5-gram character model; for each test caption, we calculate the mean similarity of the ten most similar captions. The plot shows whether both models score a point on Correct@5, binned by train-test similarity (the overall histogram is shown as the super-imposed line) when using original versus FIRE annotations. On the original annotations, CLIP4CLIP fares much better compared to TeachText when similarity is not nearly 1.0 (i.e., not overfitting).

Length	Spearman	Kendall
Word	-0.419	-0.296
Character	-0.479	0.334

Table 7: This table shows the Spearman and Kendall rank correlations between the train-test textual similarity score used to measure train-test overlap (Figure 8) and the length of captions in both words and characters. The results support our hypothesis that caption length and train-test overlap are correlated.

this, we conclude that although most positives have likely been collected, there likely remain more past rank 10, especially in MSR-VTT.

H Computational Resources

This paper was developed using two types of computational resources. To rerun text-to-video retrieval models, we trained and evaluated on a single AWS p4d compute node which has 96 vCPUs, 1152GB of RAM, and eight Nvidia A100 GPUs.¹⁴ All other experiments were run locally on a 16 inch, 2019 Macbook Pro with a 2.4GHz 8-core Intel Core i9 CPU and 32GB of RAM.

¹⁴<https://aws.amazon.com/ec2/instance-types/p4/>

Absolute Difference Between Resampled and Sampled Metric Means for Dataset = MSVD, Model = CLIP4CLIP

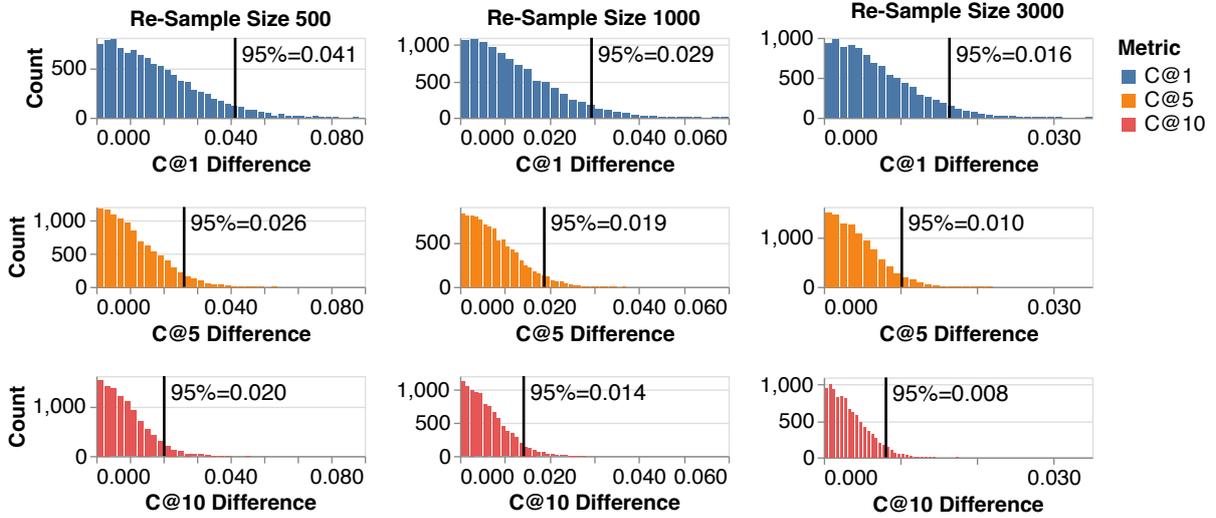


Figure 9: This figure replicates the C@1 results from Figure 4 but adds results for C@5 and C@10. The additional results are consistent in showing that differences of about 1 point are already detectable with 1,000 annotations.

Absolute Difference Between Resampled and Sampled Metric Means for Dataset = MSVD, Model = TeachText

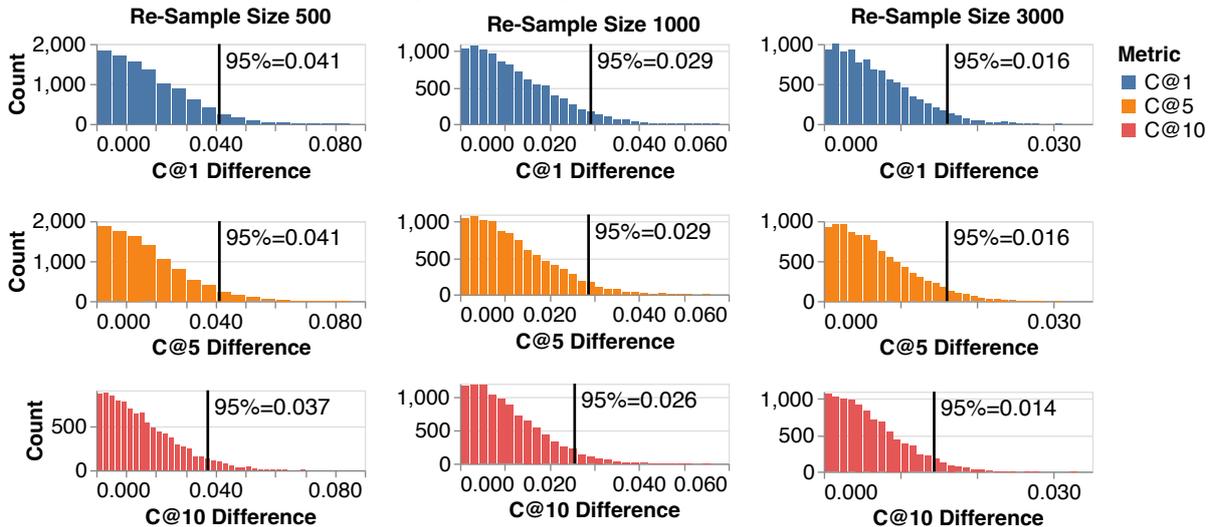


Figure 10: Similar to Figure 4, this figure shows the distribution of absolute differences between bootstrap re-sample estimates of TeachText C@1, C@5, and C@10 scores and their true sample mean (i.e., scores on the full test set). Compared to CLIP4CLIP, statistically significant differences are marginally harder to detect.

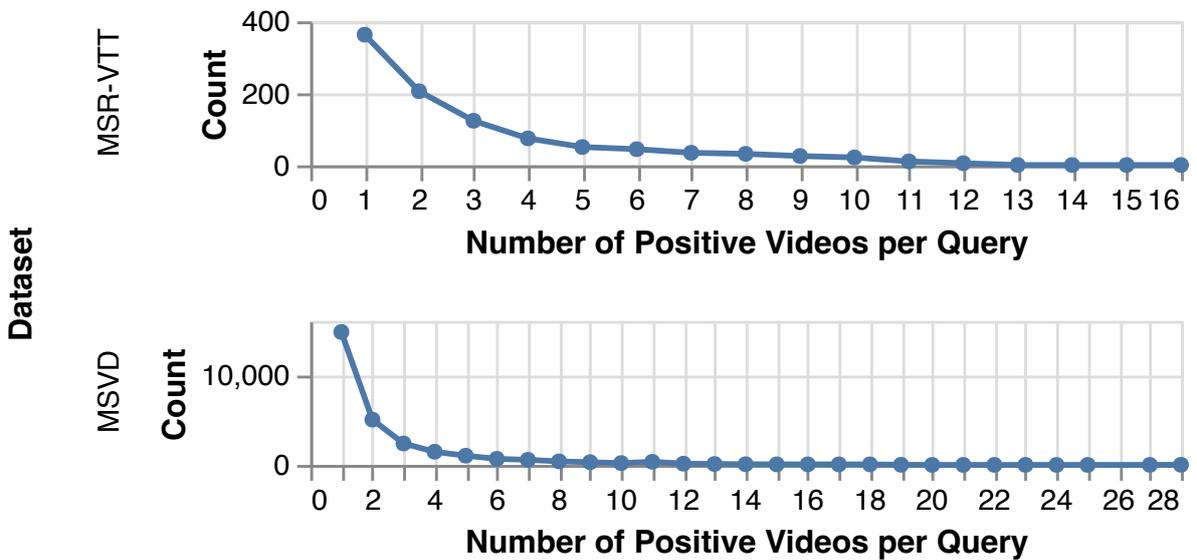


Figure 11: For MSR-VTT and MSVD, we plot the number of positive videos per test set query. While many queries across both datasets have only one known positive, many others have more than that.

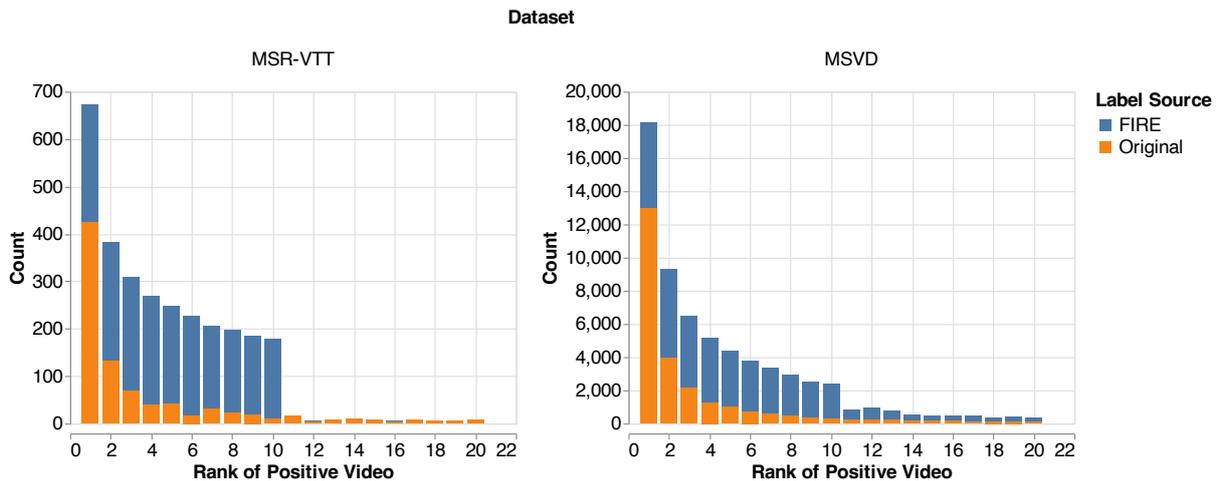


Figure 12: The figure plots the rank of positive video predictions from CLIP4CLIP versus their count. The plot displays MSR-VTT and MSVD separately, and it breaks down the source of each positive (from the original dataset versus from FIRE). While the distribution suggests most positives are found within the top 10, the long tail suggests that there are still unknown positives.

Improving Numeracy by Input Reframing and Quantitative Pre-Finetuning Task

Chung-Chi Chen,¹ Hiroya Takamura,² Ichiro Kobayashi,³ Yusuke Miyao²

¹ Artificial Intelligence Research Center, AIST, Japan

² Ochanomizu University, Japan

³ University of Tokyo, Japan

c.c.chen@acm.com, takamura.hiroya@aist.go.jp,
koba@is.ocha.ac.jp, yusuke@is.s.u-tokyo.ac.jp

Abstract

Numbers have unique characteristics to words. Teaching models to understand numbers in text is an open-ended research question. Instead of discussing the required calculation skills, this paper focuses on a more fundamental topic: understanding numerals. We point out that innumeracy—the inability to handle basic numeral concepts—exists in most pretrained language models (LMs), and we propose a method to solve this issue by exploring the notation of numbers. Further, we discuss whether changing notation and pre-finetuning along with the comparing-number task can improve performance in three benchmark datasets containing quantitative-related tasks. The results of this study indicate that input reframing and the proposed pre-finetuning task is useful for RoBERTa.

1 Introduction

Numerals are an indispensable part of narratives and provide much fine-grained information.¹ How models learn the number system has intrigued many researchers (Spithourakis and Riedel, 2018; Naik et al., 2019; Chen et al., 2019; Wallace et al., 2019; Zhang et al., 2020). Researchers have long discussed some numeracy-related properties of pretrained language models (LMs). In this study, we propose a new concept — *innumeracy*. The problem of innumeracy becomes most evident when models are faced with numerals that do not appear in training data, e.g., when the range of numerals in training data is different from that in the test data. Moreover, LMs often face difficulties understanding numbers even though the numbers are present in the training data. One possible cause of this problem is that numerals can have various notations, some of which are difficult to understand from their subwords. Another possible cause is

¹In this paper, we focus on the numerals represented by digits (0 to 9 and decimal point) and do not discuss those written in words such as “one” and “two”.

Model	Notation	Tokenized Example
BERT	<i>Org.</i>	"147", "##70", "##2"
	<i>Digit</i>	"1", "4", "7", "7", "0", "2"
	<i>SN</i>	"1", ".", "47", "##70", "##200", "##00", "##0", "##e", "+", "05"
RoBERTa	<i>Org.</i>	"147", "702"
	<i>Digit</i>	"1", "4", "7", "7", "0", "2"
	<i>SN</i>	"1", ".", "47", "70", "200000", "E", "+", "05"

Table 1: Tokenized example. *Org.* and *SN* denote original and scientific notation, respectively.

that LMs are not pretrained to deal with numbers. Therefore, in this study, we address the problem of innumeracy via input reframing and quantitative pre-finetuning tasks.

Input reframing refers to changing the notations of numbers, which can be one of the crucial clues for understanding numerals (Zhang et al., 2020; Chen et al., 2021). In addition to the original notation, we consider the digit-based and scientific notations. Table 1 lists examples of using different representations for numerals. Our experiments indicate that RoBERTa (Liu et al., 2019) performs poorly than BERT-based models (Devlin et al., 2019; Yasunaga et al., 2022) in understanding numerals. However, its performance is at par with vanilla BERT-based models with a proper input reframing method. Furthermore, in previous studies, pretraining with the self-supervised learning approach been determined to be a compelling method (Devlin et al., 2019; Yasunaga et al., 2022). However, it is costly to pretrain a new LM from scratch. Thus, an alternative way is to design pre-finetuning tasks to enhance the ability of LMs (Aghajanyan et al., 2021). Inspired by this idea, we propose a novel pre-finetuning task to enhance the ability of the models to deal with quantitative questions and improve the numeracy of the models. Specifically, the proposed method automatically generates a simple dataset for the comparing-numbers task (ComNum), and uses it to pre-finetune LMs. This study experiments with representative pretrained LMs, includ-

ing BERT, RoBERTa, and LinkBERT (Yasunaga et al., 2022), and the experimental results show that pre-finetuning with the proposed ComNum improves the performance in the Quantitative Natural Language Inference (QNLI) task regardless of the LMs used.

To evaluate the influence of the input re-framing and the quantitative pre-finetuning task, we constructed the Quantitative 101 dataset, which is a combination of three benchmark datasets: Numeracy-600K (Chen et al., 2019), EQUATE (Ravichander et al., 2019), and NumGLUE Task 3 (Mishra et al., 2022). The tasks in Quantitative 101 include Quantitative Prediction (QP), QNLI, and Quantitative Question Answering (QQA). In the future, Quantitative 101 can be used as a new collection by researchers studying the quantitative skills of LMs.²

2 Related Work

Numeracy, one of the recent hot topics in NLP, incorporates many skills such as calculation, algebra, and geometry. Some previous studies (Spithourakis and Riedel, 2018; Chen et al., 2019) have discussed the prediction of the masked number tasks, while others (Wallace et al., 2019; Naik et al., 2019; Zhang et al., 2020) have explored numeracy from the perspective of embedding properties. The math word problem (Chen et al., 2021; Mishra et al., 2022) is a high-level task requiring several numeracy skills. The textual representation of numerals, such as digit-based or scientific notations-based, is one of the possible directions for improving numeracy. Chen et al. (2021) suggested to use a digit-based encoder to encode numerals. Meanwhile, Zhang et al. (2020) used scientific notation to represent numerals and explored scale understanding tasks. In this paper, we explore the role of these notations of numbers in quantitative skill tasks.

A recent trend is to design pretraining tasks to enhance the capability of models to understand natural language. Devlin et al. (2019) proposed two pretraining tasks: masked language model (MLM) and next sentence prediction (NSP), and broadened the horizons of the transformer-based natural language processing research direction. Yasunaga et al. (2022) designed a new cross-document pretraining task, called document relation prediction (DRP), to improve the performance of LMs in sev-

²We release this dataset for academic use and follow the license of the sources (Appendix C).

Task	Question	Answer
ComNum	[Num 1] is equal to [Num 2]. [Num 1] is smaller than [Num 2]. [Num 1] is larger than [Num 2].	TRUE/FALSE
QP	FED'S DUDLEY REPEATS EXPECTS GDP GROWTH TO PICK UP IN 2014, FROM [Masked] PCT POST-RECESSION AVERAGE	1
QNLI	S1: Nifty traded above 7500, Trading Calls Today S2: Nifty above 7400	Entailment
QQA	Elliot weighs 180 pounds whereas Leon weighs 120 pounds. Who has a bigger gravity pull? Option1: Elliot Option2: Leon	Option 1

Table 2: Example for each task.

eral benchmark datasets, especially those requiring multi-hop reasoning and multi-document understanding skills. To the best of our knowledge, this is one of the earliest works proposing a tailor-made pre-finetuning task to understanding numerals. Our experimental results also support the usefulness of the proposed task, specifically in the QNLI task.

3 Datasets and Tasks

This section introduces two datasets: the Comparing Numbers Dataset (CND) and Quantitative 101, with the corresponding quantitative tasks, including ComNum, QP, QNLI, and QQA.

3.1 Comparing Numbers Dataset (CND)

Comparing numbers (ComNum) is one of the basic quantitative skills. We propose the Comparing Numbers dataset (CND) to test the ability of different pretrained LMs to perform the ComNum task. CND is an automatically created dataset, and the ComNum task is designed as a binary classification task. In essence, the models need to determine whether a given statement of comparing numbers is true or false. In the CND, there are only three templates as shown in Table 2. There is one training set and two test sets in CND. Specifically, we randomly select two numbers from 0 to 199,999 and insert them into the template. The selected numbers are deleted from the pool of numbers to avoid duplication. Finally, 100,000 instances are obtained, and the numbers in all instances are unique. Note that the distributions of each template and answers are balanced. 80% of the dataset is considered as the training set and the remaining 20% is taken as the CND-T1 test set. Next, two numbers from 4,000,000 to 5,000,000 are randomly selected for 10,000 times to construct the CND-T2 test set. Thus, the order of magnitude of the training set and the first test set (CND-T1) is from 0 to 5, and that of the other test set (CND-T2) is 6. In this study, we focused on natural numbers, and fu-

	BERT		RoBERTa		LinkBERT		FinBERT	
	CND-T1	CND-T2	CND-T1	CND-T2	CND-T1	CND-T2	CND-T1	CND-T2
<i>Original</i>	99.86	95.59 (↓ 4.27)	99.44	86.75 (↓ 12.69)	99.92	97.58 (↓ 2.34)	99.55	78.37 (↓ 21.18)
<i>Digit-based</i>	99.96	99.03 (↓ 0.93)	99.92	98.46 (↓ 1.46)	99.99	96.54 (↓ 3.45)	99.96	97.03 (↓ 2.93)
<i>Scientific Notation</i>	99.92	99.68 (↓ 0.24)	99.82	99.13 (↓ 0.69)	99.95	99.81 (↓ 0.14)	99.72	98.78 (↓ 0.94)

Table 3: Experimental results of ComNum task. The evaluation metric is Micro-average of F1 score (%).

Model	Notation	QP		QNLI				Stress Test	QQA	Score
		Comment	Headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI			
BERT	<i>Original</i>	70.44%	57.46%	64.40%	59.20%	72.29%	60.42%	99.91%	53.20%	67.17
	<i>Digit-based</i>	65.38%	54.74%	57.86%	56.46%	71.36%	60.11%	99.11%	53.75%	64.85
	<i>Scientific Notation</i>	65.31%	55.99%	64.42%	60.73%	72.23%	59.66%	99.56%	53.24%	66.39
CN-BERT	<i>Digit-based</i>	69.93%	54.84%	61.07%	60.27%	75.54%	65.39%	99.42%	52.53%	67.37
	<i>Scientific Notation</i>	64.87%	56.40%	66.39%	54.70%	75.41%	63.94%	99.42%	51.90%	66.63
LinkBERT	<i>Original</i>	68.81%	55.70%	59.94%	56.85%	73.43%	59.01%	99.91%	54.14%	65.97
	<i>Digit-based</i>	63.76%	55.41%	59.54%	57.42%	73.63%	60.17%	99.73%	53.44%	65.39
	<i>Scientific Notation</i>	65.81%	56.05%	57.00%	56.78%	75.51%	58.51%	99.82%	54.33%	65.48
CN-LinkBERT	<i>Digit-based</i>	68.61%	54.44%	63.59%	55.08%	71.21%	58.99%	100.00%	50.44%	65.30
	<i>Scientific Notation</i>	63.48%	53.15%	62.02%	59.39%	75.70%	62.61%	99.73%	52.11%	66.02

Table 4: Experimental results of the BERT-based models. The results in bold are the ones that are better than the *Original*. The score indicates Quantitative-101 Score.

ture studies can extend our results to decimals and fractions. Since natural numbers are in the infinite set, and it is impossible to let models learn with a dataset containing all magnitudes and numbers, we designed the task in the way following the human learning process because human beings do not need to learn to count from zero to trillion to get the ability to compare all numbers.

3.2 Quantitative 101

Quantitative 101 collects recent benchmark datasets and focuses on quantitative tasks. There are three tasks in Quantitative 101, including Quantitative Prediction (QP), Quantitative Natural Language Inference (QNLI), and Quantitative Question Answering (QQA). This section briefly introduces the tasks, and we further provide details in Appendix C.

QP is the task of predicting the correct magnitude of the masked numeral. Although a possible choice would be to predict the exact number given a context, doing so is often very difficult, even for a human. For example, the QP listed in Table 2, in which the correct answer is 2.2. However, making an accurate rough estimate for the magnitude would often be feasible only for seasoned experts. We attempt to test whether models can also learn such a numeracy skill after being trained with a large amount of data. Thus, we adopt Numeracy-600K (Chen et al., 2019) as the dataset for this task. Chen et al. (2019) designed this task as an eight-class classification task, which includes the magnitude from 1 to 6, decimal, and a magnitude

larger than 6. Numeracy-600K contains two subsets: market comments and blog headlines.

QNLI is the task of making natural language inferences based on quantitative clues. It is a complex version of ComNum, because the given sentences could be varied. The example of QNLI presented in Table 2 shows that models need to compare numbers based on more complex semantics. We selected EQUATE (Ravichander et al., 2019) to experiment on real-world scenarios for QNLI. EQUATE has five subsets, including RTE-QUANT, AWP-NLI, NEWSNLI, REDDITNLI, and Stress Test.

QQA is the other format for testing whether models can understand numerals and semantics. We selected the Task 3 subset of NumGLUE (Mishra et al., 2022) for the QQA experiments. Table 2 provides an example of this dataset. It is under a binary-classification setting, and each instance has two options.

We chose these three datasets to test the basic quantitative skills of models. We noticed that several instances in these datasets can be solved using only the basic ability to understand numbers. However, the other subtasks in NumGLUE required reasoning skills including the generation of equations. These tasks are not the target of this paper.

4 Methods

4.1 Notation of Numbers

The findings of previous studies (Chen et al., 2021; Zhang et al., 2020) suggest two methods that are worth trying: digit-based notation and scientific no-

Model	Notation	QP		QNLI					QQA	Score
		Comment	Headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI	Stress Test		
RoBERTa	<i>Original</i>	60.46%	58.03%	60.15%	57.64%	79.58%	58.77%	98.93%	51.96%	65.69
	<i>Digit-based</i>	69.25%	57.65%	59.40%	56.69%	78.90%	62.38%	99.91%	54.34%	67.31
	<i>Scientific Notation</i>	64.32%	55.49%	60.08%	57.41%	78.68%	60.81%	100.00%	53.67%	66.31
CN-RoBERTa	<i>Digit-based</i>	64.25%	55.92%	68.96%	58.80%	77.99%	60.99%	99.73%	50.88%	67.19
	<i>Scientific Notation</i>	60.28%	54.85%	62.15%	58.74%	65.92%	59.59%	99.47%	52.27%	64.16

Table 5: Experimental results of the RoBERTa-based models.

tation. Table 1 shows an example for each method. *Original* signifies that we did not perform any pre-processing on the input data, and the results are tokenized based on WordPiece (Schuster and Nakajima, 2012; Wu et al., 2016) and Byte-Pair Encoding (BPE) (Sennrich et al., 2016). In the *Digit-based* method, we separated a numeral into digits. In the *Scientific Notation* method, we converted numerals into scientific notation according to the method described in Zhang et al. (2020), and Table 1 provides examples to show that tokenizers provide different results in this case. Note that we pad the mantissa to 10 significant figures to retain the information of most numerals.

4.2 Pre-Finetuning Task

We pre-finetune LMs with the CND for learning the numeracy of comparing numbers. We believe that this learning process can make models aware of the numerals and may help answer the questions listed in Table 2. We further test whether the proposed pre-finetuned method is helpful in the QP, QNLI, and QQA tasks. We primarily use BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and LinkBERT (Yasunaga et al., 2022) for the experiments. Since the market comment subset for the QP task is in the financial domain, we also experiment with FinBERT (Araci, 2019) in this subset. The pre-finetuned LMs using BERT, RoBERTa, LinkBERT, and FinBERT as initial models are named CN-BERT, CN-RoBERTa, CN-LinkBERT, and CN-FinBERT, respectively. During the pre-finetuning process, we use the *Digit-based* or *Scientific Notation* reframing methods to transform the numerals in the input data. Thus, each proposed pre-finetuned LM has two versions depending on the notation of numbers.

5 Experiment

5.1 Innumeracy

Innumeracy can be tested via various experiments. In this section, we observe the innumeracy phenomenon with the empirical results of the Com-

Model	Reframing	QP-Comment
FinBERT	<i>Original</i>	65.26%
	<i>Digit-based</i>	69.89%
	<i>Scientific Notation</i>	70.03%
CN-FinBERT	<i>Digit-based</i>	68.84%
	<i>Scientific Notation</i>	69.76%

Table 6: Results of the FinBERT-based models.

Num task. We aim to answer whether LMs have different performances between CND-T1 and CND-T2. We use the micro-average of the F1 score to evaluate the results of the ComNum task. Table 3 shows the results. It is not surprising that models perform well in CND-T1. However, model performances drop when we test using CND-T2. In CND-T2, the order of magnitude of the numerals is different from that in the training set. We call this phenomenon “innumeracy”, and find that both *Digit-based* and *Scientific Notation* perform well for most pretrained LMs. In particular, using *Scientific Notation* method leads to the least performance drops with all LMs.³

5.2 Experimental Results

We follow the setting of previous studies to use the macro-average of F1 score for the QP task and the micro-average of F1 score for the QNLI and QQA tasks. Table 4 presents the results of the BERT-based models, and Table 5 presents the results of the RoBERTa-based models.⁴ To evaluate the aggregate performance, we average all results as in previous studies (Dua et al., 2019; Mishra et al., 2022), and named this score the Quantitative-101 Score. First, it can be observed that all notation methods and the pre-finetuning task improved the overall performance of RoBERTa, and lead RoBERTa to perform at par with the BERT-based LMs. Second, we observed that the proposed pre-finetuning task helped improve the QNLI task performance. Third, using a proper reframing method improved the QQA task performance. Fourth, the

³We provide more analysis on this point in Appendix B.

⁴We provide a fine-grained analysis in Appendix A for the QNLI-Stress Test.

Model	Preprocessing	QP		RTE-QUANT	QNLI			Stress Test	QQA	Score
		Comment	Headline		AWP-NLI	NEWSNLI	REDDITNLI			
RoBERTa		60.46%	58.03%	60.15%	57.64%	79.58%	58.77%	98.93%	51.96%	65.69
CN-RoBERTa	<i>Original</i>	86.86%	77.29%	62.52%	56.70%	78.82%	64.29%	99.94%	50.71%	72.14

Table 7: Results of CN-RoBERTa without input reframing.

reframing methods and the pre-finetuning task were not helpful for the BERT-based LMs in the QP task as well as the overall performance.

Table 6 shows the results of the FinBERT-based models in QP-comment. The results indicate that the performances of FinBERT can be improved with a proper reframing method. Additionally, the proposed CN-FinBERT performs better than the *Original* FinBERT.

To sum up our findings, the input reframing methods can improve the performance of RoBERTa and FinBERT. However, it does not work for BERT-based models. The proposed pre-finetuning task can improve the performance in the QNLI task regardless of the LM used.

5.3 Ablation Analysis

In this section, we train CN-RoBERTa without input reframing for ablation analysis. Table 7 shows the results. The results indicate that the performances of QP tasks were improved significantly, and the performance of QNLI tasks was also improved. These results indicate the proposed pre-finetuning task is important for the QP tasks, but input reframing is not. However, the performance of the QQA did not improve without input reframing. This result implies that, for QQA, input reframing provides some hints to the models to make predictions. Overall, this study does not find a silver bullet for solving quantitative problems, but shows that input reframing and basic quantitative pre-finetuning design are promising directions.

6 Conclusion

This study deals with the innumeracy of LMs and shows that the notation of numbers matters, especially for RoBERTa. We also propose a novel pre-finetuning task for improving the quantitative skills, and find that the performance in the QNLI task can be improved after pre-finetuning. We hope our results in Quantitative 101 lead to a more in-depth discussion on the ability of LMs to understand numerals.

Limitations

The first limitation of the paper is that we focus on the numerals represented by digits (0 to 9 and decimal point) and do not discuss those written in words such as “one” and “two”. Future work can extend the findings of this work and transfer the numeral words to digits. The second limitation of this paper is that we do not discuss long text scenarios because the length of the instances in the datasets is within 512. Future work can design quantitative-related tasks with longer documents and examine whether the proposed methods still work. The third limitation of this paper is that we do not train the model from scratch with the proposed input reframing methods. We leave it as one of the open questions for future studies. The fourth limitation of this work is that we do not experiment with all cases, including using data in several ranges and experimenting with all kinds of pretrained LMs, to prove that the innumeracy phenomenon is a general phenomenon. Instead, we present a pilot exploration of the phenomenon and further pay attention to improving the performances of other quantitative-related tasks.

Ethical Note

All datasets used in our experiment are available online, and we provide the details and the license information in Appendix C. We release the pre-finetuned LMs (CN-BERT, CN-RoBERTa, CN-LinkBERT, and CN-FinBERT) on the Hugging Face models platform.⁵ Future work can reproduce our results easily and use our pre-finetuned LMs for further research issues. Please refer to Appendix B for details.

Acknowledgements

This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

⁵<https://huggingface.co/models>

References

- Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811.
- Dogu Araci. 2019. FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.
- Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. NQuAD: 70,000+ questions for machine comprehension of the numerals in text. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2925–2929.
- Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.
- Aakanksha Naik, Abhilasha Ravichander, Carolyn Rose, and Eduard Hovy. 2019. Exploring numeracy in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3374–3380, Florence, Italy. Association for Computational Linguistics.
- Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.
- Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Georgios Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2104–2115, Melbourne, Australia. Association for Computational Linguistics.
- Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

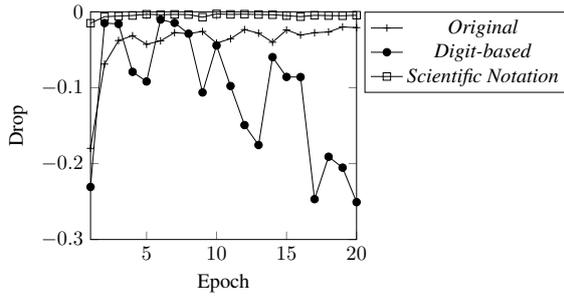


Figure 1: BERT’s innuery phenomenon. (Performance Drop between CND-T1 and CND-T2.)

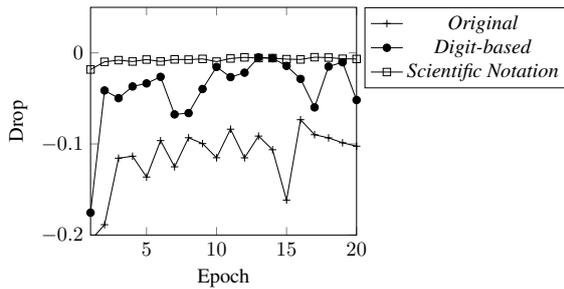


Figure 2: RoBERTa’s innuery phenomenon. (Performance Drop between CND-T1 and CND-T2.)

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020. [Do language embeddings capture scales?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4889–4896, Online. Association for Computational Linguistics.

A Analysis of QNLI-Stress Test

QNLI-Stress Test uses the data collected from AQUA-RAT, and was annotated by an automatic method (Ravichander et al., 2019). We follow the splitting method in NumGLUE Task 7 (Mishra et al., 2022) to separate it into training, development, and test sets. First, we find 316 repeated instances in both training and evaluation sets (development and test sets). We already removed these repeated instances from the training set in our experiment. Second, we check the instances by removing all numerals in each instance and find that 2,229 instances appear in both training and evaluation sets, with 1,639 appearing in the same training and test sets, and 80.17% have the same answer. That could be the reason that the models perform well in this dataset, since most instances do not need to understand numerals.

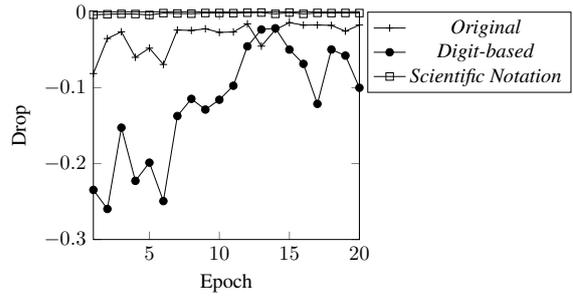


Figure 3: LinkBERT’s innuery phenomenon. (Performance Drop between CND-T1 and CND-T2.)

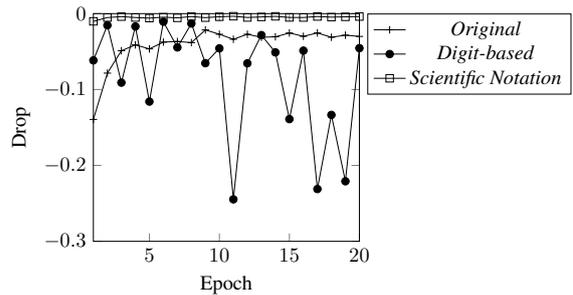


Figure 4: FinBERT’s innuery phenomenon. (Performance Drop between CND-T1 and CND-T2.)

B Implementation Detail

We used the Hugging Face transformers package (Wolf et al., 2019) for the experiment.⁶ Intel Xeon Gold CPU and Nvidia Tesla V100 w/32GB are the CPU and GPU used in our experiment. Table 8 provides the links to the LMs used in our experiment. All pre-finetuned LMs (CN-BERT, CN-RoBERTa, CN-LinkBERT, and CN-FinBERT) are released on the Hugging Face platform.

Figure 1 to 4 present the tracing results of the drop between CND-T1 and CND-T2 during the training process. It can be observed that when using Scientific Notation, the performances of LMs stabilizes more quickly. In contrast, the change of the performances with the *Digit-based* method varies, and we did not obtain stable results in some cases.

C Dataset

CND is our own generated dataset; therefore, we did not have to obtain license permissions to use it. There are three subsets in the proposed Quantitative 101. Numeracy-600K (Chen et al., 2019) for

⁶<https://huggingface.co/docs/transformers/index>

	URL
BERT (Devlin et al., 2019)	https://huggingface.co/bert-base-uncased
RoBERTa (Liu et al., 2019)	https://huggingface.co/roberta-base
LinkBERT (Yasunaga et al., 2022)	https://huggingface.co/michiyasunaga/LinkBERT-base
FinBERT (Araci, 2019)	https://huggingface.co/ProsusAI/finbert

Table 8: Reference for the models in our experiments.

Model	Reframing Method	URL
CN-BERT	<i>Digit-based</i>	https://huggingface.co/NLPFin/CN-BERT-Digit
	<i>Scientific Notation</i>	https://huggingface.co/NLPFin/CN-BERT-Sci
CN-RoBERTa	<i>Original</i>	https://huggingface.co/NLPFin/CN-RoBERTa
	<i>Digit-based</i>	https://huggingface.co/NLPFin/CN-RoBERTa-Digit
	<i>Scientific Notation</i>	https://huggingface.co/NLPFin/CN-RoBERTa-Sci
CN-LinkBERT	<i>Digit-based</i>	https://huggingface.co/NLPFin/CN-LinkBERT-Digit
	<i>Scientific Notation</i>	https://huggingface.co/NLPFin/CN-LinkBERT-Sci
CN-FinBERT	<i>Digit-based</i>	https://huggingface.co/NLPFin/CN-FinBERT-Digit
	<i>Scientific Notation</i>	https://huggingface.co/NLPFin/CN-FinBERT-Sci

Table 9: Reference for the proposed models.

QP task could be downloaded from GitHub⁷, and it is under the Creative Commons Attribution-Non-Commercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. EQUATE (Ravichander et al., 2019) for QNLI task can also be downloaded from GitHub⁸, and it is under the MIT License. NumGLUE (Mishra et al., 2022) for QQA task can be downloaded from the page of Allen Institute for AI (AI2)⁹, and it is under the ODC Attribution License (ODC-By).¹⁰ In the following subsections, we provide details of each subset. The README document of the dataset provides all details about the separation. Please download the dataset from <https://huggingface.co/datasets/NLPFin/Quantitative101>.

C.1 Quantitative Prediction

Quantitative prediction (QP) is a task to predict the correct magnitude of the masked numeral. For example, even for a human, it is difficult to predict the exact numeral (2.2) of the QP’s instance in Table 2; however, some seasoned experts can make a correct rough estimate of the magnitude. We attempt to test whether models also learn to make such predictions after being trained with a large amount of data. Thus, we adopt Numeracy-600K (Chen et al., 2019) as the dataset for this task. Chen et al.

⁷<https://github.com/aistairc/Numeracy-600K>

⁸<https://github.com/AbhilashaRavichander/EQUATE/blob/master/LICENSE>

⁹<https://allenai.org/data/numglue>

¹⁰<https://github.com/allenai/numglue/blob/main/license.txt>

(2019) designed this task as an eight-class classification task, which includes the magnitude from 1 to 6, decimal, and the magnitude larger than 6. We follow their setting in this paper. There are two subsets, including 600K market comments and 600K news headlines. We use 80%, 10%, and 10% of instances as training, development, and test sets in each subset, respectively.

C.2 Quantitative Natural Language Inference

Quantitative Natural Language Inference (QNLI) is a complex version of ComNum because the given sentences can be varied. The example of QNLI provided in Table 2 shows that models need to compare numbers based on more complex semantics. We select EQUATE (Ravichander et al., 2019) to experiment on real-world scenarios for QNLI. EQUATE has five subsets collected from different sources, including RTE-QUANT, AWP-NLI, NEWSNLI, REDDITNLI, and Stress Test. Since four of these subsets are less than 1,000 instances, we perform the 10-fold cross-validation in the experiments. For the Stress Test, which contains 7,500 instances, we follow the splitting method in NumGLUE Task 7 (Mishra et al., 2022) to separate it into training, development, and test sets. Ravichander et al. (2019) designed the QNLI task as a two or three-class classification task depending on the subset. We follow their settings for each subset.

C.3 Quantitative Question Answering

Quantitative Question Answering (QQA) is the other format for testing whether models can un-

derstand numerals and semantics. We selected the Task 3 subset of NumGLUE (Mishra et al., 2022) for the QQA experiments. Table 2 provides an example of this dataset. It is under a binary-classification setting, and each instance has two options. We follow Mishra et al. (2022) to separate the dataset into training, development, and test sets.

Visualize Before You Write: Imagination-Guided Open-Ended Text Generation

Wanrong Zhu[¶], An Yan[†], Yujie Lu[¶], Wenda Xu[¶],
Xin Eric Wang[§], Miguel Eckstein[¶], William Yang Wang[¶]

[¶]UC Santa Barbara, [†]UC San Diego, [§]UC Santa Cruz

{wanrongzhu,yujielu,wendaxu,william}@cs.ucsb.edu, ayan@ucsd.edu

xwang366@ucsc.edu, miguel.eckstein@psych.ucsb.edu

Abstract

Recent advances in text-to-image synthesis make it possible to visualize machine imaginations for a given context. On the other hand, when generating text, human writers are gifted at creative visualization, which enhances their writings by forming imaginations as blueprints before putting down the stories in words. Inspired by such a cognitive process, we ask the natural question of whether we can endow machines with the same ability to utilize visual information and construct a general picture of the context to guide text generation. In this work, we propose iNLG that uses machine-generated images to guide language models (LM) in open-ended text generation. The experiments and analyses demonstrate the effectiveness of iNLG on open-ended text generation tasks, including text completion, story generation, and concept-to-text generation in both few-shot and full-data scenarios. Both automatic metrics and human evaluations verify that the text snippets generated by our iNLG are coherent and informative while displaying minor degeneration.¹

1 Introduction

One great resource human writers cherish is the ability of imagination, with which they render mental images about an actual or vicarious experience and link knowledge that would later make the writing more concrete, sensible, and intriguing. Cognitive studies show that visual imagery improves comprehension during language processing (Gambrell and Bales, 1986; Joffe et al., 2007; Sadoski and Paivio, 2000), and that mental imagery facilitates humans’ written language expression at young ages (Gambrell and Koskinen, 2002).

When it comes to the study of Artificial Intelligence (AI), one classic challenge for AI systems is to generate informative and coherent text snippets. Open-ended text generation is such a task that provides an input context, and asks the model to

¹Our code & data: <https://github.com/VegB/iNLG>.

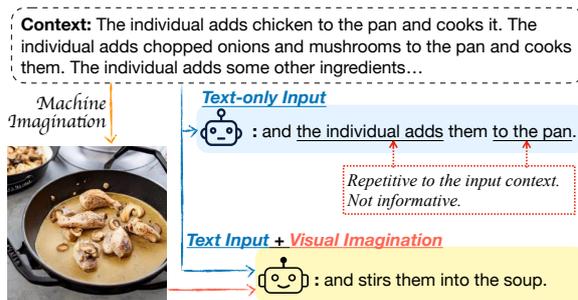


Figure 1: When performing open-ended text generation, the language models prompted with text-only input may generate repetitive or unilluminating contents, which is also known as degeneration. Hereby, we propose to use machine-generated images as additional visual supervision to guide the language models in generating more informative and coherent text with the given context.

generate a piece of text that is consistent with the context. This is the cornerstone of a wide range of downstream tasks such as text completion (Guan et al., 2019; Radford et al., 2019), story generation (Fan et al., 2018; Goldfarb-Tarrant et al., 2020; Swanson et al., 2021; Su et al., 2022b), and dialogue systems (Schatzmann et al., 2007; Wen et al., 2015, 2017; Wei et al., 2018; Wu et al., 2021), and has received much attention throughout the years. Inspired by human writers’ common practice of creative visualization, we ask the following question: Can we endow machines with the same ability to construct a general picture of the context and use it as a blueprint to guide text generation?

Recent advances in text-to-image generation make it possible to visualize machine imaginations for a given context (Ramesh et al., 2021; Rombach et al., 2022; Crowson et al., 2022; Wang et al., 2022b; Saharia et al., 2022). Moreover, this line of work shows great potential in utilizing textual information to guide image synthesis. It comes naturally that one may attempt to complete the loop by using visual supervision to guide text generation.

In this work, we propose using machine-

generated images to guide the language model (LM) in open-ended text generation. More specifically, we visualize machine imagination for the input context by rendering images with StableDiffusion (Rombach et al., 2022), a state-of-the-art text-to-image generator. The machine imagination acts as additional visual supervision to guide LMs in generating informative and coherent text in two ways. Firstly, the machine-generated images are introduced as the input to the LM in the form of the visual prefix. Secondly, we designed a contrastive training objective that enforces the generated text to be semantically similar to the visual supervision.

We conduct experiments on three open-ended text generation tasks, namely text completion, story generation, and concept-to-text generation. Extensive experiments in the few-shot settings show better or competitive performance to state-of-the-art baselines on both automatic metrics and human evaluation. Experiments with full-data settings show that introducing machine-generated visual supervision with our iNLG yields consistent improvements on various LM models including GPT-2 (Radford et al., 2019), BART (Lewis et al., 2020), and T5 (Raffel et al., 2020).

Our main contributions are as follows:

- We introduce a novel paradigm that leverages machine-generated images to guide open-ended text generation. This endows the machines with the ability of creative visualization that human writers often demonstrate.
- We distill the vision information from the pre-trained multimodal models and further construct visual prefixes to guide language models performing text generation with teacher forcing and contrastive objectives.
- Extensive experiments show the effectiveness of iNLG as a model-agnostic framework in open-ended text generation tasks, including text completion, story generation, and concept-to-text in both few-shot and full-data settings.

2 Related Work

Open-ended Conditional Text Generation is the task of generating a coherent portion of the text based on the given context. Recent advances in pre-trained models have pushed frontier in the open-ended conditional text generation, such as text completion (See et al., 2019; Ippolito et al., 2020), story generation (Guan et al., 2020; Fan

et al., 2018; Yao et al., 2019) and concept-to-text generation (Zhou et al., 2021; Liu et al., 2021). Despite the success of large language models, text degeneration and semantic coverage still remain as two core technical challenges in few-shot open-ended text generation. To improve the text coverage, StoryEndGen (Guan et al., 2019) leverages the knowledge graph to encode context sequentially. Fan et al. (2018) and Yao et al. (2019) plan the content (premise or keywords) first and then encourage the generation based on planned content. To mitigate the text degeneration, SimCTG (Su et al., 2022b) uses a contrastive training strategy to encourage the model to learn isotropic token embeddings. Similar to our approach, Wang et al. (2022a) generates a scene graph for each concept and combines them with text for the model input. Previous work has proposed to add visual information to LM by retrieving images from the Internet or large-scale image sets (Yang et al., 2020; Cho et al., 2021; Su et al., 2022a). However, the retrieved images may fail to fully incorporate the context, which will misguide the LM from yielding contextually consistent predictions.² Unlike prior work, our approach leverages images generated conditioning on the context to assist the text generation process.

Visually-aided NLP Recent work show the power of visual guidance in natural language processing, spanning from the language representation learning (Lu et al., 2019; Li et al., 2019; Sun et al., 2019; Luo et al., 2020; Chen et al., 2020; Li et al., 2020; Tan and Bansal, 2020; Lu et al., 2022), the downstream tasks (Grubinger et al., 2006; Elliott et al., 2016; Xie et al., 2019; Christie et al., 2016; Shi et al., 2019; Lu et al., 2022) and evaluation (Zhu et al., 2021). They either leverage visual information from an external vision-and-language corpus or obtain such visual knowledge from the large pre-trained model. In this line of work, imagination achieves promising performance in various NLP domains (Long et al., 2021; Zhu et al., 2021; Wang et al., 2022a; Lu et al., 2022). Previous imagination-based work in NLP either study non-generation problems (Zhu et al., 2021; Lu et al., 2022) or utilize non-visual information (Long et al., 2021; Wang et al., 2022a). Our work explores the potential of generating visual imagination to improve open-ended text generation tasks.

²Figure 8 shows examples where the image retrieved from the search engine is irrelevant with the input context.

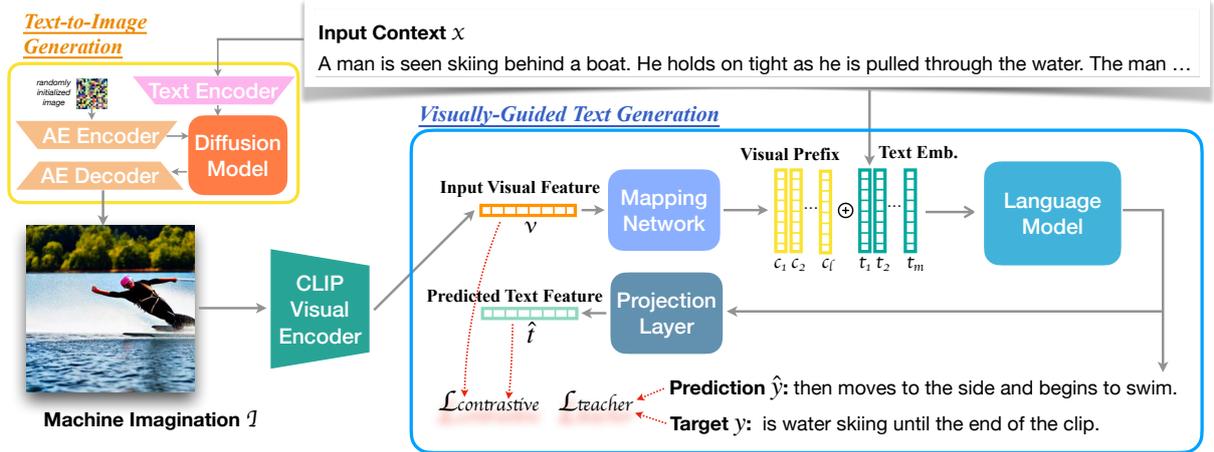


Figure 2: An overview of our iNLG. Given an input context x , we first visualize the context with the text-to-image generation model. Then we use the machine-generated image I as the additional visual supervision to guide the language model in open-ended text generation. The visual feature is provided as a source of input to the LM in the form of the visual prefix. Aside from the teacher forcing objective $\mathcal{L}_{\text{teacher}}$, we also enforce the LM to generate text that is semantically similar to the machine imagination with a contrastive training objective $\mathcal{L}_{\text{contrastive}}$.

3 Method

3.1 Overview

Open-ended text generation is a task that provides an input context, and asks the model to generate a piece of text that is consistent with the context.

This work mainly focused on introducing machine-rendered images to assist LM in performing open-ended text generation. More specifically, given the context x^i , we first use a text-to-image generator to illustrate an image I^i that depicts the input context. The LM is prompted with image I^i as the visual prefix along with the text context x^i , and will incorporate the multimodal input to generate the output text \hat{y}^i .

Figure 2 provides an overview of our iNLG framework, which mainly involves two modules. The first module is a text-to-image generator that takes in the input context and illustrates a descriptive image, which we also refer to as the machine imagination. The second module is a visually-guided language model that utilizes the machine imagination as a source of input and also a supervision that encourages the LM to generate text that is semantically similar to the visual information.

3.2 Text-to-Image Rendering

In this work, we propose to use images generated conditioning on the context by the machines as additional visual information to the LM. The text-to-image generation backbone is StableDiffusion (Rombach et al., 2022), which mainly consists

of a text encoder, a diffusion model, and an auto-encoder. The text encoder is from the frozen CLIP ViT-L/14 (Radford et al., 2021) and encodes the input text to textual embeddings. The diffusion model uses UNet (Ronneberger et al., 2015) to provide noise estimation. The UNet is modified so as to attend to the input textual embeddings. The encoder of the pretrained autoencoder encodes images into the lower-resolution latent maps z_T . At each step t , the diffusion model provides the noise estimation ϵ and modifies z_t correspondingly. The decoder of the pretrained autoencoder takes the final noise-free latent map z and generates the image prediction. StableDiffusion is trained with LAION-5B (Schuhmann et al., 2022).

3.3 Visually Guided Text Generation

Visual Prefix Construction One can encode the visual information with the pre-trained visual models. However, such visual embedding may lie in a representation space different from the LM due to the discrepancy between models. One way of introducing features extracted by another network to the current model is through feature mapping (Mokady et al., 2021). With a dataset of image-text pairs (I', x') , we can pre-train a mapping network \mathcal{F} for a given LM in an image captioning formulation. More specifically, we encode I' with the visual encoder $\text{Enc}_{\text{visual}}$ and receive its visual features v' . Then we apply the mapping network \mathcal{F} over v' , and receive a sequence of l visual prefixes:

$$c'_1, c'_2, \dots, c'_l = \mathcal{F}(v') = \mathcal{F}(\text{Enc}_{\text{visual}}(I')) \quad (1)$$

We provide the list of visual prefix as input to the LM with the corresponding text x' as the target output. Such a pre-training process enables \mathcal{F} to project visual features into the visual prefix that lies within the same embedding distributions as the LM. The mapping network is agnostic of the downstream task, and only depends on the visual source and the LM.

After generating a descriptive image I^i for the input context x^i , we use CLIP to encode I^i and receive its visual features v^i . We apply the pre-trained mapping network \mathcal{F} over v^i , and receive the visual prefix c^i of length l :

$$c^i = \{c_1^i, c_2^i, \dots, c_l^i\} = \mathcal{F}(\text{CLIP}(I^i)) \quad (2)$$

Visually-guided Language Modeling We use the visual information to guide text generation in two ways, reflected in the following two training objectives. Firstly, we directly introduce the machine-generated visual information as input to the LM. We concatenate the visual prefix c^i and the text embeddings t^i for the input context x^i with m tokens. LM input can be denoted as $[c^i; t^i] = \{c_1^i, \dots, c_l^i, t_1^i, \dots, t_m^i\}$. With $y^i = \{y_1^i, y_2^i, \dots, y_n^i\}$ denoting the target output of n tokens, and θ denoting the trainable parameters, we can list out the teacher forcing training objective as follows:

$$\mathcal{L}_{\text{teacher}} = - \sum_{j=1}^n \log p_{\theta}(y_j^i | c^i; t^i; \mathbf{y}_{<j}^i) \quad (3)$$

In addition, we design a contrastive objective to enforce the generated text to be semantically similar to the input visual supervision with the InfoNCE loss (van den Oord et al., 2018; Yan et al., 2021):

$$\mathcal{L}_{\text{contrastive}} = - \log \frac{\exp(\text{sim}(\mathbf{v}^i, \hat{\mathbf{t}}^i)/\tau)}{\sum_{j \neq i} \exp(\text{sim}(\mathbf{v}^i, \hat{\mathbf{t}}^j)/\tau)} \quad (4)$$

in which $\hat{\mathbf{t}}$ is the projected representation of the decoder’s last layer’s output, and can be viewed as the sentence-level representation of the generated text. Here $\text{sim}(\cdot, \cdot)$ first normalizes the two vectors, then compute their cosine similarity, and τ is the temperature.

3.4 Training & Inference

We first pre-train the mapping network on the pre-training dataset with the teacher-forcing objective. Such pre-training is agnostic of the downstream task, and only depends on the type of base LM.

When applying our iNLG on downstream tasks, we train the base LM with the teacher forcing objective for the first $N_{\text{no_contra}}$ epochs. Then, we introduce the contrastive objective and tune the base LM together with the mapping network and projection layer by minimizing the following loss \mathcal{L} . Here ep denotes the epoch and λ is the factor:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{teacher}}, & ep < N_{\text{no_contra}}, \\ \mathcal{L}_{\text{teacher}} + \lambda \mathcal{L}_{\text{contrastive}}, & ep > N_{\text{no_contra}}, \end{cases} \quad (5)$$

During inference, we provide the context and machine-generated image to the LM. We use beam search during decoding with a beam width of 10.

4 Experimental Setup

4.1 Tasks, Datasets, and Baselines

We apply our iNLG on three open-ended text generation setups: sentence completion, story generation, and concept-to-text generation. Table 1 shows examples for each task.

Sentence Completion is a task of finishing the sentence in a commonsense inference scenario. We conduct experiments on the ActivityNet (Heilbron et al., 2015) subset³ of HellaSwag (Zellers et al., 2019), which is a benchmark for commonsense natural language inference that ask the model to predict the most likely follow-up among several choices given a specific context. We compare with StoryEndGen (Guan et al., 2019) which encodes the given context incrementally and attends to the one-hop knowledge graph retrieved from ConceptNet for the context tokens. We implement our iNLG on top of the GPT-2 (Radford et al., 2019), which by nature, can generate the follow-up for an arbitrary input in a zero-shot manner.

Story Generation requires the model to compose a story based on the given title or context. We conduct experiments on the widely used story generation benchmark ROCStories (Mostafazadeh et al., 2016). Each data item consists of a story title and a human-written five-sentence everyday life story that incorporates commonsense related to the title.⁴ We provide the story title and the story’s first sentence as the input context, and ask the LM to predict the following four sentences. We consider the

³14740/982/2261 samples for train/validation/test.

⁴We use the split provided by Su et al. (2022a), which is based on the ROCStories Winter 2017 release and contains 49666/1500/1500 items for the train/validation/test sets.

Task	Input Context	Target Output
Text Completion	Different people are interviewed on camera while several others are shown raking up the leaves. A man is seen sitting in his car and another puts his gloves on. The camera	pans over the raked up leaves while several others discuss their hard work.
Story Generation	Live Show. Tim was in his school’s play.	He was nervous about their first show. He almost dropped out. The show went smoothly. Tim was excited for his second show.
Concept-to-Text	grow, flower, pavement	Wild flower growing through crack in the tiled pavement.

Table 1: Exemplars of the input context and corresponding target output for three open-ended text generation task covered in this study, namely story generation, text completion, and concept-to-text generation.

following methods as baselines: Action-Plan (Fan et al., 2018) first predicts the premise of a story with the convolutional LM (Dauphin et al., 2017), then use the fusion mechanism (Sriram et al., 2018) to encourage a convolutional seq2seq model (Gehring et al., 2017) to generate the story from the premise. Plan-and-Write (Yao et al., 2019) first plans a storyline that consists of keywords, then generate the story conditioned on the storyline. Its model structure is built upon GRU (Cho et al., 2014). SimCTG (Su et al., 2022b) proposes a contrastive training objective that encourages the LM to learn discriminative and isotropic token representations, and is implemented on GPT-2 (Radford et al., 2019).

Concept-to-Text is a relatively more constrained conditional text generation task involving common-sense reasoning. This task provides a set of concepts as input, and requires the model to generate a piece of text that incorporates the concepts and describes an everyday scenario. We conduct experiments on the CommonGen (Lin et al., 2020) benchmark.⁵ We compare against the following models: KG-BART (Liu et al., 2021) encompasses the relations of concepts with the knowledge graph and augments the BART (Lewis et al., 2020) encoder and decoder with graph representations. ModelAdapt (Ma et al., 2021) is built upon BART and removes the positional embedding in the encoder. Imagine-and-Verbalize (I&V) (Wang et al., 2022a) predicts a scene graph for each set of concepts, and uses it as an additional input to the LM. In contrast to I&V, we directly visualize the concepts and use the machine-generated images as the auxiliary information to assist the concept-to-text generation.

4.2 Evaluation

Automatic For sentence completion and story generation, we follow previous work and eval-

⁵We use the in-house split provided by Wang et al. (2022a), which contains 65323/2066/4018 samples for train/validation/test.

uate the quality of the generated text from the aspect of model degeneration level (rep- n , diversity, distinct- n), text distribution divergence (MAUVE), and semantic similarity (BERTScore): (1) rep- $n = 1.0 - \frac{|\text{unique } n\text{-grams}|}{|\text{total } n\text{-grams}|}$ measures sequence level repetition by computing the portion of duplicate n -grams (Welleck et al., 2020). (2) diversity = $\prod_{n=2}^4 (1 - \text{rep-}n)$ measures the diversity of n -grams (Su et al., 2022a). (3) distinct- $n = \frac{|\text{unique } n\text{-grams}|}{|\text{length of text}|}$ measures the portion of distinct n -grams in the text (Li et al., 2016). (4) MAUVE measures the learned distributions divergence between the generated text and human-written text (Pillutla et al., 2021),⁶ a low MAUVE indicates a great difference between the distributions of generated text and human text. (5) BERTScore assesses contextual text similarity between two pieces of texts by computing the cosine similarities between their tokens’ embeddings (Zhang* et al., 2020),⁷ a low BERTScore means the generated text is contextually different from the ground-truth.

For concept-to-text, following prior work, we report the metrics scores on BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), SPICE (Anderson et al., 2016), and BERTScore (Zhang* et al., 2020).

Human We also set up a human evaluation as a complementary evaluation beyond the automatic metrics. We select 100 samples from the test set for sentence completion and story generation and perform the head-to-head comparison between the text snippets generated by our iNLG and the baseline models. We invite human annotators to compare the text quality from the following three independent aspects: (1) *Coherence*: Which snippet is more semantically consistent with the context, and follows the logic of the context more naturally. (2) *Fluency*: Which snippet is more fluent in English. (3) *Informativeness*: Which snippet contains more

⁶We report MAUVE with gpt2-large as the base model.

⁷We report BERTScore with roberta-large as base model.

Task	*	Setting	rep-2 ↓	rep-3 ↓	rep-4 ↓	diversity ↑	distinct-2 ↑	MAUVE ↑	BERTScore ↑
Sentence Completion	0	Human	0.45	0.05	0.01	99.50	77.32	-	-
	1	GPT2 <i>no finetune</i> (Radford et al., 2019)	6.71	6.87	10.13	78.07	74.83	44.19	22.57
	2	StoryEndGen (Guan et al., 2019)	39.53	35.11	39.30	34.12	44.57	0.45	-47.29
	3	GPT2 <i>text-only finetune</i>	4.20	4.03	5.53	86.85	75.14	49.45	24.13
	4	GPT2 +iNLG	2.43	2.61	3.57	91.63	75.92	60.30	24.25
Story Generation	5	Human	1.76	0.38	0.15	97.71	56.34	-	-
	6	GPT2 <i>no finetune</i>	37.65	22.76	21.92	45.67	43.42	0.43	-7.77
	7	Action-Plan (Fan et al., 2018)	52.05	35.58	28.11	26.97	21.43	0.41	-18.32
	8	Plan-and-Write (Yao et al., 2019)	45.22	32.86	23.34	30.71	20.83	0.41	-37.35
	9	SimCTG (Su et al., 2022b)	28.72	24.02	20.61	43.00	42.06	0.43	18.01
	10	GPT2 <i>text-only finetune</i>	25.41	18.51	14.41	52.10	46.60	9.10	21.23
	11	GPT2 +iNLG	10.73	5.64	3.42	81.36	51.91	35.94	23.03

Table 2: Generation quality scores for few-shot text completion on the ActivityNet and few-shot story generation on ROCStories. “Human” shows the human performance and “GPT2 *no finetune*” denotes the vanilla GPT2 model without tuning. All the other listed models are trained with 1% of the training data. “+iNLG” denotes introducing machine-generated images on top of the base LM.

interesting content, and describes the scenes that are more likely to happen in real life. Three human judges rate each comparison.

4.3 Implementation Details

We use StableDiffusion-v1-1 (Rombach et al., 2022) to render a 512x512 image from the context, and use CLIP ViT/B-32 to extract features offline. The mapping network is an 8-layer Transformer, and the visual prefix length is 20. For the sentence completion and story generation tasks, the mapping network is pre-trained on the MSCOCO (Lin et al., 2014) dataset. For the concept-to-text task, the mapping network is pre-trained on VIST (Huang et al., 2016).⁸ We pre-train the mapping network for 5 epochs with a batch size of 128. Results are reported on three repeat runs. Detailed hyperparameters are listed in the Appendix.

5 Result and Analysis

5.1 Few-Shot Learning Results

Open-ended text generation is a broad topic with flexible and inexhaustible setups, many of which have low resources. Collecting annotations is often extremely expensive and time-consuming. Therefore, we first report few-shot results to check if our iNLG can rapidly adapt to new task setups with a few examples, which is more practical in real-life.

More specifically, we report few-shot open-ended text generation results with 1% of the training data. For sentence completion and story gen-

eration tasks, the base LM is GPT2-base (Radford et al., 2019). For concept-to-text, we test it with BART-base (Lewis et al., 2020) as the base LM.

Sentence Completion As shown in Table 2, StoryEndGen (#2) suffers from degeneration with the highest rep- n and the lowest diversity. Training with only 1% of the training data improves GPT2’s performance on all metrics (#3 vs. #1). Under the same few-shot setting, adding additional machine-generated images with our iNLG (#4) further alleviate model degeneration. The improvement on MAUVE also indicates that introducing visual input can aid GPT2 in generating text that is more similar to the human-written ones.

Story Generation As shown in Table 2, for the story generation task that requires the LM to compose longer text, we see the vanilla GPT2 without tuning suffering from more severe degeneration compared to rendering a sentence ending (#6 vs. #1). The high rep- n scores indicate that the two non-Transformer-based baselines Action-Plan (#7) and Plan-and-Write (#8) stammer with repetitive tokens, which greatly differs from the human-written text (leads to low MAUVE) and does not have concrete meanings (leads to low BERTScore). The models based on GPT-2 (#9-#10) yield more complete sentences with concrete meanings (BERTScore gets higher). However, they keep repeating the same sentence, which is still quite different from human language (MAUVE remains low). Applying iNLG to GPT-2 leads to minor degeneration and has the best performance on all metrics (#11). Examples of generated text snippets can be found in Figure 6 and in Appendix.

⁸CommonGen is built upon image and video captioning datasets including MSCOCO. To avoid data leakage, we choose to pre-train the mapping network on VIST, which is not revealed to CommonGen.

Task	Models	Coherence			Fluency			Informativeness		
		Win(%)	Tie(%)	Lose(%)	Win(%)	Tie(%)	Lose(%)	Win(%)	Tie(%)	Lose(%)
Sentence Completion	Ours vs. StoryEndGen	51.67	20.33	28.00	44.67	19.33	36.00	41.33	18.33	40.33
	Ours vs. GPT2 <i>no finetune</i>	51.00	22.67	26.33	45.00	22.33	32.67	41.00	21.00	38.00
	Ours vs. GPT2 <i>text-only finetune</i>	58.00	24.33	17.67	43.33	18.67	38.00	42.33	21.67	36.00
Story Generation	Ours vs. Action-Plan	51.00	24.67	24.33	54.67	16.33	29.00	52.00	15.00	33.00
	Ours vs. Plan-and-Write	45.33	25.67	29.00	53.00	16.67	30.33	54.67	17.00	28.33
	Ours vs. SimCTG	42.00	27.67	30.33	40.33	25.67	34.00	43.33	18.33	38.33
	Ours vs. GPT2 <i>no finetune</i>	43.33	24.33	32.33	43.67	20.33	36.00	44.67	19.00	36.33
	Ours vs. GPT2 <i>text-only finetune</i>	39.33	26.67	34.00	38.67	26.67	34.67	44.33	22.67	33.00

Table 3: Human evaluation results for the sentence completion task and the story generation task. The scores indicate the percentage of win, tie or lose when comparing our iNLG with the baseline models.

* Setting	B-4	M.	CIDEr	SPICE	BertS.
1 BART-base <i>text-only finetune</i>	20.72	25.47	114.49	24.58	59.76
2 +KG (Liu et al., 2021)	15.26	24.44	98.53	23.13	52.76
3 +Adapt (Ma et al., 2021)	23.11	25.96	123.44	25.14	61.53
4 +I&V (Wang et al., 2022a)	24.50	25.89	119.61	25.59	57.29
5 +iNLG	25.07	26.48	127.93	26.32	63.37

Table 4: Automatic metrics scores for few-shot concept-to-text generation on CommonGen with 1% of the training data. All listed models are implemented on BART-base. “+KG” adds knowledge graph, “+Adapt” applies model adaption, “+I&V” adds scene graph, and “+iNLG” introduces machine-generated images as input. B-4: BLEU-4; M.: METEOR; BertS.: BERTScore.

Concept-to-Text Table 4 shows that knowledge graph information may not be fully exploited under the few-shot setting (#2), while removing the information of relative positions between input concepts helps the LM write better sentences (#3). Introducing machine-generated images can improve the base LM’s performance on concept-to-text generation (#5 vs. #1). While both I&V and our iNLG involve machine “imagination”, we provide such information in different forms (scene graphs vs. images). Comparing #4 and #5, our iNLG outperforms I&V with BART-base as the base LM. This suggests that the additional information introduced by I&V and iNLG is complementary.

Human Evaluation Table 3 lists out human evaluation results on text completion and story generation. Our iNLG outperforms the compared baselines on all three criteria in the model-level head-to-head comparisons. This further verifies the effectiveness of our iNLG in generating fluent and informative text snippets that better align with the given context.

5.2 Model-Agnostic Improvement

We further report open-ended text generation results with various base LM when trained with the full set of data. For concept-to-text, we experiment

Base LM	Setting	Metrics				
<i>Concept-to-Text</i>		B-4↑	MET↑	CIDEr↑	SPICE↑	BertS.↑
BART-base	text-only	30.32	31.35	158.92	31.22	68.50
	+iNLG	30.60	31.44	160.63	31.42	69.02
BART-large	text-only	32.38	33.06	169.69	33.01	70.33
	+iNLG	32.76	33.17	171.47	33.35	70.79
T5-base	text-only	30.39	30.87	163.67	32.77	70.03
	+iNLG	31.09	31.18	165.52	32.81	70.35
T5-large	text-only	34.13	32.91	175.67	34.30	72.44
	+iNLG	34.50	33.87	177.65	35.48	72.70
<i>Sentence Completion</i>		rep-4↓	div.↑	dist-2↑	MAUVE↑	BertS.↑
GPT2-base	text-only	4.20	87.46	72.87	61.42	29.84
	+iNLG	3.95	89.33	74.09	64.01	30.10
GPT2-large	text-only	1.77	96.54	76.74	87.81	31.66
	+iNLG	2.05	95.90	76.80	89.11	32.15
<i>Story Generation</i>		rep-4↓	div.↑	dist-2↑	MAUVE↑	BertS.↑
GPT2-base	text-only	7.83	68.42	49.53	33.13	28.81
	+iNLG	6.80	71.17	49.92	38.86	29.13
GPT2-large	text-only	1.02	91.91	54.17	82.81	31.86
	+iNLG	0.85	92.51	54.54	87.83	32.03

Table 5: Automatic metric scores when trained with the full set of data with ablations of the base LM. Introducing our iNLG leads to model-agnostic improvements across the board. B-4: BLEU-4; MET.: METEOR; BertS.: BERTScore; div.: diversity; dist-2: distinct-2.

with BART-base/large (Lewis et al., 2020) and T5-base/large (Raffel et al., 2020). For sentence completion and story generation, we record results on GPT2-base/large (Radford et al., 2019). As shown in Table 5, introducing machine-generated visual supervision with our iNLG leads to model-agnostic improvements over text-only finetuning. This holds true for all the listed base LM with different architectures and verifies that our iNLG is a model-agnostic framework.

5.3 Performance Analysis

Source of Image We first perform an ablation study to understand how the source of visual information affects our iNLG framework. We compare retrieved/generated images from four sources: (1) the first returned result by Yahoo Image Search;⁹

⁹<https://images.search.yahoo.com/>

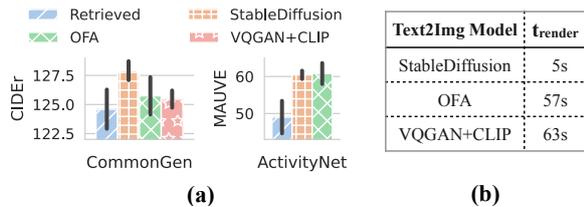


Figure 3: (a) iNLG’s performance on CommonGen and ActivityNet with visual supervisions retrieved from the web or generated by machines. Scores are reported with error bars. (b) Average time to render an image on TITAN RTX with each text-to-image generator.

(2) images rendered by VQGAN+CLIP (Crowson et al., 2022);¹⁰ (3) images rendered by OFA (Wang et al., 2022b),¹¹ and (4) images rendered by StableDiffusion (Rombach et al., 2022), with which we report the main results.

As shown in Figure 3(a), the images generated by machines act as a more effective supervision than the retrieved images. This validates our motivation of introducing machine-generated images over retrieved ones to guide LM in performing text generation. Among the three text-to-image generators, VQGAN+CLIP is slightly inferior to the other two, while StableDiffusion and OFA have mixed performance. Images generated by StableDiffusion rank first on CommonGen, while images rendered with OFA score slightly higher on ActivityNet. Figure 3(b) reports the average image rendering time, where StableDiffusion is 10× faster when rendering images than the other two.

Contrastive Training We examine the effect of the contrastive training objective on CommonGen, and the results are presented in Figure 4. We notice that introducing $\mathcal{L}_{contrastive}$ improves iNLG’s performance on 4 out of 5 listed few-shot setups, which suggests that our contrastive training objective generally can assist the LM in composing open-ended text snippets. One exception is in the extreme few-shot setting with only 0.1% of training data, where the amount of data is insufficient to let the LM form a decent representation. In this case, enforcing the sentence representation to be similar to the visual supervision with $\mathcal{L}_{contrastive}$ might misguide the LM.

Mapping Network & Visual Prefix We discuss the effects of different types of mapping networks and various visual prefix lengths. Aside from the 8-layer Transformer we used in the main experi-

¹⁰<https://github.com/nerdyrodent/VQGAN-CLIP>

¹¹<https://github.com/OFA-Sys/OFA>

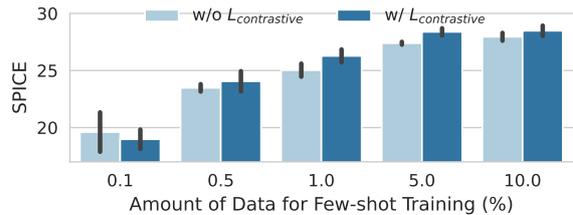


Figure 4: Performance of applying our iNLG on BART-base for few-shot concept-to-text with ablated training objective $\mathcal{L}_{contrastive}$ on various few-shot settings. Scores are reported with error bars.

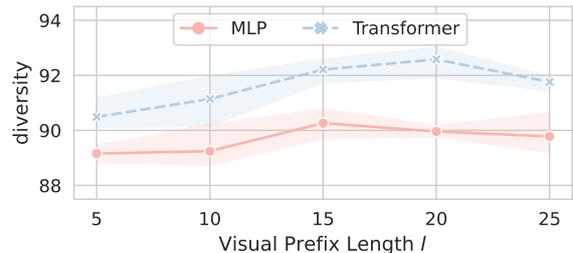


Figure 5: Performance of our iNLG on few-shot sentence completion with various visual prefix lengths and with MLP and Transformer as mapping network. Scores are reported with error bands.

ments, we also tried a simple Multi-Layer Perceptron (MLP) with two fully-connected layers. As shown in Figure 5, the Transformer-based mapping network outperforms MLP on all listed l . MLP has the best performance when visual prefix length $l = 15$, while the Transformer-based mapping network scores highest when $l = 20$.

Model Weight Tuning Table 6 compares the influence of pre-training/tuning the weights of different modules of our iNLG. Generally speaking, tuning the weights during training outperforms freezing the weights, which applies to both the base LM and the mapping network. In addition, considering our few-shot setup, pre-training the mapping network also helps our iNLG gain better performances. The best combination is applying the pre-trained mapping network, and tuning it together with the base LM on the few-shot downstream task.

Showcase Figure 6 provides two showcases on few-shot sentence completion and story generation to compare our iNLG with the GPT2-based baselines. SimCTG and GPT2 tuned with text-only corpus rendering repeated segments, either copying from the input context, or simply repeating themselves. In comparison, our iNLG has minor degeneration and writes coherent sentence endings or stories with more creative details in both tasks.

Tune LM	Pretrain Map.	Tune Map.	diversity \uparrow	MAUVE \uparrow
X	X	X	15.52	0.47
X	X	✓	78.20	33.79
X	✓	X	27.06	1.83
X	✓	✓	76.36	25.15
✓	X	X	87.45	48.06
✓	X	✓	88.68	51.81
✓	✓	X	89.05	55.61
✓	✓	✓	92.68	60.62

Table 6: Performance of our iNLG on few-shot sentence completion with ablated settings on whether to tune the LM, pretrain the mapping network (Pretrain Map.) and tune the mapping network (Tune Map.).

Context: Two girls are standing in a yard wearing cheerleading outfits. A girl...	
Generated Image:	
	GPT2 (text-only) is <u>standing in a yard</u> with a group of girls.
	GPT2 +iNLG is wearing a white bikini and blue shorts.
(a) Sentence Completion	
Context: Barry at the Mall. Barry and his friends spent a day at the mall.	
Generated Image:	
	SimCTG Barry was at the mall with his friends. They were all very excited to see him at the mall. <u>They all had a great time at the mall. They all had a great time at work. They all had a great time at school</u>
	GPT2 (text-only) <u>They went to the local mall and bought a new car. They drove to the mall and bought a new car. They drove to the mall and bought a new car.</u> Barry's friends were very happy they <u>bought a new car.</u>
	GPT2 +iNLG They were swamped by the crowds. Barry was scared to go home. They tried to avoid the mall, but they were stopped by the people at the mall.
(b) Story Generation	

Figure 6: Sentence ending and stories generated by GPT2-based methods tuned with 1% of the training data. *Repetitive contents* are underlined. The sentence ending and story written by our iNLG is coherent with the context, related to the machine-generated image, and has minor degeneration. More demonstrative examples can be found in the Appendix.

6 Conclusion

In this work, we propose iNLG, a framework that introduces machine-generated images to guide open-ended text generation. This endows the machines with the ability of creative visualization that human writers often demonstrate. We distill the vision information from the pre-trained multimodal models and further construct visual prefixes to guide language models to perform text generation with the teacher forcing and the contrastive objective. Extensive experiments show the effectiveness of iNLG in open-ended text generation tasks, including text completion, story generation, and concept-to-text generation in few-shot settings.

Limitations

This work mainly focuses on open-ended text generation, where the search space for the target output is infinite, and the language model would benefit from additional visual imagination distilled from large text-to-image generation models to produce coherent and meaningful content. However, we should note here that despite the commendable performance of text-to-image generation models, there are certain terms and concepts that are inherently challenging to visualize, such as numerical values and abstract philosophical terms. This problem itself is an interesting open research question for all tasks involving text-and-vision.

In our current approach, the images are generated offline. In future work, one may explore the integration of text-to-image and image-to-text modules in an end-to-end manner, which may be more suitable for longer text generation that is not covered in this work.

Text-to-image generation models currently have a length limit on the input text prompt, which may impede their ability to visualize long text inputs in a single image. Furthermore, as previously discussed, text-to-image models may also encounter difficulties in generating images of complex scenes or situations that are challenging to depict through a single image. Future research could explore the use of multiple images or supplementary videos as visual input in order to provide a more comprehensive representation of the scene or situation in question. The iNLG framework can be easily extended to take video representation by taking longer visual prefixes or iteratively applying visual prefixes at each step.

Ethics Statement

In this work, we use pre-trained multimodal models to visualize machine imagination. The machine-generated images may contain uncontrolled bias if any inductive bias exists from the pre-training data. Even though we do not witness such an issue in our study, this may be a potential factor that affects the quality of the generated text. We do not anticipate any major ethical concerns given that all the datasets and models used in this study have already been released to the public. We reproduce baselines with the released code repository. For human evaluation, our study is approved for IRB exempt. The estimated hourly wage paid to MTurk annotators is \$10.

Acknowledgement

The research was sponsored by the U.S. Army Research Office and was accomplished under Contract Number W911NF-19-D-0001 for the Institute for Collaborative Biotechnologies. This work was also supported by the National Science Foundation award #2048122. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

- Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. [Spice: Semantic propositional image caption evaluation](#). In *ECCV*.
- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. [Uniter: Universal image-text representation learning](#). In *ECCV*.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. [Unifying vision-and-language tasks via text generation](#). In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 1931–1942. PMLR.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. [On the properties of neural machine translation: Encoder–decoder approaches](#). In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.
- Gordon Christie, Ankit Laddha, Aishwarya Agrawal, Stanislaw Antol, Yash Goyal, Kevin Kochersberger, and Dhruv Batra. 2016. [Resolving language and vision ambiguities together: Joint segmentation & prepositional attachment resolution in captioned scenes](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1493–1503, Austin, Texas. Association for Computational Linguistics.
- Katherine Crowson, Stella Rose Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. 2022. [Vqgan-clip: Open domain image generation and editing with natural language guidance](#). In *ECCV*.
- Yann Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. [Language modeling with gated convolutional networks](#). In *ICML*.
- Desmond Elliott, Stella Frank, Khalil Sima’an, and Lucia Specia. 2016. [Multi30k: Multilingual english-german image descriptions](#). In *Proceedings of the 5th Workshop on Vision and Language, hosted by the 54th Annual Meeting of the Association for Computational Linguistics, VL@ACL 2016, August 12, Berlin, Germany*, pages 70–74. Association for Computational Linguistics (ACL). 5th Workshop on Vision and Language, VL 2016 ; Conference date: 12-08-2016 Through 12-08-2016.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#). In *ACL*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Linda B Gambrell and Ruby J Bales. 1986. Mental imagery and the comprehension-monitoring performance of fourth- and fifth-grade poor readers. *Reading Research Quarterly*, pages 454–464.
- Linda B Gambrell and Patricia S Koskinen. 2002. [Imagery: A strategy for enhancing comprehension](#). *Comprehension instruction: Research-based best practices*, pages 305–318.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. [Convolutional sequence to sequence learning](#). In *ICML*.
- Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. [Content planning for neural story generation with aristotelian rescoring](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338, Online. Association for Computational Linguistics.
- Michael Grubinger, Paul D. Clough, Henning Müller, and Thomas Deselaers. 2006. [The iapr tc-12 benchmark: A new evaluation resource for visual information systems](#).
- Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. [A knowledge-enhanced pre-training model for commonsense story generation](#). *ArXiv*, abs/2001.05139.
- Jian Guan, Yansen Wang, and Minlie Huang. 2019. [Story ending generation with incremental encoding and commonsense knowledge](#). In *AAAI*.
- Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Nieves. 2015. [Activitynet: A large-scale video benchmark for human activity understanding](#). *CVPR*, pages 961–970.

- Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. [Visual storytelling](#). In *NAACL*, pages 1233–1239, San Diego, California. Association for Computational Linguistics.
- Daphne Ippolito, David Grangier, Douglas Eck, and Chris Callison-Burch. 2020. [Toward better storylines with sentence-level language models](#). In *ACL*, pages 7472–7478, Online. Association for Computational Linguistics.
- Victoria L Joffe, Kate Cain, and Nataša Marić. 2007. [Comprehension problems in children with specific language impairment: does mental imagery training help?](#) *International Journal of Language & Communication Disorders*, 42(6):648–664.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *ACL*, pages 7871–7880, Online. Association for Computational Linguistics.
- Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020. [Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training](#). In *AAAI*, pages 11336–11344. AAAI Press.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. [A diversity-promoting objective function for neural conversation models](#). In *NAACL*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. [Visualbert: A simple and performant baseline for vision and language](#). *ArXiv*, abs/1908.03557.
- Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. [CommonGen: A constrained text generation challenge for generative commonsense reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.
- Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. [Microsoft coco: Common objects in context](#). In *ECCV*.
- Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. [Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning](#). In *AAAI*.
- Quanyu Long, Mingxuan Wang, and Lei Li. 2021. [Generative imagination elevates machine translation](#). In *NAACL*, pages 5738–5748, Online. Association for Computational Linguistics.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. [Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks](#). In *NeurIPS*.
- Yujie Lu, Wanrong Zhu, Xin Wang, Miguel Eckstein, and William Yang Wang. 2022. [Imagination-augmented natural language understanding](#). In *NAACL*, pages 4392–4402, Seattle, United States. Association for Computational Linguistics.
- Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. 2020. [Univlm: A unified video and language pre-training model for multimodal understanding and generation](#). *ArXiv*, abs/2002.06353.
- Kaixin Ma, Filip Ilievski, Jonathan Francis, Satoru Ozaki, Eric Nyberg, and Alessandro Oltramari. 2021. [Exploring strategies for generalizable commonsense reasoning with pre-trained models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5474–5483, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. [Clipcap: Clip prefix for image captioning](#). *arXiv preprint arXiv:2111.09734*.
- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. [A corpus and cloze evaluation for deeper understanding of commonsense stories](#). In *NAACL*, pages 839–849, San Diego, California. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *ACL*.
- Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. [MAUVE: measuring the gap between neural text and human text using divergence frontiers](#). In *NeurIPS*, pages 4816–4828.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *ICML*.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.

- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. [Zero-shot text-to-image generation](#). *ArXiv*, abs/2102.12092.
- Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. [High-resolution image synthesis with latent diffusion models](#). *CVPR*, pages 10674–10685.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. [U-net: Convolutional networks for biomedical image segmentation](#). In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.
- Mark Sadoski and Allan Paivio. 2000. [Imagery and text: A dual coding theory of reading and writing](#). Lawrence Erlbaum Associates Publishers.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. 2022. [Photorealistic text-to-image diffusion models with deep language understanding](#). *ArXiv*, abs/2205.11487.
- Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. [Agenda-based user simulation for bootstrapping a POMDP dialogue system](#). In *NAACL (Companion Volume, Short Papers)*, pages 149–152, Rochester, New York. Association for Computational Linguistics.
- Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. [LAION-5b: An open large-scale dataset for training next generation image-text models](#). In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D. Manning. 2019. [Do massively pretrained language models make better storytellers?](#) In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861, Hong Kong, China. Association for Computational Linguistics.
- Haoyue Shi, Jiayuan Mao, Kevin Gimpel, and Karen Livescu. 2019. [Visually grounded neural syntax acquisition](#). In *ACL*, pages 1842–1861, Florence, Italy. Association for Computational Linguistics.
- Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2018. [Cold fusion: Training seq2seq models together with language models](#). In *InterSpeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pages 387–391. ISCA.
- Yixuan Su, Tian Lan, Yahui Liu, Fangyu Liu, Dani Yogatama, Yan Wang, Lingpeng Kong, and Nigel Collier. 2022a. [Language models can see: Plugging visual controls in text generation](#). *ArXiv*, abs/2205.02655.
- Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022b. [A contrastive framework for neural text generation](#). In *NeurIPS*.
- Chen Sun, Austin Myers, Carl Vondrick, Kevin P. Murphy, and Cordelia Schmid. 2019. [Videobert: A joint model for video and language representation learning](#). *ICCV*, pages 7463–7472.
- Ben Swanson, Kory Mathewson, Ben Pietrzak, Sherol Chen, and Monica Dinalescu. 2021. [Story centaur: Large language model few shot learning as a creative writing tool](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 244–256, Online. Association for Computational Linguistics.
- Hao Tan and Mohit Bansal. 2020. [Vokenization: Improving language understanding with contextualized, visual-grounded supervision](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2066–2080, Online. Association for Computational Linguistics.
- Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. [Representation learning with contrastive predictive coding](#). *ArXiv*, abs/1807.03748.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. [Cider: Consensus-based image description evaluation](#). *CVPR*, pages 4566–4575.
- PeiFeng Wang, Jonathan Zamora, Junfeng Liu, Filip Ilievski, Muhao Chen, and Xiang Ren. 2022a. [Contextualized scene imagination for generative commonsense reasoning](#). In *ICLR*.
- Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022b. [Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework](#). *ICML*.
- Zhongyu Wei, Qianlong Liu, Baolin Peng, Huaixiao Tou, Ting Chen, Xuanjing Huang, Kam-fai Wong, and Xiangying Dai. 2018. [Task-oriented dialogue system for automatic diagnosis](#). In *ACL (Volume 2: Short Papers)*, pages 201–207, Melbourne, Australia. Association for Computational Linguistics.
- Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. [Neural text generation with unlikelihood training](#). In

ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. [Semantically conditioned LSTM-based natural language generation for spoken dialogue systems](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. [A network-based end-to-end trainable task-oriented dialogue system](#). In *ACL*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Jie Wu, Ian Harris, and Hongzhi Zhao. 2021. [Spoken language understanding for task-oriented dialogue systems with augmented memory networks](#). In *NAACL*, pages 797–806, Online. Association for Computational Linguistics.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. [Visual entailment: A novel task for fine-grained image understanding](#). *ArXiv*, abs/1901.06706.

An Yan, Zexue He, Xing Lu, Jiang Du, Eric Chang, Amilcare Gentili, Julian McAuley, and Chun-Nan Hsu. 2021. [Weakly supervised contrastive learning for chest X-ray report generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4009–4015, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengcheng Yang, Boxing Chen, Pei Zhang, and Xu Sun. 2020. [Visual agreement regularized training for multi-modal machine translation](#). In *AAAI*, volume 34, pages 9418–9425.

Lili Yao, Nanyun Peng, Ralph M. Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. [Plan-and-write: Towards better automatic storytelling](#). In *AAAI*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [HellaSwag: Can a machine really finish your sentence?](#) In *ACL*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *ICLR*.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, and Xiang Ren 0001. 2021. [Pre-training text-to-text transformers for concept-centric common sense](#). In *ICLR 2021*. OpenReview.net.

Wanrong Zhu, Xin Eric Wang, An Yan, Miguel P. Eckstein, and William Yang Wang. 2021. [Imagine: An imagination-based automatic evaluation metric for natural language generation](#). *ArXiv*, abs/2106.05970.

A Appendix

A.1 Experiment Details

Pretraining We pre-train the mapping network for GPT-2-base (Radford et al., 2019) on the MSCOCO (Lin et al., 2014) dataset with 414,113 (image, text) pairs for training. We pre-train the mapping network for BART-base (Lewis et al., 2020) on VIST (Huang et al., 2016) story-in-sequence subset, with 141,593 (image, text) pairs for training after excluding the images that the users have removed. For each setting, we pre-train the mapping network for 5 epochs with a batch size of 128, learning rate of 2e-5, weight decay of 0.01, and warmup steps of 5,000.

Few-Shot Training for Downstream Tasks Table 7 lists out the hyperparameters we used during few-shot experiments on the three open-ended text generation tasks.

Hyperparameters	Concept-to-Text	Text Completion	Story Generation
Base LM	BART-base	GPT2-base	GPT2-base
Batch Size	8	8	8
Training Epoch	20	20	20
$N_{\text{no_contra}}$	4	10	15
λ	1.5	1	0.2
Learning Rate	2e-5	2e-5	2e-5
Weight Decay	0.01	0.01	0.01
Warmup Steps	400	400	400
Max Output Length	64	100	150
Num of Beam	10	10	10

Table 7: Hyperparameter settings for few-shot open-ended text generation.

Parameter Search We tried the learning rate in the following setting: {1e-5, 2e-5, 5e-5, 1e-4}, and tried the batch size in {4, 8, 16, 32}.

Parameter Size Table 8 lists out the parameter size for the network modules used in our study.

Environment & Run Time Table 9 lists out the execution time for the three open-ended text generation tasks with 1% of the training data. Experiments are conducted on NVIDIA A100.

A.2 Human Evaluation

We invite Amazon Mechanical Turk¹² annotators to judge the quality of the generated text. Figure 7 shows an example template we use for head-to-head comparison.

¹²<https://www.mturk.com/>

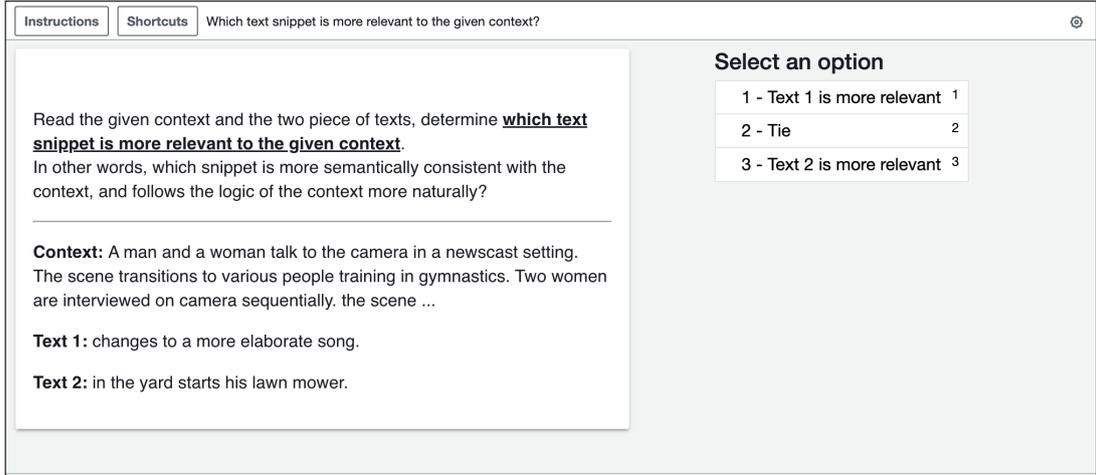


Figure 7: A screenshot of the Amazon Mechanical Turk interface for our human evaluation on text coherency.

Task	Model	Parameter Size
Sentence Completion	StoryEndGen	11M
	GPT-2 base	117M
	GPT-2 base+iNLG	160M
Story Generation	Action-Plan	43M
	Plan-and-Write	34M
	SimCTG	117M
Concept-to-Text	BART-base	110M
	KGBART	439M
	ModelAdapt	110M
	Imagine-and-Verbalize	880M
	BART-base+iNLG	153M

Table 8: Parameter sizes of the network modules used in our study.

Dataset	Text-only	+ iNLG
ActivityNet	50min	70min
ROCStories	70min	95min
CommonGen	40min	55min

Table 9: The average execution time for one single run (training + inference) on each dataset. Text generation experiments are conducted on NVIDIA A100.

A.3 More Showcases

Figure 8 compares the images retrieved from Yahoo Image Search and the images generated by StableDiffusion-v1-1 (Rombach et al., 2022), which is the text-to-image generation model we used in this work. Figure 9 and Figure 10 show more examples comparing the sentence endings and stories generated by different models.

Context 1: One of the guys hits the ball over to the other side and they hit it back. Then on the other side of the beach there is a group of women also playing volleyball. They...

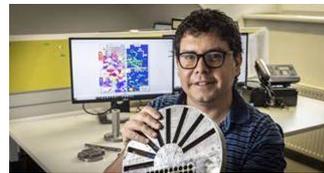


(a1) Retrieved Image



(b1) Generated Image

Context 2: A boy is talking to a camera. He goes into a bathroom and drinks a cup of mouthwash. He...



(a2) Retrieved Image



(b2) Generated Image

Figure 8: With the context as input, (a1)(a2) is the first returned image by the Yahoo image search engine,¹³ and (b1)(b2) is generated by StableDiffusion-v1-1 (Rombach et al., 2022). The two input contexts are from the ActivityNet (Heilbron et al., 2015) subset in Hel-laSwag (Zellers et al., 2019).

¹³The screenshots of the search results returned by Yahoo Image Search as of Feb.3rd 2023: [link1](#), [link2](#).

Context: A leaf blower is shown blowing a large pile of leaves across a green lawn in front of residential houses. The leaves...

Generated Image:



StoryEndGen	is her hands . the woman
GPT2 (no finetune)	are then blown by a small wind turbine.
GPT2 (text-only finetune)	are then shown in a large circle and the roof is shown in a close up.
GPT2 + iNLG	are placed on the ground and the man is shown sitting on the back.

(a)

Context: Men are standing in the edge of a trampoline preparing to make a jump into a pool in a roofed pool. People...

Generated Image:



StoryEndGen	then then the camera and then the camera and the camera and the camera
GPT2 (no finetune)	are standing in the edge of a trampoline preparing to amke a jump into a pool in a roofed pool.
GPT2 (text-only finetune)	are standing in the middle of the pool preparing to jump into a pool in a roofed pool.
GPT2 + iNLG	are swimming in the pool and throwing a ball.

(b)

Context: They mix the eggs around a bowl and place butter and milk into another bowl and mix them all together. They...

Generated Image:



StoryEndGen	the cake ups and the cake and then the cake and then the cake and the cake and then the cake and then the cake and
GPT2 (no finetune)	will be very soft and fluffy.
GPT2 (text-only finetune)	are ready to use.
GPT2 + iNLG	then put a bowl on the end of the sink and put the mixture in the sink.

(c)

Figure 9: Comparisons on few-shot sentence completion performance on ActivityNet.

Context: Sushi. Jerry loves to eat food from other nations.

Generated Image:



Action-Plan	She was to to to a . He . He was to to to the . He . She was to to to . He . She was to the the .
Plan-and-Write	the man was a lot of her . good he had to go good his new phone good friends # wanted # made # found # happy good good # home good good good # time good good good # bought # time
SimCTG	Tom was hungry. He went to the sushi place. There was a lot of sushi. Tom was hungry. He went to the sushi place
GPT2 (no finetune)	He's a big fan of the Japanese.
GPT2 (text-only finetune)	He goes to the country to buy the freshest seafood. He buys a lot of seafood. He buys more seafood and buys more seafood. Jerry buys more seafood and eats better.
GPT2 + iNLG	He was unable to find a slice he wanted. He grabbed a slice of sushi at a fast food restaurant. He was able to eat it and it was the perfect meal.

(a)

Context: Hole in the House. Tim was putting away some paperwork in the attic.

Generated Image:



Action-Plan	He was to to to the . He . He was to to the . . He was to to to the . . He was to the the .
Plan-and-Write	# i was a very hot . </s> he decided good to the store good <EOT> friends # wanted # made # found # happy <EOL> # my friend went good with good , she had good and looked in good # time # found # home
SimCTG	My neighbor's house was burglarized. I went to investigate. My neighbor's house was burglarized. My neighbor's house was burglarized. My neighbor's house was burglarized
GPT2 (no finetune)	He was trying to figure out how to get the house to be ready for the next day. He was trying to figure out how to get the house to be ready for the next day. "I'm not going to be able to do that," he said. "I'm not going to be able to do that."
GPT2 (text-only finetune)	He was trying to write a letter to his boss. He was trying to get his boss to write a letter to him. Tim was frustrated.
GPT2 + iNLG	He saw a map of the area. He went to the bathroom to check. There was nothing there. He was surprised to see it was a loophole.

(b)

Figure 10: Comparisons on few-shot story generation performance on ROCStories.

IMAGINE: An Imagination-Based Automatic Evaluation Metric for Natural Language Generation

Wanrong Zhu[¶], Xin Eric Wang[§], An Yan[†], Miguel Eckstein[¶], William Yang Wang[¶]

[¶]UC Santa Barbara, [§]UC Santa Cruz, [†]UC Santa Diego

{wanrongzhu,william}@cs.ucsb.edu, xwang366@ucsc.edu

ayan@ucsd.edu, miguel.eckstein@psych.ucsb.edu

Abstract

Automatic evaluations for natural language generation (NLG) conventionally rely on token-level or embedding-level comparisons with the text references. This is different from human language processing, for which visual imagination often improves comprehension. In this work, we propose IMAGINE, an imagination-based automatic evaluation metric for natural language generation. With the help of StableDiffusion (Rombach et al., 2022), a state-of-the-art text-to-image generator, we automatically generate an image as the embodied imagination for the text snippet and compute the imagination similarity using contextual embeddings. Experiments spanning several text generation tasks demonstrate that adding machine-generated images with our IMAGINE displays great potential in introducing multi-modal information into NLG evaluation, and improves existing automatic metrics’ correlations with human similarity judgments in both reference-based and reference-free evaluation scenarios.

1 Introduction

A major challenge for natural language generation (NLG) is to design an automatic evaluation metric that can align well with human judgments. To this end, many approaches have been investigated. Metrics that base on matching mechanisms such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), have been widely adopted in the field. Edit-distance based metrics, such as CharacTER (Wang et al., 2016), WMD (Kusner et al., 2015), SMD (Clark et al., 2019), have also been explored. Recently, BERTScore (Zhang* et al., 2020) and BLEURT (Sellam et al., 2020) attempt to leverage BERT (Devlin et al., 2019) to compare text embedding similarities, which correlates better with human judgments than previous methods. These automatic evaluation metrics make use of textual information from various angles extensively.

But what happens in our minds when we read, comprehend, and evaluate text? Research (Just et al., 2004; Eviatar and Just, 2006) has found that, unlike commonly designed automatic evaluation methods that compare the generated candidates with the references on the text domain only, humans, in contrast, leverage visual imagination and trigger neural activation in vision-related brain areas when reading text. Cognitive studies show that visual imagery improves comprehension during language processing (Gambrell and Bales, 1986; Joffe et al., 2007; Sadoski and Paivio, 2013). Inspired by this imagination-based multi-modal mechanism in human text comprehension, we ask a critical research question: *can machines create a visual picture of any underlying sentence, and use their imaginations to improve natural language understanding?* The advances of recent pre-trained vision-language models such as CLIP (Radford et al., 2021) provide an excellent opportunity for us to utilize the learned image-text representations. This enables us to explore the possibility of incorporating multi-modal information into NLG evaluation.

In this work, we propose IMAGINE, an imagination-based automatic evaluation metric for text generation. Specifically, we first use the state-of-the-art text-to-image generator StableDiffusion (Rombach et al., 2022) to visualize machine imagination from sentences, which is to generate descriptive images for the candidate text and the references. Then we receive the IMAGINE scores by computing two sets of similarity scores with the pre-trained CLIP model (Radford et al., 2021): the visual similarity of the generated images, and the cross-modal similarity between the text and the generated image. Figure 1 shows an example.

To understand the role the machine-generated images play in NLG evaluation, we conduct a series of experiments with IMAGINE on multiple NLG tasks and datasets, including machine translation, text summarization, and sentence completion for

Text for Summarization:

Kevin Garnett scored ## points in his return after a one-game suspension and the Boston Celtics ripped Detroit ##-## here Thursday in a rematch of last season's NBA semi-finals.

Reference:

Basketball: Garnett makes triumphant return as Celtics top Pistons

Hypothesis:

Celtics sink Detroit ##-## in NBA semi-final rematch

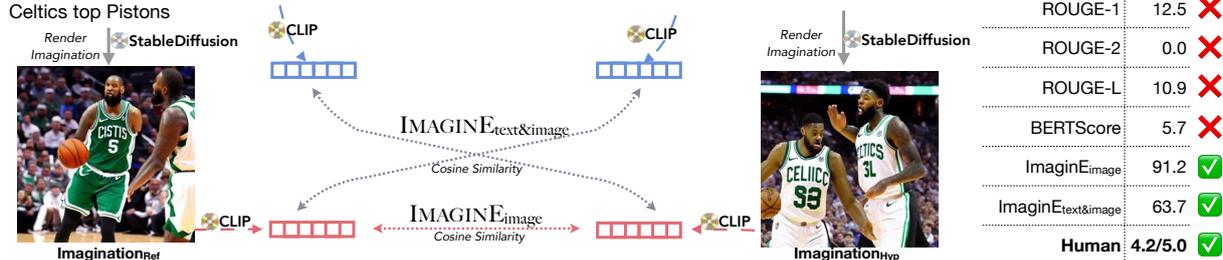


Figure 1: An evaluation example on GigaWord for text summarization. IMAGINE visualizes machine imagination with StableDiffusion (Rombach et al., 2022) and extracts textual and visual representations with CLIP (Radford et al., 2021). While traditional evaluation metrics for natural language generation rely on n -grams matching or textual embeddings comparison, IMAGINE incorporates machine-generated images into the evaluation process and enhances the understanding of the text snippet as a whole through the integration of multi-modal information.

open-ended text generation, aiming to answer the following questions:

1. *How influential is IMAGINE in NLG evaluation in terms of correlations with human judgments? Can it provide additional reference information on top of existing metrics?*
2. *What are the applicable scenarios of introducing IMAGINE to NLG evaluation? When and why do machine-generated images help?*
3. *What are the potentials and limitations of introducing machine-generated images with IMAGINE to NLG evaluation?*

Experimental results show that IMAGINE can serve as a complementary evaluation metric to text-based ones, and adding IMAGINE scores to existing metrics surprisingly improves most of the popular metrics' correlations with human performance on various text generation tasks. This holds for both reference-based evaluation and reference-free evaluation. We further conduct comprehensive quantitative analyses with case studies to verify its effectiveness. Overall, IMAGINE displays great potential in introducing multi-modal information into NLG evaluation.

2 Related Work

Automatic Metrics for Natural Language Generation Common practices for NLG evaluation compare the generated hypothesis text with the annotated references. Metric performance is conventionally evaluated by its correlation with human judgments. Existing automatic evaluation metric calculations are mainly based on three mechanisms: n -grams overlap, edit distance, and em-

bedding matching. BLEU (Papineni et al., 2002), ROUGE- n (Lin, 2004), METEOR (Banerjee and Lavie, 2005) and CIDEr (Vedantam et al., 2015) are a few widely used n -gram based metrics for text generation tasks. Another direction is based on edit distance (Tomás et al., 2003; Snover et al., 2006; Panja and Naskar, 2018; Tillmann et al., 1997; Wang et al., 2016), where they calculate the edit distance between the two text snippets with different optimizations. Embedding-based metrics (Kusner et al., 2015; Rubner et al., 1998; Clark et al., 2019; Lo, 2017, 2019) evaluate text quality using word and sentence embeddings, and more recently, with the help of BERT (Zhang* et al., 2020; Sellam et al., 2020).

Multi-Modal Automatic Metrics Aside from previous text-only metrics, some metrics utilize pre-trained multi-modal models and introduce visual features on top of text references for NLG evaluation. TIGER (Jiang et al., 2019) computes the text-image grounding scores with pre-trained SCAN (Lee et al., 2018). ViLBERTScore-F (Lee et al., 2020) relies on pre-trained ViLBERT (Lu et al., 2019) to extract image-conditioned embeddings for the text. The CLIPScore (Hessel et al., 2021) proposes a metric for image captioning by directly comparing images with captions using CLIP (Radford et al., 2021). Our method differs in that we use visual picture generation as embodied imagination and apply our metric to various text-to-text generation tasks.

Mental Imagery The debate between pictorialists and propositionalists about how imagery infor-

mation is stored in the human brain is still an open question in the neuroscience and psychology community (Troscianko, 2013). We follow the views from pictorialists that information can be stored in a depictive and pictorial format in addition to language-like forms (Kosslyn et al., 2001; Pearson and Kosslyn, 2015). In pictorialists’ model, mental imagery is constructed in the “visual buffer” either from the retinal image in seeing or from a long-term memory store of “deep representations” in the brain. Our image generation method is to mimic the generation of deep representations in machines, with the help of recent powerful text-to-image models. Inspired by empirical studies from cognitive science that visual imagination improves human text comprehension (Gambrell and Bales, 1986; Sadoski and Paivio, 1994; Nippold and Duthie, 2003; Just et al., 2004; Joffe et al., 2007; Sadoski and Paivio, 2013), we are interested in exploring if one can draw similar conclusions from automatic text evaluations by machines.

3 IMAGINE

This section describes how our IMAGINE metric evaluates the similarity between two pieces of text with the help of machine imagination. Figure 2 provides an overview of our method.

3.1 Model Details

CLIP We use the cross-modal retrieval model, CLIP (Radford et al., 2021), for our evaluation purposes. CLIP jointly trains an image encoder and a text encoder to predict the correct pairing of image-text pairs with InfoNCE (van den Oord et al., 2018) on 400M image-text pairs gathered from the web. We utilize the CLIP-ViT-B/32 variant, which consists of a 12-layer, 8-head Transformer text encoder with a hidden size of 512, and a Vision Transformer (ViT) (Dosovitskiy et al., 2021; Vaswani et al., 2017) image encoder adopting the BERT-base configuration and using a 32×32 input patch size. Both the text and image representations are normalized and projected into the multi-modal space before computing pairing likelihood through cosine similarity.

StableDiffusion We perform text-to-image generation with StableDiffusion (Rombach et al., 2022), which is a denoising diffusion probabilistic model (Ho et al., 2020). The model comprises three key components: a text encoder, a diffusion model, and an autoencoder. The text encoder,

adopted from the frozen CLIP-ViT-L/14 (Radford et al., 2021), is utilized to encode the input text into textual embeddings. The diffusion model, which leverages UNet (Ronneberger et al., 2015) for noise estimation, is modified to attend to the input textual embeddings. We conduct experiments with StableDiffusion-v1-1, which was trained with LAION (Schuhmann et al., 2022), using 256×256 images for pre-training, followed by 512×512 images for fine-tuning.

3.2 IMAGINE Similarity Score

In our proposed approach, as depicted in Figure 2, the computation of IMAGINE consists of three sequential steps. Firstly, the StableDiffusion model (Rombach et al., 2022) is utilized to generate descriptive images, referred to as machine imagination, from the two text snippets being compared. Secondly, both the text snippets and the generated images are encoded using the CLIP model (Radford et al., 2021). Finally, IMAGINE is calculated by computing the cosine similarities of the resulting text and visual features, both in a mono-modal and cross-modal manner.

Step 1: Render Imagination For each image, StableDiffusion randomly initializes a latent matrix H from the standard normal distribution and uses the encoder of the pre-trained autoencoder to encode H into the lower-resolution latent map z_T (T is the total inference steps). At each step t , the diffusion model estimates the noise, ϵ , and subtracts it from z_t . The decoder of the pretrained autoencoder takes the final noise-free latent map z and generates the image prediction I of size 512×512 .

Step 2: Extract Feature In the previous step, we generate the corresponding images I_1 and I_2 for the pair of text x_1 and x_2 for comparison with the text-to-image synthesis backbone. Then we pass the machine-generated images I_1 and I_2 and the input text x_1 and x_2 through corresponding CLIP encoders to receive the visual representations v_1 , v_2 , and the textual representation t_1 , t_2 .

Step 3: Measure Similarity With $\text{sim}(\cdot, \cdot)$ denoting the process of first normalizing the two vectors, then computing their cosine similarity, we compute two types of similarity scores for IMAGINE with the extracted textual and visual features:

(1) IMAGINE_{image} computes the visual representation similarity between v_1 and v_2 :

$$\text{IMAGINE}_{image} = \mathcal{F}(\text{sim}(v_1, v_2)) \quad (1)$$



Figure 2: Illustration of the computation process of the IMAGINE metric. Given the two pieces of text for comparison, x_1 and x_2 , we render the machine imagination by generating two images I_1 and I_2 with the pre-trained StableDiffusion (Rombach et al., 2022). We extract features of the input text and corresponding generated images with CLIP (Radford et al., 2021). We receive two variants of IMAGINE by computing the cosine similarity of the extracted features, in which $IMAGINE_{image}$ measures mono-modal similarities on the visual side, while $IMAGINE_{text\&image}$ conducts cross-modal matching.

(2) $IMAGINE_{text\&image}$ ($IMAGINE_{t\&i}$) takes both the text and the generated image into consideration, and conducts cross-modal comparisons between (t_1, v_2) , as well as (t_2, v_1) :

$$IMAGINE_{t\&i} = \mathcal{F} \left(\frac{\text{sim}(t_1, v_2) + \text{sim}(t_2, v_1)}{2} \right) \quad (2)$$

The cosine similarity between the text and image representations theoretically has a range of $[-1, 1]$. However, in practice, the IMAGINE similarity scores tend to cluster within a more narrow interval $[l, h]$. Following Hessel et al. (2021), we use a linear function \mathcal{F} to stretch the similarity score distribution to the range of $[0, 1]$, which is also the score range for most of the automatic metrics covered in this study. Eq. (3) shows how we re-scale the similarity score s into s' . Appendix Figure 6 plots the two IMAGINE variants' distributions before and after rescaling.

$$s' = \frac{s - l}{h - l},$$

$$[l, h] = \begin{cases} [0.1, 1.0], & \text{for } IMAGINE_{image}, \\ [0.1, 0.4], & \text{for } IMAGINE_{text\&image}. \end{cases} \quad (3)$$

3.3 Integration with Existing Metrics

The IMAGINE similarity scores can serve as standalone automatic metrics. Additionally, IMAGINE can be incorporated as an extension to existing metrics, as it offers multimodal references and addresses the limitations of current text-only evaluations that only compare tokens or text embeddings. This mimics the human process of comprehending

text, where both text and visual imagination are utilized. The integration of IMAGINE with other automatic metrics is straightforward, achieved by summing the IMAGINE similarity score with the other automatic metric's score for each example:

$$metric_score' += IMAGINE_{similarity_score} \quad (4)$$

4 Experimental Setup

4.1 Tasks, Datasets, and Models

We evaluate our approach on three popular natural language generation tasks: machine translation, abstractive text summarization, and open-ended text generation.

Machine Translation We use Fairseq (Ott et al., 2019) to generate English translation from German on IWSLT'14 (Cettolo et al., 2014) and WMT'19 (Barrault et al., 2019) datasets.

Abstractive Text Summarization We use the implementation of Li et al. (2017) to generate summarization on DUC2004¹ and use ProphetNet (Qi et al., 2020b) for generation on Gigaword.² Both datasets are built upon news articles.

Open-ended Text Generation We perform experiments on the ActivityNet (Heilbron et al., 2015) subset of HellaSwag (Zellers et al., 2019), which is a benchmark for commonsense natural language inference that ask the model to predict the most likely follow-up among several choices given a specific

¹<https://duc.nist.gov/duc2004/>

²<https://catalog.ldc.upenn.edu/LDC2011T07>

Metric	IWSLT'14			WMT'19		
	Original	+IE _{image}	+IE _{text&image}	Original	+IE _{image}	+IE _{text&image}
BLEU-1	21.47	21.38±1.53	21.86 ±0.82	13.74	14.71±1.19	16.40 ±0.73
BLEU-2	20.82	21.17±1.45	21.53 ±0.68	12.50	12.93±1.13	15.11 ±0.64
BLEU-3	19.17	19.88±1.39	20.31 ±0.62	11.31	12.07±1.09	13.90 ±0.58
BLEU-4	17.60	18.57±1.36	19.08 ±0.60	9.10	9.15±1.06	11.84 ±0.54
METEOR	20.60	21.44±1.54	21.30 ±0.99	13.47	14.77±1.33	16.80 ±0.91
ROUGE	20.55	20.69±1.54	21.26 ±0.80	11.40	11.58±1.16	14.34 ±0.68
CIDEr	21.98	22.12±0.24	22.25 ±0.07	11.82	11.86±0.18	12.05 ±0.07
BERTScore	23.95	24.02±1.41	24.09 ±0.65	17.01	17.08±1.22	18.88 ±0.78
BLEURT	22.93	22.99±0.64	23.40 ±0.41	18.81	19.36±0.82	19.59 ±0.37

Table 1: The effect of applying our IMAGINE similarities on automatic metrics for machine translation, reflected in the Pearson correlation with human judgments. The image generation process is conducted over five different random seeds for each piece of text. We report the mean and standard deviation of the repeated runs. IE: IMAGINE.

context. The dataset is derived from ActivityNet video captions and we use it for the task of sentence completion, where the model is given a context and asked to complete the sentence. The predicted sentence endings generated by StoryEndGen (Guan et al., 2019) and GPT-2 (Radford et al., 2019) are collected and used in the following evaluation.

4.2 Automatic Metrics

Machine Translation & Summarization In the evaluation of machine translation and text summarization tasks, it is a common practice to compare the predicted text with the reference. Adhering to previous studies, we present results using reference-based metrics. For machine translation, we present scores using BLEU- n ($n=1,2,3,4$) (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and CIDEr (Vedantam et al., 2015). Meanwhile, for text summarization, we present ROUGE- n ($n=1,2$) (Lin, 2004) precision scores. Additionally, we report the scores of ROUGE-L (Lin, 2004), BERTScore (Zhang* et al., 2020), and BLEURT (Sellam et al., 2020) for both tasks.

Open-ended Text Generation In the context of open-ended text generation, where the number of possible answers for a given scenario can be inexhaustible, evaluating the quality of generated text through a comparison with a fixed set of references is challenging. To address this issue, previous studies have proposed to utilize reference-free metrics to evaluate the quality of the generated text. In this work, we experiment with the following reference-free metrics which assess model degeneration: (1) $\text{div-}n = \frac{|\text{unique } n\text{-grams}|}{|\text{total } n\text{-grams}|}$ measures sequence level repetition by computing the portion of duplicate n -grams ($n=2,3,4$) (Welleck et al., 2020). (2) $\text{diversity} = \prod_{n=2}^4 \text{rep-}n$ measures the diversity of

n -grams (Su et al., 2022), and assesses the model degeneration. (3) $\text{distinct-}n = \frac{|\text{unique } n\text{-grams}|}{|\text{length of text}|}$ measures the portion of distinct n -grams (here $n=2$) in the text (Li et al., 2016). In addition, we report results on BERTScore (Zhang* et al., 2020) and BLEURT (Sellam et al., 2020) for comparison of contextual similarity.

4.3 Human Evaluation

We invite Amazon Mechanical Turk³ annotators to evaluate the quality of the generated text. Due to cost constraints, when conducting human evaluation, we randomly sample 1,000 test examples for each dataset, except for DUC2004 which has 500 examples in the test set. Each example is evaluated by three human judges using a 5-point Likert scale, which assessed the fluency, grammar correctness, and factual consistency of the generated text with the reference text. The overall human assessment score is calculated as the mean of the scores obtained from the three aspects. We compute the Pearson correlation (Freedman et al., 2007) between the human scores and the scores obtained from the automatic metrics, and the results are reported as a multiple of 100 for clarity.

5 Results and Analysis

5.1 Main Results

Machine Translation Table 1 presents the results of the system-level Pearson correlation with human judges when extending the IMAGINE similarity metric to various existing automatic natural language generation (NLG) metrics on the IWSLT'14 and WMT'19 German-to-English datasets. The results demonstrate that the addition of both IMAGINE_{image} and IMAGINE_{text&image}

³<https://www.mturk.com/>

Metric	DUC2004			GigaWord		
	Original	+IE _{image}	+IE _{text&image}	Original	+IE _{image}	+IE _{text&image}
ROUGE-1	13.66	16.77 ±1.31	13.45±0.80	12.90	17.52 ±0.73	16.78±0.66
ROUGE-2	9.74	15.71 ±1.65	11.19±1.08	7.75	14.26 ±0.83	13.33±0.77
ROUGE-L	13.14	16.35 ±1.47	13.17±0.95	14.31	17.44 ±0.77	16.78±0.70
BERTScore	19.44	20.60 ±1.29	20.26±0.78	19.59	20.47 ±0.64	20.10±0.57
BLEURT	23.59	25.20 ±0.72	24.46±0.42	20.23	21.08 ±0.39	20.74±0.35

Table 2: The effect of applying our IMAGINE similarities on automatic metrics for text summarization, reflected in the Pearson correlation with human judgments. The image generation process is conducted over five different random seeds for each piece of text. We report the mean and standard deviation of the repeated runs. IE: IMAGINE.

Metric	Reference-based			Reference-free		
	Original	+IE _{image}	+IE _{text&image}	Original	+IE _{image}	+IE _{text&image}
div-2	27.21	28.01±0.49	28.08 ±0.34	27.21	26.51±0.42	27.29 ±0.58
div-3	26.80	27.67±0.49	27.78 ±0.35	26.80	26.17±0.43	26.98 ±0.59
div-4	26.20	27.14±0.48	27.28 ±0.36	26.20	25.71±0.44	26.55 ±0.60
diversity	27.40	28.19±0.41	28.23 ±0.30	27.40	26.89±0.36	27.55 ±0.50
distinct-2	26.72	27.76±0.56	27.90 ±0.40	26.72	25.54±0.48	26.49 ±0.66
BERTScore	23.47	25.92 ±0.50	25.43±0.36	25.10	23.47±0.56	25.26 ±0.78
BLEURT	19.99	22.47 ±0.83	21.55±0.72	18.70	19.67±0.88	20.56 ±1.25

Table 3: The effect of applying our IMAGINE similarities on ActivityNet for open-ended text generation, reflected in the Pearson correlation with human judgments. In the ‘‘Reference-based’’ setting, we compare the predictions with the references, while in the ‘‘Reference-free’’ setting, we compare the predictions with the input contexts. The image generation process is conducted over five different random seeds for each piece of text. We report the mean and standard deviation of the repeated runs. IE: IMAGINE.

improves the Pearson correlation for all metrics listed. Among the two variants, the mean of $\text{IMAGINE}_{\text{text}\&\text{image}}$ consistently performs better on both datasets. It is observed that there is a more substantial variance in $\text{IMAGINE}_{\text{image}}$, which is attributed to the difference in the images generated by the StableDiffusion model (Rombach et al., 2022) due to varying random seed and initialization values. As a result, $\text{IMAGINE}_{\text{image}}$, which compares two machine-generated images, has a higher standard deviation compared to $\text{IMAGINE}_{\text{text}\&\text{image}}$.

Abstractive Text Summarization The results in Table 2 demonstrate the system-level Pearson correlation with human judges when incorporating our IMAGINE similarity into existing automatic NLG metrics on the DUC2004 and GigaWord datasets. In alignment with the observations made in the machine translation task, the addition of both $\text{IMAGINE}_{\text{image}}$ and $\text{IMAGINE}_{\text{text}\&\text{image}}$ results in an improvement in Pearson correlation across all metrics. On the two summarization datasets, we notice that the correlation after incorporating $\text{IMAGINE}_{\text{image}}$ exhibits higher mean values along with larger variances compared to the correlation with $\text{IMAGINE}_{\text{text}\&\text{image}}$.

Open-ended Text Generation For the sentence completion task, we conduct evaluations in two setups. In the reference-based evaluation, we com-

pare the predicted sentence ending with the ground-truth ending provided in the dataset. In reference-free evaluation, we compare the predicted sentence ending with the input context. This setup is designed to assess the coherence of the prediction with the input context, as it is hypothesized that a high-quality prediction for open-ended text generation should be consistent with the input context.

The results of extending our IMAGINE similarity metric to existing automatic NLG metrics for the sentence completion task on the ActivityNet dataset are shown in Table 3. In the reference-based setting, both IMAGINE variants demonstrate improvement over the listed metrics and exhibit comparable performances. In the reference-free setting, the introduction of $\text{IMAGINE}_{\text{text}\&\text{image}}$ continues to enhance the Pearson correlation, while the implementation of $\text{IMAGINE}_{\text{image}}$ results in a decrease in correlation. One possible reason for the decline in correlation when $\text{IMAGINE}_{\text{image}}$ is used in the reference-free setting of the sentence completion task on ActivityNet (which is comprised of video captions) is that, despite the requirement for the predicted continuation to be coherent with the given context, the visual representation of the context and continued text may differ greatly in this scenario (e.g., due to a plot twist in the video). Consequently, direct comparison of images through $\text{IMAGINE}_{\text{image}}$ may result in a decrease in correla-

Src.: Also entschied ich mich eines tages den filialleiter zu besuchen, und ich fragte den leiter, "funktioniert dieses modell, dass sie den menschen all diese möglichkeiten bieten wirklich?"

Ref.: So I one day decided to pay a visit to the manager, and I asked the manager, "is this model of offering people all this choice really working?"

Hyp.: So I decided to visit the fillaller one day, and I asked the ladder, "does this model work that you really offer to the people all these possibilities?"



Figure 3: A case study on IWSLT’14 German-to-English translation with images rendered by StableDiffusion-v2-1. Src.: input source text. Ref.: reference text. Hyp.: generated hypothesis text.

tion. However, the inherent coherence between the input text and the continued text may be captured through cross-modal comparison, which may explain why $\text{IMAGINE}_{\text{text}\&\text{image}}$ still improves the correlation for the listed metrics.

5.2 Performance Analysis

Why is ImaginE helpful? As shown in Tables 1 to 3, the incorporation of certain variants of IMAGINE improves the correlation between the reference-based and reference-free metrics and human scores in the majority of cases. This indicates the usefulness of extending text-only metrics with multi-modal knowledge. However, how do these machine imaginations actually help text understanding and evaluation? In this section, we further explore how and why IMAGINE works. We first provide a case study to show the uniqueness of IMAGINE over text-based metrics, then systematically analyze the effectiveness of our method from different perspectives.

Case Study Figure 3 shows an example in which IMAGINE effectively detects the dissimilarity in keywords between two text snippets. Despite the similarity in sentence structure between the reference and hypothesis, the crucial distinction lies in the inclusion of the terms “manager” and “ladder”. While traditional automatic metrics that rely on n -grams matching (BLEU, ROUGE) or textual embedding comparison (BERTScore, BLEURT) may exhibit high scores, the quality of the generated text remains questionable. In contrast, IMAGINE generates distinctive images and exhibits a relatively low cross-modal similarity score, which aligns with human perception.

Metric	Original	+IE _i (dVAE)	+IE _i (BigGAN)	+IE _i (VQ-GAN)	+IE _i (SD)
ROUGE-1	13.7	15.9 ± 0.9	15.7 ± 1.0	15.9 ± 0.8	16.8 ± 1.3
ROUGE-2	9.7	14.9 ± 1.2	14.6 ± 1.3	14.9 ± 1.0	15.7 ± 1.7
ROUGE-L	13.1	16.0 ± 1.0	15.8 ± 1.1	16.0 ± 0.9	16.4 ± 1.5

Table 4: The Pearson correlations with human judges when using $\text{IMAGINE}_{\text{image}}$ (IE_i) to augment ROUGE-1/2 and ROUGE-L on DUC2004. We compute four sets of $\text{IMAGINE}_{\text{image}}$ similarity scores (mean±std) with dVAE, BigGAN, VQGAN, and StableDiffusion (SD).

	dVAE	BigGAN	VQGAN	StableDiffusion
Entity Recall	88.8%	41.2%	87.2%	94.1%

Table 5: Entity recall rate on the visualizations for Flickr30k captions. We report results for images generated by dVAE, BigGAN, VQGAN, and StableDiffusion.

People sitting at a bench talking to each other by a body of water



Figure 4: An example caption from Flickr30k Entities, and images rendered by dVAE, BigGAN, VQGAN and StableDiffusion. The bounding boxes point to the visualizations of the entities marked in the same color.

Sensitivity to Different Image Generation Backbones

In previous sections, we utilize StableDiffusion (Rombach et al., 2022) as the image generation backbone for IMAGINE. Here, we examine the influence of the image generation backbone on the evaluation performance of IMAGINE by conducting experiments on the DUC2004 dataset for summarization and comparing StableDiffusion with three alternative models: dVAE (Ramesh et al., 2021), BigGAN (Brock et al., 2019), and VQGAN (Esser et al., 2021). The results, as shown in Table 4, indicate comparable performance of $\text{IMAGINE}_{\text{image}}$ with dVAE and VQGAN, both of which outperform BigGAN across all metrics. StableDiffusion achieves the highest mean value, but also displays the largest variance among the models. These findings highlight the significance of considering the image generation architecture when evaluating text, as it can result in varying machine-generated images and affect the final evaluation outcomes.

Reliability of Machine-Generated Images

The reliability of IMAGINE’s visualization capability is further evaluated on the Flickr30k Entities dataset (Plummer et al., 2015), which consists of annotated image captions. We randomly sample

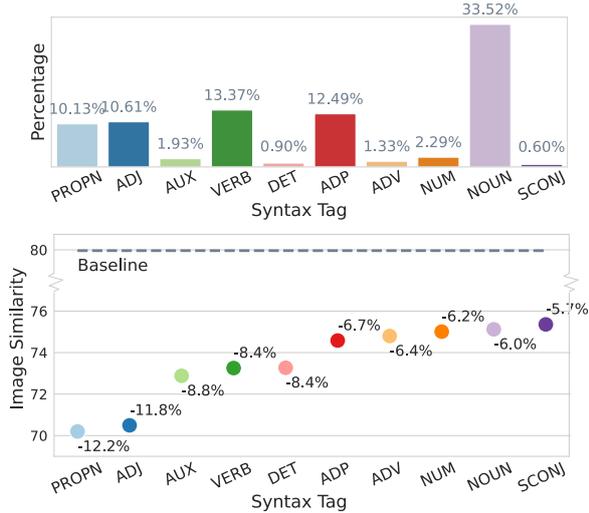


Figure 5: The influence on visualization when masking tokens of different syntax tags. Upper: The occurrence frequency of each syntax tag in DUC2004. Lower: The relative image similarity decrease after masking each syntax tag. Baseline: The average intra-group pairwise image similarity. The top-10 syntax tags that have the most significant impact on visualization are listed here.

100 captions and use the four generative backbones to render images. We present the captions and generated images to human annotators, and ask them to indicate if the entities mentioned in the captions are visually represented. The results, in terms of entity recall rates, are presented in Table 5. A higher recall rate indicates that the text-to-image generator is more capable of visualizing the content described in the text. The results show that StableDiffusion has the highest entity recall rate of approximately 94%, followed closely by dVAE and VQGAN. In contrast, BigGAN has the lowest recall rate of around 41%. An example of entity recall for a set of images generated by the four generative backbones is shown in Figure 4.

Syntax Importance to Machine-Generated Images

We evaluate the significance of different syntax tokens in the image generation process using the DUC2004 summarization dataset. We utilized the Stanza (Qi et al., 2020a) part-of-speech (POS) tagger to parse the text and created ablated examples by masking out a token of a specific syntax tag.⁴ The visual similarity of the images generated from the ablated examples is then compared to the visualization of the original text. The results, as reported in Table 5, indicated that the re-

⁴We report Universal POS tags in this study: <https://universaldependencies.org/u/pos/>

POS Tag	10 Most Frequent Tokens
NOUN	president, minister, government, space, party, station, budget, game, right, arrest
PROPN	U.S., Clinton, China, Korea, Gaza, Microsoft, Congo, Israel, Livingston, Lebanon
ADJ	new, prime, Russian, international, Asian, possible, Cambodian, first, human, economic

Table 6: The most frequent NOUN, PROPN, and ADJ tokens in DUC2004.

moval of PROPN and ADJ tags has a significant impact on the visualization results, resulting in a 12% decrease in visual similarity. Conversely, removing NOUN tokens has a comparatively smaller effect. The most frequent NOUN, PROPN, and ADJ tokens in the DUC2004 dataset were listed in Table 6. For DUC2004 built upon new clusters, PROPN and ADJ tokens cover concrete concepts such as nations, corporations, and celebrities, while NOUN tokens involve more abstract concepts such as government, party, and right. For this particular dataset, our IMAGINE approach pays more attention to PROPN and ADJ tokens that are easier to visualize by nature. Further analysis for other dataset domains can be found in the Appendix.

Which IMAGINE Variant to Report? From Tables 1 to 3, we can see a mixed trend of performance between the two IMAGINE variants. In general, $IMAGINE_{text\&image}$ has smaller variances among repeated runs. Nevertheless, we would still suggest reporting both IMAGINE variants since they conduct comparisons from different aspects, with $IMAGINE_{image}$ comparing similarity within the visual modality, while $IMAGINE_{text\&image}$ compares cross-modal similarity.

IMAGINE as a Standalone Metric Table 7 presents the Pearson correlation with human evaluations on each dataset when utilizing the two IMAGINE variants as standalone metrics. The results reveal that both IMAGINE variants demonstrate a lower correlation compared to other metrics as reported in Tables 1 to 3. Additionally, the scores produced by IMAGINE are not determinate, given the probabilistic nature of text-to-image models that generate various images with different random seeds. Hence, IMAGINE may not be an optimal choice as a standalone metric. Nonetheless, it is important to emphasize the capability of IMAGINE in introducing multimodal aspects to traditional text-only metrics. In this study, integrating IMAG-

	IWSLT'14	WMT'19	DUC2004	GigaWord	AN(w/ ref)	AN(w/o ref)
IE _i	19.1±1.5	13.8±1.7	10.6±1.5	15.9±1.1	18.9±1.5	16.8±1.9
IE _{t&zi}	18.0±1.5	12.9±1.8	9.6±1.6	15.3±1.1	18.4±1.6	18.2±1.8

Table 7: The Pearson correlation between IMAGINE variants and human assessments on each dataset. Here we use $IMAGINE_{image}$ (IE_i) and $IMAGINE_{text\&image}$ (IE_{t&zi}) as two individual metrics. AN: ActivityNet, “w/ ref”: reference-based, “w/o ref”: reference-free.

INE with text-only metrics leads to an improvement in the Pearson correlation with human evaluations. Future work may explore alternative methods of integrating multimodal information in text evaluation.

6 Conclusion

We present IMAGINE, a novel automatic evaluation metric for NLG that is based on machine imagination. Our experiments on five datasets across three different NLG tasks demonstrate the potential of incorporating IMAGINE similarity scores as a supplement to existing automatic NLG metrics, which can lead to improvement in their correlation with human evaluations in various scenarios. In the future, it is interesting to explore effective ways of visualizing abstract concepts, and how to incorporate machine imagination efficiently. We hope our work can contribute to the discussion and advancement of multi-modal studies.

Limitations

The current limitations of IMAGINE include the length constraint of the CLIP text encoder, which is limited to 77 BPE tokens (including [BOS] and [EOS]), thus limiting its applicability to longer text generation tasks such as story generation or document summarization. As a metric that relies on “machine imagination”, IMAGINE is limited by the inherent limitations of the generative models for images. The non-determined nature of machine-generated images can lead to non-determined IMAGINE scores. Possible solutions to mitigate this issue includes but are not limited to fixing the random seeds or repeating the evaluation process several times to reduce the variance effect. Additionally, it remains a challenge for machines to properly visualize certain abstract concepts or numerical values, which could limit the scope of IMAGINE’s applicability.

Ethical Statement

Our study has received IRB exempt status and the estimated hourly wage paid to MTurk annotators is \$12. It is important to note that our “imagination” approach may raise questions of fairness if the training dataset for CLIP or StableDiffusion contains any biases. This could result in a tendency for IMAGINE to generate certain types of images based on what it has seen in the training data. While we did not observe such issues in our study, it is important to consider that such unfair behavior would undermine the effectiveness of IMAGINE as an evaluation tool.

All of the datasets used in our study on machine translation, abstractive text summarization and open-ended text generation tasks are publicly available. We use the public repositories to implement IMAGINE. The implementations of image generators used in our study are DALL-E(dVAE+CLIP),⁵ Big-Sleep(BigGAN+CLIP),⁶ VQGAN+CLIP,⁷ and StableDiffusion.⁸

Acknowledgement

The research was sponsored by the U.S. Army Research Office and was accomplished under Contract Number W911NF-19-D-0001 for the Institute for Collaborative Biotechnologies. This work was also supported by the National Science Foundation award #2048122. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Loïc Barrault, Ondrej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Gra-

⁵<https://github.com/openai/DALL-E>

⁶<https://github.com/lucidrains/big-sleep>

⁷<https://github.com/nerdyrodent/VQGAN-CLIP>

⁸<https://huggingface.co/CompVis/stable-diffusion-v1-1>

- ham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 1–61. Association for Computational Linguistics.
- Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*.
- Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*.
- Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. Sentence mover’s similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Patrick Esser, Robin Rombach, and Björn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 12873–12883. Computer Vision Foundation / IEEE.
- Zohar Eviatar and Marcel Adam Just. 2006. Brain correlates of discourse processing: An fmri investigation of irony and conventional metaphor comprehension. In *Neuropsychologia*, volume 44, pages 2348–2359. Elsevier.
- David Freedman, Robert Pisani, and Roger Purves. 2007. *Statistics (international student edition)*.
- Linda B Gambrell and Ruby J Bales. 1986. Mental imagery and the comprehension-monitoring performance of fourth-and fifth-grade poor readers. In *Reading Research Quarterly*, pages 454–464. JSTOR.
- Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6473–6480. AAAI Press.
- Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Nieves. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970.
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.
- Ming Jiang, Qiuyuan Huang, Lei Zhang, Xin Wang, Pengchuan Zhang, Zhe Gan, Jana Diesner, and Jianfeng Gao. 2019. TIGER: Text-to-image grounding for image caption evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2141–2152, Hong Kong, China. Association for Computational Linguistics.
- Victoria L Joffe, Kate Cain, and Nataša Marić. 2007. Comprehension problems in children with specific language impairment: does mental imagery training help? In *International Journal of Language & Communication Disorders*, volume 42, pages 648–664. Wiley Online Library.
- M. Just, S. Newman, T. Keller, A. McEleney, and P. Carpenter. 2004. Imagery in sentence comprehension: an fmri study. In *NeuroImage*, volume 21, pages 112–124.
- Stephen M Kosslyn, Giorgio Ganis, and William L Thompson. 2001. Neural foundations of imagery. In *Nature reviews neuroscience*, volume 2, pages 635–642. Nature Publishing Group.
- Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*

- 2015, Lille, France, 6-11 July 2015, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 957–966. JMLR.org.
- Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2020. ViLBERTScore: Evaluating image caption using vision-and-language BERT. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 34–39, Online. Association for Computational Linguistics.
- Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, volume 11208 of *Lecture Notes in Computer Science*, pages 212–228. Springer.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2091–2100. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Chi-kiu Lo. 2017. MEANT 2.0: Accurate semantic MT evaluation for any output language. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 589–597. Association for Computational Linguistics.
- Chi-kiu Lo. 2019. Yisi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 507–513. Association for Computational Linguistics.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23.
- Marilyn A Nippold and Jill K Duthie. 2003. Mental imagery and idiom comprehension: a comparison of school-age children and adults. In *Journal of Speech, Language, and Hearing Research*. ASHA.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Joybrata Panja and Sudip Kumar Naskar. 2018. ITER: improving translation edit rate through optimizable edit costs. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 746–750. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.
- Joel Pearson and Stephen M Kosslyn. 2015. The heterogeneity of mental representation: Ending the imagery debate. In *Proceedings of the National Academy of Sciences*, volume 112, pages 10089–10092. National Acad Sciences.
- Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, J. Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *International Journal of Computer Vision*, volume 123, pages 74–93.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020a. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020b. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2401–2410. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language

- supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Alec Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.
- Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.
- Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. 1998. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV-98), Bombay, India, January 4-7, 1998*, pages 59–66. IEEE Computer Society.
- Mark Sadoski and Allan Paivio. 1994. A dual coding view of imagery and verbal processes in reading comprehension. In *Theoretical Models and Processes of Reading*, pages 582–601. International Reading Association.
- Mark Sadoski and Allan Paivio. 2013. *Imagery and text: A dual coding theory of reading and writing*. Routledge.
- Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Matthew G. Snover, Bonnie J. Dorr, Richard M. Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, AMTA 2006, Cambridge, Massachusetts, USA, August 8-12, 2006*, pages 223–231. Association for Machine Translation in the Americas.
- Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems*.
- Christoph Tillmann, Stephan Vogel, Hermann Ney, A. Zubiaga, and Hassan Sawaf. 1997. Accelerated DP based search for statistical translation. In *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997*. ISCA.
- Jesús Tomás, Josep Àngel Mas, and Francisco Casacuberta. 2003. A quantitative method for machine translation evaluation. In *Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing: are evaluation methods, metrics and resources reusable?*, pages 27–34.
- Emily T Troscianko. 2013. Reading imaginatively: the imagination in cognitive science and cognitive literary studies. In *Journal of Literary Semantics*, volume 42, pages 181–198. De Gruyter Mouton.
- Aäron van den Oord, Y. Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. In *ArXiv*, volume abs/1807.03748.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.
- Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. Character: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation, WMT 2016, collocated with ACL 2016, August 11-12, Berlin, Germany*, pages 505–510. The Association for Computational Linguistics.

Sean Welleck, Ilya Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

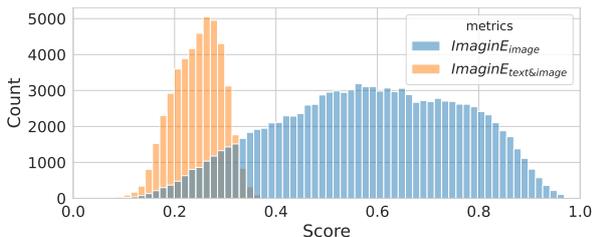
Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

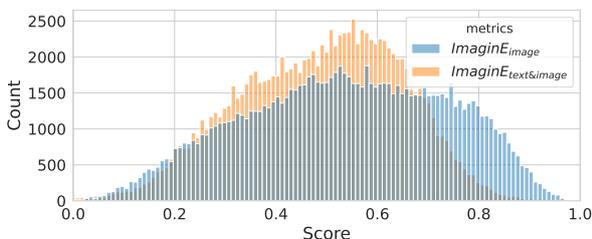
A Appendix

A.1 Score Distributions

In this study, we use cosine similarity to evaluate the similarity between features, which yields a score distribution in the range of $[-1, 1]$. However, our results indicate that negative scores were not observed when computing the similarities between the features generated by CLIP. The score distributions of the two IMAGINE variants are depicted in Figure 6. Prior to re-scaling, the scores generated by $\text{IMAGINE}_{\text{image}}$ typically fall within the range of $[0.1, 0.4]$, while those generated by $\text{IMAGINE}_{\text{text}\&\text{image}}$ are within $[0.1, 1.0]$. Following re-scaling, both IMAGINE metrics are linearly transformed to lie within the range $[0, 1]$.



(a) Before re-scaling



(b) After re-scaling

Figure 6: The score distributions of $\text{IMAGINE}_{\text{image}}$ and $\text{IMAGINE}_{\text{text}\&\text{image}}$ before and after re-scaling.

A.2 Syntax Importance to Imaginations

In Section 5.2, we discussed the impact of DUC2004 Part-of-Speech (POS) tags on the quality of generated images. In this section, we extend our examination to another dataset domain, the Flickr30k Entities dataset (Plummer et al., 2015), which is an image captioning corpus. While the domain of the Flickr30k Entities dataset is distinct from that of the DUC2004 (based on news articles), similar trends are observed. The results displayed in Figure 7 also suggest that concrete concepts are easier to be visualized and play a more significant role in the visualization process, similar to the results observed in Figure 5.

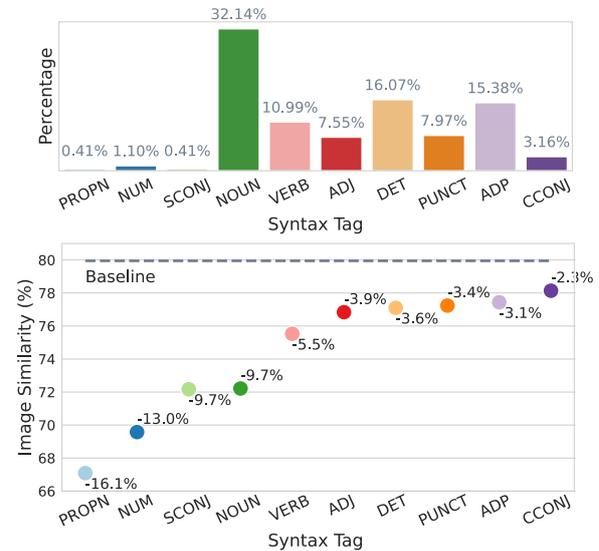


Figure 7: The influence on visualization when masking tokens of different syntax tags. Upper: The occurrence frequency of each syntax tag in Flickr30k. Lower: The relative image similarity decrease after masking each syntax tag. Baseline: The average intra-group pairwise image similarity. The top-10 syntax tags that have the most significant impact on visualization are listed here.

Entity-Aware Dual Co-Attention Network for Fake News Detection

Sin-Han Yang¹, Chung-Chi Chen², Hen-Hsen Huang³, Hsin-Hsi Chen¹

¹ Department of Computer Science and Information Engineering,
National Taiwan University, Taiwan

² AIST, Japan

³ Institute of Information Science, Academia Sinica, Taiwan
b08202029@ntu.edu.tw, c.c.chen@acm.org
hhhuang@iis.sinica.edu.tw, hhchen@ntu.edu.tw

Abstract

Fake news and misinformation spread rapidly on the Internet. How to identify it and how to interpret the identification results have become important issues. In this paper, we propose a Dual Co-Attention Network (Dual-CAN) for fake news detection, which takes news content, social media replies, and external knowledge into consideration. Our experimental results support that the proposed Dual-CAN outperforms current representative models in two benchmark datasets. We further make in-depth discussions by comparing how models work in both datasets with empirical analysis of attention weights.¹

1 Introduction

The development of the Web and social media platforms helps us obtain news quickly, but also provides a gateway for spreading false information. The impact of false information is wide, and the spread speed might be even faster than the actual one (Vosoughi et al., 2018). For example, fake news is proven empirically to influence the 2016 U.S. presidential election (Bovet and Makse, 2019; Grinberg et al., 2019; Budak, 2019). Given the impact of false information, previous studies paid a lot of effort to detect it from different aspects, including (1) news content only (Santos et al., 2020; Kim and Ko, 2021), (2) the combination of news articles and social media replies (Li et al., 2020; Lu and Li, 2020), and (3) additional publisher/user information (Long et al., 2017; Yuan et al., 2020; Del Tredici and Fernández, 2020). In this work, we focus on using both news contents and social media replies, and further add external knowledge to enhance the model’s ability to capture critical entities.

Named entities play an important role in document understanding and influence text generation

¹Code repository: <https://github.com/SinHanYang/Dual-CAN>

performances (Narayan et al., 2021, 2022). Inspired by this notion, we design a novel model, named Dual Co-Attention Network (Dual-CAN), which takes entities’ descriptions into consideration to enhance the background knowledge of the model. The proposed Dual-CAN is modified based on one of the representative fake news detection models, dDEFEND (Shu et al., 2019a). There are three major improvements in the proposed Dual-CAN: (1) Inspired by Hu et al. (2021), we add entities’ descriptions for enhancing the performance. (2) Instead of using LSTM-based architectures (Shu et al., 2019a; Lu and Li, 2020), we adopt attention architecture (Vaswani et al., 2017) as the backbone. (3) We further tailor-made a co-attention layer for comparing the given news article with entity descriptions. In sum, in addition to adopting entity descriptions from Wikipedia, we design a new architecture to fusion all information. Our main contribution is providing a novel model for fake news detection and pointing out a new direction for enhancing performance.

2 Related Works

Previous works in fake news detection mainly focused on two aspects: news content based and social context based. Rashkin et al. (2017) focus on the linguistic characteristics of the news content to detect fake news, and find that fake news often contain specific kinds of words. Ma et al. (2016) use recurrent neural networks (RNN) to learn the hidden representations from the contextual information of relevant posts over time. Monti et al. (2019) analyze social graph and user profile to predict fake news. Shu et al. (2019b) find that user profile features are useful in fake news detection. Shu et al. (2019a) and Lu and Li (2020) use co-attention model to leverage news content and social context. Their models not only have better performance but also provide interpretability to their models. Several works also use external

knowledge to improve model’s predictions. Wang et al. (2020) and Hu et al. (2021) use entity linking method to capture entity descriptions and leverage them in their models. Inspired by these works, we use external knowledge for entities to enhance performance, and use both news content and social media context in the proposed model.

3 Method

Figure 1 shows the architecture of the proposed Dual-CAN. This section describes the details of the proposed Dual-CAN model, which is composed of five components.² The first one is *news content encoder*, which employs word-level attention network and sentence-level encoder to generate features for the corresponding news contents. The second is *entity description encoder*. For each entity in news content, entity description encoder grabs its descriptions from the external knowledge base and creates features to represent them. The third is *user engagement encoder*, which employs the same method as *news content encoder* to create features to represent user comments. The fourth is *dual co-attention component*, which captures the relation between (news content, entity description) and (news content, user engagement) pairs. The last is *prediction component*, which combines all information from the previous components to make the final predictions.

3.1 News Content Encoder

A news story is composed of a sequence of sentences $\mathbf{S} = [s_1, s_2, \dots, s_N]$, and a sentence is composed of up to M words $s_i = [w_{i1}, w_{i2}, \dots, w_{iM}]$. Here, N is the maximum number of sentences in a piece of news, and M is the maximum number of words in a sentence. We perform padding to control the maximum number of sentences and words in news content. To create features to represent a news story, we use *word-level attention network* to encode each sentence, and use *sentence-level encoder* to encode all sentences in news content.

3.1.1 Word-Level Attention Network

We use Glove (Pennington et al., 2014) to create word embedding of d dimensions during the pre-processing stage for each word in sentences. For a sentence $s \in \mathbb{R}^{d \times M}$, we utilize bi-directional Gating Recurrent Units (GRU) (Chung et al., 2014) to learn the word-level representation. The output

²The hyperparameters are reported in Appendix B.

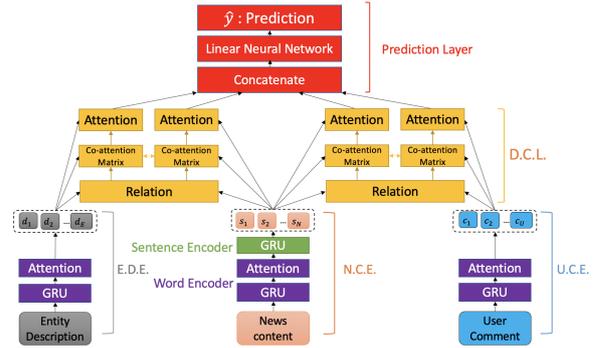


Figure 1: Architecture of Dual-CAN. D.C.L., E.D.E., N.C.E., and U.C.E. stand for dual co-attention layer, entity description encoder, news content encoder, and user comment encoder, respectively.

of the BiGRU is $\mathbf{v}_i = BiGRU(w_i) \in \mathbb{R}^{2h}, i \in \{1, 2, \dots, M\}$, where h is the dimension of the GRU. Next, we perform the basic attention mechanism to increase performance and interpretability (Lu and Li, 2020) of the word encoder. Attention weight α_i shows the importance of the i th word. The word-level attention network generates the representation of a sentence vector $\mathbf{v}' \in \mathbb{R}^{2h \times 1}$ calculated as follows:

$$\mathbf{v}' = \sum_{i=1}^M \alpha_i \mathbf{v}_i \quad (1)$$

where α_i is:

$$\mathbf{k}_i = \tanh(\mathbf{P}_n \mathbf{v}_i + \mathbf{b}_n)$$

$$\alpha_i = \frac{\exp(\mathbf{u}_n \mathbf{k}_i)}{\sum_{j=1}^M \exp(\mathbf{u}_n \mathbf{k}_j)} \quad (2)$$

$\mathbf{P}_n \in \mathbb{R}^{2h \times h}$, $\mathbf{u}_n \in \mathbb{R}^{h \times 1}$ are learnable parameter. We perform a linear layer on \mathbf{v}_i , and use a parameter \mathbf{k}_j to calculate the attention weight.

3.1.2 Sentence-Level Encoder

We use BiGRU again to encode sentences in a news story. A sentence vector $s_i \in \mathbb{R}^{2h \times 1}$ is calculated from the output of *word-level attention network*:

$$s_i = BiGRU(\mathbf{v}'_i), i \in \{1, 2, \dots, N\} \quad (3)$$

Finally, single news content is represented by a list of sentence vectors $\mathbf{S} = [s_1, s_2, \dots, s_N] \in \mathbb{R}^{2h \times N}$.

3.2 Entity Description Encoder

For each news content, we identify entities in it and grab their descriptions from Wikipedia using tools TAGME (Ferragina and Scaiella, 2010). For each entity description, we only use the first E

sentences. With the word-level attention network in Section 3.1.1, we create features that describe entity descriptions $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_E]$. Finally, entity descriptions for a piece of news is represented by a list of sentence vectors $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_E] \in \mathbb{R}^{2h \times E}$.

3.3 User Comment Encoder

For all user comments related to a news story, we only use the first U sentences. We extract features to describe user comments $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_U]$ with the word-level attention network in Section 3.1.1. Finally, user comments for a news story are represented by a list of sentence vectors $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_U] \in \mathbb{R}^{2h \times U}$.

3.4 Dual Co-Attention Component

Because we want to know whether the entity description confirms/refutes the news content and whether user comments reflect the character of the news content, we adopt co-attention network for capturing the relationship between news content and entity descriptions, and another co-attention network for linking the relationship between news content and user comments. Given news content feature vectors $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N] \in \mathbb{R}^{2h \times N}$, entity description feature vectors $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_E] \in \mathbb{R}^{2h \times E}$, and user comments feature vectors $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_U] \in \mathbb{R}^{2h \times U}$, we use *dual co-attention mechanism* for interpreting model predictions.

3.4.1 Entity Description Co-attention

First, we compute a relation matrix \mathbf{F}

$$\mathbf{F} = \tanh(\mathbf{D}\mathbf{W}_r\mathbf{S}) \in \mathbb{R}^{E \times N} \quad (4)$$

to capture the relationship between news content and entity descriptions, where $\mathbf{W}_r \in \mathbb{R}^{2h \times 2h}$ is a learnable parameter. Second, we calculate interaction maps for news content H_s and entity description H_c ,

$$\begin{aligned} \mathbf{H}_s &= \tanh(\mathbf{W}_s\mathbf{S} + \mathbf{W}_d\mathbf{D}\mathbf{F}^T) \\ \mathbf{H}_d &= \tanh(\mathbf{W}_d\mathbf{D} + \mathbf{W}_s\mathbf{S}\mathbf{F}) \end{aligned} \quad (5)$$

where $\mathbf{W}_s, \mathbf{W}_d \in \mathbb{R}^{2h \times 2h}$ are learnable parameters. Third, we calculate attention weights on each sentence in news content and entity descriptions.

$$\begin{aligned} \mathbf{a}_{s_1} &= \text{softmax}(\mathbf{w}_{hs}\mathbf{H}_s) \\ \mathbf{a}_d &= \text{softmax}(\mathbf{w}_{hd}\mathbf{H}_d) \end{aligned} \quad (6)$$

where \mathbf{w}_{hs} and $\mathbf{w}_{hd} \in \mathbb{R}^{1 \times 2h}$ are learnable parameters. After we get attention weights $\mathbf{a}_{s_1} \in \mathbb{R}^{1 \times N}$, $\mathbf{a}_d \in \mathbb{R}^{1 \times E}$, we generate new feature vectors for news contents and entity descriptions:

$$\begin{aligned} \hat{\mathbf{s}}_1 &= \mathbf{a}_{s_1}\mathbf{S}^T \\ \hat{\mathbf{d}} &= \mathbf{a}_d\mathbf{D}^T \end{aligned} \quad (7)$$

Finally, we represent news content in a feature vector $\hat{\mathbf{s}}_1 \in \mathbb{R}^{1 \times 2h}$, and entity descriptions in a feature vector $\hat{\mathbf{d}} \in \mathbb{R}^{1 \times 2h}$.

3.4.2 User Comment Co-attention

We apply co-attention model as shown in Section 3.4.1 to news content and user comments. We represent news content in a feature vector $\hat{\mathbf{s}}_2 \in \mathbb{R}^{1 \times 2h}$, and user comments in a feature vector $\hat{\mathbf{c}} \in \mathbb{R}^{1 \times 2h}$. The attention weights vector for news content and user comments are $\mathbf{a}_{s_2} \in \mathbb{R}^{1 \times N}$ and $\mathbf{a}_c \in \mathbb{R}^{1 \times U}$.

3.5 Prediction Component

Our task is a binary classification task with real/fake labels. First, we concatenate all feature vectors $\mathbf{f} = [\hat{\mathbf{s}}_1, \hat{\mathbf{d}}, \hat{\mathbf{s}}_2, \hat{\mathbf{c}}]$, and feed the result into a 2-layer linear neural network. It is calculated by:

$$\hat{y} = \mathbf{W}_2(\mathbf{W}_1\mathbf{f} + \mathbf{b}_1) + \mathbf{b}_2 \quad (8)$$

where \mathbf{W}_1 and \mathbf{W}_2 are learnable parameters and $\mathbf{b}_1, \mathbf{b}_2$ are bias terms. The prediction result $\hat{y} = [y_0, y_1]$ indicates the probabilities of label 0 is y_0 , and label 1 is y_1 . We choose cross entropy as our loss function:

$$\mathcal{L}(\theta) = -y \log(\hat{y}_1) - (1 - y) \log(1 - \hat{y}_0) \quad (9)$$

where θ is all parameters in our model. We choose Adam optimizer (Kingma and Ba, 2014) to optimize all parameters θ .

4 Experiments

4.1 Datasets

We adopt two datasets in our experiment. The first dataset is GossipCop (Shu et al., 2018), which collects both news content and social context from fact-checking website. The second dataset is CoAID (Cui and Lee, 2020), which is a benchmark dataset for COVID-19 misinformation. Please refer to Appendix A for the statistics of the datasets. We follow the evaluation settings as previous studies (Shu et al., 2018; Cui and Lee, 2020) to use (Accuracy, F1, Precision, Recall) for *GossipCop* and use (PR-AUC, F1, Precision, Recall) for *CoAID*.

Model (Input) (# of Parameters)	Accuracy	GossipCop			CoAID			
		F1	Precision	Recall	PR-AUC	F1	Precision	Recall
BiGRU (N+C+E) (28M)	0.580	0.367	0.290	0.500	0.876	0.782	0.769	0.804
BERT (N+C+E) (339M / 110M)	0.787	0.776	0.787	0.771	0.940	0.877	0.901	0.859
RoBERTa (N+C+E) (384M / 125M)	0.894	0.890	0.896	0.887	0.918	0.877	0.901	0.859
LinkBERT (N+C+E) (330M / 110M)	0.824	0.811	0.841	0.802	0.927	0.880	0.903	0.863
dEFEND (N+C) (5M)	0.771	0.758	0.771	0.754	0.749	0.799	0.792	0.808
Dual-CAN (N+E) (33M)	0.895	0.891	0.901	0.885	0.853	0.884	0.905	0.868
Dual-CAN (N+C) (33M)	0.914	0.912	0.913	0.911	0.937	0.887	0.907	0.872
Dual-CAN (N+C+E) (33M)	0.949	0.947	0.946	0.949	0.954	0.884	0.905	0.868

Table 1: Experimental results. N, C, and E denote news content, user comments, and entity description, respectively. BERT-based models are implemented in two methods (details in Appendix B) with different number of parameters.

4.2 Results

We compare the results with the following representative models: BiGRU (Chung et al., 2014), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), LinkBERT (Yasunaga et al., 2022), and dEFEND (Shu et al., 2019a).³ Table 1 shows our experimental results. Our Dual-CAN outperforms all baselines in both datasets. In addition, our Dual-CAN uses fewer parameters than BERT-based models. Our approach also performs better than dEFEND (Shu et al., 2019a) when no entity descriptions are provided. This is because we use different preprocessing methods, and the differences between two model architectures. The bottom half of Table 1 shows ablation analysis of the proposed model. The results indicate the importance of adding entity information to the proposed model, especially in *GossipCop*. However, only a few improvements in PR-AUC when experimenting with *CoAID*. *CoAID* usually are short posts that contain few entities, which results in the limitation of the proposed entity-aware concept. The main source to predict whether a piece of news is fake is the news content itself. Therefore, N+E, N+C, and N+C+E results only have small differences because they both contain N. The roles of C and E are to improve the predictions.

5 Interpretability

We examine attention weights $[a_{s_1}, a_d, a_{s_2}, a_{s_c}]$ to find those sentences that the proposed model is focusing on when making predictions. Figure 2 illustrates the results. We find that our model pays a certain degree of attention to the first sentence in the entity descriptions of both datasets (Figure 2a,2c).

³Because Shu et al. (2019a) did not release the information for dataset separation, we use the same hyperparameter reported in their work to reproduce the results. All implemental details are provide in Appendix B

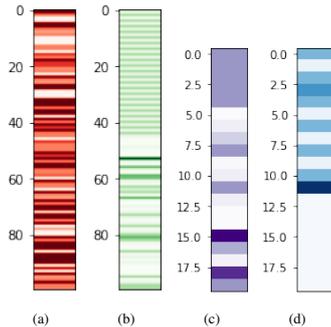


Figure 2: Attention weights of: (a) *GossipCop* entity description, (b) *GossipCop* user comments, (c) *CoAID* entity description, and (d) *CoAID* user comments. Dark colors means higher attention weights. The vertical axis means the index of the sentence.

Our intuition about this phenomenon is that the first sentence always provides a brief definition of the entity, and it would be helpful for models to understand the given entity. On the other hand, model’s attention weights on user comments of both datasets are in the middle replies, as shown in Figure 2b and Figure 2d. It follows our intuition because the sentences like “FYI. It’s a fake news.” for clarifying the given news/post is fake news always appears later than some discussions. Based on Figure 2d, we also find that models give little attention weight to the twelfth or later sentences. Besides weight distributions studies, we also did some case studies in Appendix C. The results show that attention weights do reflect the important parts of the input, which help us interpret the model better. For example, we understood the importance and usage of entity descriptions from attention weights.

6 Conclusion

We propose a dual co-attention network for fake news detection, which improves the previous representative model, dEFEND, by (1) adding entity

description as external knowledge and (2) redesigning co-attention architecture for using all input information. Our results support the usefulness of the proposed Dual-CAN model. The interpretability based on the attention weight is also discussed.

Limitations

The major limitation of the proposed model is that when the given text (news article or social media post) is short, and the performance of adding entity description may not be significantly improved. It is because such text provides few entities in the narrative, and it will limit the proposed entity-aware concept.

Ethical Statement

We will follow the licenses of GossipCop (Shu et al., 2018) and CoAID (Cui and Lee, 2020) to share the training, development, and test datasets in our experiments.

Acknowledgments

This research is supported by National Science and Technology Council, Taiwan, under grants 110-2221-E-002-128-MY3, 110-2634-F-002-050-, and 111-2634-F-002-023-.

References

- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Alexandre Bovet and Hernán A Makse. 2019. Influence of fake news in twitter during the 2016 us presidential election. *Nature communications*, 10(1):1–14.
- Ceren Budak. 2019. What happened? the spread of fake news publisher content during the 2016 us presidential election. In *The World Wide Web Conference*, pages 139–150.
- Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.
- Limeng Cui and Dongwon Lee. 2020. [CoAID: Covid-19 healthcare misinformation dataset](#).
- Marco Del Tredici and Raquel Fernández. 2020. [Words are the window to the soul: Language-based user representations for fake news detection](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5467–5479, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Paolo Ferragina and Ugo Scaiella. 2010. [TAGME: On-the-fly annotation of short text fragments \(by wikipedia entities\)](#). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, page 1625–1628, New York, NY, USA. Association for Computing Machinery.
- Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. 2019. Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378.
- Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan, and Ming Zhou. 2021. Compare to the knowledge: Graph neural fake news detection with external knowledge. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 754–763.
- Gihwan Kim and Youngjoong Ko. 2021. [Graph-based fake news detection using a summarization technique](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3276–3280, Online. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#).
- Jiawen Li, Yudianto Sujana, and Hung-Yu Kao. 2020. [Exploiting microblog conversation structures to detect rumors](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5420–5429, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. [Fake news detection through multi-perspective speaker profiles](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 252–256, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Yi-Ju Lu and Cheng-Te Li. 2020. [GCAN: Graph-aware co-attention networks for explainable fake news detection on social media](#). In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 505–514, Online. Association for Computational Linguistics.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. 2019. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

Shashi Narayan, Gonalo Simões, Yao Zhao, Joshua Maynez, Dipanjan Das, Michael Collins, and Mirella Lapata. 2022. [A well-composed text is half done! composition sampling for diverse conditional generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1319–1339, Dublin, Ireland. Association for Computational Linguistics.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. [Truth of varying shades: Analyzing language in fake news and political fact-checking](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.

Roney Santos, Gabriela Pedro, Sidney Leal, Oto Vale, Thiago Pardo, Kalina Bontcheva, and Carolina Scarton. 2020. [Measuring the impact of readability features in fake news detection](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1404–1413, Marseille, France. European Language Resources Association.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019a. [defend: Explainable fake news detection](#). In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 395–405.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2018. [FakeNewsNet: A data repository with news content, social context and spatio-temporal information for studying fake news on social media](#).

Datasets	GossipCop		CoAID	
	Glove 100d	Glove 300d	Glove 100d	Glove 300d
Word Embedding				
Max sentence length	120		120	
Max sentence number per news N	40		4	
Max sentence number per entity description	4		4	
Max sentence number per user comment	2		2	
Max sentence number of total entity description E	100		20	
Max sentence number of total user comment U	100		20	
Embedding dimension	100	300	100	300
h	100	300	100	300
Batch size	16		32	
learning rate	0.001		0.001	

Table 2: Model parameters.

Kai Shu, Xinyi Zhou, Suhang Wang, Reza Zafarani, and Huan Liu. 2019b. The role of user profiles for fake news detection. In *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, pages 436–439.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. volume 30.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Youze Wang, Shengsheng Qian, Jun Hu, Quan Fang, and Changsheng Xu. 2020. Fake news detection via knowledge-driven multimodal graph convolutional networks. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 540–547.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. [LinkBERT: Pretraining language models with document links](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. 2020. [Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5444–5454, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Dataset

Table 3 reports the statistics of the datasets. We will release the datasets for reproduction, and follow the same license of GossipCop

Datasets	GossipCop	CoAID
Total news	4,273	2,162
True news	2,562	1,590
Fake news	1,711	572
User Comments	309,059	37,187
Entity Descriptions	95,150	5,666

Table 3: Dataset statistics.

and CoAID. Due to the size limits, we cannot upload the dataset via the submission system. Please download it via the following anonymous link: <https://drive.google.com/file/d/1QuZeINFHqy80F1Av5627zTyyVVg7g2HD/view?usp=sharing>.

B Implementation Detail

Below are the implementation details of the baseline models:

- **BiGRU** (Chung et al., 2014): We use Glove 300d for word embedding of news content, entity descriptions and user comments. The word embedding of three resources are feed into BiGRU and concatenate their results $\mathbf{T} = [v_n, v_d, v_c]$. Second, we feed \mathbf{T} into linear neural network described in Section 3.5 to get final result.
- Pretrained language models (**BERT** (Devlin et al., 2019), **RoBERTa** (Liu et al., 2019), **LinkBERT** (Yasunaga et al., 2022)): We adopt three representative pretrained language models for comparison, and implemented in two different ways.

– Method 1

For each experiment, we feed news content, entity descriptions, and user comments into three tokenizers respectively. Afterwards, we feed input id and attention masks of each resource into three pretrained language models respectively. Each pretrained model handles one resource. Finally, we concatenate the outputs of three pretrained models, and pass a linear layer to output probability of two labels \hat{y} . This method is used for *GossipCop* dataset in Table 1.

– Method 2

The number of parameters in **Method 1** is huge, but it’s necessary for *GossipCop* dataset. We tried another method

to reduce the number of parameters. We concatenate all three resources and feed into one tokenizer. Second, we feed input id and attention masks into one pretrained models. The final procedure is same as the previous method. The experiment results for *CoAID* dataset are in Table 1, and the results are better than **Method 1**’s. The experiment results for *GossipCop* dataset is in Table 4. The performances are worse than **Method 1**’s. We believe it’s because *GossipCop* dataset’s data are too long for a single pretrained model. Therefore, we tried **Longformer** (Beltagy et al., 2020) which accept longer input. The performance becomes better, but this methods uses more parameters.

- **dEFEND** (Shu et al., 2019a): dEFEND is one of the representative fake news detection methods. It is based on co-attention model to increase explainability.⁴

Table 2 reports the hyperparameters used in the proposed Dual-CAN. In the ablation study, we remove the original data of entity description E or user comments C, and replace them with padding token <PAD>. Therefore, the model architecture remains the same as Section 3 stated. We have submitted the code for review, and it will be released on GitHub.

C Case Study of Interpretability

We analyzed individual sentences and words which have higher attention weight, in order to figure out the explainability of the attention weight.

For sentence-level analysis, entity descriptions that define an entity would have higher attention weights. Here are two example entity descriptions that have higher attention weights:

1. **{Dataset: GossipCop, id: 587, attention weight: 0.036 >average 0.01}**: “*IMDb (an abbreviation of Internet Movie Database) is an online database of information related to films, television series, home videos,...*”

⁴Because Shu et al. (2019a) did not release the information for dataset separation, we use the same hyperparameter reported in their work to reproduce the results. We will release the datasets for reproduction.

Model (Input) (# of Parameters)	Accuracy	GossipCop		
		F1	Precision	Recall
BERT (N+C+E) (110M)	0.643	0.587	0.645	0.599
RoBERTa (N+C+E) (125M)	0.698	0.631	0.774	0.646
LinkBERT (N+C+E) (110M)	0.702	0.694	0.694	0.693
Longformer (N+C+E) (148M)	0.752	0.742	0.758	0.723

Table 4: **Method 2** experiment results of GossipCop dataset. N, C, and E denote news content, user comments, and entity description, respectively.

2. **{Dataset: CoAID, id: 48, attention weight: 0.182 > average 0.05}**: *“The Centers for Disease Control and Prevention (CDC) is the national public health agency of the United States.”*
3. **{Dataset: CoAID, id: 1304, attention weight: 0.119 > average 0.05}**: *“Getty Images, Inc. is a British-American visual media company and is a supplier of stock images, editorial photography, video and music for business and consumers, with a library of over 477 million assets.”*

Moreover, we can see some correlation between highlighted entity descriptions and news content that contain them. For example, the news sentence which contains entity (2,3), both have higher attention weight than average.

1. **{Dataset: CoAID, id: 48, attention weight: 0.33 > average 0.25}**: *“enters for disease control and prevention, **cdc** twenty four seven, saving lives protecting people centers for disease control and prevention”*
2. **{Dataset: CoAID, id: 1304, attention weight: 0.33 > average 0.25}**: *“**getty images** the antimalarial drug hydroxychloroquine is being widely promoted as a cure for covid-19 but we still lack good data on its true benefits.”*

Case studies indicate that our model performs like it is doing “fact-checking”, which is an useful and important strategy for fake news detection. Meanwhile, entity descriptions are essential for fact-checking. Therefore, with the good usage of entity descriptions, fake news detection can achieve better performance, same as the ablation studies in Section 4.2 shown.

For word-level analysis, we discovered similar results as (Lu and Li, 2020) did. Some fake news contains emotional words or words that catch people’s attention like “Breaking” or “warn”.

CIKQA: Learning Commonsense Inference with a Unified Knowledge-in-the-loop QA Paradigm

Hongming Zhang^{1,2}, Yintong Huo^{3*}, Yanai Elazar^{4,5},
Yangqiu Song¹, Yoav Goldberg^{4,5}, Dan Roth²

¹HKUST, ²UPenn, ³CUHK, ⁴Bar Ilan University, ⁵AI2

{hzhangal, yqsong}@cse.ust.hk, ythuo@cse.cuhk.edu.hk
{yanaiela, yoav.goldberg}@gmail.com, danroth@seas.upenn.edu

Abstract

We propose a new commonsense reasoning benchmark to motivate commonsense reasoning progress from two perspectives: (1) Evaluating whether models can distinguish knowledge quality by predicting if the knowledge is enough to answer the question; (2) Evaluating whether models can develop commonsense inference capabilities that generalize across tasks. We first extract supporting knowledge for each question and ask humans to annotate whether the auto-extracted knowledge is enough to answer the question or not. After that, we convert different tasks into a unified question-answering format to evaluate the models' generalization capabilities. We name the benchmark Commonsense Inference with Knowledge-in-the-loop Question Answering (CIKQA). Experiments show that with our learning paradigm, models demonstrate encouraging generalization capabilities. At the same time, we also notice that distinguishing knowledge quality remains challenging for current commonsense reasoning models.

1 Introduction

Understanding human language requires both language knowledge (e.g., grammar and semantics) and world knowledge, which can be further divided into factual and commonsense knowledge (Katz and Fodor, 1963). Recently, the community has made great progress in helping machines acquire and apply language and factual knowledge. However, how to help machines acquire and infer over commonsense is still unclear. To answer this question, many commonsense reasoning datasets (Roemmele et al., 2011; Sakaguchi et al., 2020; Talmor et al., 2019; Zellers et al., 2019; Lin et al., 2020) have been proposed. Even though they target different knowledge types, modalities, and formats, they often

follow a standard supervised learning setting that aims at helping machines solve a specific task with training data. However, two limitations of this learning paradigm have restricted the development of commonsense reasoning systems.

First, there is no clear separation between knowledge and inference. As discussed in Elazar et al. (2021), a common phenomenon is that larger training data will lead to better performance, mainly because richer knowledge is covered. However, due to the large scale of commonsense knowledge, it is infeasible to annotate a large enough training set for each task, and the responsibility of the training data should be teaching models how to make inferences rather than acquire commonsense knowledge. Several recent works have explored using structured knowledge for commonsense reasoning tasks (Lin et al., 2019; Lv et al., 2020; Paul and Frank, 2020). However, as these works did not clearly analyze the coverage of the structured knowledge (i.e., knowledge graphs (KGs)), it is still unclear what the performance means, better knowledge coverage, or better inference capability. To investigate what is behind this learning process, we propose to equip each question with auto-extracted knowledge and ask humans to annotate whether the knowledge is sufficient to answer the question. By doing so, we could evaluate whether models can know if the provided knowledge is good or not and how well they can conduct inference over the provided knowledge to solve the task.

Second, supervised learning may force the model to learn the distribution of the training data rather than a universal inference model. As a result, the model may perform well on the test set that follows the same distribution but fail to generalize (Kejriwal and Shen, 2020). Previously, as different tasks have different formats, it is hard to evaluate the generalization ability of commonsense reasoning models. Following the trend of

* This work was done when the second author was visiting HKUST.

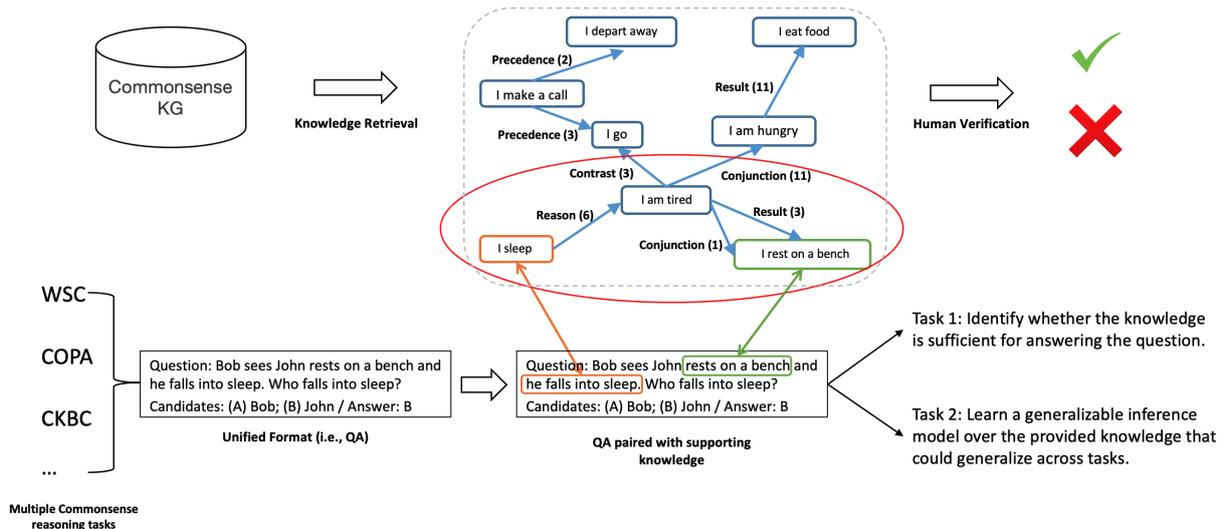


Figure 1: **CIKQA** demonstration. All tasks are converted into a unified format such that we can easily evaluate the generalization capability of all models. We also equip all questions with auto-extracted knowledge graphs from existing KGs and ask humans to annotate whether the knowledge is gold or not. In this example, we expect models to first identify the quality of the knowledge and then conduct inference over the knowledge to solve the question.

using a unified format (i.e., question answering) for different tasks (Khashabi et al., 2020), we propose to convert various commonsense reasoning tasks into a unified QA format such that we can easily and fairly evaluate the generalization ability of learned commonsense reasoning models.

Combining these two lines of effort, we propose a new commonsense inference benchmark Commonsense Inference with Knowledge-in-the-loop QA (**CIKQA**). An example is shown in Figure 1. We first convert several popular commonsense reasoning tasks into a unified QA format and equip them with the relevant knowledge from existing commonsense knowledge graphs. We leverage human annotation to label whether the provided knowledge is correct and enough¹ to answer the question. The **CIKQA** benchmark can motivate us to answer two questions: (1) Whether current models can distinguish the knowledge is gold or not; (2) Can current commonsense inference models generalize across different commonsense reasoning tasks?

Experiments with several recent knowledge-based commonsense reasoning models show that even though current deep models could learn to conduct simple inferences after training with a few examples when gold knowledge is provided, they still cannot learn to distinguish gold knowledge very well. Moreover, although current mod-

els demonstrate encouraging generalization ability across the three tasks we consider, they still struggle with complex inference (e.g., abductive reasoning). We hope that our benchmark² can motivate more advanced commonsense inference methods in the future.

2 Dataset Construction

In **CIKQA**, to encourage a generalizable commonsense inference model, we follow previous work (Khashabi et al., 2020; Cohen et al., 2020; Wu et al., 2020; Du and Cardie, 2020) to unify all selected tasks as a binary question answering problem, and equip each question with a supporting knowledge graph G retrieved from existing commonsense KGs. We leverage crowd-sourcing workers to annotate whether the knowledge is gold (i.e., accurate and enough) for answering the question. With that, we can evaluate whether models know how to distinguish gold and knowledge and whether they can learn the generalizable inference with the help of the knowledge. In total, **CIKQA** contains 15 thousand instances from four kinds of commonsense reasoning tasks. Details about task selection, format unification, knowledge extraction, and annotation are as follows.

2.1 Task Selection

In **CIKQA**, we select the following four popular commonsense reasoning tasks:

¹In the rest of the paper, we denote such knowledge as the gold knowledge.

²Available at <https://github.com/CogComp/CIKQA>.

Task Name	Original Assertion	Transformed Question	Answer
HardPCR	The fish ate the worm. It was hungry.	The fish ate the worm. It was hungry. What was hungry?	(A) Fish ; (B) Worm
CommonsenseQA	What is a place that someone can go buy a teddy bear?	What is a place where someone can go buy a teddy bear?	(A) Toy store ; (B) Shelf
COPA	I drank from the water fountain.	I drank from the water fountain. What was the cause of this?	(A) I was thirsty. ; (B) I felt nauseous.
ATOMIC	PersonX buys the bike.	Before PersonX buys the bike, what did PersonX want?	(A) To be social. ; (B) To have transportation.

Table 1: Demonstration of the original assertion, transformed questions, and answers. Correct and wrong answers are indicated in blue and red, respectively.

1. HardPCR (Zhang et al., 2021): The hard pronoun coreference resolution (HardPCR) task is one of the most famous commonsense reasoning tasks. For each question, a target pronoun and two candidate mentions are provided, and the task is to select the correct mention that the pronoun refers to. Careful expert annotations are conducted to get rid of the influence of all simple linguistic rules, and the models are required to solve the problem with commonsense reasoning. We include instances from WSC (Levesque et al., 2012), DPR (Rahman and Ng, 2012), and WinoGrande (Sakaguchi et al., 2020). To create a question regarding the target pronoun, we first find the sentence that contains the target pronoun and then determine whether the participating pronoun refers to a person or an object.
2. CommonsenseQA (Talmor et al., 2019) is a multiple-choice question answering dataset. For each question-answer pair, four relevant but wrong concepts are used as the other candidates, and the models are required to select the correct one out of five candidates. In **CIKQA**, we randomly sample a negative answer to make it a binary choice task, which is consistent with other datasets.
3. COPA (Roemmele et al., 2011) focuses on evaluating the understanding of event causality. Two follow-up events are provided for a target event, and models are asked to predict the one caused by or the reason for the target event.
4. ATOMIC (Sap et al., 2019): is a commonsense knowledge graph, which we convert into a completion problem. Given a head concept (e.g., “eat food”) and a relation (e.g., “cause”), we want to predict the tail concept, focusing on predicting the edges of ATOMIC.

In COPA and ATOMIC, where the task is to predict the relations between two events or states (e.g., “PersonX eats”-*Causes*-“PersonX is full”), for each triplet, we randomly sample another event or state as the negative tail and ask the model to select the correct one. To make the task challenging and avoid sampling irrelevant events or states, we restrict the sampled negative event or state to be connected with the head of a different triplet (e.g., “PersonX is hungry” from the triplet “PersonX eats”-*CausedBy*-“PersonX is hungry”). For each relation, we write a pattern to generate the question. For example, for the “Causes” relation, we will ask “What can be caused by the event ‘PersonX eats’?”. Examples of instances in the original datasets and their transformed questions and candidate answers are presented in Table 1.

2.2 Supporting Knowledge Extraction

As discussed in Section 1, a limitation of existing commonsense reasoning benchmarks is that there is no clear boundary between knowledge and inference. As such, it is unclear what is learned from the training data, the knowledge, how to perform inference, or a combination of both. We propose to equip each question with supporting knowledge to address this issue and encourage models to learn inference rather than knowledge from the training data. The question is selected as part of the dataset only if we find supporting knowledge to answer the question. Note that this procedure serves as an improved evaluation setup than purely supervised learning and not as a solution to commonsense reasoning. This section introduces the selected commonsense knowledge graphs and then introduces how we extract the corresponding commonsense knowledge for each question.

2.2.1 Commonsense KG Selection

Many commonsense knowledge graphs were developed to enhance machines’ commonsense reasoning abilities, including ConceptNet (Liu and Singh, 2004), ATOMIC (Sap et al., 2019), GLUCOSE (Mostafazadeh et al., 2020), and ASER (Zhang et al., 2020a). Among these four, ConceptNet, ATOMIC, and GLUCOSE were constructed via crowd-sourcing, while ASER was constructed automatically with information extraction techniques. Besides ATOMIC, which is used as one of the tasks, we use the other KBs as supporting knowledge resources.

2.2.2 Supporting Graph Extraction

Here we introduce how to extract the supporting knowledge from external commonsense knowledge bases. For each question, we need to obtain a sub-graph from supporting knowledge graphs to contain the relevant commonsense knowledge about the question. The sub-graph extraction process includes the following three steps: (1) Pre-processing: Convert each question into several key sentences; (2) Matching: Match the sentences into nodes in the KG; (3) Extraction: Retrieve the relevant sub-graphs from the entire KG. In what follows, we give some more details on each of the steps.

Data Pre-processing: For each question and the associated candidate answers, we first replace the question words (e.g., “What”) with the two candidate answers such that they become two declarative sentences. For instance, if the question is “The fish ate the worm. It was hungry. Who is hungry?” and the candidates are “Fish” and “Worm,” we will convert the question into the declarative sentence: “The fish is hungry” and “The worm is hungry.” As a result, we will get three sentences for this question: “The fish ate the worm,” “The fish is hungry,” and “The worm is hungry.”

KG Matching: After getting the declarative sentences containing the question and key answers, we map them to nodes in knowledge graphs to extract the relevant knowledge. Considering that each sentence may have multiple words and it is often hard to find an exact match, we adopt an embedding-based fuzzy matching technique. For each sentence and node in the KG, we treat them as a sentence and get the corresponding representations with SimCSE (Gao et al., 2021). For each input sentence, SimCSE encodes the sentence into a vector. A close distance between two

vectors indicates that the two sentences are similar to each other. We use cosine similarity on the obtained representations to measure the similarity between two sentences.³ Since there are 287 thousand nodes in GLUCOSE and 194 million nodes in ASER, it is computationally infeasible to compute the cosine similarity between sentences pair by pair. Thus we use an approximation. For each extracted sentence, we first apply Faiss (Johnson et al., 2017), a large-scale similarity-based matching algorithm that first clusters all KG nodes in the vector space to increase the matching efficiency when finding the top N nodes in the KG. We encode all the nodes of the graph and index them using Faiss (Johnson et al., 2017). Then, we can perform fast and quick retrieval of the most-similar nodes with each query sentence. After that, we sort the N nodes based on the cosine similarity to find the top K similar nodes. We set N and K to be 60 and 1, respectively. On average, it takes 25 seconds to retrieve the relevant nodes for each question.

Graph Extraction: Next, we extract the sub-graph that contains all the relevant nodes. We denote the extracted m nodes as n_1, n_2, \dots, n_m , and for each of them, we find K similar nodes from the KG. The resulting matched node sets are denoted as $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_m$. For any pair of nodes $n \in \mathcal{N}_i$ and $n' \in \mathcal{N}_j$ ($i \neq j$), if there exists a path in the KG between n and n' , we will keep that path. After adding all paths together, we will get the final sub-graph. On average, constructing a graph for each question takes less than two seconds.

Knowledge Quality Annotation: Since our extraction method is automatic, some of the sub-graphs may be irrelevant or insufficient for answering the questions. We use crowdsourcing to annotate whether the extracted knowledge is gold (i.e., accurate and enough), five per example. The average Inter-annotator agreement (Cohen’s kappa statistic) is 0.83, indicating our annotation’s high quality. In the end, we apply a strict standard (at least four of five annotators need to vote for gold) to select the gold knowledge.

2.3 CIKQA Statistics

We report the dataset statistics in Table 2. In total, CIKQA contains 14,599 instances, among which Hard PCR and ATOMIC provide the most

³We also tried other techniques such as string match, ROUGE (Lin, 2004), and BLEURT (Sellam et al., 2020), but found them to be either inaccurate or too slow for our scale.

Task Name	# Instance by Knowledge Resource			# Total Instance	Avg Sub-graph Size	# Gold Instance
	ASER	ConceptNet	GLUCOSE			
HardPCR	2,030	202	2,143	4,375	2.85	670
CommonsenseQA	530	31	37	598	3.19	59
COPA	103	41	149	293	3.03	78
ATOMIC	5,655	212	3,466	9,333	2.67	2,200
Total	8,318	486	5,795	14,599	2.75	3,007

Table 2: **CIKQA** statistics. “Avg Sub-graph Size” is the average graph size measured by the number of edges. “# Gold Instance” means the number of instances supported by different knowledge resources and annotated gold (i.e., Accurate and Enough) knowledge.

questions because their original datasets are much larger than others. According to the annotation, 20.59% of the instances contain gold knowledge. Based on our analysis, annotators hold a very strict standard for selecting the gold knowledge. We randomly split the dataset into training, development, and testing sets for each task with a standard 8:1:1 splitting. As a result, we get 11,678 training, 1,459 development, and 1,462 testing instances.

3 Experiment Setup

We present the performance of the following commonsense inference models on **CIKQA**:

(1) Vanilla LM: We use the language model (LM) based multiple-choice (MC) model as the basic baseline. For each candidate answer, we concatenate it with the question and feed it to the model. After getting the sentence representation, a linear layer is used to obtain a score and trained with a cross-entropy loss.

(2) KagNet: As one of the pioneering works that utilized structured knowledge for solving commonsense reasoning tasks, KagNet (Lin et al., 2019) first uses a graph convolution network to encode the knowledge graph and then apply an LSTM based hierarchical attention mechanism to encode the knowledge paths that start with the nodes corresponding to the question and end with nodes corresponding to the answer. At the same time, KagNet encodes the question and answers with pre-trained LMs. In the end, it concatenates all representations for the final prediction.

(3) Graph-Based Reasoning (GBR): Instead of only encoding paths starting with the question nodes and ending with answer nodes, in GBR (Lv et al., 2020), they propose to run a depth-first algorithm over the knowledge graph to generate a sequence of paths as the supporting knowledge paths.

(4) Multi-Head Knowledge Attention (MHKA):

To further utilize the knowledge, MHKA (Paul and Frank, 2020) uses a transformer network to model the paths from the question nodes and answer nodes, then concatenates the knowledge and context representation for the final prediction.

(5) Graph-to-Text (G2T): In the end, we also evaluate a simple yet effective approach of combining structured knowledge and language models: Graph-to-Text (Bian et al., 2021), which first verbalizes knowledge into a sentence and then concatenates the knowledge sentence and target question together. On top of that, a transformer-based model is used to encode the input sentence and make the final prediction.

Implementation Details We implement all experiments with Huggingface (Wolf et al., 2019). We select BERT-base (Devlin et al., 2019) as the base language model for all models. The batch size is set to 16. All models are trained for 10,000 steps⁴, and the best-performing checkpoints on the dev set are evaluated. For our model, we set both the number of random walk paths and the walk length to five. Considering that the auto-extracted knowledge could contain noise or miss certain knowledge, we add a “gold knowledge” setting, where only examples with the gold knowledge are used for training and testing, for all models as the upper bound of their model. All other hyper-parameters are the same as the base language model. All models are trained with GTX 2080, and the average running time is 12 hours.

4 Result Analysis

We first conduct analysis experiments to evaluate to what extent the provided knowledge could help existing models. For each model, we train it with different numbers of training instances and report the average performance and standard de-

⁴All models converge at 10,000 steps.

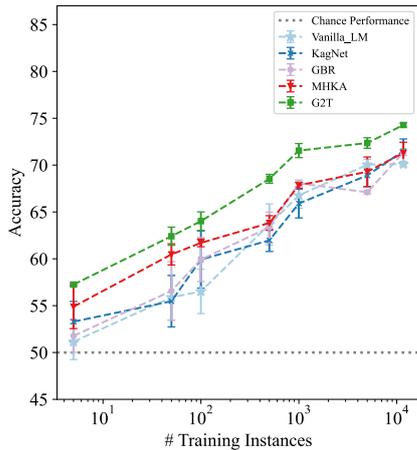


Figure 2: Learning curves of all evaluated models on all instances of **CIKQA**. We evaluate all models with the full dataset.

violation of five trials. Experiment results with all instances and the gold subset of **CIKQA**, where only instances with gold knowledge are used for training and testing, are presented in Figure 2 and 3, respectively. From the results, we can make the following observations. First, when explicitly including the knowledge, all inference models outperform the baseline model with no supporting knowledge, especially G2T. When the auto-extracted and gold knowledge is provided, G2T outperforms the baseline Vanilla LM model by 4.17 and 15.34 accuracy, respectively. It supports our assumption that learning all knowledge from the limited training data is hard, and external structured knowledge could help. At the same time, we also notice a significant gap between auto-extracted knowledge and gold knowledge. For example, if gold knowledge is available, models could learn to answer the questions with only a few examples. This indicates that the knowledge quality can significantly impact models’ performance, which further shows the importance of distinguishing whether the knowledge is gold or not automatically. Last but not least, we can see that G2T outperforms other inference models in most settings, which shows that with the help of current large-scale LMs, jointly encoding questions and knowledge is more efficient and a more effective strategy than acquiring them separately. Due to the simplicity and efficiency of G2T, we will conduct the rest analysis experiments with G2T.

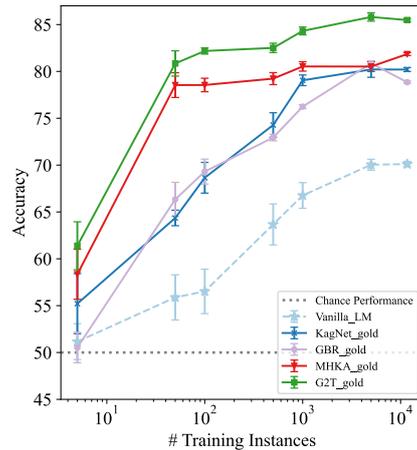


Figure 3: Learning curves of all evaluated models on the gold subset of **CIKQA**, where only instances with gold knowledge are used for training and testing.

4.1 Distinguishing the Gold Knowledge

Humans can say “I do not know” when they find out that they cannot answer a question with their knowledge. To investigate whether current deep models have a similar capability, we use G2T as an example to test whether these deep models can distinguish the gold knowledge. For each (question, answer, and knowledge) triplet, we train and test G2T with annotated knowledge quality labels. To address the imbalanced distribution problem, we randomly select the same number of “Not Gold” examples as the “Gold” ones to make the dataset balanced. From the results in Figure 4, we can see that the performance of G2T can be improved slightly with the increase of training data. However, after seeing thousands of examples, it still can only achieve 0.65 accuracy on a binary classification problem. It shows that knowing when to say “I do not know” is still a challenging task for current deep models, which is consistent with the observations in previous literature that deep models cannot understand the reasons and knowledge they used to answer questions (Zhang et al., 2020b; Sanh et al., 2022). We hope that **CIKQA** could motivate more future work on this important research problem.

4.2 Generalization Ability

An important assumption and motivation behind the unified problem design of **CIKQA** is that even though the commonsense could be enormous, the inference rules over commonsense knowledge can

Training Task	Testing Task			
	Hard PCR	CommonsenseQA	COPA	ATOMIC
Hard PCR	-	37.50 → 52.30	75.00 → 53.24	44.13 → 53.32
CommonsenseQA	50.00 → 50.14	-	62.50 → 56.67	56.34 → 70.56
COPA	45.95 → 51.26	62.50 → 58.33	-	49.77 → 62.96
ATOMIC	39.19 → 50.76	50.00 → 76.67	62.50 → 73.33	-

(a) Full Dataset (Vanilla LM (without knowledge) → G2T (with knowledge))

Training Task	Testing Task			
	Hard PCR	CommonsenseQA	COPA	ATOMIC
Hard PCR	-	46.67 → 51.67	63.33 → 56.67	51.85 → 55.78
CommonsenseQA	49.32 → 50.32	-	50.00 → 75.00	60.39 → 91.08
COPA	52.51 → 54.79	56.67 → 87.50	-	53.01 → 76.06
ATOMIC	50.46 → 51.35	68.33 → 93.75	56.67 → 87.50	-

(b) Gold Subset (Vanilla LM (without knowledge) → G2T (with knowledge))

Table 3: Generalization ability demonstration. We report the performance on both the full dataset and the gold dataset (i.e., only questions with gold knowledge are selected for training and testing) to show the generalization ability. Strong and moderate generalization settings are indicated with the **green** and **orange** background, respectively.

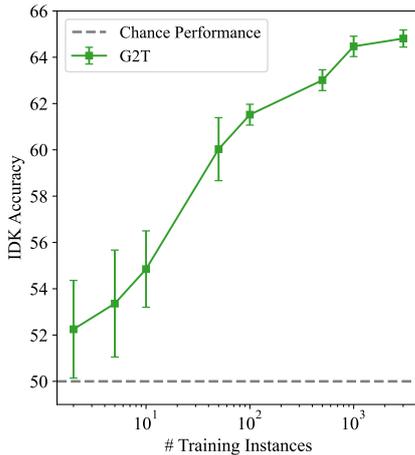


Figure 4: The learning curve of G2T on the gold knowledge identification task.

be limited. As a result, even though we could not learn all the commonsense from limited training data, we can learn how to conduct inference with several tasks and then generalize to others. In this section, we conduct experiments with both the “Without Knowledge” and “With Knowledge” models to show that we can gain such generalization ability across different tasks with our unified formulation. We conduct experiments on two settings: (1) Full Set: We train and test the model with the whole dataset; (2) Gold Subset: We only train and test the model on questions where the supporting graph is annotated as gold. We train the model with questions from a specific task and

test it on all tasks. The results are in Table 3.

From the results, we can see that the knowledge can help models to generalize well among CommonsenseQA, COPA, and ATOMIC. The only exception is HardPCR. This is mainly because the inference needed for solving HardPCR is more complex than the other tasks, where we not only need to find the relevant knowledge but also need to replace the target pronouns with the entity in the provided knowledge. As shown in Figure 5, two paths can be found relevant to question: (1) “I am drunk” → *Co_Occurrence* → “I hit someone”; (2) “I am drunk” → *Co_Occurrence* → “That is not fair” → *Co_Occurrence* → “You kick me”. For the correct inference, we need to know when there is a conflict, we should trust the one-hop inference more because the additional node in the two-hop path may introduce extra noise. As a comparison, for other tasks, the main inference we need is to find the relevant paths, which is relatively easy. How to train a model that can learn to conduct such complex reasoning is a problem worth exploring in the future.

In general, the observed generalization ability is encouraging because if we can learn a good model on **CIKQA**, based on the assumption that there are limited types of inference, we can potentially solve any commonsense reasoning task as long as the needed inference types are covered by **CIKQA**. At the same time, we also notice that models typically generate better when gold knowledge is provided, further proving the importance

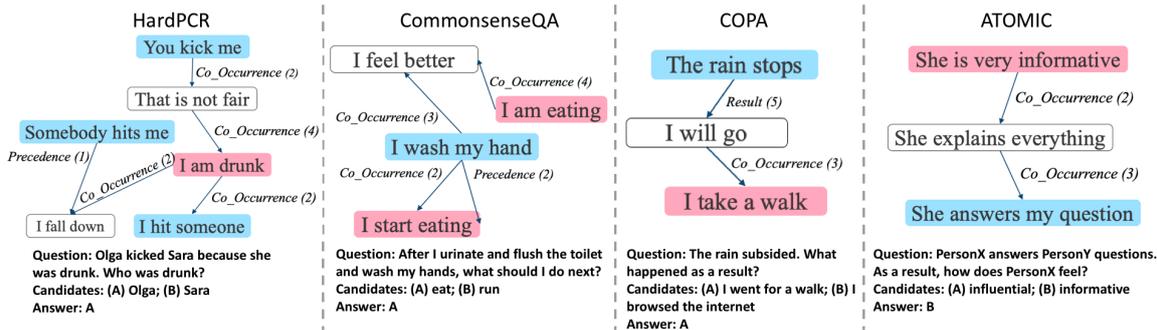


Figure 5: **CIKQA** Case Study. Mapped nodes for the question/answers are in blue/pink. Other nodes are white. Edge weights are in brackets. We only show the relevant parts of the graphs for clear representation.

of the gold knowledge identification task.

5 Related Work

To help machines understand commonsense, the community has devoted great efforts to constructing commonsense knowledge bases with either crowdsourcing (e.g., ConceptNet (Liu and Singh, 2004) and ATOMIC (Sap et al., 2019)) or information extraction techniques (e.g., ASER (Zhang et al., 2020a)). Typically, crowd-sourced knowledge bases are of higher quality, and the auto-constructed ones have broader coverage. Besides acquiring commonsense knowledge, the community also developed many commonsense reasoning datasets to train and test models’ commonsense reasoning abilities. Even though these datasets may have different *formats* (e.g., slot fitting in Winogrande (Sakaguchi et al., 2020) and question answering in CommonsenseQA (Talmor et al., 2019)), *knowledge types* (e.g., causal commonsense in COPA (Roemmele et al., 2011) and numerical commonsense in NumerSense (Lin et al., 2020)), or *modalities* (e.g., visual commonsense in VCR (Zellers et al., 2019) and textual commonsense in many others), they follow a standard supervised learning setting, and aim at helping machines to solve a specific commonsense task in an end-to-end manner. Given this setting, it is often difficult to tell what has been learned during the training. Was it used to acquire commonsense knowledge, learn to conduct commonsense inference, or both? Such ambiguity limits our progress in solving these commonsense reasoning tasks. In this work, we connect the efforts on commonsense acquisition and inference by creating a commonsense inference benchmark **CIKQA**, where models can focus on learning to identify the gold knowledge and perform inference over the sup-

porting commonsense knowledge.

Answering questions in natural language based on a knowledge base (KB) is a mature research topic in the NLP community, which is also known as the KBQA problem (Clark et al., 1999; Yih et al., 2015, 2016; Usbeck et al., 2017; Cui et al., 2017). Previous work mainly focuses on factual knowledge, which is stored in the triplets format. The main challenge is to parse the question and then precisely and effectively identify the correct path over a large-scale KB to make the inference. Compared with inference over factual knowledge, inference over commonsense knowledge brings the following unique challenges: (1) Commonsense is a kind of preference rather than fixed knowledge. As a result, the ideal commonsense reasoning process could involve the comparison of multiple candidates. For example, both “drink coffee” and “drink bear” could happen in the morning, but a normal person will prefer “drink coffee;” (2) Beyond named entities, commonsense knowledge also covers daily entities and events, and thus it is difficult to find an exact node from the commonsense KB that matches the question, and we may need to conduct inference based on the partial match (i.e., the extracted nodes are relevant but not identical).

6 Conclusion

In this paper, we present **CIKQA**, a unified commonsense inference benchmark. Specifically, we first convert several popular commonsense tasks into a unified QA format and then equip each question with a supporting commonsense knowledge graph. We also leverage humans to annotate the quality of auto-extracted knowledge. Experiments show that even though models can better learn how to perform commonsense inference with a few ex-

amples and significantly outperform the baseline method that does not use structured knowledge in the data-scarce setting, identifying the gold knowledge is still a challenging problem. More interestingly, with our unified formulation, models demonstrate the encouraging generalization ability across tasks. As both the format unification and supporting graph extraction are automatic, we can easily extend to other commonsense reasoning tasks in the future.

Acknowledgements

The authors of this paper were supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the BETTER Program, and by contract FA8750-19-2-1004 with the US Defense Advanced Research Projects Agency (DARPA). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This paper was also supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20) and the GRF (16211520 and 16205322) from RGC of Hong Kong, the MHKJFS (MHP/001/19) from ITC of Hong Kong and the National Key RD Program of China (2019YFE0198200) with special thanks to HKMAAC and CUSBLT, and the Jiangsu Province Science and Technology Collaboration Fund (BZ2021065). We also thank the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08). Yanai Elazar is grateful to be supported by the PBC fellowship for outstanding Ph.D. candidates in Data Science and the Google Ph.D. fellowship.

Limitations

A common limitation of existing semi-parametric models is the coverage of knowledge resources. **CIKQA** also faces this limitation. Based on our analysis, the largest commonsense knowledge bases can still cover part of the questions in existing commonsense benchmarks. How to populate these commonsense knowledge graphs is an important research question in the future.

References

- Ning Bian, Xianpei Han, Bo Chen, and Le Sun. 2021. [Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation](#). In *Proceedings of AAAI 2021*, pages 12574–12582. AAAI Press.
- Peter Clark, John Thompson, and Bruce Porter. 1999. [A knowledge-based approach to question-answering](#). In *Proceedings of AAAI 1999*, pages 43–51.
- Amir D. N. Cohen, Shachar Rosenman, and Yoav Goldberg. 2020. [Relation extraction as two-way span-prediction](#). *CoRR*, abs/2010.04829.
- Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2017. [KBQA: learning question answering over QA corpora and knowledge bases](#). *Proceedings of VLDB 2017*, 10(5):565–576.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of NAACL 2019*, pages 4171–4186.
- Xinya Du and Claire Cardie. 2020. [Event extraction by answering \(almost\) natural questions](#). In *Proceedings of EMNLP 2020*, pages 671–683.
- Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021. [Back to square one: Artifact detection, training and commonsense disentanglement in the winograd schema](#). In *Proceedings of EMNLP 2021*, pages 10486–10500. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [Simcse: Simple contrastive learning of sentence embeddings](#). In *Proceedings of EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. [Billion-scale similarity search with gpus](#). *CoRR*, abs/1702.08734.
- Jerrold Katz and Jerry Fodor. 1963. [The structure of a semantic theory](#). *Language*, 39:170–210.
- Mayank Kejriwal and Ke Shen. 2020. [Do fine-tuned commonsense language models really generalize?](#) *CoRR*, abs/2011.09159.
- Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hananeh Hajishirzi. 2020. [Unifiedqa: Crossing format boundaries with a single QA system](#). In *Proceedings of EMNLP 2020 Findings*, pages 1896–1907.
- Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. [The winograd schema challenge](#). In *Proceedings of KR 2012*.

- Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. [Kagnet: Knowledge-aware graph networks for commonsense reasoning](#). In *Proceedings of EMNLP-IJCNLP 2019*, pages 2829–2839.
- Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. [Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models](#). In *Proceedings of EMNLP 2020*, pages 6862–6868.
- Chin-Yew Lin. 2004. [Rouge: A package for automatic evaluation of summaries](#). In *Text summarization branches out*, pages 74–81.
- Hugo Liu and Push Singh. 2004. [Conceptnet: a practical commonsense reasoning tool-kit](#). *BT technology journal*, 22(4):211–226.
- Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2020. [Graph-based reasoning over heterogeneous external knowledge for commonsense question answering](#). In *Proceedings of AAAI 2020*, pages 8449–8456.
- Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. [GLUCOSE: Generalized and Contextualized story explanations](#). In *Proceedings of EMNLP 2020*, pages 4569–4586.
- Debjit Paul and Anette Frank. 2020. [Social commonsense reasoning with multi-head knowledge attention](#). In *Proceedings of the EMNLP 2020, Findings*, pages 2969–2980.
- Altaf Rahman and Vincent Ng. 2012. [Resolving complex cases of definite pronouns: The winograd schema challenge](#). In *Proceedings of CoNLL 2012*, pages 777–789.
- Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. [Choice of plausible alternatives: An evaluation of commonsense causal reasoning](#). In *Proceedings of AAAI 2011 Spring Symposium*, pages 90–95.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. [Winogrande: An adversarial winograd schema challenge at scale](#). In *Proceedings of AAAI 2020*, pages 8732–8740.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. [Multitask prompted training enables zero-shot task generalization](#). In *Proceedings of ICLR 2022*.
- Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. [ATOMIC: an atlas of machine commonsense for if-then reasoning](#). In *Proceedings of AAAI 2019*, pages 3027–3035.
- Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. [BLEURT: learning robust metrics for text generation](#). In *Proceedings of ACL 2020*, pages 7881–7892.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [Commonsenseqa: A question answering challenge targeting commonsense knowledge](#). In *Proceedings of NAACL 2019*, pages 4149–4158.
- Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 2017. [7th open challenge on question answering over linked data \(QALD-7\)](#). In *Proceedings of 4th SemWebEval Challenge at ESWC 2017*, pages 59–69.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [Huggingface’s transformers: State-of-the-art natural language processing](#). *CoRR*, abs/1910.03771.
- Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. [Corefqa: Coreference resolution as query-based span prediction](#). In *Proceedings of ACL 2020*, pages 6953–6963.
- Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. [Semantic parsing via staged query graph generation: Question answering with knowledge base](#). In *Proceedings of ACL 2015*, pages 1321–1331.
- Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. [The value of semantic parse labeling for knowledge base question answering](#). In *Proceedings of ACL 2016*, pages 201–206.
- Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [From recognition to cognition: Visual commonsense reasoning](#). In *Proceedings of CVPR 2019*, pages 6720–6731.
- Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020a. [ASER: A large-scale eventuality knowledge graph](#). In *Proceedings of WWW 2020*, pages 201–211.

Hongming Zhang, Xinran Zhao, and Yangqiu Song.
2020b. [Winowhy: A deep diagnosis of essential commonsense knowledge for answering winograd schema challenge](#). In *Proceedings of ACL 2020*, pages 5736–5745.

Hongming Zhang, Xinran Zhao, and Yangqiu Song.
2021. [A brief survey and comparative study of recent development of pronoun coreference resolution in English](#). In *Proceedings of CRAC@EMNLP 2021*, pages 1–11.

Data-Efficient Methods For Improving Hate Speech Detection

Sumegh Roychowdhury *
IIT Kharagpur, India
sumegh-tech@gmail.com

Vikram Gupta
ShareChat, India
vikramgupta@sharechat.co

Abstract

Scarcity of large-scale datasets, especially for resource-impo-verished languages encouraged exploration of data-efficient methods for hate speech detection. In this work, we progress *implicit* and *explicit* hate speech detection using an input-level data augmentation technique, task reformulation using entailment and cross-learning across five languages. Our proposed data augmentation technique `EasyMixup`, improves the F1 performance across languages by **0.5-9%**. We also observe substantial F1 gains of **1-8%** by reformulating hate speech detection as `Entailment-style` problem. We further probe the contextual models and observe that higher layers encode *implicit* hate while lower layers focus on *explicit hate*, highlighting the importance of token-level understanding for *explicit* and context-level for *implicit* hate speech detection.¹

1 Introduction

Deep learning based methods (Badjatiya et al., 2017; Zhang et al., 2018; Kshirsagar et al., 2018) have shown impressive results in detecting hate speech. Transformer based models (Caselli et al., 2021; Tekiroğlu et al., 2020; Aluru et al., 2020; Mozafari et al., 2019; Dutta et al., 2022) have further pushed the state-of-the-art by leveraging large amount of unlabeled data in a self-supervised manner. Various hate speech detection datasets have been contributed in textual (Gibert et al., 2018; Davidson et al., 2017; Founta et al., 2018), audio (Gupta et al., 2022) and visual (Gomez et al., 2020) domains. However, these algorithms are data-hungry and motivate development of algorithms which are data-efficient.

To tackle this, we introduce an input-level data augmentation technique `EasyMixup` and improve hate speech detection in monolingual and

multilingual settings. `EasyMixup` is inspired by *mixup* based augmentation techniques which are broadly categorized into input-level mixup (Yun et al., 2019; Kim et al., 2020; Uddin et al., 2021; Walawalkar et al., 2020) and hidden-level mixup (Verma et al., 2019). `EasyMixup` follows the input-level paradigm and leverages a simple observation that the label of a hateful instance is preserved on concatenation with a hateful or non-hateful instance. Similarly, label of a non-hateful instance does not change on concatenation with another non-hateful instance.

We also study the efficacy of reformulating hate speech detection as `Entailment-style` problem. We extend the work by (Wang et al., 2021) and perform detailed experiments under *implicit*, *explicit* and *multilingual* settings. We observe that monolingual entailment performs better than English based entailment. This observation is intuitive because the models are pretrained using pair of sentences from same language and monolingual entailment reflects the same settings.

Majority of the existing textual datasets focus on *explicit* hate speech where *swear*, *cuss*, *abusive* words are used to express the hateful intent. In contrast, *implicit* hate speech employs subtle, indirect and contextual ways for expressing hate speech making it extremely harmful and difficult, as shown in (ElSherief et al., 2021). Acknowledging this difference of expression, we explore the relationship between *explicit* and *implicit* hate speech using cross-learning and observe strong correlations. We also perform probing experiments and observe that lower layers focus on *explicit* hate speech while higher layers are responsible for encoding *implicit* hate speech. This alludes to the hypothesis that *implicit* hate speech is more contextual in nature and requires more understanding, while *explicit* hate speech can be detected by leveraging lower-level information.

In summary, our main contributions are:

*Work done during internship at ShareChat

¹Code and Dataset splits - https://github.com/Sumegh-git/data_efficient_hatedetect

- We propose input-level data augmentation technique `EasyMixup` which outperforms previous methods for our task.
- We show performance gains by reformulating hate speech detection as monolingual `Entailment-style` problem.
- We probe contextual models and observe that higher layers encode *implicit* hate speech while lower layers focus on *explicit hate* speech.
- We show that correlations exist between *explicit* and *implicit* hate speech and leverage that for improving hate speech detection.

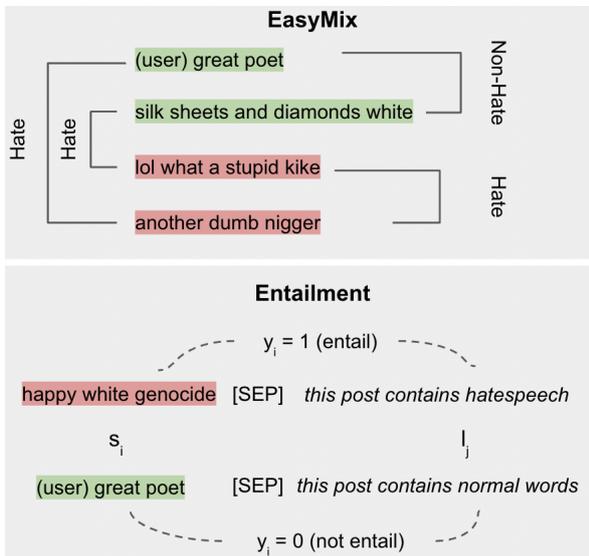


Figure 1: (top) Mixing hateful (red) samples with either hate (red) or non-hate (green) samples doesn’t change the final label. Similarly mixing two non-hate (green) samples preserve the final label. (bottom) Posing hatespeech classification as entailment task. [Best viewed in color]

2 Methodology

2.1 EasyMixup

`EasyMixup` is an input-level data augmentation technique where we leverage the observation that ground truth label of a hateful sample does not change on concatenation with another hateful or non-hateful samples. Similarly, concatenation of a non-hate sample with another non-hate sample results in a novel non-hate sample as shown in Figure 1(top). More formally, let’s say (s_i, y_i) is the sentence and it’s corresponding label $y \in \{hate, non-hate\}$ in a minibatch S and D is the entire

dataset,

$$S = \{(s_0, y_0), (s_1, y_1), \dots, (s_n, y_n) | (s_i, y_i) \in D\}$$

For every sample in the batch, $s_i \in S$, we randomly select $(\bar{s}_i, \bar{y}_i) \in D$ with $\bar{s}_i \neq s_i$ and augmentation probability p_{aug} to create new augmented sample:

$$s_{i_{aug}} = \phi(s_i, \bar{s}_i), y_{i_{aug}} = y_i \vee \bar{y}_i$$

where ϕ is defined as :

$$\phi(s_i, \bar{s}_i) = \begin{cases} concat(s_i; \bar{s}_i) & \bar{p} > p_{flip} \\ concat(\bar{s}_i; s_i) & , otherwise \end{cases}$$

where, p_{flip} is the sentence flipping probability and $concat()$ refers to concatenation. Flipping introduces more augmentation and prevents the model from learning positional bias. Finally, we get the updated minibatch \bar{S} by replacing original with augmented samples $(s_{i_{aug}}, y_{i_{aug}})$.

2.2 Entailment-style

We reformulate hate speech classification task as an entailment-style task (Wang et al., 2021). The (input, target) for the contextual model is: $(s_i[sep]l_j, y_i)$, where, s_i is the original sentence, l_j is the label-prompt, $[sep]$ is the separator and $y_i \in \{0, 1\}$ as shown in Figure 1 (bottom). Label-prompt represents the ground-truth label of the sentence in textual format. For example, *this post contains hatespeech* / *this post contains normal words* can be used as label-prompt for *hate* and *non-hate* sentences respectively (Table B). The target to the model, $y_i = 0$ indicates that the sentence, s_i and label-prompt, l_j do not entail each other. $y_i = 1$ indicates entailment. We extend analysis of `Entailment-style` for multiple languages using monolingual and multilingual label-prompts.

2.3 Explicit and Implicit Hate Speech

In this section, we study the correlation between *explicit* and *implicit*. As discussed previously, *explicit* hate speech comprises of *cuss*, *swear*, *abusive*, *profane* words but *implicit* hate speech is more contextual and indirect. While the manner of expression is different, the intent behind both these modes is similar. To leverage this, we pretrain on the task of *explicit* hate speech detection and finetune it on *implicit* hate speech dataset and vice-versa and observe consistent gains. We probe the

Model	Acc	F1	Δ F1
RoBERTa-base	68.61	67.20	-
RoBERTa-Tw	69.18	67.64	+0.44
RoBERTa-TwS	69.54	67.88	+0.24
RoBERTa-TwS-EasyMixup	69.80	68.33	+0.45
Mathew et al. (2021)	69.00	67.40	-

Table 1: Explicit Hate: Accuracy and F1 score on HateXplain dataset averaged over 3 runs.

Model	Acc	F1	Δ F1
RoBERTa-base	76.91	74.09	-
RoBERTa-Tw	77.86	75.77	+0.68
RoBERTa-TwS	78.36	76.13	+0.36
RoBERTa-TwS-EasyMixup	78.38	76.66	+0.53
ElSherief et al. (2021)	77.50	70.40	-

Table 2: Implicit Hate: Accuracy and F1 score on LatentHatred dataset averaged over 3 runs.

layers of contextual models by extracting the features from each layer and training a classifier over these representations to understand how contextual models encode the information about hate speech and observe that *explicit* and *implicit* hate speech is encoded differently.

3 Dataset and Models

Explicit: We experiment with HateXplain (HX)(Mathew et al., 2021) dataset for *explicit* hate speech study. HateXplain (HX) captures explicit lexicon based hate speech posts collected from popular social media sites like Twitter and Gab.

Implicit: For *implicit* hate speech, we use LatentHatred (LH)(ElSherief et al., 2021), which comprises of *implicit* hate speech containing indirect/coded language.

Multilingual: We also experiment with *explicit* hate speech datasets in French (FR), Spanish (ES), Arabic (AR) and Portuguese (PT)² for evaluating our methodology for different languages. Since the taxonomy was different for each label, we focus on the datapoints annotated with *hate* and *non-hate* labels only (Poletto et al., 2021). In Appendix Section A, we summarize the details and statistics

²hatespeechdata.com

Model	Accuracy	F1 Score	Δ F1
RoBERTa-Tw	69.18	67.64	-
RoBERTa-Tw-IH	70.74	68.88	+1.24
RoBERTa-Tw	77.86	75.77	-
RoBERTa-Tw-EH	78.38	75.95	+0.18

Table 3: Cross-Learning results between *explicit* and *implicit* hate speech detection.

Lang	DL	XLM-R	XLM-Tw	XLM-TwS	EM-mo	EM-mu
FR	65.95	64.48	68.36	72.73	78.58	81.16
ES	73.29	76.99	77.27	77.87	79.23	80.66
AR	83.20	82.36	83.57	84.50	84.80	85.60
PT	69.41	71.83	72.35	72.76	73.60	74.09

Table 4: F1 score on two-way classification (hate, non-hate) for different languages using adaptation and monolingual (EM-mo) and multilingual (EM-mu) variations of EasyMixup augmentation. DL((Aluru et al., 2020))

	Baseline		+ prompt-en		+ prompt	
	Acc	F1	Acc	F1	Acc	F1
HX	69.85	68.36	72.97	71.39	72.97	71.39
LH	77.81	74.42	78.57	75.97	78.57	75.97
FR	88.46	84.62	88.55	84.64	94.23	92.83
ES	76.13	75.87	77.06	76.74	80.44	79.97
AR	89.67	78.09	89.30	78.51	90.41	82.03
PT	72.19	66.50	75.00	67.98	79.23	71.04

Table 5: F1 score on entailment task for all datasets using english prompts (prompt-en) and language-specific prompts (prompt). Baseline corresponds to BERT-base for HX, LH and mBERT for rest. For English datasets, prompt is equivalent to prompt-en.

of all the datasets.

Models: We consider RoBERTa-base (Liu et al., 2019) and XLM-R (Conneau et al., 2020) as the baseline model for English and other languages respectively. For exploring the impact of domain adaptive models, we experiment with RoBERTa-Tw and XLM-Tw models. For the multilingual experiments, we use XLM-TwS, which is the XLM-Tw model finetuned on the UMSAB dataset (Barbieri et al., 2021). More details in Appendix Section C.

4 Results

Explicit: In Table 1, we report the results on HateXplain dataset. We observe that RoBERTa-Tw improves upon the results of RoBERTa-base model. This shows that the pre-training over similar domain (social media) helps in achieving better performance. RoBERTa-TwS which has been trained for sentiment detection demonstrates further improvement highlighting the correlation between sentiments and hate-speech detection. On adding our augmentation (RoBERTa-TwS-EasyMixup), we notice further performance gains demonstrating the benefits of EasyMixup augmentation. Overall, our results improve upon the previously reported baseline (Mathew et al., 2021).

Implicit: We conduct similar experiments on LatentHatred dataset. We notice gains by using the domain adapted RoBERTa-Tw model.

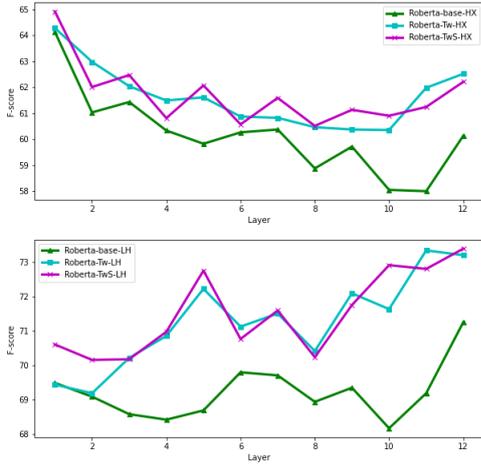


Figure 2: Layer-wise probing results on HateXplain (top) and LatentHatred (bottom) datasets for RoBERTa-base, RoBERTa-Tw and RoBERTa-TwS [Best viewed in color].

RoBERTa-TwS does not improve the accuracy but improves upon the F1 score which is a better metrics due to data imbalance. Addition of EasyMixup (RoBERTa-TwS-EasyMixup) further improves the performance. Our results improve upon the previously reported state-of-the-art results 2.

Explicit-Multilingual: We evaluate our method on 4 more languages in Table 4 and observe similar trends. For all the languages, multilingual domain (XLM-Tw) and task adapted (XLM-TwS) models perform better than the base model (XLM-R). On integration of EasyMixup, we further note improvements. We also experiment with sampling augmented samples from other languages (EM-mu) and notice further gains highlighting the cross learning between languages by 1-3%. We compare EasyMixup with state-of-the-art method SSMixup in Table 6 and observe that EasyMixup improves the performance by 1-2% for both implicit and explicit hate-speech detection. **Entailment-style:** In Table 5, we report the results using monolingual³ and English prompts and observe that monolingual prompts outperform English prompts. This is not surprising considering that models are trained on pairs of sentences from same language only. We use mBERT/BERT-base for this study as it has been trained with NSP task which aligns with Entailment-style. Check Appendix B for more details.

Implicit-Explicit Correlation: We finetune the RoBERTa-Tw model on *implicit* hate speech

³We used Google Translate to obtain monolingual prompt

Model	Acc	F1
LatentHatred		
BERT-base	76.51	73.70
+SSMixup (Yoon et al., 2021)	77.30	74.76
+EasyMixup	77.52	75.28
HateXplain		
BERT-base (Mathew et al., 2021)	69.00	67.40
+SSMixup	69.59	67.72
+EasyMixup	69.70	68.66

Table 6: Comparing EasyMixup with SSMixup (Yoon et al., 2021)

(RoBERTa-Tw-IH) before training it for implicit hate speech and observe the F1 improvement from 67.64 to 68.88 in Table 3. This shows that *implicit* hate speech detection benefits the task of *explicit* hate speech. Similarly, F1 score of *implicit* hate speech detection improves from 75.77 to 75.95 by finetuning using *explicit* hate speech dataset.

Probing: In Figure 2, we plot the F1 score of RoBERTa-base and RoBERTa-Tw for *explicit* and *implicit* hate speech across different layers of the contextual model. We note that lower layers show higher F1 for *explicit* hate speech detection (expected layer = 0.98), while higher layers demonstrate better *implicit* hate detection performance (expected layer = 5.12). This alludes to the hypothesis that *implicit* hate speech is contextual in nature while *explicit* hate speech can be detected by using token-level information also. Training details are described in Appendix Section D.

5 Conclusion

In this work, we introduced a novel input-level data-augmentation technique, EasyMixup which shows performance gains over monolingual and multilingual settings. We also explored reformulation of hate speech classification as Entailment-style problem and achieved substantial performance gains using monolingual entailment. We also performed layer probing to find that higher layers encode *implicit* hate information, while lower layers are more focused on *explicit* hate speech highlighting the contextual nature of *implicit* and token-level dependence of *explicit* hate speech. In future work, we would like to explore how EasyMixup and Entailment-style perform when ensembled together in both mono, multi-lingual settings.

6 Limitations

One limitation would be that EasyMixup won't be applicable in tasks like sentiment analysis where the final mixed label might not be binary.

References

- Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection. In *ECML-PKDD*.
- Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. [Deep learning for hate speech detection in tweets](#). In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, page 759–760, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Francesco Barbieri, José Camacho-Collados, Leonardo Neves, and Luis Espinosa Anke. 2020. Tweeteval: Unified benchmark and comparative evaluation for tweet classification. *Findings of EMNLP*.
- Francesco Barbieri, Luis Espinosa-Anke, and Jose Camacho-Collados. 2021. A Multilingual Language Model Toolkit for Twitter. In *arXiv preprint arXiv:2104.12250*.
- Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. [SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter](#). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. [HateBERT: Retraining BERT for abusive language detection in English](#). In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.
- T. Davidson, D. Warmley, M. W. Macy, and I. Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media*, 512–515. Montreal, Québec, Canada: AAAI Press.
- Parag Dutta, Souvic Chakraborty, Sumegh Roychowdhury, and Animesh Mukherjee. 2022. [Crush: Contextually regularized and user anchored self-supervised hate speech detection](#). *Findings of NAACL*.
- Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. *EMNLP*.
- Paula Fortuna, João Rocha da Silva, Juan Soler-Company, Leo Wanner, and Sérgio Nunes. 2019. A hierarchically-labeled portuguese hate speech dataset. In *Proceedings of the 3rd Workshop on Abusive Language Online (ALW3)*.
- Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. *ICWSM*.
- Ona Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*.
- Raul Gomez, Jaume Gibert, Lluís Gómez, and Dimosthenis Karatzas. 2020. [Exploring hate speech detection in multimodal publications](#). *WACV*.
- Vikram Gupta, Rini Sharon, Ramit Sawhney, and Debdoot Mukherjee. 2022. Adima: Abuse detection in multilingual audio. *arXiv preprint arXiv:2202.07991*.
- Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. 2020. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *ICML*.
- Rohan Kshirsagar, Tyrus Cukovac, Kathy McKeown, and Susan McGregor. 2018. [Predictive embeddings for hate speech detection on twitter](#). In *Abusive Language Online Workshop, EMNLP 2018*, pages 26–32.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arxiv.1907.11692*.
- B. Mathew, R. Dutt, P. Goyal, , and A Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science*, 173–182.
- Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *AAAI*.
- Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2019. [A bert-based transfer learning approach for hate speech detection in online social media](#). *CoRR*, abs/1910.12574.

- Hala Mulki, Hatem Haddad, Chedi Bechikh Ali, and Halima Alshabani. 2019. L-hsab: A levantine twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 111–118.
- N. Ousidhoum, Z. Lin, H. Zhang, Y. Song, and D.-Y. Yeung. 2019a. Multilingual and multi-aspect hate speech analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4667–4676.
- Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019b. Multilingual and multi-aspect hate speech analysis. In *Proceedings of EMNLP*. Association for Computational Linguistics.
- Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2021. Resources and benchmark corpora for hate speech detection: a systemic review. *Language Resources and Evaluation volume 55*, pages 477–523.
- Lara Quijano-Sanchez, Juan Carlos Pereira Kohatsu, Federico Liberatore, and Miguel Camacho-Collados. 2019. Haternet a system for detecting and analyzing hate speech in twitter. In *Zenodo*.
- Serra Sinem Tekiroğlu, Yi-Ling Chung, and Marco Guerini. 2020. [Generating counter narratives against online hate speech: Data and strategies](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1177–1190, Online. Association for Computational Linguistics.
- F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, and Sung-Ho Bae. 2021. Saliency mix: A saliency guided data augmentation strategy for better regularization. In *ICLR*.
- Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *ICML*.
- Devesh Walawalkar, Zhiqiang Shen, Zechun Liu, and Marios Savvides. 2020. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. In *arXiv:2003.13048*.
- Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner. *arXiv preprint arXiv:2104.14690*.
- Soyoung Yoon, Gyuwan Kim, and Kyumin Park. 2021. Ssmix: Saliency-based span mixup for text classification. In *Findings of ACL*.
- Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*.
- Ziqi Zhang, D. Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *Extended Semantic Web Conference, ESWC 2018*.

A Dataset

In Table 7, we note the dataset size and source of the datasets used in our study. Majority of the datasets are source from Twitter and have data imbalance.

Explicit Hate (HX): HateXplain dataset has been sourced from Twitter and Gab. The lexicon set from (Davidson et al., 2017), (Ousidhoum et al., 2019a) & (Mathew et al., 2019) is combined to sample 1% tweets in the period Jan-2019 to Jun-2020. For Gab, they use the dataset provided by (Mathew et al., 2019). All posts containing embedded links, pictures, videos were removed and usernames were anonymized by replacing with *user* token. Each post in the dataset is labelled into 3 categories: *Normal*, *Offensive* or *Hateful*. For the annotation task, Amazon Mechanical Turk (MTurk) workers are used where each post is labelled by 3 annotators and the ground truth class is chosen by majority voting. Finally, 19,229 posts were annotated of which 5,935 were hateful, 5,480 were offensive and 7,814 were normal. For the rest 919 posts the annotators provided 3 different classes and hence these were discarded.

Implicit Hate (LH): LatentHatred introduces a theoretically-justified taxonomy of implicit hate-speech with fine-grained labels on eight ideological clusters of US hate groups as given by the SPCL report - *Black Separatist*, *White Nationalist*, *Neo-Nazi*, *Anti-Muslim*, *Racist Skinhead*, *Ku Klux Klan*, *Anti-LGBT* and *Anti-Immigrant*. For high-level categorization, the tweets were categorized into *explicit hate*, *implicit hate* & *non-hateful*. Overall, the dataset contains 21,480 tweets, where 7,100 were implicit hate, 1,089 explicit hate and 13,291 non-hateful. Using majority vote, labels were obtained for 19,112 tweets of which 4,909 were implicit hate, 13,291 non-hateful and rest 933 explicit hate were discarded. For a finer categorization, 6 labels were chosen representing principal axes of implicit hate - *White Grievance*, *Incitement*, *Inferiority*, *Irony*, *Stereotypes & Threatening*. The 4,909 implicit hate tweets labeled in the high-level stage were further annotated using the above mentioned fine-grained labels.

Multilingual: We collected 6 publicly available datasets in 4 different languages - French, Spanish, Arabic and Portuguese and combined them individually. Each dataset had a variety of labels - *hate*, *abusive*, *profanity*, *offensive* etc. Since the taxonomy is different for each label, we focus on the

Dataset	Source	#datapoints	%hate
HateXplain	Twitter, Gab	19,229	30.86
LatentHatred	Twitter	20,391	34.82
Arabic	Twitter	5,418	17.07
Portuguese	Twitter	5,670	31.53
Spanish	Twitter	11,150	33.29
French	Twitter	1,028	20.14

Table 7: Dataset Statistics

datapoints annotated with *hate* and *non-hate* labels. We describe each dataset in following section.

- **Arabic (AR):** Mulki et al. (2019) contains Syrian/Lebanese political tweets labeled as abusive, normal or hate. (Ousidhoum et al., 2019b) consists of multi-labeled tweets based on attributes like hostility, target, directness, etc.
- **Spanish (ES):** Basile et al. (2019) provided a multilingual hatespeech dataset against women & immigrants. Quijano-Sanchez et al. (2019) collected a small hatespeech dataset in spanish with hate/non-hate labels.
- **Portuguese (PT):** Fortuna et al. (2019) provided a hierarchically labeled hatespeech dataset of which we use only the binary labels for our task.
- **French (FR):** Ousidhoum et al. (2019b) consists of multi-labeled tweets based on attributes like hostility, target, directness, etc.

B Prompts used for Entailment-style task

Refer to Table 8.

C Model Details

ROBERTa-Tw is based on ROBERTa-base model trained on 60M English tweets. XLM-Tw (Barbieri et al., 2021) is a XLM-R model trained on 200M tweets retrieved from 30+ languages. For task-adaptive models, we take ROBERTa-TwS and ROBERTa-Tw-EH which are initialized with the ROBERTa-Tw model and further finetuned using Sentiment and Hatespeech classification data from the TweetEval (Barbieri et al., 2020) benchmark.

D Implementation Details

We perform all experiments with 3 different seeds on a single NVIDIA V100 GPU and report the

Language	Label Description
HateXplain	<i>this post contains hate speech / this post contains {offensive,normal} words</i>
LatentHatred	<i>this is implicit hate / this is normal</i>
French	<i>c'est odieux / c'est normal</i>
Spanish	<i>esto es odioso / esto es normal</i>
Arabic	<small>هذا المنشور يحتوي على كلمات / هذا المنشور يحتوي على كلمات عادية / غير الكراهة</small>
Portuguese	<i>este post contém discurso de ódio / este post contém palavras normais</i>

Table 8: Prompts used across various datasets for Entailment-style task.

average score. We use a batch size of 16 and maximum sequence length of 128. We choose initial learning rate from $\{3e-5, 4e-5, 5e-5\}$ and perform linear decay after 10% warmup steps. We use the AdamW optimizer and train our models for 5 epochs. The classifier head consists of a 2-layer MLP with ReLU activation. We choose the best checkpoint using validation metrics every epoch. From our experiments, we found best reported results were obtained by combining *offensive+normal* & *hate+normal* classes for HateXplain and *hate+normal* classes for LatentHatred and keeping $p_{aug} = 0.2$ and $p_{flip} = 0.5$.

For the probing experiments, we train the 2-layer MLP probe classifier for 50 epochs with batch size 64 and learning rate $1e-3$.

For the entailment experiments, we use a batch size 128 (required for entailment method to get good gains) consistently for all methods and learning rate $3e-5$.

E Effect of Length

We used the max sequence length of 128 in our experiments. $< 1\%$ of samples exceed this limit across all datasets - HateXplain, LatentHatred, MultilingualHate. Thus, length of 128 tokens does not degrade Entailment-style performance. However, in case of EasyMixup, length of concatenated sentences could exceed 128 tokens. To evaluate the impact, we repeat experiments using best performing model - RoBERTa-TwS-EasyMixup (averaged over 3 random seeds) keeping maximum sequence length as 512. For HateXplain, Δ Accuracy / F1 $\sim 0.00 / -0.03 \%$ and for LatentHatred Δ Accuracy / F1 $\sim +0.21 / -0.05 \%$. As we can see there is no significant impact from the reported results. This can be attributed to the fact that we do probabilistic mixup in EasyMixup ($p_{aug} = 0.2$ and $p_{flip} = 0.5$). Thus the model sees all type of examples during the training phase.

F Ethical Considerations

All the datasets that we use are publicly available. We report only aggregated results in the main paper. We have not or do not intend to share any Personally Identifiable Data with this paper. We release the code and data associated with this paper as well - https://anonymous.4open.science/r/data_efficient_hatedetect/

Learning the Effects of Physical Actions in a Multi-modal Environment

Gautier Dagan, Frank Keller, Alex Lascarides

School of Informatics

University of Edinburgh, UK

gautier.dagan@ed.ac.uk, {keller, alex}@inf.ed.ac.uk

Abstract

Large Language Models (LLMs) handle physical commonsense information inadequately. As a result of being trained in a disembodied setting, LLMs often fail to predict an action’s outcome in a given environment. However, predicting the effects of an action before it is executed is crucial in planning, where coherent sequences of actions are often needed to achieve a goal. Therefore, we introduce the multi-modal task of predicting the outcomes of actions solely from realistic sensory inputs (images and text). Next, we extend an LLM to model latent representations of objects to better predict action outcomes in an environment. We show that multi-modal models can capture physical commonsense when augmented with visual information. Finally, we evaluate our model’s performance on novel actions and objects and find that combining modalities help models to generalize and learn physical commonsense reasoning better.

1 Introduction

Large Language Models (LLMs) are trained on large corpora of disembodied texts. They are typically pre-trained on a masked language modeling task: the model must predict a masked word in a text given its context. LLMs have achieved state-of-the-art performance on many NLP tasks (Devlin et al., 2019; Brown et al., 2020), but they can also fail on seemingly easy and obvious tasks and in unpredictable ways (McCoy et al., 2020; Bommasani et al., 2021). Commonsense knowledge is shared knowledge and is often so obvious that it is absent from the LLMs’ training data: people don’t mention what is already known to their interlocutors. This includes physical commonsense information, including how executed actions affect the physical attributes of objects; e.g., shape and weight (Forbes et al., 2019). Humans may learn such knowledge from their embodied environment. But LLMs, being trained on disembodied text, can make incorrect

predictions about physical attributes and how these change when actions occur. For instance, when asked what the weight of a 150 grams potato after it is sliced, GPT-3 (Brown et al., 2020) incorrectly answers 75 grams (see Appendix A for the exact prompt). GPT-3 is an LLM with 175 billion parameters, and nonetheless its disembodied existence limits its physical commonsense estimates.

Zellers et al. (2021) inject physical commonsense information into LLMs via their model PIGLeT—a modified LLM that is trained on their PIGPeN simulated 3D environment dataset. PIGLeT estimates how an environment changes as a result of specific actions. In training and testing, the model uses ground-truth symbolic representations of the environment but not the images: it ignores visual sensory observations. These symbolic representations of objects in an environment are chosen to capture the possible effects of actions, and include attributes like weight, size and temperature. However, in an embodied situation, an agent needs to use visual perception to estimate its interpretation of the scene. Therefore, the symbolic representations should be treated as latent rather than observed.

We propose an alternative to the PIGLeT model, PIGLeT-Vis, which uses images directly as input into a multi-modal LLM to ground the model to its physical environment. We compare our approach to the original PIGLeT model and evaluate the generalization capabilities gained from using image inputs. At test time, our model foregoes symbolic labels: only the images and the name of the action are observed. Thus our model tackles a more challenging task than the original PIGLeT model in that it must not only predict the effect of actions but also (indirectly) estimate the symbolic representations of objects in the images. We also evaluate a model for predicting the effects of actions that trains on PIGPeN’s images and their associated natural language (NL) descriptions, eliminating the need for

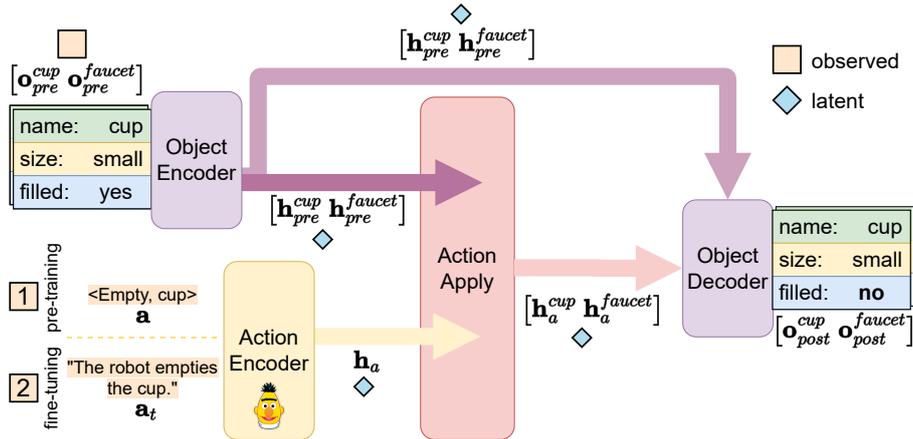


Figure 1: **Original PIGLeT Physical Dynamics Model (Zellers et al., 2021)**. During pre-training the model receives as input the full symbolic representation of two objects (\mathbf{o}_{pre}^0 and \mathbf{o}_{pre}^1) before the action is taken and the symbolic representation of the action itself (\mathbf{a}) and is tasked with predicting the attributes of the objects after the action (\mathbf{o}_{post}^0 and \mathbf{o}_{post}^1). During fine-tuning, the action encoder is replaced by an LLM to process a natural language description of the action being taken and with what objects.

formal symbolic representations.

Our contributions are three-fold. First, we show that it is possible to predict the physical effects of actions from visual data. Second, we show that it is possible to learn the task on training data where formal symbolic representations, which are unobservable in real-world settings, are replaced with NL descriptions (which can be observed through natural interaction). Third, we evaluate all our models in a stricter zero-shot setup to promote ways to train agents that generalize. Overall our work paves the way for multi-modal models that learn the effects of actions in realistic environments.

2 Related Work

Commonsense reasoning has been highlighted as a potential weak point of LLMs in recent years (Shen and Kejriwal, 2021; Forbes et al., 2019; Bisk et al., 2020). Datasets such as PIGPeN (Zellers et al., 2021), commonsenseQA (Talmor et al., 2019), VCR (Zellers et al., 2019) and GD-VCR (Yin et al., 2021) help evaluate different aspects of commonsense reasoning in modern LLMs. In this paper, we focus on physical commonsense reasoning, which involves understanding the (often) unexpressed rules of the physical world.

Forbes et al. (2019) reported that neural representations found it challenging to infer the link between actions and what they imply about the attributes of objects. Accordingly, Zellers et al. (2019) introduced the Visual Commonsense Reasoning (VCR) task to test how images can inform

question answering models that tackle commonsense information. Bisk et al. (2020) designed the PIQA benchmark to evaluate physical commonsense reasoning in LLMs through question answering. Sampat et al. (2021) proposed an extension to the CLEVR dataset, where an agent must reason and answer questions about a scene after a hypothetical action is taken.

Multiple approaches can improve the capabilities of LLMs in commonsense reasoning, such as using handcrafted knowledge graphs (Hwang et al., 2021) or leveraging simulated environments (Zellers et al., 2021). PIGLeT, in particular, combines a traditional LLM and a “Physical Dynamics” model to ground an LLM (Zellers et al., 2021). The Physical Dynamics model enhances the commonsense knowledge of an LLM by fine-tuning it, using trajectories sampled from a realistic environment (see Figure 1). Trajectories are an action and a pair of environment states (before and after the action) expressed in a formal symbolic representation. Zellers et al. (2021) found that fine-tuning LLMs with symbolic data from the simulated environment helped them outperform other models in physical commonsense reasoning tasks: in particular, predicting the effects of an action when executed in a particular state.

Image inputs offer a way to ground an LLM, as they only require general alignment with a text or symbolic input and do not require the comprehensive environment ground-truth labels that PIGLeT uses. Gao et al. (2018) used multi-modal web data to learn actions and their effects from images

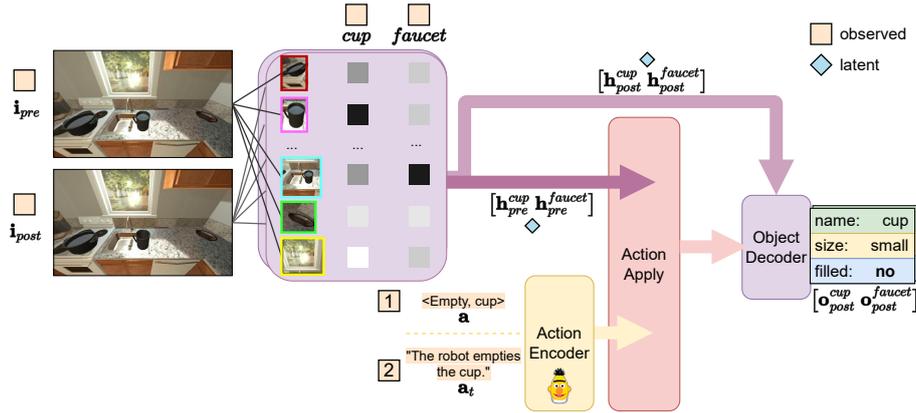


Figure 2: **PIGLeT-Vis**. We introduce PIGLeT-Vis, where we modify the PIGLeT architecture to replace its Symbolic Object Encoder with a vision component that makes use of images of the environment before and after an action is taken to predict the symbolic representation of objects post-action. We use an attention mechanism over the extracted bounding boxes to obtain a visual hidden representation of an object given its name. The only remaining symbolic inputs during pre-training are the action description and object names.

and corresponding text descriptions. Zellers et al. (2019) used an off-the-shelf ResNet50 model (He et al., 2016) to augment an existing BERT language model (Devlin et al., 2019) with vision capabilities. Transformer models such as UNITER (Chen et al., 2020), ERNIE-ViL (Yu et al., 2021), VisualBERT (Li et al., 2020), and ViLBert (Lu et al., 2019) have been applied to visual commonsense reasoning. These models use a joint transformer backbone for images and text and vary their pre-training objectives. However, most of these models are trained on static text-image pairs: they aren’t designed to capture the dynamics of an environment, particularly how object attributes change with actions. Notably, recent work by Hanna et al. (2022) uses CLIP (Radford et al., 2021) and MOCA (Singh et al., 2021) embeddings to predict a post-action image given a set of possible images. In contrast, we focus on adapting an LLM with a vision-based component to predict the consequences of actions on the environment.

3 Method

We propose PIGLeT-Vis (Figure 2) for learning the effects of actions on objects from images. We use a pre-trained vision backbone, DETR (Carion et al., 2020), as a Vision Object Encoder and combine it with a RoBERTa LLM (Liu et al., 2019) as an Action Encoder. We experiment with different configurations of inputs to measure the impact of the various components of our architecture. In particular, we test a variation in which we remove the formal symbolic labels even in training, replacing

them with NL text labels. To evaluate our models, we use the PIGPeN dataset (Zellers et al., 2021), which consists of a symbolic and visual representation of an environment before and after an action is taken. However, we filter PIGPeN to create a viable testing ground for visual grounding of physical actions and more accurately measure generalization capabilities of models.

3.1 Architecture

PIGLeT-Vis (shown in Figure 2) consists of separate components, which can combine multi-modal inputs in different ways. Through this modular approach, we can turn off specific components to evaluate how different inputs and model structures affect performance on the task. We test models with and without symbolic inputs and image inputs. For all components, we use a dropout of $p = 0.1$ in between layers and a default hidden layer size of $h = 64$.

3.1.1 Object Encoder

We reproduce Zellers et al. (2021), where all actions are assumed to involve two objects, \mathbf{o}^0 and \mathbf{o}^1 , and the symbolic representation of objects are encoded in an Object Encoder model. The symbolic representation of an object before the action is represented by \mathbf{o}_{pre} . Both objects (\mathbf{o}_{pre}^0 and \mathbf{o}_{pre}^1) in the environment are described by a vector of 38 attributes, chosen on the basis that they are the kinds of physical attributes that are influenced by actions. They describe an object as small/large, cold/hot, empty/full, etc.

We first embed these symbolic object attributes

using an embedding layer $\mathbf{E}^{e \times h}$, where $e = 329$ is the total number of unique attributes and h is our hidden size. For an object k :

$$\hat{\mathbf{o}}_{pre}^k = \mathbf{E}(\mathbf{o}_{pre}^k) \quad (1)$$

The Object Encoder $\mathbf{O}_{encoder}$ takes in the embedded object attributes through a set of multi-head attention layers to encode the symbolic representation of each object. We use the default Pytorch implementation of the Transformer Encoder (Paszke et al., 2019) with three layers and 4 heads. The first encoded output of each object sequence is used for representing the entire object.

$$\mathbf{h}_{pre}^k = \mathbf{O}_{encoder}(\hat{\mathbf{o}}_{pre}^k) \quad (2)$$

3.1.2 Action Encoder

Actions are encoded either as a symbolic triplet (action, action object, action receptacle) or as an annotated text describing an action being taken (e.g., “robot empties the cup”).

During pre-training, the Action Encoder $\mathbf{A}_{pretrain}$ uses an action embedding layer \mathbf{E}' to embed the first dimension of the action, and re-uses the object embedding layer \mathbf{E} to embed the action object name a_o and action receptacle name a_r . The action embedding layer \mathbf{E}' has dimensionality $10 \times h$ for the 10 distinct actions. The three embedded representations are summed and passed to the Action Encoder’s linear layers to produce \mathbf{h}_a (see equation 3). Similarly to Zellers et al. (2021), a tanh activation is applied after each linear layer.

$$\mathbf{h}_a = \mathbf{A}_{pretrain}(\mathbf{E}'(\mathbf{a}) + \mathbf{E}(a_o) + \mathbf{E}(a_r)) \quad (3)$$

When fine-tuning on the annotated dataset, the action input is text and therefore we switch out the Action Encoder $\mathbf{A}_{pretrain}$ for $\mathbf{A}_{finetune}$ —our text-based Action Encoder. $\mathbf{A}_{finetune}$ uses a RoBERTa-base¹ model (Liu et al., 2019) to process a tokenized version of the text input \mathbf{a}_t . The first token ([CLS]) of the RoBERTa output layer is used to represent the action sequence and then passed through a linear layer to map the dimensionality of the hidden states from 256 to h .

$$\mathbf{h}_a = \mathbf{A}_{finetune}(\mathbf{a}_t) \quad (4)$$

¹Implementation and pre-trained model weights are taken from the Huggingface library (Wolf et al., 2019).

3.1.3 Vision Object Encoder

The Vision Object Encoder takes in images (\mathbf{i}_{pre} and \mathbf{i}_{post}) to provide a visual representation of each object k before and after (\mathbf{h}_{pre}^k and \mathbf{h}_{post}^k). We use the DETR¹ (Carion et al., 2020) model as a backbone to predict N bounding boxes in a pair of images (pre- and post-action). As DETR is pre-trained on the COCO object detection dataset (Lin et al., 2014), its predicted object labels do not align with those in PIGPeN. Therefore, we instead learn a mapping between the predicted bounding box representations and the PIGPeN objects. For each image, we obtain a hidden representation \mathbf{h}_b of dimensionality $N \times 256$ where $N = 100$.

We use an attention mechanism over the bounding boxes’ hidden representation, conditioned on the object names. For a given object o^k , its conditional representation \mathbf{h}_c^k is the encoded name of the object: $\mathbf{E}(o_{name}^k)$. We can therefore obtain the attention score of a given object o^k and image \mathbf{i}_m by calculating the alignment between the conditional representation \mathbf{h}_c^k and the hidden representations of bounding boxes \mathbf{h}_{b_m} :

$$\mathbf{h}_{b_m} = \text{DETR}(\mathbf{i}_m) \quad (5)$$

$$\alpha_m^k = \text{Softmax} \left(\sum_{i=1}^h (\mathbf{h}_c^k \mathbf{h}_{b_m})_i \right) \quad (6)$$

We obtain the final representation for a given object and image by multiplying our attention scores α with the extracted output representation from DETR and summing along the bounding box axis:

$$\mathbf{h}_{o_m}^k = \mathbf{W} \left(\sum_{j=1}^b (\alpha_m^k \mathbf{h}_{b_m})_j \right) \quad (7)$$

We use a final output layer \mathbf{W} to decrease the dimensionality of \mathbf{h}_o from the DETR dimensionality of 256 to h .

Through the Vision Object Encoder, we replace the previously symbolic inputs with images and can extract $[\mathbf{h}_{pre}^0 \mathbf{h}_{pre}^1]$ and $[\mathbf{h}_{post}^0 \mathbf{h}_{post}^1]$ from \mathbf{i}_{pre} and \mathbf{i}_{post} respectively. Note that we make the implicit assumption that \mathbf{i}_{pre} and \mathbf{i}_{post} contain the information necessary to predict object attributes of the objects post-action.

3.1.4 Action Apply

The Action Apply Model β is a simple fuse operation (concatenation in the hidden dimension) followed by three linear layers, which combine

the action representation \mathbf{h}_a and an object representation of the scene pre-action \mathbf{h}_{pre}^k . The model outputs an object’s representation \mathbf{h}_a^k , containing information conditioned all inputs:

$$\mathbf{h}_a^k = \beta(\mathbf{h}_a, \mathbf{h}_{pre}^k) \quad (8)$$

3.2 Object Decoder

Finally, the Object Decoder is a transformer module that maps the object representations h_o from the pre-action state back to 38 symbolic attributes. It uses a default three layer Transformer Decoder (Paszke et al., 2019) that takes the hidden representation from the Action Apply \mathbf{h}_a^k as an encoded memory state and \mathbf{h}_{pre}^k as the source sequence to predicts a label for each attribute.

$$\dot{\mathbf{o}}_{post}^k = \mathbf{O}_{decoder}(\mathbf{h}_a^k, \mathbf{h}_{pre}^k) \quad (9)$$

When we use image inputs, we also have access to the post-action visual representation and can therefore use $\mathbf{h}_{pre}^k + \mathbf{h}_{post}^k$ instead of \mathbf{h}_{pre}^k .

The output has post-action object states $\dot{\mathbf{o}}_{post}^k$ which are compared to the ground truth \mathbf{o}_{post}^k to calculate cross-entropy. As an additional loss, we also use the cross-entropy between $\dot{\mathbf{o}}_{pre}^k$ and \mathbf{o}_{pre}^k by passing an empty \mathbf{h}_a^k to force the Object Decoder to recreate the attributes in the pre-action state. We weight both losses equally.

3.3 Evaluation Metrics

Since our task involves predicting 38 attributes for two different objects per example, we follow Zellers et al. (2021) and report different types of accuracy metrics on the test set (after fine-tuning). We measure the overall accuracy by scoring how many objects have all attributes correctly predicted (exact match). Note that this is a high bar for a model where the symbolic representations are latent: to predict an object correctly, our model must first estimate its attributes before the action and then estimate whether and how these change given an action. So we also measure the attribute-level and action-level accuracies of each model, so as to explore which attributes and actions are more difficult to predict than others.

3.4 PIGPeN-Vis Dataset Split

To evaluate physical commonsense reasoning using PIGLeT-Vis, we filter PIGPeN (Zellers et al., 2021) to create a subset (PIGPeN-Vis) which we use for all our experiments. We motivate PIGPeN-Vis as

a way to isolate the effects of adding our vision component, because while PIGPeN already has images, these images were not used in PIGLeT.

The PIGPeN dataset consists of trajectories of an environment before (*pre*) and after (*post*) an action is taken. Each trajectory contains representations of two distinct objects before and after. One of the objects is usually targeted by the action, while the other acts as a distractor. In addition, image pairs $(\mathbf{i}_{pre}, \mathbf{i}_{post})$ for each trajectory are provided, where each image is snapshot of the simulated photo-realistic 3D environment which contains the objects in view (see Appendix B for an example). Each image is an RGB image of dimensions 640×385 .

The original dataset is separated into two distinct sets:

1. A *pre-training* set of 278,009 trajectories, which includes the symbolic representations of objects \mathbf{o} before and after a symbolic action \mathbf{a} is taken. A separate validation set of 33,042 examples is also included.
2. A *fine-tuning* set of 1,000 trajectories which has been annotated to replace the symbolic action \mathbf{a} with a textual representation \mathbf{a}_t describing the action. Separate validation and test sets of 500 examples each are also included. All metrics are reported on the test set.

In PIGPeN, the object states \mathbf{o}_{pre} and \mathbf{o}_{post} contained 40 different attributes and 13 different actions \mathbf{a} . Attributes range from intrinsic such as name or moveable to stateful such as distance or isCooked. In forming PIGPeN-Vis, we remove two attributes and three actions from the dataset to obtain 38 attributes and 10 possible actions (see Appendix B for more details).

3.4.1 Viewpoint and Action Filtering

Since the PIGPeN images were not generated with the goal of being used as input data, we identified several issues with the quality of certain scenes. A notable difficulty is that in some cases, the before and after images are not captured from the same camera angle or they have different lighting conditions. Changing orientations and lighting conditions makes it difficult to use an image pair $(\mathbf{i}_{pre}, \mathbf{i}_{post})$ to isolate the outcome of an action. Conversely, image pairs with too few perceivable differences also break our assumption that the changes in

the environment are perceivable. Therefore, we filter the dataset using pixel statistics to remove image pairs that have either large perceivable differences (likely due to changes in viewpoint) or small perceivable differences (where the action’s results are not visually salient enough) (see Appendix B.2). We exclude 15.4% of the total dataset through visual filtering of the original dataset.

3.4.2 Zero-Shot Filtering

To evaluate the generalization capabilities gained from a vision component, we further filter the dataset to exclude a subset of training examples. Unlike the original PIGPeN dataset which only tested for zero-shot generalization at the level of the fine-tuning data, we remove all instances with selected specific objects or action-object pairs from all training and validation sets. To minimize the effect of removing examples from the dataset, we pick objects and action-object pairs with an already low number of samples in the training sets. In total, we exclude 14 objects and 27 action-object pairs, which amounts to less than 3% (6,816 samples) of the remaining training sets (see Appendix B.3). These zero-shot examples comprise around 10% of the test set.

After both filtering stages, PIGPeN-Vis contains a pre-training dataset of 232,625 trajectories with a validation set of 26,823, and a fine-tuning training set of 750 examples with a validation set of 367 examples and a test set of 398 examples.

3.5 Training Configurations

We evaluate the impact of the vision component on PIGPeN-Vis through five different setups:

- **base:** We implement a baseline model without symbolic object inputs. Our implementation removes the Object Encoder entirely, such that the model must predict the attributes of objects solely from knowing the action and the object names that it relates to. This model acts as a lower bound on the capabilities of the vision model: its performance would match the vision model if images are irrelevant to solving the task.
- **base+symbolic:** This is our implementation of the original Zellers et al. (2021) PIGLeT model, shown in Figure 1. This model acts as an upper bound on the capabilities of the vision model since it observes the true symbolic

representations of objects before the action (which the vision model must estimate).

- **base+images:** This is our proposed PIGLeT-Vis, shown in Figure 2, where the Vision Object Encoder replaces the previously symbolic Object Encoder. This model leverages the before and after images of the environment as well as the name of the objects to extract representations of the object attributes.
- **base+symbolic+images:** We sum the hidden symbolic representations of objects with their visual representations in a unified model. Through this setup, we evaluate whether images can provide additional information to the already comprehensive symbolic representations.
- **base+images+text-labels:** We convert the symbolic representations of the labels for the object names and actions to their text label and encode them using a frozen LLM during pre-training. We use the same LLM to encode the text labels that we later use in the fine-tuning stage. This setup replaces all symbolic inputs from the pre-training stage to only language and image inputs.

Note that there are a few differences between the original Zellers et al. (2021) model and our implementation of base+symbolic. For instance, for simplicity, we opted to use an off-the-shelf RoBERTa-base (Liu et al., 2019) model instead of training our own custom GPT2 (Radford et al., 2019). Additionally, we also reduce the dimensionality of the PIGLeT layers from $h = 256$ to $h = 64$. We found that not only does this allow faster training times as it shrinks the Physical Dynamics model from 11.9 million parameters to 2 million parameters, it also improves the overall accuracy by a small margin (+1.51%).

We train each model for 80 epochs with a batch size of 256 using the Pytorch implementation of the Adam optimizer (Kingma and Ba, 2014) and a learning rate of 10^{-3} during pre-training and 10^{-5} during fine-tuning. We run each setup over 10 different seeds and report the average and standard deviation for each metric (see Appendix C.1 for more details).

	Accuracy (% \pm σ)	
	Overall	Zero-Shot
base	21.23 \pm 0.72	5.34 \pm 2.77
base+symbolic (PIGLeT)	85.03 \pm 0.45	39.04 \pm 3.37
base+symbolic+images	86.01 \pm 0.89	35.89 \pm 3.47
base+images (PIGLeT-Vis)	45.47 \pm 1.50	7.53 \pm 2.60
base+images+text-labels	47.55 \pm 2.10	8.90 \pm 3.24

Table 1: Overall and zero-shot accuracies (PIGPeN-Vis)

4 Results and Discussion

We evaluate all models on our PIGPeN-Vis split and report the overall (exact match), zero-shot, action-level, and attribute-level accuracy results for all setups in Tables 1 and 2. For completeness, we also evaluate models on the original PIGPeN to contrast the effects of our filtering operations (see §3.4 and Appendix D) and find PIGPeN-Vis is a more challenging subset for all models.

The base model provides a low bar estimate of what is achievable using only the action encoder inputs. Unsurprisingly, the base model performs worst on overall accuracy, which demands an exact match of all attributes. It does relatively well on (individual) attribute-level accuracy, primarily because it predicts the most common attribute for each object. Some actions are also easier than others—for instance, the model reaches 27.38% accuracy on ToggleOn from only knowing the action and object names. This is likely because ToggleOn is constrained to a small set of objects and effects.

Our base+symbolic model obtains similar results to the original implementation by Zellers et al. (2021), with an overall accuracy of 85.03%. However, it performs much worse on the zero-shot split (39.04%) than the original PIGLeT model reported (80.2%) (Zellers et al., 2021). This disparity can be explained by the fact that the original zero-shot PIGPeN dataset was not a true zero-shot dataset, because the Physical Dynamics model was exposed to the “unseen” objects in its pre-training. The base+symbolic model provides a high bar estimate of what could be achievable if: (i) i_{pre} and i_{post} capture the symbolic environment; and (ii) the Vision Object Encoder can subsequently extract these features. However, as we will argue in Section 6, both (i) and (ii) are unrealistic given the constraints of both the dataset and the model.

Our base+images (PIGLeT-Vis) model scores 45.28% in overall accuracy but only 7.53% on the zero-shot set. Nevertheless, it outperforms the base model in overall accuracy ($p < 0.0001$) and in zero-shot accuracy ($p = 0.08$), which demon-

strates that the images improve the prediction of the effects of actions. The base+images model also performs significantly better than base on difficult attribute-level accuracies such as distance ($p < 0.0001$). However, as before, accuracy on individual attributes benefits from the skewed distributions of their values and does not necessarily translate to high scores on predicting all 38 attributes correctly.

Utilizing both images and symbolic representations as inputs helps the base+symbolic+images model outperform purely symbolic inputs in overall accuracy, from 85.03% to 86.01% ($p < 0.01$). However, image inputs also decrease the model’s zero-shot performance from 39.04% to 35.89%, although this isn’t statistically significant ($p = 0.05$) due to high variance. We suspect that this high variance is caused by an increase in noise in the model resulting from adding images to the symbolic model. However, the overall picture is more complicated, as images can also provide gains on certain actions (e.g., PickUp accuracy increases from 80.48% to 86.14%) even though it causes a decrease in many other cases (e.g., ToggleOn).

Finally, when we utilize NL descriptions to replace the formal symbolic inputs (action name and object names), base+images+text-labels improves overall accuracy when compared to base+images from 45.47% to 47.55% ($p = 0.02$). Text inputs appear to improve zero-shot accuracy, but not by a statistically significant margin ($p = 0.31$). Accuracy also improves in most actions, for instance the Slice accuracy improves from 41.64% to 45.57% ($p = 0.03$). So the NL descriptions inform the task in a beneficial way, over and above the raw images. But encoding the labels as text rather than formal symbolic representations also adds noise.

Nevertheless, text labels improve accuracy on actions where the semantic information contained in the label provides a richer context to help generalize to similar objects. For instance, a “cup” and a “mug” are semantically close, and thus learning the effects of actions on a “cup” might help the model predict the same effects on a “mug” even if the word forms are different. In contrast, the formal symbolic representations treat the predicate symbols cup and mug as unrelated, and so don’t benefit from the lexical relationships that the LLM captures. Fully removing the symbolic representations allows us to adapt our model

	Action Accuracy (%)				Attribute Accuracy (%)		
	Open	Pickup	ToggleOn	Slice	size	distance	temperature
base	8.33	10.96	27.38	22.13	73.78	51.01	95.91
base+symbolic (PIGLeT)	85.73	80.48	96.90	75.41	94.98	95.13	99.85
base+symbolic+images	88.75	86.14	92.86	81.31	96.35	96.13	99.59
base+images (PIGLeT-Vis)	20.83	33.49	70.24	41.64	87.03	76.62	96.10
base+images+text-labels	22.92	40.12	67.14	45.57	87.89	78.06	96.72

Table 2: Action and attribute specific accuracies for a subset of actions and attributes; for a comprehensive table with standard deviations see Appendix D. size and distance each have eight possible classes while temperature has three.

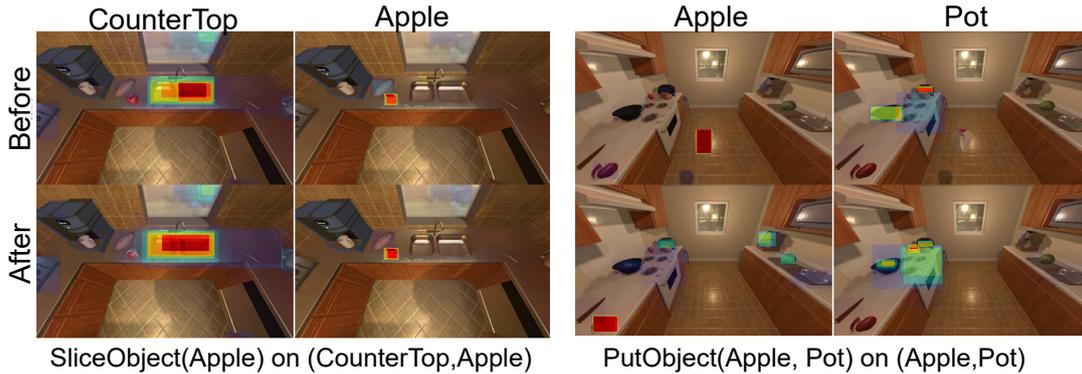


Figure 3: We visualize the attention of the Vision Object Encoder from a trained base+images model on two different actions and environments. The left grid focuses on the effect of Slice(Apple) on CounterTop and Apple, while the right grid focuses on the effects of Slice(Apple) on Apple and Pot objects.

to any possible unseen object during test time. base+images+text-labels is adaptable to general settings without knowing the symbolic mapping of objects and actions in the environment.

The results of both base+symbolic+images and base+images+text-labels make the case multi-modal modeling of commonsense reasoning, as both language and images are complementary to generalize to unseen settings.

4.1 Qualitative Attention Maps

Visualizing attention is another benefit of a vision component, as we can see what the model focuses on and partially explain its predictions. Figure 3 shows two separate examples and corresponding attention maps. In the left example, base+images is tasked with predicting the attributes of CounterTop and Apple after the Slice action is applied on the Apple. In the right example, the Put action is applied on the Apple, and the model must predict the attributes of the Apple and the distractor object Pot. The two rows are the before and after images (i_{pre} and i_{post}), and the two columns are the two objects used to condition the attention. The attention maps display the strength of the attention for each bounding box given an object name.

Both examples in Figure 3 show that the Vision

Object Encoder can map known objects to relevant bounding boxes. The model successfully tracks the Apple in both cases by placing the most weight on the bounding box targeting the Apple. However, these examples also show the difficulty of this task—the environments are realistic and can be filled with more than one instance of an object.

5 Conclusion

In this paper, we tackle the task of predicting the effects of actions on objects’ physical attributes. In contrast to (Zellers et al., 2021), our model does not treat the formal symbolic representation of the images as observed. Instead, PIGLeT-Vis supports inference when the inputs are images alone or images plus NL descriptions and a phrase denoting the action (e.g., “the robot empties the cup”). While PIGPeN offers challenges for applying a multi-modal approach, our model can extract useful information from images, opening the door for generalizing learning physical commonsense to real-world data. Importantly, our PIGPeN-Vis split can be used to evaluate the zero-shot capabilities of different model configurations. Moreover, while base+symbolic still outperforms base+images, it does so without estimating the attributes of ob-

jects and thus solves a much easier but unrealistic task. Through `base+images+text-labels`, we show that, when replacing symbolic inputs, the best solution is to complement image inputs with NL descriptions to leverage information from both modalities. Finally, our results show the need to improve the generalization capabilities of multimodal models such that they can learn and adapt to unseen situations.

6 Limitations

There are several limitations to our approach that result directly from the inherent limitations of PIGPeN and our proposed Vision Object Encoder respectively.

PIGPeN was not originally designed for testing commonsense reasoning using images and contains numerous inconsistencies which cannot all be solved with the PIGPeN-Vis split obtained from filtering (Section 3.4.1). Given the presence of non-physically salient attributes such as temperature, images are not guaranteed to fully capture their symbolic representations. PIGPeN includes certain attributes which are not discernible from images, e.g., even humans would be unable to tell a hot plate from a cold plate from vision alone. The images in PIGPeN can also contain more than one object (e.g., more than one cup) without ever specifying which one the symbolic representation refers to. This causes difficulty for our approach because judging specific attributes such as distance is impossible if there are two cups at different distances from the viewpoint. Additionally, PIGPeN also discretizes continuous variables such as distance into categories which can be hard to disambiguate.

To approach the accuracy of `base+symbolic` with our vision component, we also need a vision representation from which to correctly estimate all latent attributes. Even if images are assumed to be perfect representations of the symbolic environment, the model still has to extract each of the 38 attributes correctly for both objects using only two images. It is possible (and likely) for the vision detection backbone to miss the target object entirely because it is not trained to detect the specific object in question. We see this effect in Figure 3, where the model falls back to using a bounding box around the sink area to describe the CounterTop object. The DETR vision model used to extract bounding boxes was pre-trained on the COCO dataset (Lin et al., 2014) which does not

contain CounterTop as an object. PIGLeT-Vis is therefore ultimately limited by the capabilities of its vision backbone.

Ethics Statement

While this work does not introduce new data or involve human participants, we use the PIGPeN dataset which contains human-labelled data. The fine-tuning portion of the dataset was annotated through MTurk by Zellers et al. (2021) and they report following best practices (paying decent wages, providing feedback and using a qualification test) in their data collection. We filter and use a subset of PIGPeN and introduce methods to learn the effects of actions in a multimodal setting. We, therefore, believe that our work does not raise any ethical concerns.

Acknowledgements

This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) at the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences and by the UKRI-funded TAS Governance Node (grant number EP/V026607/1).

References

- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, page 1877–1901. Curran Associates, Inc.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey

- Zagoruyko. 2020. [End-to-end object detection with transformers](#). *CoRR*, abs/2005.12872.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. [Uniter: Universal image-text representation learning](#). In *ECCV*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. [Do neural language representations learn physical commonsense?](#) In *CogSci*.
- Qiaozhi Gao, Shaohua Yang, Joyce Chai, and Lucy Vanderwende. 2018. [What action causes this? towards naive physical action-effect prediction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 934–945, Melbourne, Australia. Association for Computational Linguistics.
- Michael Hanna, Federico Pedeni, Alessandro Suglia, Alberto Testoni, and Raffaella Bernardi. 2022. [ACT-thor: A controlled benchmark for embodied action understanding in simulated environments](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5597–5612, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. [Deep residual learning for image recognition](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. [\(comet-\) atomic 2020: On symbolic and neural commonsense knowledge graphs](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392.
- Diederik Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#). *International Conference on Learning Representations*.
- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2020. [What does BERT with vision look at?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5265–5275, Online. Association for Computational Linguistics.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. [Microsoft COCO: Common objects in context](#). In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. [Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. [BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance](#). In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). *arXiv:2103.00020 [cs]*. ArXiv: 2103.00020.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI blog*.
- Shailaja Keyur Sampat, Akshay Kumar, Yezhou Yang, and Chitta Baral. 2021. [CLEVR_HYP: A challenge dataset and baselines for visual question answering with hypothetical actions over images](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3692–3709, Online. Association for Computational Linguistics.
- Ke Shen and Mayank Kejriwal. 2021. [On the generalization abilities of fine-tuned commonsense language representation models](#). In *Artificial Intelligence XXXVIII*, page 3–16. Springer International Publishing.

Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. 2021. Factorizing perception and policy for interactive instruction following. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1888–1897.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [CommonsenseQA: A question answering challenge targeting commonsense knowledge](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [Huggingface’s transformers: State-of-the-art natural language processing](#). *CoRR*, abs/1910.03771.

Da Yin, Liunian Harold Li, Ziniu Hu, Nanyun Peng, and Kai-Wei Chang. 2021. [Broaden the Vision: Geo-Diverse Visual Commonsense Reasoning](#). In *EMNLP*.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. In *AAAI*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [From recognition to cognition: Visual commonsense reasoning](#). In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 6713–6724. IEEE.

Rowan Zellers, Ari Holtzman, Matthew Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. 2021. Piglet: Language grounding through neuro-symbolic interaction in a 3d world. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

A GPT-3 Example of Physical Reasoning

The weight of the potato is 150 grams.
 The robot then slices the potato into thin slices.
 The weight of the potato is now 75 grams.

Figure 4: Example of incorrect physical commonsense by an LLM. When predicting what comes after the **input text**, the large 175 billion parameter GPT-3 (Brown et al., 2020) predicts that the weight of the potato halves after a slicing action is taken.

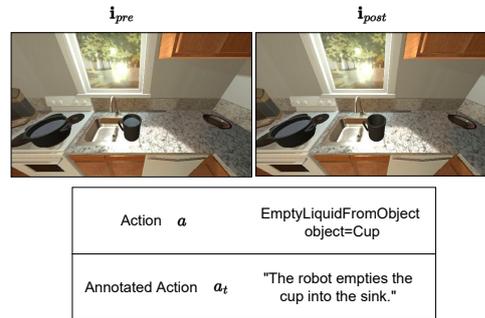


Figure 5: Image pair and actions for a selected PIGPeN example.

	<i>pre</i>		<i>post</i>	
	\mathbf{o}_{pre}^{cup}	$\mathbf{o}_{pre}^{faucet}$	\mathbf{o}_{post}^{cup}	$\mathbf{o}_{post}^{faucet}$
ObjectName	Cup	Faucet	Cup	Faucet
Contained Objects				
Is contained in...				
Mass	1 to 2lb	Massless	1 to 2lb	Massless
Size	small	medium	small	medium
Temperature	RoomTemp	RoomTemp	RoomTemp	RoomTemp
Distance	1 to 2ft	3 to 4 ft	1 to 2ft	3 to 4 ft
Breakable	Yes	No	Yes	No
Cookable	No	No	No	No
CanBecomeDirty	Yes	No	Yes	No
IsBroken	No	No	No	No
IsCooked	No	No	No	No
IsDirty	No	No	No	No
IsFilledWithLiquid	Yes	No	No	No
IsOpen	No	No	No	No
IsPickedUp	Yes	No	Yes	No
IsSliced	No	No	No	No
IsToggled	No	No	No	No
Moveable	No	No	No	No
Openable	No	No	No	No
Pickupable	Yes	No	Yes	No
CanHoldItems	Yes	No	Yes	No
Sliceable	No	No	No	No
Toggleable	No	Yes	No	Yes
Materials	Ceramic		Ceramic	

Table 3: Attributes for a selected PIGPeN example. The total number of attributes is 38 as the Materials attribute is a multi-hot encoding.

B PIGPeN-Vis

We select an example from PIGPeN to display in Figure 5 and Table 3.

From the original dataset, we remove two attributes (`isUsedUp` and `salientMaterials_Organic`) because they are unchanged in all examples. We also remove 3 actions (`ThrowObject10`, `ThrowObject100` and `ThrowObject1000`) which are all related to throwing an object across a certain distance. These actions account for only a small subset of the dataset and create inconsistent image pairs due to the agent’s momentum being captured in the images. The angle of the camera changes as a result of `ThrowObject` and this breaks our assumption that the difference between \mathbf{i}_{pre} and \mathbf{i}_{post} solely reflects the effects of the action on the environment (and not on the viewer). We therefore reduce the total number of symbolic attributes per object to 38 and the number of possible actions to 10.

B.1 Attributes

The following 38 symbolic attributes are used to describe an object in PIGPeN:

ObjectName,	parentReceptacles,
receptacleObjectIds,	distance, mass, size,
ObjectTemperature,	breakable, cookable,
dirtyable, isBroken,	isCooked, isDirty,
isFilledWithLiquid,	isOpen, isPickedUp,
isSliced, isToggled,	moveable, openable,
pickupable, receptacle,	salientMaterials_Ceramic,
salientMaterials_Fabric,	salientMaterials_Food,
salientMaterials_Glass,	salientMaterials_Leather,
salientMaterials_Metal,	salientMaterials_Paper,
salientMaterials_Plastic,	
salientMaterials_Rubber,	salientMaterials_Soap,
salientMaterials_Sponge,	salientMaterials_Stone,
salientMaterials_Wax,	salientMaterials_Wood,
sliceable, toggleable	

B.2 Filtering Statistics

We initially filter the PIGPeN dataset using two main strategies to remove images with too much or too little change between the pre and post images. In both cases, the goal is to remove pairs of images in which it would be impossible for a vision model to predict what has changed.

Images with too many changes are often images taken from different viewpoints or with different lighting conditions. We filter these images by look-

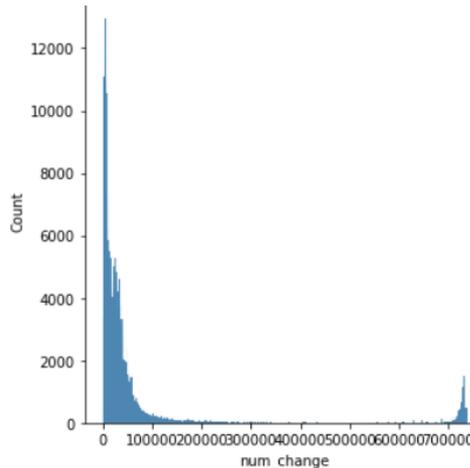


Figure 6: Distribution of the number of pixels changed per image in the PIGPeN dataset.

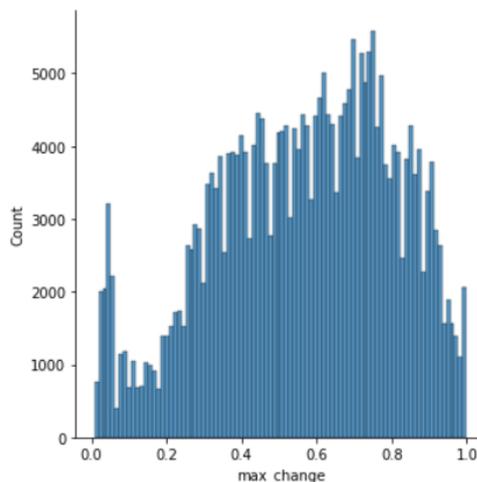


Figure 7: Distribution of the maximum pixel value changed per image in the PIGPeN dataset.

ing at the number of pixels changed between \mathbf{i}_{pre} and \mathbf{i}_{post} . We show the distribution of the number of pixels changed per image over the training dataset in Figure 6. Using this visualization we can clearly see a small peak at the extreme - where almost all the pixels in \mathbf{i}_{post} are different from \mathbf{i}_{pre} . Note that since each image is an RGB image of dimensions 640×385 , the max number of change is $640 \times 385 \times 3 = 739,200$ (we also compare pixels across color channels). We opt to remove all images with more than 400,000 changes, which corresponds to around 6.2% of the training dataset.

Images with too little change could be examples of where the action has no visual outcome and \mathbf{i}_{pre} and \mathbf{i}_{post} are indistinguishable. To filter these images we measure the maximum magnitude of change in each pixel and each color channel be-

tween the pairs of images. We visualize the max change across the training dataset in Figure 7. Here a low values implies almost no salient change, and as max change approaches zero - it becomes unlikely that a human would be able to perceive the difference between the pair of images. We opt for to keep images with a max change greater than 0.2 which corresponds to excluding 7.8% of the training dataset.

Filtering on the number of changed pixels lead to the exclusion of around 13.89% of the training dataset.

B.3 Zero-shot Filtering

We remove the following 14 objects from both the train and validation (3, 401 examples total):

HandTowel, Towel, Plunger, Watch, CD, SoapBottle, Pen, RemoteControl, SoapBar, Box, Bottle, CreditCard, Statue, KeyChain

We remove the following 27 action-object pairs from both the train and validation (3, 278 examples total):

(CloseObject, Toilet),
 (DirtyObject, Pan), (DirtyObject, Pot),
 (EmptyLiquidFromObject, Bottle),
 (EmptyLiquidFromObject, Pot), (OpenObject, Toilet),
 (PickupObject, Box), (PickupObject, CellPhone),
 (PickupObject, CreditCard),
 (PickupObject, KeyChain), (PutObject, CD),
 (PutObject, CreditCard), (PutObject, HandTowel),
 (PutObject, Laptop), (PutObject, Lettuce),
 (PutObject, Pen), (PutObject, Plunger),
 (PutObject, Pot), (PutObject, RemoteControl),
 (PutObject, SoapBar), (PutObject, SoapBottle),
 (PutObject, Statue), (PutObject, ToiletPaper),
 (PutObject, Towel), (PutObject, Watch),
 (ToggleOff, CellPhone), (ToggleOff, Television)

C Code Release and Training

Our full code, models, and PIGPeN-Vis split can be found at github.com/gautierdag/piglet-vis.

C.1 Additional Training Details

As previously mentioned, there are a few differences between the original Zellers et al. (2021) model and our implementation of base+symbolic. We use an off-the-shelf RoBERTa-base (Liu et al., 2019) model instead of a custom GPT2 (Radford et al., 2019). Additionally, we also reduce the dimensionality of the PIGLeT layers from $h = 256$ to $h = 64$. This shrinks the overall model (ex-

cluding the LLM) from 11.9 million parameters to less than 2 million parameters during pre-training and improves the overall accuracy by a small margin (+1.51%). We do not run any other hyperparameter search throughout our experiments and wherever possible use the same hyper-parameters as PIGLeT. We also reduce the batch size from 1024 to 256 because we use a mix of NVIDIA GTX 1080 and NVIDIA A100 GPUs and wish to keep batch size constant.

The +images models use the extracted representations from a frozen off-the-shelf DETR model (41.3 million parameters), however it is ran only once over all images as we cache its predictions. We do not use the “NO OBJECT” predictions from DETR, and simply pass all 100 bounding boxes representations to the attention mechanism. Since we do not have access to the true bounding boxes in PIGPeN, we do not fine-tune DETR and therefore ignore its prediction heads which have also been trained on COCO and mismatch our possible objects.

The +symbolic models use the Symbolic Object Encoder which is an additional 800, 000 parameters on its own. During fine-tuning all models use a RoBERTA-base model (+120 million parameters) in the Action Encoder. The +text-label model also uses the RoBERTA-base model during pre-training, but again this is frozen and its outputs are cached for the full dataset.

We pre-train each model for 80 epochs and fine-tune for 60 epochs. For all setups, pre-training takes between 1 to 2 hours and fine-tuning takes less than 1 hour on an NVIDIA A100 GPU. We use the Pytorch implementation of the Adam optimizer (Kingma and Ba, 2014) and a learning rate of 10^{-3} during pre-training and 10^{-5} during fine-tuning. We use early stopping on the validation loss with a patience of 10 epochs. We run each setup over 10 different seeds ($s \in [1, 2, \dots, 10]$) and report the average and standard deviation for each metric.

D Accuracy Results

D.1 Comparing PIGPeN and PIGPeN-Vis

Table 4 compares the overall accuracy on the original PIGPeN dataset with our proposed PIGPeN-Vis split. We find that our PIGPeN-Vis split is consistently harder to solve than the original PIGPeN dataset. We explain the increased accuracy in the original dataset with the fact that some of the filtered out actions (see Appendix B) are easy to

	Overall Accuracy (% $\pm \sigma$)		
	PIGPeN	PIGPeN-Vis	Δ
base	29.18 \pm 0.34	21.23 \pm 0.72	-7.95%
base+symbolic (PIGLeT)	86.39 \pm 0.79	85.03 \pm 0.45	-1.36%
base+symbolic+images	87.45 \pm 0.66	86.01 \pm 0.89	-1.44%
base+images (PIGLet-Vis)	49.13 \pm 1.53	45.47 \pm 1.50	-3.66%
base+images+text-labels	51.28 \pm 1.68	47.55 \pm 2.10	-3.73%

Table 4: Overall Accuracies comparing full PIGPeN with the PIGPeN-Vis split across 10 seeds.

	Overall Accuracy (% $\pm \sigma$)	
	validation	test
base	23.85 \pm 0.95	21.23 \pm 0.72
base+symbolic (PIGLeT)	88.08 \pm 0.50	85.03 \pm 0.45
base+symbolic+images	89.49 \pm 0.82	86.01 \pm 0.89
base+images	50.73 \pm 2.97	45.47 \pm 1.50
base+images+text-labels	53.33 \pm 3.15	47.55 \pm 2.10

Table 5: Validation and test overall accuracies. Note the zero-shot accuracy is not calculated on the validation set since there are no unseen examples in the validation set to prevent leakage.

solve from knowing the object name and action: e.g., most of the images we exclude due to little salient changes are appliances like stoves being turned on or off. However, it is easy for a model to predict the post-condition attributes of a stove, which are mostly static, across all examples given an action such as ToggleOn, which always has the same effect.

D.2 Complete Accuracy Results on PIGPeN-Vis

Table 5 shows the overall accuracies for both the test and validation sets. The full accuracy results for all actions in Table 6 and for all attributes in Table 7.

E Additional Attention Maps

We plot additional attention visualizations for all three image models base+images, base+symbolic+images, and base+images+text-labels in Figures 8, Figures 9, and Figures 11. Since the DETR object detector remains frozen, all models have access to the same bounding boxes and bounding box representations. Qualitatively, we find that the attention weights of base+images and base+images+text-labels both learn to map to globally relevant bounding boxes given an objects. We also find the attention maps in base+images+text-labels to be less confident overall than base+images, likely due to the noise introduced by the semantic text inputs. As a result, base+images+text-labels makes less mistakes

by not focusing too much attention to the wrong bounding box.

On the other hand, base+symbolic+images focuses on seemingly random bounding boxes. Since base+symbolic+images already receives the full representation of each objects, it does not learn to complement the object’s representation with accurate visual information. While base+symbolic+images extracts 1% of additional overall accuracy from image inputs when compared to base+symbolic, it does so by falling back to vision for visually salient actions such as Pickup. base+symbolic+images focuses only a narrow set bounding boxes with overconfidence with no regard for whether or not the bounding box relates to the object. We posit that the model might use vision to better estimate more difficult attributes to predict such as distance in some contexts. Note Pickup is a salient action because when the agent in the environment picks an object up, the object is placed directly in the middle of its field of vision (as if the agent were holding the object in front of it).

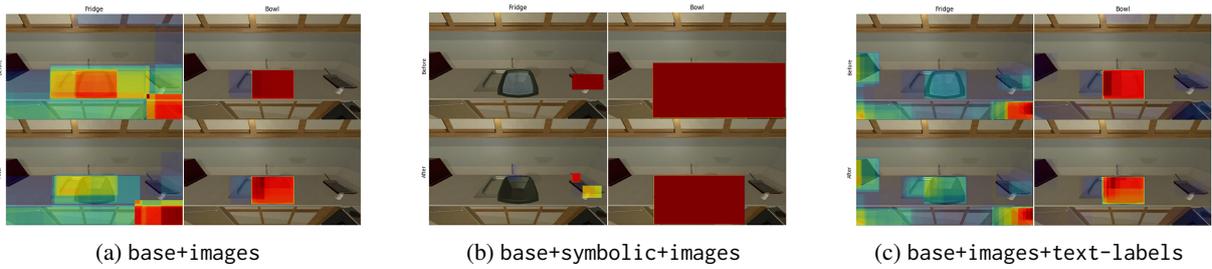


Figure 8: Attention maps for the effects of the EmptyLiquid action on Bowl with objects Fridge and Bowl. The top row of each grid maps to the before environment and the bottom row maps to the after environment. The columns map to each respective object. The Fridge object appears in the lower left of the image, and is only correctly identified by base+images+text-labels, even though the model does place more weight to the bounding box of the stove (lower right).

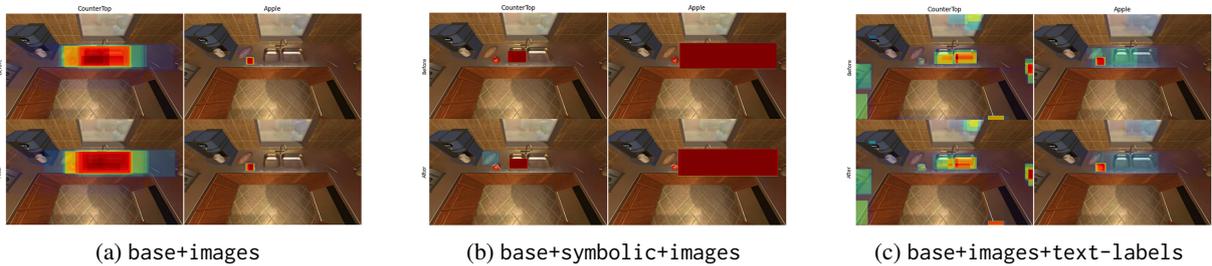


Figure 9: Attention maps for the effects of the Slice action on Apple with objects CounterTop and Apple. The top row of each grid maps to the before environment and the bottom row maps to the after environment. The columns map to each respective object.

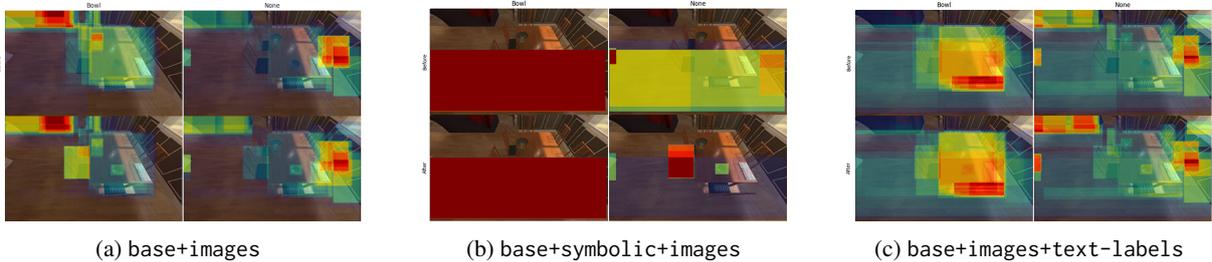


Figure 10: Attention maps for the effects of the Dirty action on Bowl with objects Bowl and None. The top row of each grid maps to the before environment and the bottom row maps to the after environment. The columns map to each respective object. None can be an object in PIGPeN, but we do not predict its attributes and exclude it in all model predictions.

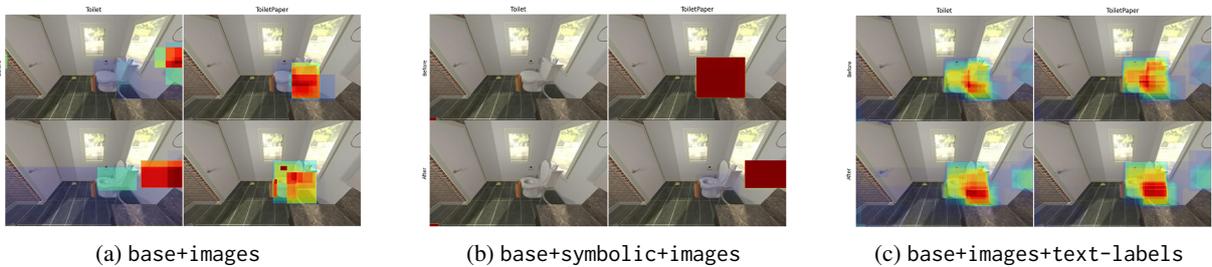


Figure 11: Attention maps for the effects of the Open action on Toilet with objects Toilet and ToiletPaper. The top row of each grid maps to the before environment and the bottom row maps to the after environment. The columns map to each respective object. This particular set example is an unseen combination of action and object that has been excluded from the training and validation set.

Action Accuracy (% \pm σ)					
	Close	Dirty	EmptyLiquid	HeatUpPan	Open
base	13.20 \pm 1.06	17.71 \pm 1.20	24.75 \pm 5.75	36.33 \pm 4.14	8.33 \pm 1.84
base+symbolic	85.98 \pm 1.77	94.00 \pm 3.42	99.34 \pm 1.15	100.00 \pm 0.00	85.73 \pm 0.99
base+symbolic+images	86.80 \pm 3.29	90.29 \pm 5.90	99.02 \pm 2.07	99.17 \pm 1.62	88.75 \pm 3.02
base+images	27.42 \pm 3.71	58.57 \pm 2.78	69.34 \pm 4.17	68.67 \pm 3.75	20.83 \pm 4.63
base+images+text-labels	28.87 \pm 3.19	57.71 \pm 3.24	70.16 \pm 3.17	74.00 \pm 3.16	22.92 \pm 5.79
	Pickup	Put	Slice	ToggleOff	ToggleOn
base	10.96 \pm 1.92	27.95 \pm 1.19	22.13 \pm 0.86	30.83 \pm 3.39	27.38 \pm 2.57
base+symbolic	80.48 \pm 2.88	58.39 \pm 1.94	75.41 \pm 1.89	99.40 \pm 0.84	96.90 \pm 1.61
base+symbolic+images	86.14 \pm 2.56	57.59 \pm 2.31	81.31 \pm 3.96	99.05 \pm 0.75	92.86 \pm 5.14
base+images	33.49 \pm 3.45	34.91 \pm 2.43	41.64 \pm 3.80	71.43 \pm 2.75	70.24 \pm 16.00
base+images+text-labels	40.12 \pm 2.61	38.30 \pm 3.11	45.57 \pm 3.85	69.05 \pm 5.81	67.14 \pm 16.53

Table 6: Full accuracy results table including the standard deviation over 10 seeds for all actions and setups.

Attribute Accuracy (% \pm σ)										
Name	Temperature	attribute	breakable	cookable	dirtyable	distance	isBroken	isCooked	isDirty	
base	99.66 \pm 0.07	95.91 \pm 0.41	96.12 \pm 0.07	91.46 \pm 0.36	99.95 \pm 0.07	99.95 \pm 0.10	51.01 \pm 0.93	99.86 \pm 0.00	98.60 \pm 0.06	97.93 \pm 0.19
base+symbolic	99.64 \pm 0.12	99.85 \pm 0.04	99.48 \pm 0.03	99.84 \pm 0.09	100.00 \pm 0.00	100.00 \pm 0.00	95.13 \pm 0.35	100.00 \pm 0.00	99.85 \pm 0.04	99.71 \pm 0.14
base+symbolic+images	99.62 \pm 0.09	99.59 \pm 0.27	99.48 \pm 0.04	99.78 \pm 0.10	100.00 \pm 0.00	99.97 \pm 0.09	96.13 \pm 0.40	100.00 \pm 0.00	99.85 \pm 0.04	99.52 \pm 0.32
base+images	97.34 \pm 0.65	96.28 \pm 0.74	97.25 \pm 0.13	92.63 \pm 0.75	99.91 \pm 0.10	99.62 \pm 0.20	76.90 \pm 1.05	99.85 \pm 0.05	98.68 \pm 0.19	97.87 \pm 0.34
base+images+text-labels	98.44 \pm 0.35	96.05 \pm 1.23	97.46 \pm 0.13	93.19 \pm 0.31	99.96 \pm 0.09	99.93 \pm 0.10	78.56 \pm 1.16	99.84 \pm 0.09	98.19 \pm 0.84	97.78 \pm 0.24
	isFilledWithLiquid	isOpen	isPickedUp	isSliced	isToggled	mass	moveable	openable	parentReceptacles	pickupable
base	96.79 \pm 0.50	98.84 \pm 0.23	94.83 \pm 0.82	97.99 \pm 0.09	98.36 \pm 0.23	96.51 \pm 0.15	99.90 \pm 0.09	99.97 \pm 0.06	87.44 \pm 0.42	99.84 \pm 0.09
base+symbolic	99.93 \pm 0.12	98.95 \pm 0.09	99.27 \pm 0.31	100.00 \pm 0.00	99.88 \pm 0.12	99.33 \pm 0.14	99.99 \pm 0.04	99.97 \pm 0.06	97.78 \pm 0.47	99.90 \pm 0.11
base+symbolic+images	99.84 \pm 0.19	98.67 \pm 0.38	98.96 \pm 0.31	99.97 \pm 0.06	99.74 \pm 0.30	99.59 \pm 0.09	100.00 \pm 0.00	99.99 \pm 0.04	97.26 \pm 0.44	99.88 \pm 0.10
base+images	96.88 \pm 0.55	98.81 \pm 0.97	97.43 \pm 0.37	98.28 \pm 0.30	97.92 \pm 0.83	96.41 \pm 0.41	99.79 \pm 0.21	99.74 \pm 0.20	91.05 \pm 0.77	99.59 \pm 0.17
base+images+text-labels	97.25 \pm 0.45	98.11 \pm 1.14	97.54 \pm 0.53	98.34 \pm 0.29	98.06 \pm 0.55	96.74 \pm 0.24	99.89 \pm 0.09	99.95 \pm 0.10	92.49 \pm 0.69	99.70 \pm 0.09
	receptacleIds	receptacle	Ceramic	Fabric	Food	Glass	Leather	Metal	Paper	Plastic
base	84.20 \pm 0.61	99.85 \pm 0.10	98.26 \pm 0.17	99.55 \pm 0.07	99.99 \pm 0.04	98.91 \pm 0.13	99.89 \pm 0.06	98.69 \pm 0.15	99.73 \pm 0.00	98.30 \pm 0.10
base+symbolic	96.36 \pm 0.18	99.90 \pm 0.09	100.00 \pm 0.00	99.96 \pm 0.07	100.00 \pm 0.00	99.99 \pm 0.04	100.00 \pm 0.00	99.99 \pm 0.04	100.00 \pm 0.00	99.97 \pm 0.06
base+symbolic+images	96.13 \pm 0.30	99.92 \pm 0.10	99.99 \pm 0.04	99.85 \pm 0.10	99.99 \pm 0.04	99.97 \pm 0.06	100.00 \pm 0.00	100.00 \pm 0.00	100.00 \pm 0.00	99.96 \pm 0.07
base+images	82.87 \pm 0.55	99.47 \pm 0.21	99.03 \pm 0.22	99.50 \pm 0.19	99.92 \pm 0.10	99.16 \pm 0.21	99.97 \pm 0.06	98.31 \pm 0.37	99.67 \pm 0.21	98.83 \pm 0.31
base+images+text-labels	83.91 \pm 0.56	99.69 \pm 0.11	99.36 \pm 0.19	99.44 \pm 0.12	99.96 \pm 0.09	99.37 \pm 0.24	99.95 \pm 0.10	98.69 \pm 0.30	99.56 \pm 0.19	99.08 \pm 0.20
	Rubber	Soap	Sponge	Stone	Wax	Wood	size	sliceable	toggleable	
base	100.00 \pm 0.00	99.99 \pm 0.04	100.00 \pm 0.00	99.34 \pm 0.09	100.00 \pm 0.00	99.51 \pm 0.16	73.78 \pm 0.29	98.02 \pm 0.12	99.95 \pm 0.07	
base+symbolic	100.00 \pm 0.00	100.00 \pm 0.00	100.00 \pm 0.00	99.99 \pm 0.04	100.00 \pm 0.00	99.99 \pm 0.04	94.98 \pm 0.19	100.00 \pm 0.00	99.99 \pm 0.04	
base+symbolic+images	99.97 \pm 0.06	99.99 \pm 0.04	100.00 \pm 0.00	99.99 \pm 0.04	100.00 \pm 0.00	100.00 \pm 0.00	96.35 \pm 0.20	99.99 \pm 0.04	99.96 \pm 0.09	
base+images	99.88 \pm 0.08	99.89 \pm 0.11	99.92 \pm 0.10	99.48 \pm 0.14	99.92 \pm 0.10	99.25 \pm 0.22	87.03 \pm 1.15	98.32 \pm 0.32	99.81 \pm 0.17	
base+images+text-labels	99.85 \pm 0.08	99.92 \pm 0.10	99.88 \pm 0.14	99.60 \pm 0.19	99.95 \pm 0.07	99.37 \pm 0.22	87.89 \pm 1.11	98.32 \pm 0.36	99.95 \pm 0.07	

Table 7: Full accuracy results table including the standard deviation over 10 seeds for all attributes and setups.

FVQA 2.0: Introducing Adversarial Samples into Fact-based Visual Question Answering

Weizhe Lin

Department of Engineering
University of Cambridge
United Kingdom
wl356@cam.ac.uk

Zhilin Wang

Department of Linguistics
University of Washington
United States
zhilinw@uw.edu

Bill Byrne

Department of Engineering
University of Cambridge
United Kingdom
bill.byrne@eng.cam.ac.uk

Abstract

The widely used Fact-based Visual Question Answering (FVQA) dataset contains visually-grounded questions that require information retrieval using common sense knowledge graphs to answer. It has been observed that the original dataset is highly imbalanced and concentrated on a small portion of its associated knowledge graph. We introduce FVQA 2.0 which contains adversarial variants of test questions to address this imbalance. We show that systems trained with the original FVQA train sets can be vulnerable to adversarial samples and we demonstrate an augmentation scheme to reduce this vulnerability without human annotations.

1 Introduction

Knowledge-based Visual Question Answering (KB-VQA) lies at the intersection of Computer Vision, Natural Language Processing, and Information Retrieval. A KB-VQA system must access external knowledge sources to find a correct and complete answer, a task that is sometimes hard for humans.

Fact-based Visual Question Answering (FVQA) (Wang et al., 2017) is a VQA task in which visually-grounded questions and answers about images are grounded by knowledge-graph (KG) triplets taken from several ‘common sense’ knowledge bases, such as ConceptNet (Speer et al., 2017), Webchild (Tandon et al., 2017), and DBpedia (Auer et al., 2007). For instance, “Question: Which thing in the image can be used for scooping food? Answer: spoon” is associated with the KG triplet “spoon - UsedFor - scooping food”. These questions are challenging in that retrieving information from external KGs is necessary.

The original FVQA dataset (Wang et al., 2017) has several readily observed limitations. First, the dataset is small (5486 samples) and the annotations are limited to a single answer per question, ignoring other correct answers. This limitation arises

from the FVQA creation process in which annotators were first asked to select a KG triplet on which they would ask a question about an image. This approach prevented the annotators from labeling other valid KG triplets. Secondly, the dataset is highly imbalanced. Some triplets and answers are frequently used, but other KG triplets and answers are severely underrepresented in training. For example, there are 1,129 possible answers in total, but over 90% of questions focus on only a half of them; 792 (70%) answers appear less than 3 times; only 4,216 out of ~220k triplets are used.

These limitations lead to a potential problem: KB-VQA systems trained on this dataset overfit on these frequently used triplets and perform poorly on variants that contain other valid triplets or other images. Also, extensive overlap between training and test can lead to an unrealistically high question answering baseline performance. We noted that a question with a triplet unseen in training is often answered with ‘person’, since it is the most frequent answer in the original data distribution.

To overcome these limitations, we introduce an enlarged test set that contains two types of adversarial samples (as shown in Fig. 1): (1) *FixQ*: the question remains the same, but is associated with a different image and a different correct answer. This ensures that a system is less able to achieve high performance if it is biased by language patterns in questions; (2) *FixA*: the answer remains the same, but the question is asked in a different way. This favours systems that do more than make straightforward associations between questions and answers based on the training data. In contrast to the original test set, this new set further challenges KB-VQA system to retrieve knowledge from KBs and answer questions without being biased towards frequent answers in the original dataset. We show that models trained on the original FVQA training sets are significantly less robust on these adversarial test samples.

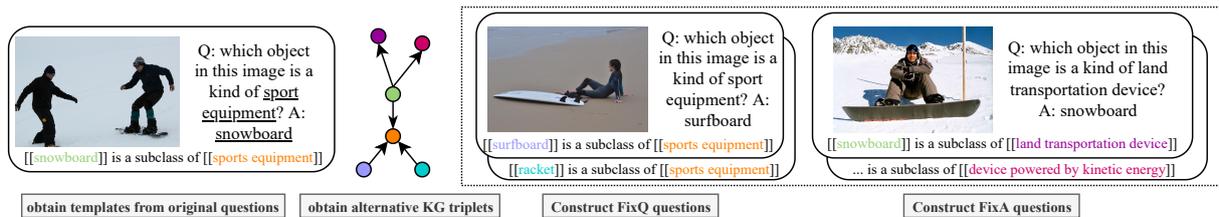


Figure 1: The workflow of constructing adversarial samples (FixQ and FixA questions) from the original test set questions.

Given that it is hard to guarantee a good triplet coverage during annotation, we explore an augmentation scheme to address this problem without costly human annotation of large-scale adversarial training samples. Our scheme generates slightly noisy adversarial samples that improve the coverage of valid KG triplets to enhance model training.

Our contributions are:

(1) We introduce FVQA 2.0, which adds an adversarial test set that challenges KB-VQA system robustness to adversarial variants of questions.

(2) We demonstrate the performance gap between the original test set and the adversarial test set, showing that considering adversarial samples is important for better realistic KB-VQA performance.

(3) To further demonstrate the importance of adversarial samples, we leverage a semi-automated augmentation scheme to improve system robustness on the adversarial test through the creation of large-scale noisy adversarial examples.

2 Related Work

KB-VQA questions can focus on facts and concepts, as in FVQA (Wang et al., 2017) and OK-VQA (Marino et al., 2019). Such questions challenge the information retrieval ability of systems. KB-VQA questions can also require common-sense reasoning, as in parts of OK-VQA and A-OKVQA (Schwenk et al., 2022). In particular, S3VQA (Jain et al., 2021) is an augmented version of OKVQA, improving both the quantity and quality of some question types. A-OKVQA has shifted its core task to “reasoning questions”. Only 18% of questions in A-OKVQA require answers from an external knowledge base.

VQA 2.0 (Goyal et al., 2017) collects ‘complementary images’ such that each question is associated with a pair of images that result in different answers. Jain et al. (2021) derive new S3VQA questions from manually defined question templates. They annotated spans of objects that could be replaced, and then substituted them with a com-

plicated substitute-and-search system. In contrast to their labour-intensive annotation work, our adversarial samples are collected through a semi-automatic approach that fully leverages the structural information in KGs to significantly reduce the human work required.

More broadly, in Knowledge-Graph Question Answering (KG-QA), work has exploited KG to generate synthetic data in unseen domains (Linjordnet, 2020; Trivedi et al., 2017; Linjordnet and Balog, 2020). Our work extends visually-grounded questions with valid common sense KG triplets.

3 Method

Extracting Question Templates. We extract question templates that can be used to reconstruct new questions using other valid KG triplets. We apply a rule-based system to replace KG entities that appear in the questions. For example, ‘what is used for storing liquid in this image?’ is transformed to ‘what is used for <t> in this image?’ given that the associated KG triplet is “bottle (<h>) - /r/UsedFor (<r>) - store liquid (<t>)”.

For each template, we construct new question-answer pairs by exploring the node structure of the KG. For example, “bottle - /r/UsedFor - hold water” is also a valid triplet from ConceptNet, whose head and relation are the same as the original triplet. A new question “Q: what is used for holding water in this image? A: bottle” can now be constructed.

Template Filtering. We focus on questions about object concepts that are transferable to other images, ignoring a small portion (<10%) of FVQA questions to which the answers are based on particular scenes (e.g. ‘what can you often find in the place shown in this picture?’).

Human annotators are employed to filter out non-transferable templates, such as questions that contain specific object positioning (“what is the object in the lower right of this image used for?”). This process takes around 1 hour with two annotators to obtain 440 valid templates after removing highly similar templates.

Matching Suitable Images. We use 619 of FVQA images¹ that are also present in the Visual Genome dataset (Krishna et al., 2017). Using the object annotations of the VG dataset to determine if an image contains the object being asked, we employ a rule-based system to assign a suitable image to each generated adversarial sample, within which process all images are assigned to approximately the same number adversarial samples by a simple approach described in Appendix B. We limit the number of FixQ and FixA questions generated by each template to 5, which guarantees a reasonable dataset size. 3,805 questions are generated.

Manual Verification. We conduct manual verification to rule out samples that are incorrectly generated. 432 counter-intuitive KG triplets are removed in this step. Finally, we obtain 2,820 adversarial samples, offering 1,671 new valid triplets from the KG. Around 75% samples are verified as correct, showing that the rule-based generation works well. The remainder are discarded.

The official FVQA evaluation performs 5-fold validation: each split preserves around half its samples for testing. As a naming convention, under each split, the templates extracted from the original training samples are called ‘train templates’ while the rest are ‘test templates’. Since the train templates may contain language patterns that have been learned in training, we ensure that only questions derived from test templates are used in the adversarial testing. As a result, we have 1,376 adversarial test samples per split on average, with 1,129 FixQ and 246 FixA questions.

Augmentation with Adversarial Data. We explore an augmentation scheme to augment the training data with slightly noisy but auto-generated adversarial samples, which avoids heavy annotation work. In each split, **only the train templates** (defined in the above paragraph) are used to generate adversarial samples for training such that no information of test samples is leaked to training. This avoids biasing the training to the test sets, which would make the test sets less indicative of true system performance. We obtain an augmentation set with 2,262 questions per split on average semi-automatically, which would otherwise cost hundreds of hours to build from scratch. The origins of these adversarial samples are referred to as ‘*Originating Questions*’. There are 435 such

questions per split. In training, these questions are randomly replaced by their adversarial variants.

4 FVQA 2.0 Statistics

Set Name	#Samples	std
Standard Train Set	2,927	69
Standard Test Set	2,899	69
Originating Questions Set	435	52
Adversarial Test Set	1,376	193
- FixA Questions	1,129	157
- FixQ Questions	246	38
Augmentation data	2,262	267

Table 1: Dataset Statistics. #Samples: average number of samples across 5 folds; std: the standard deviation over 5 folds.

The numbers of samples in each set are provided in Table 1. The official FVQA dataset creates 5 folds by splitting the images being used. Half of these images are used in training while the other half are reserved for testing. In all our new sets, under each split, questions for training are not leaked to testing. The ‘Originating Question Set’ is a subset of Standard Test Set by its definition (Sec. 3). The Adversarial Test Set is formed by FixA questions and FixQ questions; it is created by automatically generating adversarial question variants from the questions in the ‘Originating Question Set’. It covers relationships such as */r/RelatedTo*, */r/IsA*, */r/PartOf*, */r/HasA*, */r/UsedFor*, */r/CapableOf*, */r/AtLocation*, */r/Desires*, */r/MadeOf*. The augmentation data consists of adversarial variants that are derived from the questions in the Standard Train Set.

5 Experiments

Baseline Systems We use several FVQA systems for comparison²: FVQA (Wang et al., 2017), the baseline system provided in the official FVQA dataset paper; GCN (Narasimhan et al., 2018), a model that leverages graph convolutional networks (GCNs) to aggregate features from visual/language/fact modalities; Mucko (Zhu et al., 2020), the current state-of-the-art system that uses GCNs to combine visual, fact, and semantic graphs.

We test our augmentation scheme on several systems that have code available: **RAVQA-NoDPR**

¹FVQA images are from Microsoft COCO (Lin et al., 2014) and ILSVRC (Russakovsky et al., 2015).

²Since many recent FVQA systems are not open-sourced, we additionally include KB-VQA systems from OKVQA.

and **RAVQA-DPR** (Lin and Byrne, 2022), T5 (Rafel et al., 2020)-based models that transform images into texts (e.g. objects, attributes, and image captions) and the DPR version additionally uses Dense Passage Retrieval (Karpukhin et al., 2020) to retrieve documents from knowledge bases³; **TRiG** (Gao et al., 2022), a model that is similar to RAVQA-DPR but different in embedding fusion; **ZS-F-VQA** (Chen et al., 2021), an FVQA system that obtains the final prediction by fusing the individual predictions in answer/fact/relation graphs.

Metrics. We report accuracy and standard deviation over 5 splits (Sec. 4). In calculating accuracy for open-ended generation systems (RAVQA/TRiG), a question is considered successfully answered if the generated answer string is an exact match to the ground-truth answer node, which is the closest KG node to the ground-truth answer string (shortest in Levenshtein distance computed from node names).

Performance and Discussion. Table 2 shows that the systems used for evaluating the new adversarial set are sufficiently strong (e.g. 69.56% accuracy by RAVQA-DPR) in comparison with the three models that do not have code available, which achieve 58.76% (FVQA), 69.35% (GCN), and 73.06% (Mucko, current state-of-the-art) respectively. RAVQA-NoDPR achieves 84.59% accuracy on the originating questions but obtains only 71.48% accuracy on the adversarial samples derived from them. Such performance gaps are readily observed on all systems. Systems trained on the original training sets fail to perform equally well on the two sets, showing that the original FVQA training data does not contain adversarial variants and the resulting systems are vulnerable to them.

By incorporating adversarial variants in training, all systems achieve much better performance on the challenging adversarial set, e.g. RAVQA-NoDPR is improved from 71.48% to 82.38% (+10.9%). The performance on the standard and adversarial test sets now match well, with the gap reduced from more than 10% to \sim 3%, showing that the augmentation scheme significantly improves systems’ reliability and robustness. The relative improvement is slightly less (+8.1%) for RAVQA-DPR, which is expected given that it is a retrieval-based system designed to answer both seen and unseen questions with its strong retrieval ability. ZS-F-VQA benefits

³In our experiments, the knowledge base consists of surface texts of triplets (e.g. “[car] has [4 wheels]”).

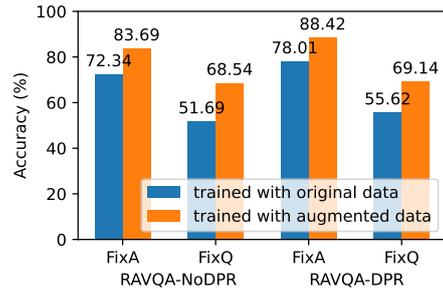


Figure 2: Performance on FixQ and FixA questions.

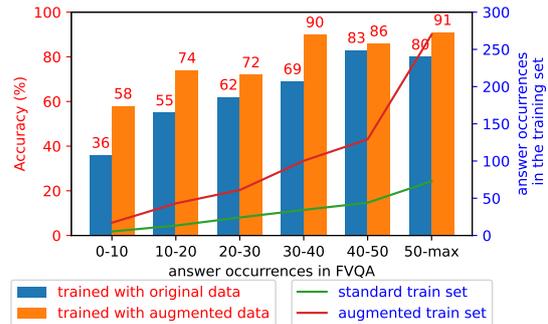


Figure 3: RAVQA-DPR accuracy on adversarial questions and answer occurrences in the standard/augmented training sets. They are grouped by the number of answer occurrences in the original FVQA dataset. For example, a question is counted towards the ‘0-10’ group if its answer appears less than 10 times in the original dataset.

greatly from augmentation: its adversarial performance is improved by 24.09%. This is because its model size is much smaller and it can easily be biased by language patterns, images, and frequent answers seen in training.

In summary, systems trained on the original training sets are vulnerable to adversarial variants of the test questions. We show that through generating adversarial samples for data augmentation, systems become much more robust to these variants.

Analysis of Model Vulnerability. As shown in Fig. 2, RAVQA systems trained with original training sets perform better on FixA questions (\sim 88%) than on FixQ questions (\sim 69%). This suggests that systems perform worse when asked the same questions on different images. This is potentially because the language patterns seen in training bias the models to frequent choices, lowering the FixQ generalizability. In contrast, systems are less distracted by different ways of asking for the same answer, potentially due to the strong language modelling capability of T5 used by them. The augmentation scheme improves systems on both types of questions significantly (by \sim 10% on each), showing the value of adversarial samples in training.

Test on:	Standard Test Set		Originating Question Set		Adversarial Test Set	
Trained on:	Original	Augmented	Original	Augmented	Original*	Augmented (improv. over *)
ZS-F-VQA	48.16 \pm 1.03	48.57 \pm 1.00	63.67 \pm 0.88	64.63 \pm 0.81	49.97 \pm 2.37	74.06 \pm 1.92 +24.09
TRiG	64.94 \pm 0.93	65.73 \pm 0.33	81.67 \pm 1.12	83.48 \pm 1.89	68.86 \pm 3.26	79.79 \pm 1.34 +10.93
RAVQA-NoDPR	66.19 \pm 1.15	66.70 \pm 1.00	84.59 \pm 1.24	85.75 \pm 0.90	71.48 \pm 2.08	82.38 \pm 1.65 +10.90
RAVQA-DPR	69.56 \pm 0.78	69.90 \pm 0.56	87.52 \pm 1.68	88.33 \pm 1.40	76.91 \pm 1.93	85.05 \pm 1.15 +8.14

Table 2: Model performance on the standard test set, originating questions (from which the adversarial questions are derived), and adversarial test set. Results are reported as the average of 5 folds with standard deviations.

Fig. 3 plots the RAVQA-DPR performance on the adversarial test set questions that are grouped by their answer occurrences in the original FVQA dataset. The answer distribution of the original dataset affects adversarial performance greatly: systems perform much worse on questions whose answers appear less frequently in FVQA. In contrast, the performance deterioration that arises from answer rarity is mitigated significantly after augmentation. The augmentation scheme (red v.s. green curve in Fig. 3) compensates for the imbalanced answer distribution by providing more question variants so that systems are trained on both popular and rare answers.

6 Conclusion

We show that the FVQA test sets are not sufficiently indicative of true system performance through providing a new human-verified adversarial test set that contains adversarial variants of the original test set questions. We show the value of adversarial samples in KB-VQA datasets by showing an augmentation scheme that leverages structural information in KGs to create augmentation questions for training, which improves models’ robustness to adversarial variants.

We release the dataset and the codes in Github (<https://github.com/LinWeizheDragon/Retrieval-Augmented-Visual-Question-Answering>).

7 Limitations

The adversarial test set was firstly generated from the original FVQA dataset by a rule-based system and then filtered by human annotators. As a result, the new set is limited with respect to the question types, language patterns, and knowledge triplets used in FVQA. One potential solution to overcome this limitation is to invest more human effort to generate adversarial questions from scratch, which is, however, much more expensive than the semi-automatic approach presented here.

The proposed augmentation approach also relies on the relationships encoded in the knowledge base (e.g. ConceptNet (Speer et al., 2017)). These will influence the quality and diversity of the augmented data, with the expectation that improvements in KG scope and quality will improve data augmentation.

The number of adversarial examples introduced in this work is sufficiently large for investigating the performance discrepancies (on the original and adversarial test sets) and demonstrating the necessity of KB-VQA adversarial samples. However, it is considered beneficial to introduce adversarial samples on a larger scale by considering them in the design of future KB-VQA datasets.

8 Ethics Statement

Our dataset was created semi-automatically from the FVQA dataset and ConceptNet, a crowd sourced common sense knowledge graph. Though we have included human annotators in the loop to remove sexual, offensive, and other inappropriate data samples that were automatically generated (we removed \sim 200 inappropriate knowledge graph triplets during annotation), we recognize that the dataset may still contain a small number of inappropriate samples. Any developers who replicate the semi-automatic methodology described in the paper to extend the datasets should include a similar review step in the manual work flow. We also recognize that the systems trained on this dataset may convey such inappropriate information to users in real-life applications. Therefore, extra care must be taken when using this dataset in applications that interact directly with real users.

9 Acknowledgement

W. Lin was supported by a Research Studentship funded by Toyota Motor Europe (RG92562(24020)). We thank our colleagues, Daniel Olmeda Reino (Toyota Motor Europe) and Jonas Ambeck (Toyota Motor Europe), who provided insight and expertise in this project.

We thank Alexandru Coca (University of Cambridge) for comments that greatly improved the manuscript. We would also like to thank all the reviewers for their knowledgeable reviews.

References

- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.
- Zhuo Chen, Jiaoyan Chen, Yuxia Geng, Jeff Z Pan, Zonggang Yuan, and Huajun Chen. 2021. Zero-shot visual question answering using knowledge graph. In *International Semantic Web Conference*, pages 146–162. Springer.
- Feng Gao, Qing Ping, Govind Thattai, Aishwarya Reganti, Ying Nian Wu, and Prem Natarajan. 2022. Transform-retrieve-generate: Natural language-centric outside-knowledge visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5067–5077.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913.
- Aman Jain, Mayank Kothiyari, Vishwajeet Kumar, Preethi Jyothi, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2021. Select, substitute, search: A new benchmark for knowledge-augmented visual question answering. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2491–2498.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.
- Weizhe Lin and Bill Byrne. 2022. Retrieval augmented visual question answering with outside knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11238–11254, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Trond Linnjordet. 2020. Neural (knowledge graph) question answering using synthetic training data. In *Proceedings of the 29th ACM International Conference on Information Knowledge Management, CIKM ’20*, page 3245–3248, New York, NY, USA. Association for Computing Machinery.
- Trond Linnjordet and Krisztian Balog. 2020. Sanitizing synthetic training data generation for question answering over knowledge graphs. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, pages 121–128.
- Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3195–3204.
- Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. 2018. Out of the box: Reasoning with graph convolution nets for factual visual question answering. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.
- Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. *arXiv preprint arXiv:2206.01718*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Niket Tandon, Gerard De Melo, and Gerhard Weikum. 2017. Webchild 2.0: Fine-grained commonsense knowledge distillation. In *Proceedings of ACL 2017, System Demonstrations*, pages 115–120.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218. Springer.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. 2017. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. *Transformers: State-of-the-art natural language processing*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zihao Zhu, Jing Yu, Yujing Wang, Yajing Sun, Yue Hu, and Qi Wu. 2020. Mucko: Multi-layer cross-modal knowledge reasoning for fact-based visual question answering. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 1097–1103. International Joint Conferences on Artificial Intelligence Organization. Main track.

A Training Details

ZS-F-VQA: The experiments were performed on $1 \times$ Nvidia RTX 3090. We used the code from the official repository⁴. The original paper dropped questions that have rare answers. For fair comparison with other models, we added these rare answers back and performed training and testing. We chose to report the performance of the system which uses ‘SAN’ as the base model (details are in the paper and the repository), since this setting has achieved the best performance. The hyperparameters for training are kept the same as the original paper. In testing, we selected $k_e = 10$; $k_r = 1$; $score = 10$ by grid search (search range: $0 \leq k_e \leq 20$; $0 \leq k_r \leq 20$; $0 \leq score \leq 20$).

⁴<https://github.com/China-UK-ZSL/ZS-F-VQA>

RA-VQA-NoDPR/RAVQA-DPR/TRiG: All experiments were performed on $1 \times$ Nvidia A-100 GPU. We chose Adam (Kingma and Ba, 2015) as the optimizer. When the model has a DPR component, we trained the DPR component for 4 epochs with a constant learning rate 10^{-5} . In training the answer generator, the learning rate linearly decays from 6×10^{-5} to 0 after 10 epochs, as suggested in the original paper. For each split, the checkpoints at global step 2k (around 3.5 epochs) were used in testing. We retrieve 5 best documents when predicting the answer ($K_{train} = 5$), since this number was reported to best balance the computation and performance (Lin and Byrne, 2022).

We obtained the pre-trained model parameters (T5-large and BERT-base) from Huggingface (Wolf et al., 2020). These systems are implemented with Huggingface Python libraries (under Apache License 2.0). The FAISS (Johnson et al., 2019) system is under MIT License.

B Balancing Images in Adversarial Variants

In assigning suitable images to question templates, it is necessary to ensure the diversity of images being used. We achieve this by controlling the number of assignments per image with a simple approach so that the numbers are approximately the same for all images.

In the process, for each new question-answer pair that needs an image, we rank all the images that contain the object being asked in the the question by their current total number of assignment. We select the image that satisfies the conditions as well as having the fewest number of assignment as the associated image of the new sample. We found that by applying this simple yet effective strategy, the assigned images present a good diversity.

C Annotation Details

Two annotators (volunteers in the research group) worked independently to rule out incorrectly generated examples. An example was accepted only if the two annotators achieved consensus. The annotators attempted to fix grammar errors that caused severe misunderstanding, while mild errors were kept (for example, ‘is used for carry people’ does not prevent models/people from understanding the question, and thus the annotators are not required to fix them).

In particular, questions that might contain information of individuals / private information were dropped, though it is a very rare case.

Questions with multiple answers: when a question can be answered with multiple instances in an image, all possible answers are included. During annotation, incorrect answers were dropped from the list. In evaluation, answering any correct answer is considered successful. There are around 11% multiple-answer questions at the end.

D Additional Results

We include some additional baseline performance in Table 3. It can be easily seen that the performance on originating questions (the original FVQA questions that are used to derive the adversarial samples) is very high even when images are excluded. This further supports our argument that the original dataset is heavily biased to frequent answers. The performance on the adversarial set is lower, showing that this new test set is more challenging and less biased toward language patterns.

E More Examples of FVQA 2.0

We demonstrate some more examples from the new Adversarial Test Set in Fig. 4.

Models	Standard Test Set	Originating Question Set	Adversarial Test Set
RAVQA-DPR	69.56 \pm 0.78	87.52 \pm 1.68	76.91 \pm 1.93
(without triplets)	66.19 \pm 1.15	84.59 \pm 1.24	71.48 \pm 2.08
(without images)	43.83 \pm 0.68	57.53 \pm 2.93	50.02 \pm 1.00
(without triplets and images)	40.29 \pm 1.60	51.41 \pm 3.25	42.55 \pm 0.90

Table 3: The performance of some additional baseline systems on the standard test set, originating questions (from which the adversarial questions are derived), and adversarial test set. Results are reported as the average of 5 folds with standard deviations.

Originating Question Set	Adversarial Test Set (FixA)
 <p>Question: which object in this image can hold liquid? Triplet: [[A glass]] can [[hold liquid]] Answer: glass</p>	 <p>Question: which object in this image can break easily? Triplet: [[glass]] can [[break easily]] Answer: glass</p>
 <p>Question: which object in this image is used for travel around town? Triplet: You can use [[a bus]] to [[travel around town]] Answer: bus</p>	 <p>Question: which object in this image is used for carry person? Triplet: [[A bus]] is used to [[carry people]] Answer: bus</p>
Originating Question Set	Adversarial Test Set (FixQ)
 <p>Question: which object in this image is hollow? Triplet: [[Tennis balls]] are [[hollow]] Answer: tennis ball</p>	 <p>Question: which object in this image is hollow? Triplet: [[a bowl]] is [[hollow]] Answer: bowl</p>
 <p>Question: which object in this image has a frame? Triplet: [[bicycle]] has [[frame]] Answer: bicycle</p>	 <p>Question: which object in this image has a frame? Triplet: [[A frame]] is part of [[a bed]] Answer: bed</p>

Figure 4: More examples taken from the FVQA 2.0 adversarial test set. The questions in the left column are from the official FVQA test set. They are used to derive the adversarial questions in the right column. FixA: the answer remains the same while the way of asking for the answer is different; FixQ: the question remains the same, but the answer changes in a different image. More details are presented in Sec. 1.

Revisiting Intermediate Layer Distillation for Compressing Language Models: An Overfitting Perspective

Jongwoo Ko¹
KAIST AI

Seungjoon Park¹
KAIST AI

Minchan Jeong¹
KAIST AI

Sukjin Hong²
KT

Euijai Ahn²
KT

Du-Seong Chang²
KT

Se-Young Yun¹
KAIST AI

¹{jongwoo.ko, sjoon.park, mcjeong, yunseyoung}@kaist.ac.kr

²{sukjin.hong, euijai.ahn, dschang}@kt.com

Abstract

Knowledge distillation (KD) is a highly promising method for mitigating the computational problems of pre-trained language models (PLMs). Among various KD approaches, Intermediate Layer Distillation (ILD) has been a *de facto standard* KD method with its performance efficacy in the NLP field. In this paper, we find that existing ILD methods are prone to overfitting to training datasets, although these methods transfer more information than the original KD. Next, we present the simple observations to mitigate the overfitting of ILD: distilling only the last Transformer layer and conducting ILD on supplementary tasks. Based on our two findings, we propose a simple yet effective consistency-regularized ILD (CR-ILD), which prevents the student model from overfitting the training dataset. Substantial experiments on distilling BERT on the GLUE benchmark and several synthetic datasets demonstrate that our proposed ILD method outperforms other KD techniques. Our code is available at <https://github.com/jongwooko/CR-ILD>.

1 Introduction

Recent advances in NLP have shown that using PLMs such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) on downstream tasks is effective. Although these models achieve state-of-the-art performances in various domains, the promising results of PLMs require numerous computation and memory costs. Deploying such large models on resource-constrained devices such as mobile and wearable devices is impractical. It is thus crucial to train computationally efficient small-sized networks with similar performance to that of large models.

KD is promising model compression technique where knowledge is transferred from a large and high-performing model (teacher) to a smaller model (student). KD has been shown to be reliable in reducing the number of parameters and

computations while achieving competitive results on downstream tasks. Recently, KD has attracted more attention in the NLP field, especially due to large PLMs. However, it is clear that the original KD (Hinton et al., 2015) is not performing well in terms of maintaining the performance of compressed PLMs and that it needs to have additional auxiliary training objectives (Sun et al., 2019; Jiao et al., 2020).

ILD methods (Jiao et al., 2020; Wang et al., 2020), which encourage the student model to extract knowledge from the Transformer layers of the teacher network, have demonstrated efficacy in improving student model performance and have become a *de facto standard* in KD. Despite of success of ILD methods, many research have been proposed to design layer mapping functions (Li et al., 2020; Wu et al., 2020) or new training objective (Park et al., 2021) to transfer the teacher’s knowledge better. These ILD methods transfer more knowledge to the student model from the intermediate Transformer layers of the teacher model. However, we find that the use of ILD in fine-tuning may induce performance degradation in some cases. As shown in Figure 1, while existing ILD methods such as TinyBERT (Jiao et al., 2020) and BERT-EMD (Li et al., 2020) work well on standard GLUE benchmark (Wang et al., 2019), we observe that these methods have performance degradation compared to original KD on ill-conditioned datasets such as those with few-samples and label noise. Because few-sample (Zhang et al., 2021) or heterogeneous datasets (Jin et al., 2021; Liu et al., 2022) can be easily found in real-world datasets, the existing ILD methods, which show performance reduction in Figure 1, are hard to use in real-world applications.

To mitigate such performance degradation, we identify the main problem as that intermediate Transformer knowledge can incur overfitting on the training dataset of the student model. We further

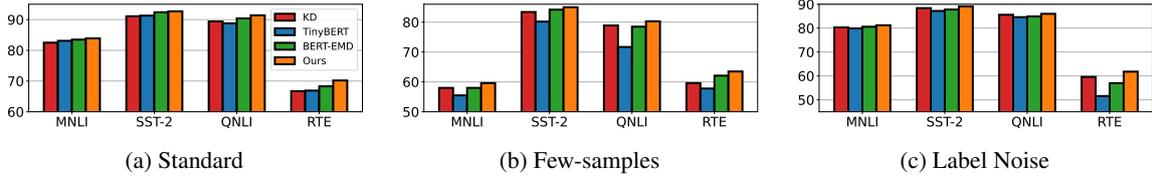


Figure 1: The motivation of our work. While existing ILD methods (Jiao et al., 2020; Li et al., 2020) work well on the standard GLUE benchmark (Wang et al., 2019), we observe that the existing ILD methods are problematic under few-samples training datasets or the presence of label noise. However, our proposed method shows robustly higher performance than the original KD for all datasets. We use BERT_{Small} (Turc et al., 2019) as the student model and BERT_{BASE} (Devlin et al., 2019) as the teacher model. The detailed descriptions for dataset are in Appendix B.

discover that distilling only the last Transformer layer knowledge and using supplementary tasks can alleviate the overfitting. Through our observations, we finally propose a simple yet effective method, consistency-regularized ILD (CR-ILD) with several analyses. Our main contributions are:

- We design and conduct comprehensive experiments to identify that overfitting is one of the main problems for performance degradation of ILD in fine-tuning. To the best of our knowledge, this is the first study to find that existing ILD methods have overfitting issues.
- Based on our findings, we propose the consistency regularized ILD (CR-ILD) that a student self-regularized itself from risk of overfitting from ILD. We further provide empirical (and theoretical) analyses for our proposed method.
- We experimentally demonstrate that our proposed method achieves state-of-the-art performance on both standard GLUE and ill-conditioned GLUE (few samples and label noise), despite its simplicity.

2 Related Works

Model Compression of LMs. Transformer encodes contextual information for input tokens (Vaswani et al., 2017). In recent years, from the success of Transformer, Transformer-based models such as GPT (Radford et al., 2018), BERT (Devlin et al., 2019), and T5 (Raffel et al., 2020) have become a new state of the arts, driving out recurrent or convolutional networks on various language tasks. However, the promising results of these models are accompanied by numerous parameters, which necessitate a high computation and memory cost for inference. Existing compression techniques can be categorized as low-rank matrix factorization (Mao et al., 2020), quantization (Bai et al., 2021), and KD (Sun et al., 2019).

Knowledge Distillation for LMs. KD is one of the most well-known neural model compression techniques. The goal of KD is to enable the student model with fewer parameters to achieve similar performance to that of the teacher model with a large number of parameters. In the recent few years, a wide range of different methods have been developed that apply data augmentation (Jiao et al., 2020; Liang et al., 2021), adversarial training (Rashid et al., 2021), and loss terms re-weighting (Jafari et al., 2021) to reduce the performance gap between the teacher and the student. In another line in the NLP field, ILD-based methods have exhibited higher effectiveness over original KD (Hinton et al., 2015) methods for compression PLMs. Sun et al. (2019) proposed the BERT-PKD to transfer representations of the [CLS] token of the teacher model. Jiao et al. (2020) proposed TinyBERT, which performed Transformer distillation in both pre-training and fine-tuning. Wang et al. (2020) distilled the self-attention module of the last Transformer layer of the teacher. Li et al. (2020) leveraged earth mover’s distance (EMD) to determine the optimal layer mapping between the teacher and student networks. Park et al. (2021) presented new KD objectives that transfer contextual knowledge via two types of relationships.

3 Observations: Two Things Everyone Should Know to Mitigate Overfitting

In this section, we identify that overfitting is the main problem for performance degradation while conducting ILD in fine-tuning. This overfitting problem can occur even in the standard GLUE benchmark. Moreover, the ill-conditioned dataset, where overfitting problems can occur more easily, induces a larger performance reduction. Furthermore, we investigate that this overfitting problem is able to be reduced by (1) distilling the last Transformer layer and (2) conducting ILD on supplemen-

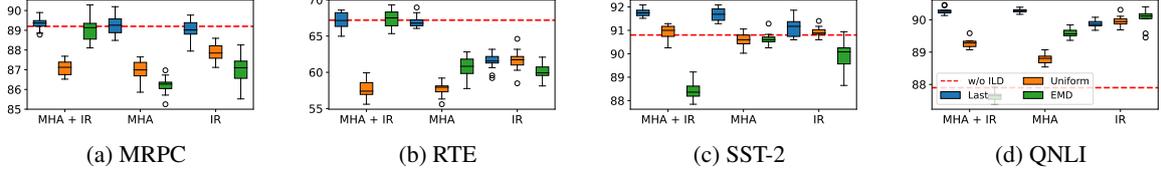


Figure 2: Performance distribution box plot across 20 random trials and the four datasets with different distillation methods. As the student model, we apply Truncated BERT (Sun et al., 2019) which initialized as the bottom 6 layers from BERT_{BASE}. Distilling knowledge of the last Transformer layer enhances generalization and reduces the variance of fine-tuning. The red-dotted lines are baseline performances that only use prediction layer KD.

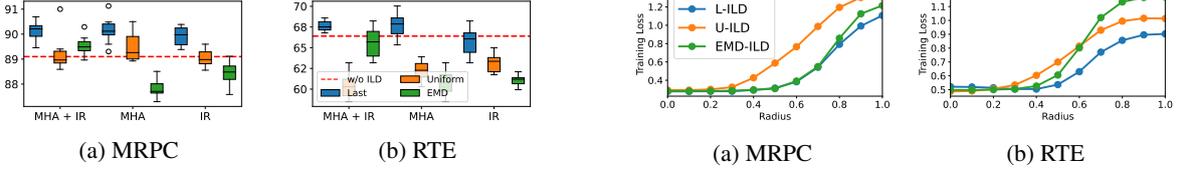


Figure 3: Performance distribution box plot across 20 random trials for MRPC and RTE when BERT_{Small} (Turc et al., 2019) is used as the student model. Well-pretrained student models have more consistent performance to the choice of layer mappings.

Figure 4: Training loss of different distillation approaches (L-ILD, U-ILD, EMD-ILD) with increasing Gaussian noise: models trained with L-ILD are more tolerant of noise, which proves that our L-ILD leads models to be more general.

tary tasks. While our suggested findings already have worked well in various domains (Wang et al., 2020; Phang et al., 2018), these previous works under-explored the effects of the techniques. However, this is the first work to use such techniques with empirical justification for mitigating overfitting problems.

Among the various ILD objectives, we focus on the two most commonly used distillation objectives: multi-head attention (MHA) and intermediate representations (IR). Formally, for the student’s layer $\ell^S \in [1, M]$, the loss function of MHA and IR are as follows:

$$\mathcal{L}_{\text{MHA}}^{\ell^S} = \frac{1}{A_h} \sum_{a=1}^{A_h} \text{KLD}(\mathbf{A}_{m(\ell^S),a}^T \| \mathbf{A}_{\ell^S,a}^S) \quad (1)$$

$$\mathcal{L}_{\text{IR}}^{\ell^S} = \text{MSE}(\mathbf{H}_{m(\ell^S)}^T, \mathbf{W}^H \mathbf{H}_{\ell^S}^S), \quad (2)$$

where $m(\cdot)$ is layer mapping function that returns teacher layer $m(\ell^S) \in [1, L]$. Note that KLD and MSE are Kullback-Leibler divergence and mean squared error, respectively. We denote \mathbf{A} and \mathbf{H} as MHA and IR. T and S are superscripts for the teacher and student model, and a and A_h indicate the index and the total number of multi-attention heads, respectively. Note that \mathbf{W}^H is a learnable weight matrix for matching the dimension between representations of the teacher and student. Consistent with previous studies (Sun et al., 2020; Jiao

et al., 2020), we observe that sequential training of ILD and original KD (Hinton et al., 2015) shows better than joint training of ILD and original KD. We conduct an experimental study on sequential training of ILD and original KD from our preliminary experiments. All the detailed descriptions of the scope of our empirical study are in Appendix C.1.

3.1 Layer Mapping: Distill Only the Last Transformer Layer

One of the biggest challenges of ILD methods is establishing a proper layer mapping function that determines layers of the teacher and student models to transfer knowledge. In this section, we observe that transferring layer-to-layer information leads student models to overfit training samples and is the primary reason for the degradation of student performance. Based on our findings, we suggest that the last layer distillation (Wang et al., 2020, 2021) is promising layer mapping method. Our empirical analyses can explain the suggested technique’s success in terms of mitigating overfitting.

Main Observations. We compare three distillation strategies: last Transformer layer distillation (L-ILD), layer-to-layer distillation using uniform layer mapping (U-ILD), and optimal many-to-

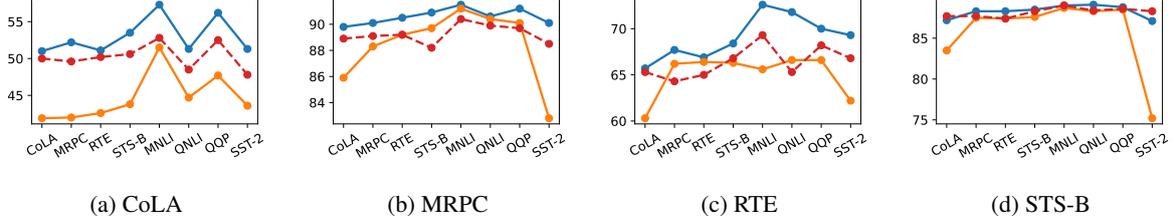


Figure 5: Comparisons for performance of ILD on different supplementary tasks. All students are $BERT_{Small}$, distilled MHA and IR from $BERT_{BASE}$ teachers with L-ILD (blue) and U-ILD (orange). We present the results of prediction layer KD on the supplementary tasks in red dotted lines. All results are averaged over 20 runs.

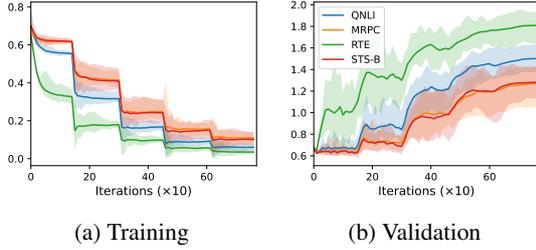


Figure 6: The mean (solid lines) and range (shaded region) of training and validation loss during fine-tuning BERT after conducting ILD on different supplementary tasks, across 20 random trials.

many layer mapping using the EMD (EMD-ILD) proposed in Li et al. (2020). In Figure 2, L-ILD (blue box) outperforms other baselines on all four datasets (MRPC, RTE, SST-2, QNLI) in terms of the test performance and variance reduction over the random trials. Note that the U-ILD, which is a commonly used mapping function (Sun et al., 2019; Jiao et al., 2020), leads to performance degradation in most fine-tuning tasks.

We conduct same experiments on the student model with different initialization ($BERT_{Small}$; Turc et al. 2019) as shown in Figure 3. We observe that L-ILD has a higher performance regardless of the size of the dataset or initialized point. However, the performance gap between L-ILD and other mapping functions gets smaller when the dataset size becomes larger, and the student model is well pre-trained. On the other hand, although EMD-ILD alleviates the difficulties in layer mapping between the teacher and student, it exhibits lower performance than L-ILD. We find that performances of EMD-ILD vary across the pre-trained methods while performances of L-ILD are not. These results validate that the inaccurate layer mapping between the intermediate Transformer layers is not the primary problem of ILD; instead, intermediate Transformer layer distillation itself is the main problem in the fine-tuning stage.

Analysis. To better understand about the performance degradation of distilling the knowledge of intermediate Transformer layers, we evaluate the generalizability of the student models of different layer mapping functions by following Zhang et al. (2019); Jeong and Shin (2020). We add Gaussian noise over $\mathcal{N}(0, \sigma^2 I)$ with different noise radius σ to the embedding vectors of the three models (L-ILD, U-ILD, EMD-ILD) and then evaluate their cross-entropy loss on the training set. More generalizable models are robust to the noisy embeddings, hence they have a lower training loss although the magnitude of noise becomes larger.

As shown in Figure 4, transferring knowledge of the intermediate Transformer layers leads the student model to the flat minima that are robust of noise and more generalizable (Hochreiter and Schmidhuber, 1997; Keskar et al., 2016). We further conduct the loss surface (Zhang et al., 2021) and linear probing (Aghajanyan et al., 2021a) analyses for evaluating the generalizable representations of PLMs during fine-tuning and report the results in Appendix E.1.

3.2 Training Data: Use Supplementary Tasks

In this section, we investigate the performance of ILD in terms of training datasets for transferring knowledge from teacher to student model. We observe that conducting ILD even on the last Transformer layer has the risk of overfitting to the training dataset of target task (TT). The Previously suggested augmentation module in Jiao et al. (2020) generates 20 times the original data as augmented samples, requiring massive computational overhead for generating. From our observation, we find that conducting ILD via supplementary tasks (ST, Phang et al. 2018) is a simple and efficient method for overfitting problem. Based on our observation, we study to find the condition for appropriate ST, which robustly improves the performance of ILD.

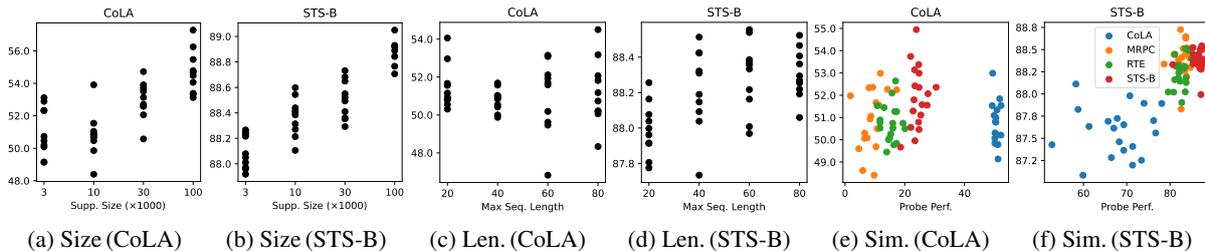


Figure 7: The conditions for appropriate supplementary tasks. The student models trained on supplementary tasks that have large datasets, longer effective sequence length, and high similarity with the target task tend to have higher performances. Size, Len., and Sim. denote the size of the dataset, effective sequence length, and similarity.

Main Observations. As shown in Figure 5, in most downstream tasks, except for STS-B, the performance of combining ILD with other STs is superior to that when using the original dataset. Among the tasks with small dataset (CoLA, MRPC, RTE, STS-B), although STS-B exhibits superiority as an ST for ILD, all student models with ILD on CoLA exhibit the worst performance for all TTs. For large tasks (MNLI, QNLI, QQP, SST-2) as STs, student models trained on MNLI and SST-2 exhibit the best and worst performance for all TTs.

Analysis. To understand performance gain from using STs, we compare the loss dynamics for fine-tuning of RTE task using the cross-entropy loss after conducting ILD on the TT (RTE) and STs (MRPC, STS-B, QNLI). Notably, the student model with ILD on RTE shows a faster decrease and increase in the training and validation loss, respectively, than the student model with ILD on the STs, as shown in Figure 6. From the results, we verify that conducting ILD over TT incurs memorization of the student model to training data of TT while performing ILD over ST prevents this memorization yet effectively transfers knowledge of the teacher model.

3.2.1 Ablation Study

Although the combination of ST with ILD generally improves the performances of student models, decreased performances are observed in some cases. These results emphasize the need to select appropriate ST. In this section, we present exploratory experiments on synthetic datasets extracted from the English Wikipedia corpus to provide further intuition for the conditions of convincing STs.

Dataset Size. According to the results in Figure 5, student models trained on STs with large datasets, such as MNLI and QQP, perform better. We conducted experiments on synthetic datasets extracted

from the Wikipedia corpus with different dataset sizes to validate our observations. The results in Figure 7a and 7b indicate that as the size of the synthetic datasets grows larger, the performance of the student models improves.

Effective Sequence Length. A surprising result of Figure 5 is that ILD on single sentence tasks such as SST-2 or CoLA exhibits lower performances than those of the smaller sentence pair tasks. This phenomenon is much more evident in U-ILD. Motivated by these results, we conducted experiments on synthetic datasets with the same dataset size of 30k and different effective sequence lengths (measured without considering [PAD] tokens). Figures 7c and 7d show that as the effective sequence length of the datasets increases, so do the performances of the student models.

Task Similarity. Finally, we investigate the effect of task similarity between TTs and STs. We only use datasets in the GLUE benchmark for computing the similarity and do not use synthetic Wikipedia datasets. To measure the task similarity, we use the probing performance of the TT after performing ILD for each ST, following Pruksachatkun et al. (2020). We conduct ILD on different STs and then conduct probing and fine-tuning on the TT. Figure 7e and 7f summarize the correlation between the probing and fine-tuning performances for CoLA and STS-B as the TT. The fine-tuning performances get better as the probing performances get better, and it is proven that ILD is better when done on an ST that has a high correlation with the TT.

4 Method: Consistency Regularized ILD

In this section, we propose a simple yet effective ILD method for improving the robustness of the student models called consistency regularized ILD (CR-ILD) that applies interpolation-based regularization (Sohn et al., 2020; Zheng et al., 2021) on

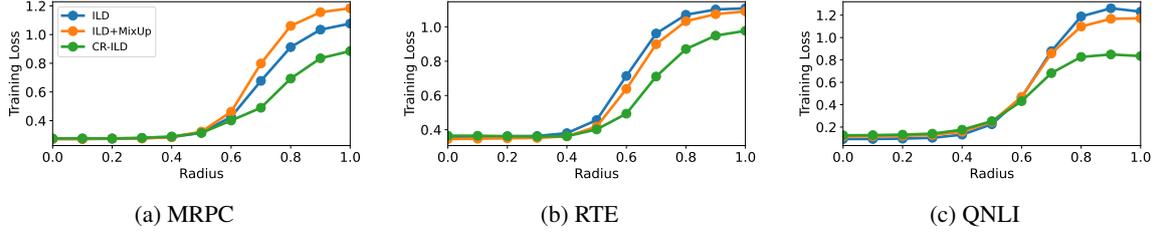


Figure 8: Comparison of training loss of different distillation approaches (ILD, ILD+MixUp, and CR-ILD) with increasing Gaussian noise: models trained with CR-ILD are more tolerant to noise which verify that our CR-ILD leads model to flat minima which have higher generalization.

Algorithm 1 Consistency Regularized ILD

Input: embedding layers $\mathbf{W}_e^T, \mathbf{W}_e^S$, model parameters Θ_T, Θ_S , training dataset \mathcal{D} , MixUp hyperparameter α , warmup iteration T , regularization coefficient $w_{\text{MHA}}^{\text{CR}}, w_{\text{IR}}^{\text{CR}}$

Output: Θ_S

- 1: initialize $t \leftarrow 0$
 - 2: **for** each minibatch \mathbf{B} **do**
 - 3: sample $|\mathbf{B}|$ pairs of $(\mathbf{x}_i, \mathbf{x}_j)$ for $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{B}$
 - 4: sample $\lambda \sim \text{Beta}(\alpha, \alpha)$
 - 5: $\mathbf{h}_i^S = \mathbf{W}_e^S \mathbf{x}_i, \mathbf{h}_j^S = \mathbf{W}_e^S \mathbf{x}_j$
 - 6: $\mathbf{h}_i^T = \mathbf{W}_e^T \mathbf{x}_i, \mathbf{h}_j^T = \mathbf{W}_e^T \mathbf{x}_j$
 - 7: $\tilde{\mathbf{h}}_i^S = \text{Mix}_\lambda(\mathbf{h}_i^S, \mathbf{h}_j^S)$
 - 8: $\tilde{\mathbf{h}}_i^T = \text{Mix}_\lambda(\mathbf{h}_i^T, \mathbf{h}_j^T)$
 - 9: compute R_{MHA} and R_{IR} from $\tilde{\mathbf{h}}_i^S, \mathbf{h}_i^S, \mathbf{h}_j^S$
 - 10: compute \mathcal{L}_{MHA} and \mathcal{L}_{IR} from $\tilde{\mathbf{h}}_i^S, \tilde{\mathbf{h}}_i^T$
 - 11: $\tilde{w}_{\text{MHA}}^{\text{CR}} = \max(\frac{t}{T}, 1) \cdot w_{\text{MHA}}^{\text{CR}}$
 - 12: $\tilde{w}_{\text{IR}}^{\text{CR}} = \max(\frac{t}{T}, 1) \cdot w_{\text{IR}}^{\text{CR}}$
 - 13: $\mathcal{L} \leftarrow \sum_{k \in \{\text{MHA}, \text{IR}\}} \mathcal{L}_k^M + \tilde{w}_k^{\text{CR}} R_k$
 - 14: update Θ_S using gradient descent methods
 - 15: update $t \leftarrow t + 1$
 - 16: **end for**
-

MHA and IR of the student models. Our method efficiently enhances the generalization by leading the student model to the flat minima (Section 3.1) and introducing appropriate ST (Section 3.2). We first introduce the proposed method and then provide analyses of CR-ILD.

4.1 Proposed Method: CR-ILD

To implement the CR, we apply MixUp (Zhang et al., 2018), which is an interpolation-based regularizer to improve the robustness in NLP (Chen et al., 2020). The direct application of MixUp to NLP is not as straightforward as images, because the input sentences consist of discrete word tokens. Instead, we perform MixUp on the word

embeddings at each token by following Chen et al. (2020); Liang et al. (2021). Thus, MixUp samples with embeddings $\mathbf{h}_i, \mathbf{h}_j$ from sentences $\mathbf{x}_i, \mathbf{x}_j$ and $\lambda \in [0, 1]$ are generated as:

$$\text{Mix}_\lambda(\mathbf{h}_i, \mathbf{h}_j) = \lambda \cdot \mathbf{h}_i + (1 - \lambda) \cdot \mathbf{h}_j,$$

Note that $\lambda \sim \text{Beta}(\alpha, \alpha)$ is randomly sampled value from Beta distribution with hyperparameter $\alpha \in (0, \infty)$ for every batch.

Then, we introduce our CR-ILD, as follows:

$$R_{f_\theta} = d(f_\theta(\text{Mix}_\lambda(\mathbf{h}_i, \mathbf{h}_j)), \text{Mix}_\lambda(f_\theta(\mathbf{h}_i), f_\theta(\mathbf{h}_j))),$$

where f_θ denotes the Transformer layer outputs (e.g., MHA and IR) of the model with parameter θ and embedded input $\mathbf{h}_i, \mathbf{h}_j$. Note that $\text{Mix}_\lambda(f_\theta(\mathbf{h}_i), f_\theta(\mathbf{h}_j)) = \lambda \cdot f_\theta(\mathbf{h}_i) + (1 - \lambda) \cdot f_\theta(\mathbf{h}_j)$ is interpolation of outputs from $\mathbf{h}_i, \mathbf{h}_j$. $d(\cdot, \cdot)$ is a distance metric for regularization, with KLD for MHA and MSE for IR. For example, we have:

$$R_{\text{MHA}} = \text{KLD}(\text{MHA}(\text{Mix}_\lambda(\mathbf{h}_i, \mathbf{h}_j)) \parallel \text{Mix}_\lambda(\text{MHA}(\mathbf{h}_i), \text{MHA}(\mathbf{h}_j)))$$

$$R_{\text{IR}} = \text{MSE}(\text{IR}(\text{Mix}_\lambda(\mathbf{h}_i, \mathbf{h}_j)), \text{Mix}_\lambda(\text{IR}(\mathbf{h}_i), \text{IR}(\mathbf{h}_j)))$$

for CR terms of MHA and IR. Hence, the overall loss function of CR-ILD is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{MHA}}^M + \mathcal{L}_{\text{IR}}^M + \tilde{w}_{\text{MHA}}^{\text{CR}} R_{\text{MHA}} + \tilde{w}_{\text{IR}}^{\text{CR}} R_{\text{IR}},$$

where $\tilde{w}_{\text{MHA}}^{\text{CR}}$ and $\tilde{w}_{\text{IR}}^{\text{CR}}$ are coefficients for regularization. As the student models are underfitted to training dataset in the early training phase, we first set the coefficients to zero and gradually increase the values to $w_{\text{MHA}}^{\text{CR}}$ and $w_{\text{IR}}^{\text{CR}}$, respectively. Note that both $\mathcal{L}_{\text{MHA}}^M$ and $\mathcal{L}_{\text{IR}}^M$ are computed by outputs from the teacher and student model with the same MixUp samples as inputs through Eq. (1) and Eq. (2). All ILD loss and CR term are computed from the last Transformer layer outputs based on Section 3. We describe the overall algorithm of CR-ILD in Algorithm 1.

Table 1: 6-layer student results on GLUE development set averaged over 4 runs. † indicates reported results from the Park et al. (2021). Other results are from our re-implementation based on officially released code of original works (Sun et al., 2019; Jiao et al., 2020; Li et al., 2020).

Model	#Params	#FLOPs	Speedup	CoLA	MNLI	SST-2	QNLI	MRPC	QQP	RTE	STS-B	AVG
BERT _{BASE}	110M	22.5B	1.0x	59.9	84.6	92.2	91.5	90.9	91.2	70.8	89.5	83.8
<i>Truncated BERT (Sun et al., 2019) as student model initialization</i>												
KD	67.5M	11.3B	2.0x	36.7	82.1	90.0	88.9	89.2	90.4	65.7	88.5	78.9
PKD	67.5M	11.3B	2.0x	37.4	82.2	90.2	89.1	89.3	90.3	66.3	87.4	79.0
TinyBERT	67.5M	11.3B	2.0x	31.4	81.3	89.2	86.7	87.1	90.2	57.2	84.8	76.0
BERT-EMD	67.5M	11.3B	2.0x	34.6	81.5	88.5	87.9	89.1	90.2	66.4	87.9	78.3
Ours	67.5M	11.3B	2.0x	40.4	82.3	91.1	90.1	89.6	90.7	67.9	89.0	80.1
<i>BERT_{Small} (Turc et al., 2019) as student model initialization</i>												
KD†	67.5M	11.3B	2.0x	-	82.5	91.1	89.4	89.4	90.7	66.7	-	-
PKD†	67.5M	11.3B	2.0x	45.5	81.3	91.3	88.4	85.7	88.4	66.5	86.2	79.2
TinyBERT†	67.5M	11.3B	2.0x	53.8	83.1	92.3	89.9	88.8	90.5	66.9	88.3	81.7
BERT-EMD	67.5M	11.3B	2.0x	50.5	83.5	92.4	90.4	89.4	90.8	68.3	88.5	81.7
CKD†	67.5M	11.3B	2.0x	55.1	83.6	93.0	90.5	89.6	91.2	67.3	89.0	82.4
Ours	67.5M	11.3B	2.0x	55.6	83.9	92.7	91.4	90.5	91.2	70.2	88.8	83.0

4.2 Analysis on CR-ILD

In this section, we provide analytical results of CR-ILD to obtain further intuition on our proposed methods. Our CR-ILD regularizes the student model to not learn an undesirable bias by (1) encouraging generalizable student via incurring consistent predictions between MixUp and original samples and (2) generating appropriate ST through MixUp operation.

To validate that our CR-ILD makes more generalizable functions empirically, we conduct a similar experiment with Figure 4 for comparing three models (ILD, ILD+MixUp, CR-ILD) as shown in Figure 8. ILD+MixUp is the simple combination of ILD and MixUp, which is the same as CR-ILD with w_{MHA}^{CR} , and w_{IR}^{CR} for zero. Note that we only use the last Transformer layer for all ILD methods in Figure 8. From the results, we obtain that our CR-ILD effectively regularizes the student model not to overfit training data and to be robust to noise injected in embedding spaces. Moreover, it is noteworthy that this smooth regularization is from CR-ILD, whereas the naive application of MixUp does not regularize the student model efficiently.

Here, we introduce our theoretical analysis that CR-ILD explicitly leads the functions (i.e., MHA, IR) to be convex which is smooth for all data points.

Theorem 4.1 (Informal). *Assume that f_θ satisfies the Assumption A.2. With the second order Taylor approximation for λ in Definition A.1, the \mathcal{L}_{mix}*

becomes $\hat{\mathcal{L}}_{mix}$ which can be represented as:

$$\hat{\mathcal{L}}_{mix} = \mathcal{L}_{std} - \frac{2\alpha + 1}{(4\alpha + 4)|I|} \sum_{j \in I} D_{\ell,j} H_{\ell,j}^{-1} D_{\ell,j}^\top, \\ + \frac{\alpha + 1}{(8\alpha + 4)|I|^2} \sum_{i,j \in I} R^*(f_\theta(\mathbf{h}_i), f_\theta(\mathbf{h}_j), \mathbf{y}_i, \mathbf{y}_j)$$

where $H_{\ell,j} = \text{Hess}_\ell(f_\theta(\mathbf{h}_j), \mathbf{y}_j)$, and $D_{\ell,j} = D_\ell(f_\theta(\mathbf{h}_j), \mathbf{y}_j)$.

The detailed form of $R^*(f_\theta(\mathbf{h}_i), f_\theta(\mathbf{h}_j), \mathbf{y}_i, \mathbf{y}_j)$ can be found in Appendix A. Theorem 4.1 states that the regularization effect of CR-ILD that makes the significant performance gain of CR-ILD. When we assume that the Hessian can be approximated by the gradient square or outer product of the gradients as in the Gauss-Newton method, the first negative term can be treated as nearly constant. We have the positive term, which performs regularization, and the near-constant negative term. As we discussed earlier, the trainable part of regularizing term reduces the offset related to curvature information. Furthermore, the regularization scheme of CR-ILD can be explained variously. If we assume that the set of data has a non-empty interior, $f(\mathbf{h})$ becomes a linear function, therefore, we can say there is a trend that the function is regularized as a simple smooth function.

Moreover, thanks to MixUp (Zhang et al., 2018; Liang et al., 2021) operation, we can effectively generate the appropriate ST (Section 3.2.1) via:

- From the MixUp operation, the possible number of MixUp samples can be increased infinitely with the choice of original samples

Table 2: The performance averaged over 4 runs on the GLUE development set of 6-layer student models, which were trained on a 1k down-sampled GLUE training set or a GLUE training set under symmetric label noise. We use officially released codes for the re-implementation of PKD (Sun et al., 2019), TinyBERT (Jiao et al., 2020), and BERT-EMD (Li et al., 2020). For label noise experiments, we do not consider STS-B for computing average values.

Model	#Params	#FLOPs	Speedup	CoLA	MNLI	SST-2	QNLI	MRPC	QQP	RTE	STS-B	AVG
<i>1k down-sampled (Zhang et al., 2021) for few-samples experiments</i>												
BERT _{BASE}	110M	22.5B	1.0x	41.6	61.1	85.8	80.8	88.2	75.9	66.1	87.6	73.4
KD	67.5M	11.3B	2.0x	17.6	58.0	83.4	78.9	86.2	74.8	59.6	83.9	67.8
PKD	67.5M	11.3B	2.0x	17.7	57.8	83.8	75.2	86.3	73.9	59.1	83.4	67.2
TinyBERT	67.5M	11.3B	2.0x	9.3	55.5	80.2	71.7	85.2	72.0	57.8	82.1	64.2
BERT-EMD	67.5M	11.3B	2.0x	18.8	58.0	84.2	78.5	86.3	74.3	62.1	84.8	68.4
Ours	67.5M	11.3B	2.0x	20.1	59.6	85.0	80.3	87.2	75.7	63.5	85.8	69.7
<i>Under the presence of uniform (symmetric) label noise (Jin et al., 2021; Liu et al., 2022) with 30% noise rate</i>												
BERT _{BASE}	110M	22.5B	1.0x	39.6	81.7	90.4	86.4	82.3	86.3	57.0	-	74.8
KD	67.5M	11.3B	2.0x	37.3	80.3	88.4	85.6	81.3	86.1	59.6	-	74.1
PKD	67.5M	11.3B	2.0x	36.8	80.0	87.6	85.4	81.1	86.2	56.2	-	73.3
TinyBERT	67.5M	11.3B	2.0x	29.7	79.9	87.2	84.6	81.2	85.7	51.6	-	71.4
BERT-EMD	67.5M	11.3B	2.0x	38.5	80.6	87.8	84.9	81.2	86.0	57.0	-	73.7
Ours	67.5M	11.3B	2.0x	39.6	81.2	89.1	86.0	82.3	86.9	61.8	-	75.3

and λ . This operation increases the dataset size with high task similarity since the MixUp samples are created from the interpolation of the original target task.

- If sentence \mathbf{x}_i contains more word tokens than sentence \mathbf{x}_j , then the extra word embeddings are mixed up with embeddings of [PAD] tokens. This operation lengthens the effective sequence length of the dataset in Section 3.2.1, which improves the performance of ILD.

From our analysis, we verify that our proposed CR-ILD can effectively transfer the knowledge of teacher models with less overfitting on the training dataset.

5 Experiments

To verify the effectiveness of CR-ILD, we compare the performance of ours with previous distillation methods on the standard GLUE and ill-conditioned GLUE benchmark. The descriptions for experimental setup are in Appendix B and C.

5.1 Main Results

Standard GLUE. Following the standard setup (Sun et al., 2019), we use the BERT_{BASE} as the teacher and 6-layer Truncated BERT (Sun et al., 2019) and BERT_{Small} (Turc et al., 2019) as the student models. Table 1 summarizes that Ours consistently achieve state-of-the-art performances for almost GLUE benchmark, except for SST-2 and STS-B for BERT_{Small}. Despite the simplicity

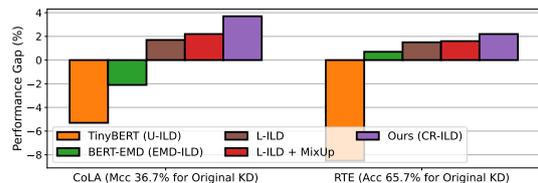


Figure 9: Ablation study on the standard GLUE (CoLA, RTE) with Truncated BERT and BERT_{BASE} as student and teacher models, respectively.

and efficiency of our proposed method, we obtain strong empirical performance.

Ill-conditioned GLUE. To verify the robustness of our proposed method, we further conduct the experiments on ill-conditioned GLUE, a synthetic dataset with downsampling or injecting label noise to the GLUE benchmark. Since STS-B is a regression task, we cannot inject noise into STS-B. Hence, we do not consider the STS-B task in label noise experiments. The detailed descriptions for ill-conditioned GLUE are in Appendix B. Table 2 demonstrate that our proposed method alleviates the overfitting and enhances the performance of the student model under few-samples training datasets or the presence of 30% of label noise. The results for other noise rate are in Appendix C. The experimental results encourage us to use our method on real-world applications which have a high risk of overfitting on the training datasets. Notably, our proposed method achieve higher performance than the teacher model under the presence of label noise.

5.2 Ablation Study

To obtain further intuition on CR-ILD, we conduct an ablation study on each component (*i.e.*, L-ILD, ST through MixUp, and CR) of our method. Our experiments are conducted on the standard GLUE benchmark with Truncated BERT (Sun et al., 2019) as the student models and BERT_{BASE} as the teacher models. Figure 9 summarizes that all our findings are meaningful, as the performance improves with each addition of a component.

6 Conclusion

This paper introduces a better use of ILD that transfer knowledge by using outputs of Transformer layers of the teacher and the student models. We found that existing ILD methods may lead the student model to overfit the training dataset of target tasks and degenerate the generalizability. Furthermore, we investigated that conducting the ILD (1) only for the last Transformer layer and (2) on supplementary tasks can alleviate the overfitting problems. Based on our observations, we proposed consistency-regularized ILD that incurs smoother functions and enhance the generalizability of the student models. Our proposed method effectively distills the knowledge of teacher models by (1) encouraging the flat minima of function from consistency regularization between original embeddings and MixUp embeddings of the student models and (2) efficiently generating appropriate supplementary tasks demonstrated in our findings via MixUp operation. The experimental results showed that our proposed method could achieve state-of-the-art performance on various datasets, such as the standard and ill-conditioned GLUE benchmarks.

Limitations

Our work handles the over-fitting of the student network caused by the layer mapping between the teacher and the student networks, which is widely used in Jiao et al. (2020); Li et al. (2020). Although we show that our proposed regularization technique can mitigate the over-fitting of the student, the relationship between layers inside the model and the hidden state of tokens in one layer (Park et al., 2021) was not sufficiently considered. In addition, we back up our proposed idea with theoretical analysis and extensive experiments in sentence classification. We plan to perform token classification and question-answering experiments to expand our methods to other tasks.

Ethics Statement

Our work complies with all ethical considerations. We hope our work contributes to environmental issues by reducing the computation cost of large PLMs.

Acknowledgment

This work was supported by the “Research on model compression algorithm for Large-scale Language Models” project funded by KT (KT award B210001432, 50%). Also, this work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by Korea government (MSIT) [No. 2021-0-00907, Development of Adaptive and Lightweight Edge-Collaborative Analysis Technology for Enabling Proactively Immediate Response and Rapid Learning, 45%] and [No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST), 5%].

References

- Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021a. [Better fine-tuning by reducing representational collapse](#). In *International Conference on Learning Representations*.
- Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021b. [Better fine-tuning by reducing representational collapse](#). In *International Conference on Learning Representations*.
- Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2021. [BinaryBERT: Pushing the limit of BERT quantization](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4334–4348, Online. Association for Computational Linguistics.
- Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. [Mix-Text: Linguistically-informed interpolation of hidden space for semi-supervised text classification](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

- Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Flat minima. *Neural computation*, 9(1):1–42.
- Jeremy Howard and Sebastian Ruder. 2018. **Universal language model fine-tuning for text classification**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. 2021. Annealing knowledge distillation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2493–2504.
- Jongheon Jeong and Jinwoo Shin. 2020. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 33:10558–10570.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. **TinyBERT: Distilling BERT for natural language understanding**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.
- Lifeng Jin, Linfeng Song, Kun Xu, and Dong Yu. 2021. Instance-adaptive training with noise-robust losses against noisy labels. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5647–5663.
- Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2016. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.
- Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. 2020. **BERT-EMD: Many-to-many layer mapping for BERT compression with earth mover’s distance**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3009–3018, Online. Association for Computational Linguistics.
- Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2021. **Mix{kd}: Towards efficient distillation of large-scale language models**. In *International Conference on Learning Representations*.
- Bo Liu, Wandu Xu, Yuejia Xiang, Xiaojun Wu, Lejian He, Bowen Zhang, and Li Zhu. 2022. **Noise learning for text classification: A benchmark**. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4557–4567, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yihuan Mao, Yujing Wang, Chufan Wu, Chen Zhang, Yang Wang, Quanlu Zhang, Yaming Yang, Yunhai Tong, and Jing Bai. 2020. **LadaBERT: Lightweight adaptation of BERT through hybrid model compression**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3225–3234, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Geondo Park, Gyeongman Kim, and Eunho Yang. 2021. **Distilling linguistic context for language model compression**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 364–378, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.
- Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. **Intermediate-task transfer learning with pretrained language models: When and why does it work?** In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Ahmad Rashid, Vasileios Lioutas, and Mehdi Rezagholizadeh. 2021. Mate-kd: Masked adversarial text, a companion to knowledge distillation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1062–1071.
- Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus

- Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332.
- Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170.
- Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. **GLUE: A multi-task benchmark and analysis platform for natural language understanding**. In *International Conference on Learning Representations*.
- Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. **MiniLMv2: Multi-head self-attention relation distillation for compressing pre-trained transformers**. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.
- Yimeng Wu, Peyman Passban, Mehdi Rezagholizade, and Qun Liu. 2020. Why skip if you can combine: A simple knowledge distillation technique for intermediate layers. *arXiv preprint arXiv:2010.03034*.
- Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. **mixup: Beyond empirical risk minimization**. In *International Conference on Learning Representations*.
- Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722.
- Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. **Revisiting few-sample {bert} fine-tuning**. In *International Conference on Learning Representations*.
- Bo Zheng, Li Dong, Shaohan Huang, Wenhui Wang, Zewen Chi, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. 2021. **Consistency regularization for cross-lingual fine-tuning**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3403–3417, Online. Association for Computational Linguistics.

Appendix

Revisiting Intermediate Layer Distillation for Compressing Language Models: An Overfitting Perspective

A Theoretical Analysis of CR-ILD

This section gives the theoretical argument that CR-ILD gives additional explicit regularization. We analyze the effect of the MixUp objective function beyond the standard loss function when the CR condition is satisfied. We use the below formulation for objective functions. **For readability, we partially apply one column style for this section.**

Definition A.1 (Objective Functions). Let us define $\mathcal{D}_\lambda := \text{Beta}(\alpha, \alpha)$, $\tilde{\mathbf{h}}_{ij} := \lambda \mathbf{h}_i + (1 - \lambda) \mathbf{h}_j$, and $\tilde{\mathbf{y}}_{ij} := \lambda \mathbf{y}_i + (1 - \lambda) \mathbf{y}_j$. Consider the index set I . Then the objective functions can be written as:

$$\mathcal{L}_{\text{std}} := \frac{1}{|I|} \sum_{i \in I} \ell(f_\theta(\mathbf{h}_i), \mathbf{y}_i) \quad \mathcal{L}_{\text{mix}} := \mathbb{E}_{\lambda \sim \mathcal{D}_\lambda} \left[\frac{1}{|I|^2} \sum_{i, j \in I} \ell(f_\theta(\tilde{\mathbf{h}}_{ij}), \tilde{\mathbf{y}}_{ij}) \right],$$

We assume that CR loss is always optimized during training. That is, if CR loss is 0, each pair of function values in the loss coincides. Therefore, we can write the first assumption as follows:

Assumption A.2 (Continuation of f_θ (\spadesuit)). we assume that f_θ has continuation on the expanded domain $\{\lambda \mathbf{h}_i + (1 - \lambda) \mathbf{h}_j : \lambda \in [0, 1], i, j \in I\}$ and the for any convex combination, function value becomes:

$$f_\theta(\lambda \mathbf{h}_i + (1 - \lambda) \mathbf{h}_j) = \lambda f_\theta(\mathbf{h}_i) + (1 - \lambda) f_\theta(\mathbf{h}_j)$$

Note also that this continuation can always be well defined if $\{\mathbf{h}_i\}_{i \in I}$ are in general position. Under this assumption, the MixUp loss possesses a regularization effect, which stabilizes the functional outcomes.

Theorem A.3. Assume that f_θ satisfies the Assumption A.2. With the second order Taylor approximation for λ , the \mathcal{L}_{mix} becomes $\hat{\mathcal{L}}_{\text{mix}}$ which can be represented as:

$$\begin{aligned} \hat{\mathcal{L}}_{\text{mix}} = & \mathcal{L}_{\text{std}} + \frac{\alpha + 1}{(8\alpha + 4)|I|^2} \sum_{i, j \in I} \left\| (f_\theta(\mathbf{h}_j), \mathbf{y}_j) - (f_\theta(\mathbf{h}_i), \mathbf{y}_i) + (2\alpha + 1) H_{\ell, j}^{-1} D_{\ell, j}^\top \right\|_{H_{\ell, j}}^2 \\ & - \frac{2\alpha + 1}{(4\alpha + 4)|I|} \sum_{j \in I} D_{\ell, j} H_{\ell, j}^{-1} D_{\ell, j}^\top, \end{aligned}$$

where $H_{\ell, j} = \text{Hess}_\ell(f_\theta(\mathbf{h}_j), \mathbf{y}_j)$, and $D_{\ell, j} = D_\ell(f_\theta(\mathbf{h}_j), \mathbf{y}_j)$.

Note also that the expectation on higher order of λ exponentially decreases as $\mathbb{E}_{\mathcal{D}_\lambda}[\lambda^n] \sim 2^{-n}$, if α is sufficiently large. The above formulation indicates that the MixUp training with consistency regularization gives further regularization terms, which stabilizes function values $f_\theta(\mathbf{h}_i)$.

A.1 Derivation of the Theorem A.3

Let us write $\mathbf{v}_{\theta, ij}^x = f_\theta(\mathbf{h}_i) - f_\theta(\mathbf{h}_j)$, $\mathbf{v}_{\theta, ij} = (\mathbf{v}_{\theta, ij}^x, \mathbf{y}_i - \mathbf{y}_j)$. We first state the second-order Taylor approximation of loss function ℓ :

$$\begin{aligned} \mathcal{L}_{\text{mix}} &= \mathbb{E}_{\lambda \sim \mathcal{D}_\lambda} \left[\frac{1}{|I|^2} \sum_{i, j \in I} \ell(f_\theta(\tilde{\mathbf{h}}_{ij}), \tilde{\mathbf{y}}_{ij}) \right] \\ &\stackrel{\spadesuit}{=} \mathbb{E}_{\lambda \sim \mathcal{D}_\lambda} \left[\frac{1}{|I|^2} \sum_{i, j \in I} \ell(\lambda f_\theta(\mathbf{h}_i) + (1 - \lambda) f_\theta(\mathbf{h}_j), \lambda \mathbf{y}_i + (1 - \lambda) \mathbf{y}_j) \right] \\ &\stackrel{\text{Taylor}}{=} \underbrace{\frac{1}{|I|} \sum_{i \in I} \ell(f_\theta(\mathbf{h}_i), \mathbf{y}_i)}_{=: \mathcal{L}_{\text{std}}} + \frac{1}{|I|^2} \sum_{i, j \in I} \left[\frac{1}{2} D_{\ell, j} \mathbf{v}_{\theta, ij} + \frac{1}{2} \frac{\alpha + 1}{4\alpha + 2} \mathbf{v}_{\theta, ij}^\top H_{\ell, j} \mathbf{v}_{\theta, ij} \right], \end{aligned}$$

since $\mathbb{E}_{\lambda \sim \mathcal{D}_\lambda}[\lambda] = 1/2$ and $\mathbb{E}_{\lambda \sim \mathcal{D}_\lambda}[\lambda^2] = (\alpha + 1)/(4\alpha + 2)$. Then,

$$\begin{aligned}
& \mathcal{L}_{\text{mix}} \mathcal{L}_{\text{std}} + \frac{1}{2|I|^2} \sum_{i,j \in I} \left[D_{\ell,j} \mathbf{v}_{\theta,ij} + \frac{1}{2} \frac{\alpha + 1}{2\alpha + 1} \mathbf{v}_{\theta,ij}^\top H_{\ell,j} \mathbf{v}_{\theta,ij} \right] \\
&= \mathcal{L}_{\text{std}} + \frac{1}{2|I|^2} \sum_{i,j \in I} \left[-\frac{2\alpha + 1}{2\alpha + 2} D_{\ell,j} H_{\ell,j}^{-1} D_{\ell,j}^\top + \right. \\
&\quad \left. \frac{1}{2} \frac{\alpha + 1}{2\alpha + 1} \left(\mathbf{v}_{\theta,ij} + \frac{2\alpha + 1}{\alpha + 1} H_{\ell,j}^{-1} D_{\ell,j}^\top \right)^\top H_{\ell,j} \left(\mathbf{v}_{\theta,ij} + \frac{2\alpha + 1}{\alpha + 1} H_{\ell,j}^{-1} D_{\ell,j}^\top \right) \right] \\
&= \mathcal{L}_{\text{std}} + \frac{\alpha + 1}{(8\alpha + 4)|I|^2} \sum_{i,j \in I} \|\mathbf{v}_{\theta,ij} + (2\alpha + 1) H_{\ell,j}^{-1} D_{\ell,j}^\top\|_{H_{\ell,j}}^2 - \frac{2\alpha + 1}{(4\alpha + 4)|I|} \sum_{j \in I} D_{\ell,j} H_{\ell,j}^{-1} D_{\ell,j}^\top.
\end{aligned}$$

B Dataset Description

Standard GLUE. The GLUE benchmark (Wang et al., 2019) cover four tasks: natural language inference (RTE, QNLI, MNLI), paraphrase detection (MRPC, QQP, STS-B), sentiment classification (SST-2), and linguistic acceptability (CoLA). We mainly focus on four tasks (RTE, MRPC, STS-B, CoLA) that have fewer than 10k training samples. While BERT fine-tuning on these datasets is known to be unstable, the ILD on few samples is under-explored. The evaluation metrics for each task of GLUE benchmark are accuracy (MNLI, SST-2, QNLI, QQP, RTE), Mcc (CoLA), F1 score (MRPC), and spearman correlation (STS-B). We utilize original split of train, validation (development) dataset for our experiments.

Ill-conditioned GLUE. We use two types of modification on GLUE benchmark, including down-sampling for few-sample GLUE and injecting label noise for corrupted GLUE. For generating few-samples GLUE, we randomly down-sample 1k-sized dataset for each task by following Zhang et al. (2021). For corrupted GLUE, we follow the experimental setups of Jin et al. (2021) and inject uniform randomness into a fraction of labels. All other attributes are same for the standard GLUE. Also, we do not modify the development dataset of GLUE benchmark.

Extracted Wiki Corpus in Section 3.2.1 To generate synthetic data, we randomly generate the sample which is consist of two sentences from the Wikipedia corpus (version: enwiki-20200501 from Huggingface). We filter the generated sample by sequence length (for experiments of effectiveness of sequence length). We generate new dataset for every single experiment instead of conducting numerous experiment trials to reduce the randomness.

C Additional Description for Experiments

C.1 Scope of Empirical Study in Section 3

Transformer-based Language Models. Transformer encodes contextual information for input tokens (Vaswani et al., 2017). We denote the concatenation of input vectors $\{\mathbf{x}_i\}_{i=1}^{|\mathbf{x}|}$ as $\mathbf{H}_0 = [\mathbf{x}_1, \dots, \mathbf{x}_{|\mathbf{x}|}]$. Then, the computation for encod-

ing vectors via stacked Transformer layers is via:

$$\mathbf{H}_\ell = \text{Transformer}_\ell(\mathbf{H}_{\ell-1}), \ell \in [1, L].$$

The attention mechanism in Transformer improves the performance of NLP significantly and becomes essential. For the ℓ -th Transformer layer, the output for a self-attention head $\mathbf{O}_{\ell,a}$, $a \in [1, A_h]$ is via:

$$\begin{aligned} \mathbf{Q}_{\ell,a} &= \mathbf{H}_{\ell-1} \mathbf{W}_{\ell,a}^Q, \mathbf{K}_{\ell,a} = \mathbf{H}_{\ell-1} \mathbf{W}_{\ell,a}^K, \\ \mathbf{A}_{\ell,a} &= \text{SoftMax}\left(\frac{\mathbf{Q}_{\ell,a} \mathbf{K}_{\ell,a}^\top}{\sqrt{d_k}}\right), \\ \mathbf{V}_{\ell,a} &= \mathbf{H}_{\ell-1} \mathbf{W}_{\ell,a}^V, \mathbf{O}_{\ell,a} = \mathbf{A}_{\ell,a} \mathbf{V}_{\ell,a}, \end{aligned}$$

where the previous layer’s outputs $\mathbf{H}_{\ell-1} \in \mathbb{R}^{|\mathbf{x}| \times d_h}$ are linearly projected to a triple of queries, keys, and values using parameter matrices $\mathbf{W}_{\ell,a}^Q, \mathbf{W}_{\ell,a}^K, \mathbf{W}_{\ell,a}^V \in \mathbb{R}^{d_h \times d_k}$, respectively. Note that A_h is the number of attention heads.

Multi-Head Attention. Many approaches (Jiao et al., 2020; Sun et al., 2020; Wang et al., 2020) train the student, making the MHA of the student (\mathbf{A}^S) imitate the MHA of the well-optimized teacher (\mathbf{A}^T).

$$\mathcal{L}_{\text{MHA}}^{\ell^S} = \frac{1}{A_h} \sum_{a=1}^{A_h} \text{KLD}(\mathbf{A}_{m(\ell^S),a}^T \| \mathbf{A}_{\ell^S,a}^S),$$

where KLD is KL-divergence as the loss function. Note that $m(\cdot)$ is the layer mapping function for input as student layer $\ell^S \in [0, M]$ and output as teacher layer $m(\ell^S) \in [1, L]$. We compare the KLD and mean squared error (MSE) for the loss function, and report the results that KLD shows better performance in Table 3.

Table 3: Comparison between KLD and MSE as the loss function for MHA distillation.

	CoLA	MRPC	RTE	STS-B
MHA (KLD)	38.1 (1.5)	89.3 (0.5)	67.0 (0.8)	89.1 (0.1)
MHA (MSE)	37.6 (0.7)	89.1 (0.5)	66.5 (1.3)	89.0 (0.1)
MHA (KLD) + IR	38.4 (1.3)	89.3 (0.3)	67.2 (1.1)	89.1 (0.1)
MHA (MSE) + IR	38.0 (1.7)	89.1 (0.3)	66.3 (0.9)	89.1 (0.1)

Intermediate Representation. Additionally, we study IR, common distillation objective regardless of the network architectures. The MSE between the IR of the teacher (\mathbf{H}^T) and student (\mathbf{H}^S) is used as the knowledge transfer objective:

$$\mathcal{L}_{\text{IR}}^{\ell^S} = \text{MSE}(\mathbf{H}_{m(\ell^S)}^T, \mathbf{W}^H \mathbf{H}_{\ell^S}^S).$$

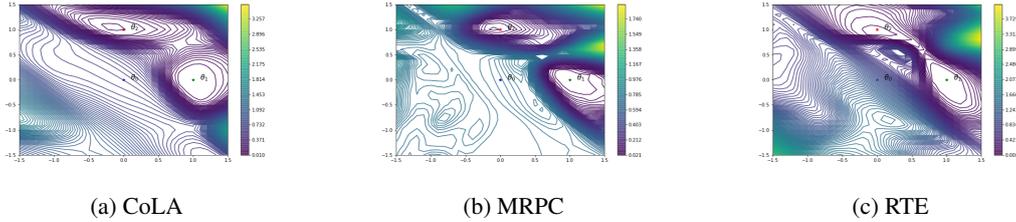


Figure 10: 2D loss surfaces in the subspace spanned by $\delta_1 = \theta_1 - \theta_0$ and $\delta_2 = \theta_2 - \theta_0$ on MRPC and RTE. $\theta_0, \theta_1, \theta_2$ denote the parameters of the Truncated BERT (blue), Last model (green) and Uniform model (red).

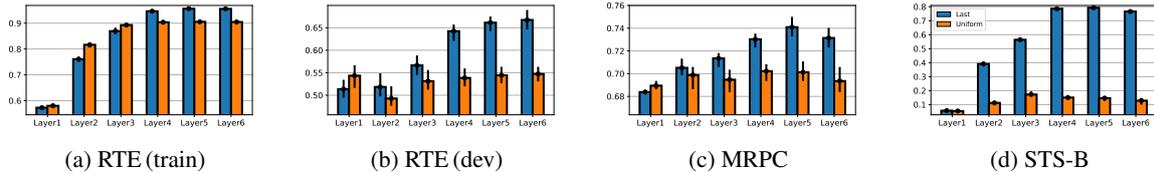


Figure 11: Results from our layer-wise (x -axis) probe comparing student models trained on RTE with L-ILD and U-ILD., respectively. The student model trained with L-ILD have more generalizable representations than U-ILD.

Note that \mathbf{W}^H is learnable weight matrix for matching the dimension between representations of the teacher and student. We further compare the IR and patience (Sun et al., 2019) in Table 3.

Table 4: Comparisons between Pool and Patience as representation for IR distillation.

	RTE (RTE)	STS-B (STS-B)	RTE (MNLI)	STS-B (MNLI)
Pool	60.6 (1.2)	86.2 (0.3)	69.9 (0.8)	89.2 (0.1)
Patience	66.2 (1.0)	88.3 (0.4)	68.8 (0.5)	88.8 (0.2)
Pool + MHA	67.2 (1.1)	89.1 (0.1)	70.6 (1.0)	89.6 (0.1)
Patience + MHA	66.5 (1.5)	88.4 (0.2)	68.7 (1.2)	88.4 (0.2)

Prediction Layer. The most standard form of KD is logit-based KD (Hinton et al., 2015) for training prediction layer.

$$\mathcal{L}_{PL} = \text{CE}(\mathbf{z}^T/t, \mathbf{z}^S/t).$$

We use the cross-entropy (CE) as the loss function with inputs \mathbf{z}^S and \mathbf{z}^T as the logit vectors of the student and teacher. We compare the sequential and joint training ILD (i.e., MHA, IR) and prediction layer distillation (PLD) and report the results that sequential training shows better in Table 5.

Table 5: Comparisons between Sequential and Joint

	RTE (RTE)	STS-B (STS-B)	RTE (MNLI)	STS-B (MNLI)
Sequential	67.2 (1.1)	89.1 (0.1)	70.6 (1.0)	89.6 (0.1)
Joint	66.7 (1.5)	88.8 (0.2)	68.9 (1.4)	89.3 (0.2)

D Experimental Setup

In this section, we describe the setup for our experimental results. Note that all single experiments

are conducted on a single NVIDIA GeForce RTX 2080Ti GPU.

D.1 Setup for Section 3 and Section 4

For teacher model, we fine-tune the uncased, 12-layer BERT_{BASE} model with batch size 32, dropout 0.1, and peak learning rate 2×10^{-5} for three epochs. For student model, we mainly use with 6-layer BERT model with initialize point as Truncated BERT (Sun et al., 2019) and BERT_{Small} (Turc et al., 2019). For fine-tuning student model, under the supervision of a fine-tuned BERT_{BASE}, we firstly perform ILD for 20 epochs with batch size 32 and learning rate 5×10^{-5} as follows Jiao et al. (2020). Then, we conduct prediction layer distillation (PLD) for 4 epochs with choosing batch size 16 and learning rate from 2×10^{-5} . Unlike the logit-based KD, we only use PLD term and do not use supervision from true labels. while We utilize GLUE (Wang et al., 2019) benchmark for exploratory experiments and set the maximum sequence length is set to 128 for all tasks.

D.2 Setup for Section 5

For achieve higher performance with our methods, we conduct hyper-parameter search as follows:

- Peak learning rate (ILD): [2×10^{-5} , 5×10^{-5}]
- Batch size (PLD): [16, 32]
- MixUp parameter (α): [0.5, 1.0, 2.0, 3.0]

For other hyper-parameter settings are not in the list, we use same parameter values as described in

Table 6: The performance averaged over 4 runs on the GLUE development set of 6-layer student models, which were trained on GLUE training set under symmetric label noise (10% and 20%). We use officially released codes for the re-implementation of PKD (Sun et al., 2019), TinyBERT (Jiao et al., 2020), and BERT-EMD (Li et al., 2020). For label noise experiments, we do not consider STS-B for computing average values.

Model	#Params	#FLOPs	Speedup	CoLA	MNLI	SST-2	QNLI	MRPC	QQP	RTE	STS-B	AVG
<i>Under the presence of uniform (symmetric) label noise (Jin et al., 2021; Liu et al., 2022) with 10% noise rate</i>												
BERT _{BASE}	110M	22.5B	1.0x	54.0	83.1	91.1	90.0	90.6	89.7	67.5	-	80.9
KD	67.5M	11.3B	2.0x	44.9	81.6	90.6	88.9	88.7	89.6	65.0	-	78.5
PKD	67.5M	11.3B	2.0x	45.2	81.2	90.5	89.0	89.1	89.4	65.4	-	78.5
TinyBERT	67.5M	11.3B	2.0x	35.4	81.9	90.1	88.3	88.3	89.6	59.9	-	76.2
BERT-EMD	67.5M	11.3B	2.0x	48.2	81.3	90.5	88.0	89.2	89.1	66.1	-	78.9
Ours	67.5M	11.3B	2.0x	50.1	82.0	90.7	89.2	89.2	89.6	66.5	-	79.6
<i>Under the presence of uniform (symmetric) label noise (Jin et al., 2021; Liu et al., 2022) with 20% noise rate</i>												
BERT _{BASE}	110M	22.5B	1.0x	50.8	82.4	90.0	88.6	87.7	87.9	63.2	-	78.7
KD	67.5M	11.3B	2.0x	42.7	81.5	90.1	88.4	87.6	88.1	64.6	-	77.6
PKD	67.5M	11.3B	2.0x	41.8	81.4	89.4	87.9	87.5	88.0	63.0	-	77.3
TinyBERT	67.5M	11.3B	2.0x	31.6	81.8	89.0	87.7	87.6	88.0	56.7	-	74.6
BERT-EMD	67.5M	11.3B	2.0x	40.7	81.0	89.7	87.6	88.0	87.9	64.6	-	77.1
Ours	67.5M	11.3B	2.0x	44.6	81.9	89.8	88.6	88.1	88.2	65.2	-	78.1

main text or Appendix D.1. We find that 2×10^{-5} is the best peak learning rate of ILD for all tasks except for STS-B. For batch size of PLD stage, RTE, MNLI and QNLI shows higher performance with batch size of 32 and other tasks shows higher performance with batch size of 16. For α , a hyperparameter for MixUp operation in CR-ILD, we choose the value of 1.0 by the result of our hyperparameter search. All hyperparameter search are conducted by using **grid search** with **averaged three runs**.

E Further Experiments on BERT

E.1 Further Observation for Section 3.1

Loss Surface Analysis. To get further intuition about the performance degradation of distilling the knowledge of intermediate Transformer layers, we provide loss surface visualizations of the U-ILD and L-ILD settings. The parameters of the Truncated BERT, the Last model (student model trained with L-ILD), and the Uniform model (student model trained with U-ILD) are $\theta_0, \theta_1, \theta_2$, respectively. In the subspace spanned by $\delta = \theta_1 - \theta_0$ and $\delta = \theta_2 - \theta_0$, we plot two-dimensional loss surfaces $f(\alpha, \beta) = \mathcal{L}(\theta_0 + \alpha\delta_1 + \beta\delta_2)$ centered on the weights of Truncated BERT θ_0 . As shown in Figure 10, transferring knowledge of the intermediate Transformer layers leads the student model to sharp minima, which results in poorer generalization (Hochreiter and Schmidhuber, 1997; Keskar et al., 2016). Thus, the knowledge from the intermediate Transformer layer causes the student

model to overfit the training dataset and reduce the generalization.

Linear Probing Analysis. Probing experiments can be used for evaluating the degradation of the generalizable representations of PLMs during fine-tuning. Similar to Aghajanyan et al. (2021b), we conduct the probing method by first freezing the representations from the model trained on one downstream task, and then fine-tuning linear classifiers on top of all Transformer layers to measure the generalization performance of the layers of the teacher and student models.

Through probing experiments, we observe that the lower-level representations of the student model related to U-ILD are overfitted to the training dataset of the target task. Figure 11a shows that the probing performances for 1 to 3 layers of the student model with U-ILD are higher than those of the Last model on the training set of RTE. According to Howard and Ruder (2018); Zhang et al. (2021), it is crucial to train PLMs so that lower layers have general features and higher layers are specific to target tasks. The overfitting of lower layers to the target task leads to performance degradation in the higher layers, as illustrated in Figure 11b. Moreover, for the other tasks, the student models with L-ILD have higher probing performance for all layers than the Uniform models, except for the performance of the first layer on MRPC as indicated in Figure 11c and 11d.

E.2 Experimental Results for Different Label Noise Ratio

We conduct additional experiments on the GLUE benchmark with different label noise ratios (10% and 20% of uniform label noise) as shown in Table 6. While BERT-EMD (Li et al., 2020) shows the second best performance in small noise ratio (10%) and achieve better performance than the original KD, the original KD and PKD (Sun et al., 2019) present the higher performance in severe noise rate (20% in Table 6 and 30% in Table 2) than BERT-EMD. Surprisingly, our CR-ILD (Ours) shows the best performance for all noise ratios consistently which verifies that our proposed method encourages the distilling of the knowledge effectively and prevents overfitting on the training datasets.

F Further Experiments on Encoder-Decoder Models

F.1 T5: Study on Encoder-Decoder Models

In this section, we apply our approaches to T5 to generalize our result from the encoder-based model to the encoder-decoder model. First, we explain our experimental setup in the experiments conducted on T5. Secondly, we examine (1) two findings (last Transformer layer, supplementary task) and (2) our proposed method, CR-ILD suggested with the experiments on BERT can boost the performance of T5 model as well as the encoder-based model.

F.2 Experimental setup

We experiment with our proposed training strategies on the encoder-decoder model. As a teacher model, we use T5_{BASE} fine-tuned to the target task with batch size 8, learning rate 1×10^{-3} for ten epochs, which follows a training scheme for fine-tuning T5 on an individual GLUE task proposed in (Raffel et al., 2020). As a student model, we use the pre-trained T5_{Small}. During the distillation, we distill the knowledge from the teacher model to the student model consecutively, similar to the training scheme described in the experimental setup of BERT distillation. We first distill the knowledge using the given distillation objective (i.e., attention, intermediate states) depending on the task. Unlike the BERT experiments, we fine-tune the T5 model on the target task after the ILD since the performance decreases in a few tasks when we apply logit-based KD (Hinton et al., 2015). To distill the transformer layers and the intermediate states, we use methods proposed by (Wang et al.,

2021) and (Jiao et al., 2020). Specifically, before distilling the attention scores, we applied relation heads proposed in (Wang et al., 2021) and calculated attention scores since the number of attention heads of the student and the teacher differs. After matching the number of relation heads, we distill attention scores of relation head and the hidden states, using the methods of (Jiao et al., 2020). Regarding the supplementary tasks, we use the same hyperparameters as the ILD experiments. In CR-ILD experiments, we set w_{MHA}^{CR} as 0.2 and 0.3 for the MRPC and RTE task individually.

F.3 Experimental Results: Last Transformer Layer and Supplementary Task

In this section, we focus on whether two findings from the experiments on BERT show consistent results in the experiments on T5.

Last Transformer Layer. We evaluate the superiority of distilling the last Transformer layer knowledge in T5 models. Unlike BERT, T5 has an additional Transformer layer of the decoder network and cross-attention (CA). Therefore, we also conduct additional comparisons between the distillation on the decoder network and the distillation on both the encoder and decoder network, as well as the comparison between the last Transformer layer mapping and uniform layer mapping. Furthermore, we examine the effectiveness of the distillation on the cross-attention when we distill the knowledge in the decoder network.

In Figure 12a, 12b, and 12c, the blue boxes, and the orange boxes denote the distillation on the decoder network, the distillation on both the encoder and decoder network, respectively. In most cases, distilling only from the decoder network tends to show higher results than distilling from the encoder and decoder network. In addition, distilling the last Transformer layer shows better performance than the distilling Transformer layers uniformly. Lastly, compared to distilling the self-attention and the cross-attention of the last Transformer decoder layer (green bar in Figure 12), distilling only the self-attention of the last Transformer decoder layer (the first blue bar) shows better performance. In conclusion, We observe that distilling knowledge from only the last layer of the decoder network shows the highest performance across the target tasks. This result is consistent with the previous results of the experiments on BERT.

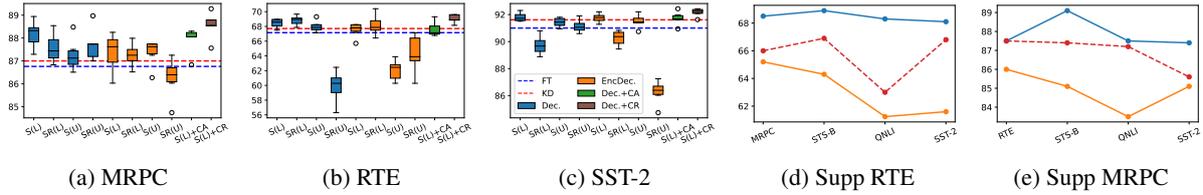


Figure 12: We compare the performance for different layer mapping functions and distillation objectives in (a)-(c). The distillations of the last layer, uniform layer mapping, self-attention, and IR are denoted by L, U, S, and R, respectively. The blue (Enc.) and orange (Dec.) bar denote the application of ILD to the decoder network only and to both encoder and decoder network, respectively. We also denote ILD with CA and CR as the green and brown bars in (a)-(c), respectively. (d)-(e) are results for using ST for TTs of MRPC and RTE. We only distill self-attention of the last layer of the decoder when using ILD with ST and CR

Supplementary Task We further evaluate the effectiveness of the supplementary tasks on ILD for the encoder-decoder models. Figure 12d and 12e summarize the performance of RTE and MRPC tasks, depending on the supplementary task initialization. Blue, red and orange lines denote distilling self-attention of the last Transformer layer, logit-based distillation, and fine-tuning, respectively. Using the distillation on the self-attention of the last Transformer layer, initialization from the supplementary task training shows better performance than PLM initialization regardless of the supplementary task.

F.4 Experimental Results: CR-ILD

In this section, we examine whether our CR-ILD method could mitigate the over-fitting of the student model when the teacher and the student are T5 models. In Figure 12a, 12b, and 12c, the brown box denotes to distill the self attention of last Transformer decoder layer with the consistency regularization, CR-ILD. In order to see the difference according to the presence or absence of the consistency regularization, we compare the brown box and the first blue box, which denotes to distill the self attention of last Transformer decoder layer without CR-ILD. In the all tasks (MRPC, RTE, and SST-2), the consistency regularization boost the performance of the student model. That is, the effect of the consistency regularization is consistent with the result of the experiment on BERT.

Implicit Temporal Reasoning for Evidence-Based Fact-Checking

Liesbeth Allein, Marlon Saelens, Ruben Cartuyvels, Marie-Francine Moens

Department of Computer Science

KU Leuven, Belgium

{liesbeth.allein,ruben.cartuyvels,sien.moens}@kuleuven.be

Abstract

Leveraging contextual knowledge has become standard practice in automated claim verification, yet the impact of temporal reasoning has been largely overlooked. Our study demonstrates that time positively influences the claim verification process of evidence-based fact-checking. The temporal aspects and relations between claims and evidence are first established through grounding on shared timelines, which are constructed using publication dates and time expressions extracted from their text. Temporal information is then provided to RNN-based and Transformer-based classifiers before or after claim and evidence encoding. Our time-aware fact-checking models surpass base models by up to 9% Micro F1 (64.17%) and 15% Macro F1 (47.43%) on the MultiFC dataset. They also outperform prior methods that explicitly model temporal relations between evidence. Our findings show that the presence of temporal information and the manner in which timelines are constructed greatly influence how fact-checking models determine the relevance and supporting or refuting character of evidence documents.¹

1 Introduction

Automatically verifying information and flagging engineered falsities have been high on the political, media, and - subsequently - research agenda for quite some *time* (European Commission, 2022). However, the role of time in machine-assisted fact-checking has been inadequately investigated. Time can affect the veracity of previously uttered claims and the relevance of supporting or refuting evidence. This is evident in research, for example, where newly acquired knowledge may question, confirm, or refute established facts. This study proposes to ground claims and associated evidence in

¹The code of this paper is publicly available: <https://github.com/Marlon668/VerificationClaimsWithTimeAttribution>.

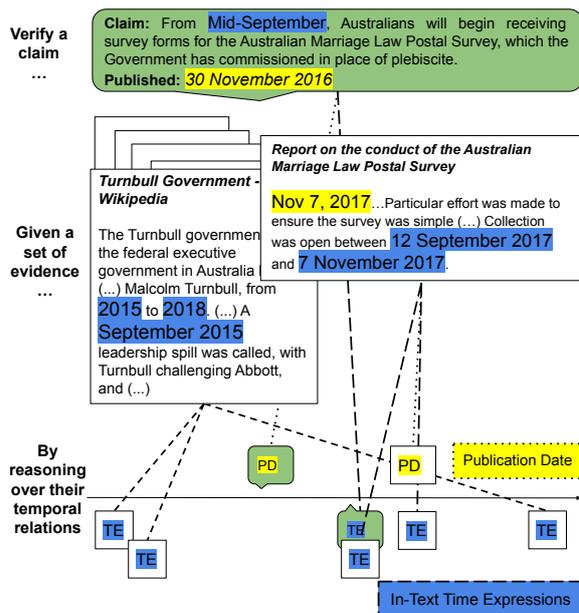


Figure 1: An evidence-based fact-checking model verifies a given claim against a set of Web documents serving as supporting or refuting evidence. In this study, we let the model implicitly reason over the temporal aspects of the claim and evidence, and their relations. For this, both inputs are grounded at two levels on a shared timeline: at the document level using their publication dates (in yellow, dotted line) and at the content-level using time expressions in their text (in blue, dashed line).

time and incorporate temporal reasoning abilities in the claim verification process of computational fact-checking models (Figure 1). Here, temporal reasoning is implicit since the models are not expected to make explicit predictions about time. They instead learn from data how to leverage temporal information.

Grounding a claim or evidence document in time is a complex task. On the one hand, it can be achieved through document-level grounding, which involves positioning the entire document on a timeline based on its publication date. On the other hand, a document may discuss several events that have occurred in the past, present, or future. To fa-

Facilitate more fine-grained grounding on the content level, time expressions in the text are used to place the document on multiple positions on a timeline. Such expressions can be explicit (e.g., 27 June 2022), implicit (e.g., Christmas 2022), and relative (e.g., mid-September), which may require additional temporal information for grounding (Strötgen and Gertz, 2013; Leeuwenberg and Moens, 2019). In this study, we ground claims and evidence on both the document and content level. This is accomplished by extracting and normalising their publication date and in-text time expressions, and subsequently relating them in terms of distance in time. This enables fact-checking models to reason over the temporal relations between a claim and its evidence on more than one level.

Contributions This study demonstrates that reasoning over temporal aspects and relations of claims and evidence not only improves fact-checking models’ prediction performance but also influences their estimation of the relevance and the supporting/refuting character of the evidence. The effects on performance are even reinforced when claims and evidence are grounded at both the document and content level, showing the appropriateness of multi-level temporal reasoning in automated fact-checking.

2 Related Work

Automated fact-checking is usually a two-phase process consisting of claim detection/selection and claim verification (Zeng et al., 2021; Guo et al., 2022). Time is arguably important in both phases. When detecting and ultimately selecting claims to fact-check, fact-checkers heed the current interest of the public in certain topics and election cycles, and rank the claims accordingly (Allein and Moens, 2020). Moreover, many selected claims mention dates or time periods (Hidey et al., 2020). Shaar et al. (2020) looked in the past and filtered out claims that are semantically similar to previously fact-checked claims to expedite the claim selection process.

While evidence-based claim verification has been widely studied (Zhong et al., 2020; Liu et al., 2020; Chen et al., 2021; Si et al., 2021; Jin et al., 2022; Xu et al., 2022; Hu et al., 2022), few studies explicitly focused on incorporating temporal reasoning in the verification process. Zhou et al. (2020) constructed (entity, value, time)-tuples representing supposedly temporal facts and verified

their correctness using probabilistic graphical models. Allein et al. (2021) constrained the evidence ranking in fact-checking models on time using silver-standard evidence rankings respecting four assumptions on temporal relevance. Instead of verifying the temporal correctness of claim tuples or explicitly enforcing time-dependent evidence rankings, we let fact-checking models reason *implicitly* over temporal aspects of claims and evidence in natural language when checking the claims.

3 Task Description

Classifier f takes a textual claim c and an associated set of N text documents $\{e_i\}^N$ serving as evidence of c , and returns a claim veracity label y .

$$f : c, \{e_i\}^N \rightarrow y \quad (1)$$

To allow f to reason over temporal aspects of c and e_i , we extract and normalise publication dates and time expressions in c and e_i , and assign them to time buckets. Temporal representations c_t and $e_{i,t}$ are sequences of time bucket indices and are given as additional input to f :

$$f : c, c_t, \{e_i\}^N, \{e_{i,t}\}^N \rightarrow y \quad (2)$$

4 Two-Level Grounding and Reasoning

To obtain temporal representations c_t and $e_{i,t}$, we ground c and e_i in time by positioning them on a joint timeline using either their *publication date* ($c_t = c_t^{doc}$; $e_{i,t} = e_{i,t}^{doc}$) or *in-text time expressions* ($c_t = c_t^{con}$; $e_{i,t} = e_{i,t}^{con}$). A fact-checking model can then reason over their temporal aspects and relations at the *document level* or the *content level*, respectively (Figure 2).

4.1 Reasoning at Document Level

The publication date of c serves as reference point for grounding e_i . This way, we lay bare the temporal relation between c and e_i at the document level. We adopt the approach of Allein et al. (2021) and compute the distance in days $\Delta_{pub} \in \mathbb{Z}$ between the publication date of c and that of e_i , where $\Delta_{pub} < 0$ indicates that e_i was published before c , $\Delta_{pub} = 0$ indicates that e_i and c were published on the same day, and $\Delta_{pub} > 0$ indicates that e_i was published after c . The publication date of e_i is then assigned to a time bucket $b_{pub} \in T^{doc}$ given Δ_{pub} . Ultimately, the document-level temporal representation of e_i , $e_{i,t}^{doc}$, is a sequence of indices corresponding to b_{pub} in T^{doc} , with $|e_{i,t}^{doc}| = 1$ since e_i

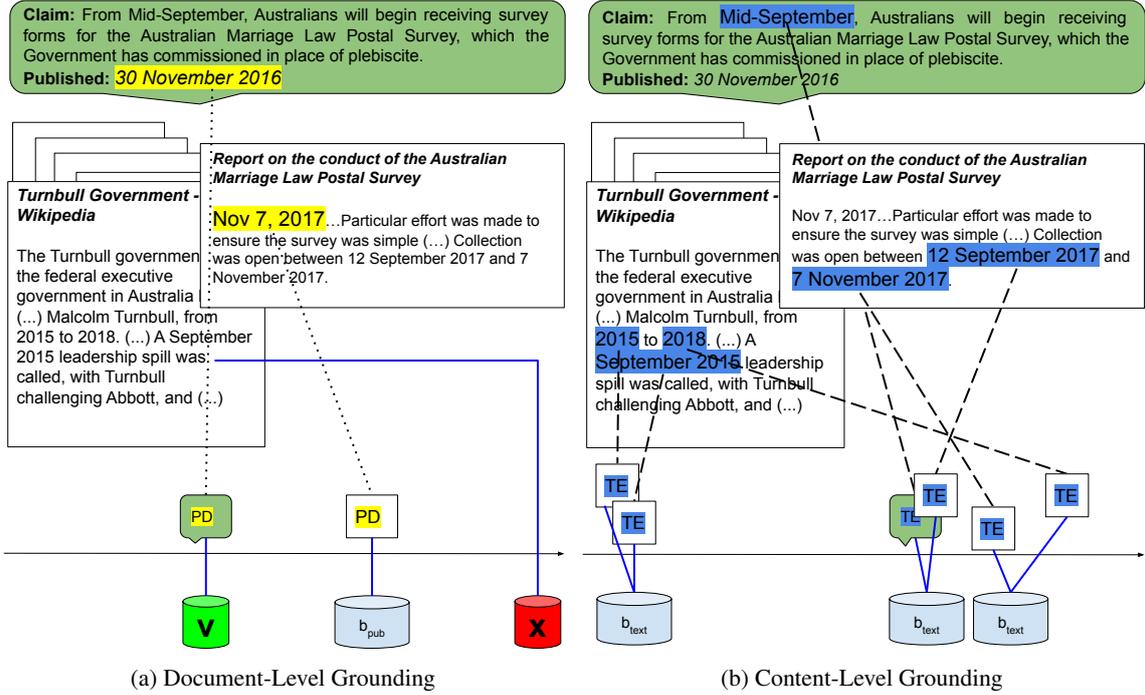


Figure 2: Illustration of two-level grounding: (a) at the document level using publication dates (PD) and (b) at the content level using in-text time expressions (TE). All PD and TE are assigned to time buckets b_{pub} and b_{text} , respectively. \checkmark means that a publication date was found (only for claims) and \times that no publication date was found.

has only one publication date. When a publication date for e_i could not be extracted, $e_{i,t}^{doc}$ corresponds to the index of a dedicated time bucket indicating date unavailability. Lastly, the document-level temporal representation of the claim, c_t^{doc} , merely indicates the availability of a publication date for the claim. We motivate and discuss the choice of T^{doc} in Section 4.3.

4.2 Reasoning at Content Level

While the document-level approach grounds c and e_i as whole documents, the content-level approach places them on various positions on a timeline using time expressions found in their text. Each time expression in e_i and c is first extracted and normalised, and its distance in days $\Delta exp \in \mathbb{Z}$ to the publication date of c is computed. They are then assigned to time buckets $b_{text} \in T^{con}$ given Δexp . The choice of T^{con} is discussed in Section 4.3. The content-level temporal representation of e_i is $e_{i,t}^{con}$ is a sequence of indices where each index corresponds to a $b_{text} \in T^{con}$. The length of $e_{i,t}^{con}$ equals the number of time expressions found in e_i , and the j^{th} element of $e_{i,t}^{con}$ corresponds to the index of the time bucket of the j^{th} time expression in e_i . A time bucket index can occur multiple times in $e_{i,t}^{con}$. The same grounding procedure is applied to obtain

content-level temporal representation c_t^{con} for c . In contrast to c_t^{doc} , c_t^{con} does not merely reflect the availability of a publication date for c but grounds time expressions in the claim text with respect to the claim’s own publication date. The content-level grounding approach allows a fact-checking model to reason over the temporal aspects of the events discussed in e_i and c , and their temporal relation to the publication date of c .

4.3 Creating Time Buckets

Time buckets $b_{pub} \in T^{doc}$ and $b_{text} \in T^{con}$ represent time intervals with respect to the publication date of c (e.g., $b_{pub} = [1, 4]$ indicates that e_i was published between 1 and 4 days after c had been published). Following the cluster hypothesis of [Jardine and van Rijsbergen \(1971\)](#) which states that documents in a cluster contain similar information, the similar information in a bucket is the distance in time to c . For document-level grounding and reasoning, the construction and choice of T^{doc} goes as follows: (1) Δpub for each e_i in the training set is computed; (2) all Δpub are ordered in ascending order; (3) and, finally, all Δpub are subdivided in 20 quantiles, containing a similar number of e_i ($\mu = 8530.5, \sigma = 266.87$). Each quantile represents one bucket b_{pub} . Various numbers of quantiles

were tested, and 20 returned the best performance on the validation set. Three buckets denoting a lacking publication date for e_i , an available publication for c , and a lacking publication date for c are added; hence, $|T^{doc}| = 23$. A similar procedure is applied for constructing T^{con} using Δexp ($|T^{con}| = 24$, $\mu = 13390.75$, $\sigma = 2050.4$). However, no extra buckets b_{text} denoting (un)availability of date are added. An overview of all b_{pub} and b_{text} can be found in Appendix A. Note that the intervals of b_{pub} and b_{text} become smaller when its bounds approach 0, allowing for more fine-grained reasoning for evidence published around or at the same time as the claim. Time buckets approaching 0 (i.e., e_i situates around the same time as c) have smaller intervals than those far from 0, with even a dedicated time bucket for those evidence published or discussing events happening on the same day as the claim. The advantage of using such time buckets is that the model is more robust against bias towards larger buckets. In the fact-checking models, each bucket corresponds to a unique embedding stored in a randomly-initialised time embedding matrix, which is updated during model training.

5 Methodology

5.1 Fact-Checking Model

We take the Joint Veracity Prediction and Evidence Ranking model introduced in [Augenstein et al. \(2019\)](#) as base model (Figure 3). Taking c and e_i represented by their word embeddings $w \in \mathbb{R}^{D_1}$, the text encoder encodes them to their latent representations $h(c)$ and $h(e_i) \in \mathbb{R}^{D_2}$. Metadata m linked to c is encoded in parallel, yielding $g(m)$. Next, $h(c)$, $h(e_i)$, and $g(m)$ are combined into a joint claim-evidence representation s_i using the matching approach introduced by [Mou et al. \(2016\)](#):

$$s_i = [h(c); h(e_i); h(c) - h(e_i); h(c) \cdot h(e_i); g(m)] \quad (3)$$

with $[\cdot]$ denoting concatenation, and $[\cdot]$ the dot product. The evidence scorer projects each s_i to $o_i \in \mathbb{R}$, forming evidence score vector $o \in \mathbb{R}^N$. The label scorer projects each s_i to its label score vector $q_i \in \mathbb{R}^L$ forming scoring matrix $Q \in \mathbb{R}^{N \times L}$, with L the number of veracity labels. $o^T \cdot Q$ gives a final score vector for all labels L , to which a softmax is applied to obtain a probability distribution over all veracity labels.

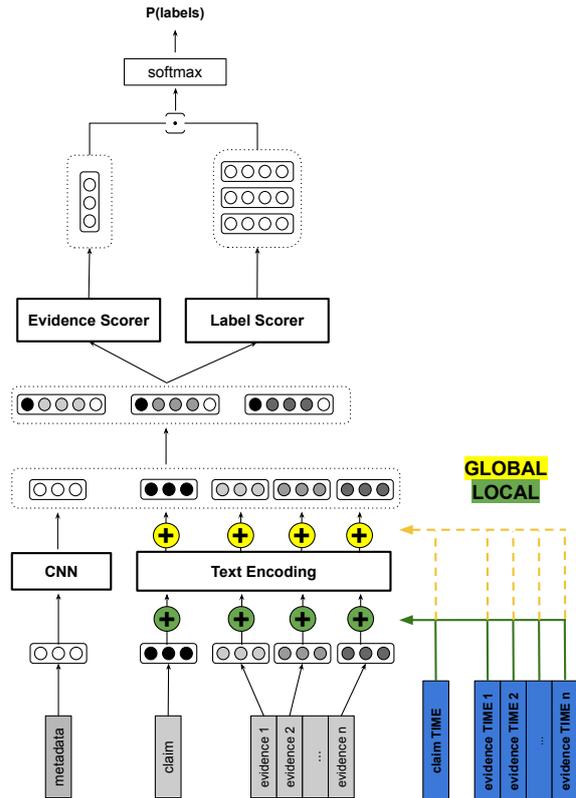


Figure 3: Overview of the fact-checking model, where temporal information on claim and evidence (in blue) is integrated before text encoding (local level; in green) or after text encoding (global level; in yellow).

5.2 Incorporating Temporal Reasoning

Temporal representations c_t and $e_{i,t}$ are transformed to their time embeddings, \hat{c}_t and $\hat{e}_{i,t}$, and given as additional model input. The embedding dimensions depend on the stage at which they are integrated in the model.

Local integration c_t and $e_{i,t}$ are integrated before encoding claim and evidence c and e_i to $h(c)$ and $h(e_i)$. Time embeddings $\hat{c}_t, \hat{e}_{i,t}$ for each time bucket index in c_t and $e_{i,t}$ are taken from the embedding matrix and projected onto the same dimension as the word embeddings $w \in \mathbb{R}^{D_1}$ of tokens in c and e_i using a linear transformation layer l .

For **document-level reasoning** (DL_{loc} , eq. 4), the embeddings (there is max. one publication date; hence one time embedding per document) are prepended to those of c and e_i . These are then sent to the text encoder.

$$\begin{aligned} c &= [l(\hat{c}_t^{doc}); w_0, \dots, w_{|c|}] \\ e_i &= [l(\hat{e}_{i,t}^{doc}); w_0, \dots, w_{|e_i|}] \end{aligned} \quad (4)$$

For **content-level reasoning** (CL_{loc} , eq. 5), local integration is more complex. Firstly, c_t^{con} and $e_{i,t}^{con}$ may refer to more than one time bucket as there may be more than one time expression in c and e_i . Secondly, the position of a time expression and the predicate it belongs to may provide rich information about a mentioned event. We first identify the type of each token in c and e_i (see Table 1).

Position, predicate, and time expression marking									
Tokens	Storm	Al-	berto	expected	to	make	landfall	to-	morrow
Type	O	O	O	O	O	B-PRED	O	B-TIME	TIME
Pos	/	/	/	/	/	+2	/	/	/
Time	/	/	/	/	/	/	/	0	/

Table 1: Additional sentence preprocessing when integrating c_t^{con} and $e_{i,t}^{con}$ at the local level. The predicate (PRED) and the time expressions (TIME) are marked, with B indicating their first token, and the token distance between B-TIME and B-PRED is computed.

We then introduce three new embeddings: predicate embedding $pr \in \mathbb{R}^{D_1}$ marks the predicate, position embedding $po \in \mathbb{R}^{D_1}$ marks the position of the predicate, and expression embedding $te \in \mathbb{R}^{D_1}$ marks the time expression. These additional embeddings are learned during training. The word embedding w of a token in c depends on that token’s type (same for e_i and $e_{i,t}^{con}$):

$$w = \begin{cases} \gamma w + (1 - \gamma)(l(\hat{c}_{t,j}^{con}) + te) & \text{if B-TIME} \\ \gamma w + (1 - \gamma)te & \text{if TIME} \\ \gamma w + (1 - \gamma)(pr + po) & \text{if B-PRED} \\ \gamma w + (1 - \gamma)pr & \text{if PRED} \\ w & \text{otherwise} \end{cases} \quad (5)$$

Embedding $\hat{c}_{t,j}^{con}$ refers to the embedding of time bucket b_{text} to which the j^{th} time expression in c refers.

Global integration c_t and $e_{i,t}$ are integrated after c and e_i have been transformed by the text encoder to their latent representations $h(c)$ and $h(e_i) \in \mathbb{R}^{D_2}$. An embedding for each time bucket in c_t and $e_{i,t}$ is taken and projected onto the same embedding space \mathbb{R}^{D_2} using a linear transformation layer k . If c_t or $e_{i,t}$ are represented by more than one time bucket, the embeddings are averaged. Fusion is performed using a weighted sum. For **document-level reasoning** (DL_{glob}):

$$\begin{aligned} h(c) &= \alpha h(c) + (1 - \alpha)k(\hat{c}_t^{doc}) \\ h(e_i) &= \alpha h(e_i) + (1 - \alpha)k(\hat{e}_{i,t}^{doc}) \end{aligned} \quad (6)$$

And for **content-level reasoning** (CL_{glob}):

$$\begin{aligned} h(c) &= \alpha h(c) + (1 - \alpha)\text{Avg}(k(\hat{c}_t^{con})) \\ h(e_i) &= \alpha h(e_i) + (1 - \alpha)\text{Avg}(k(\hat{e}_{i,t}^{con})) \end{aligned} \quad (7)$$

with Avg the average. We also experiment with a **combination of document-level and content-level reasoning** ($DL+CL_{glob}$, eq. 8) where temporal information from both levels is provided to the model:

$$\begin{aligned} h(c) &= \alpha h(c) + \beta k(\hat{c}_t^{doc}) \\ &+ (1 - \alpha - \beta)\text{Avg}(k(\hat{c}_t^{con})) \\ h(e_i) &= \alpha h(e_i) + \beta k(\hat{e}_{i,t}^{doc}) \\ &+ (1 - \alpha - \beta)\text{Avg}(k(\hat{e}_{i,t}^{con})) \end{aligned} \quad (8)$$

6 Experiments

6.1 Dataset

Experiments are conducted on MultiFC² (Augenstein et al., 2019), a large-scale dataset containing 34,924 English claims from various fact-checking websites (= ‘domains’) where each claim is associated with at most 10 *a posteriori* retrieved Web documents (319,721 documents in total). It also provides metadata on speaker, category, tags, and linked entities regarding the claim. We refer to Augenstein et al. (2019) for a more detailed description of the data. Although other datasets for fact-checking have been proposed (Zeng et al., 2021), they either lack naturally occurring claims, publication dates, or multiple evidence documents (Thorne et al., 2018; Jiang et al., 2020; Ostrowski et al., 2021; Schuster et al., 2021). Nonetheless, the large size, wide diversity of topic and data sources, and high quality of the MultiFC dataset should be sufficient for showcasing the appropriateness of our approach.

6.2 Time Extraction and Normalisation

In this section, we discuss the procedure for extracting and normalising publication dates and in-text time expressions.

6.2.1 Publication Dates

The dataset provides the publication date of a claim as structured metadata. The date is represented as Year-Month-Day using rule-based temporal tagger HeidelTime (Strötgen and Gertz, 2013). The publication date of an evidence document, however,

²The data is publicly available on CodaLab.

is not given in the metadata. Since its publication date is often communicated before the ellipsis ('...') at the beginning of its text, we can automatically extract the date from the text (Allein et al., 2021). If we cannot extract a date at that position, we look for occurrences of 'published' or 'posted' in combination with a date. We again use HeidelTime for structuring the publication dates. In total, we obtain a publication date for 34,808 (99.67%) claims and 213,165 (66.67%) evidence documents.

6.2.2 In-Text Time Expressions

Extracting and normalising in-text time expressions is more challenging as they can be implicit or relative. Since in-text time expressions are usually not annotated in datasets used for fact-checking, we need to reside to pretrained methods for extracting them. We implement the Open Information Extraction (OIE) model of Stanovsky et al. (2018), which parses a sentence and labels its arguments. In this work, we focus on temporal arguments (*ArgM - TMP*). Since inaccurate use or absence of capital letters has been shown to decrease the performance of OIE models (Alam and Awan, 2018), the OIE model is expected to return a high number of inaccurate parses for capitalised news headlines – which make up a large portion of the claims in the data. We therefore implement a pretrained Named Entity Recognition (NER) model (Peters et al., 2017) to first detect people, locations, and organisations in the text. Then, the first token of each entity is capitalised while all other tokens are lowercased. Although capitalised temporal expressions such as weekdays and holidays are automatically lowercased too, we observed a higher quality of OIE parses when adopting this approach. We normalise the extracted temporal expressions using HeidelTime. The document creation time (DCT) of a piece of information, in this study the publication date, is used as reference point for normalising in-text temporal expressions. In total, we obtain 321,278 in-text time expressions.

Quality assessment Implementing pretrained extraction and normalisation models inevitably introduces noise in the data. We therefore manually assess the quality of the NER, OIE, and HeidelTime models to ensure that the noise is limited. The assessment is performed on a randomly selected set of 10 claims and their accompanying evidence documents (104 in total) from the dataset, and performance is measured using precision (P),

recall (R), and F1. Regarding NER, we investigate whether all entities have been recognised and completely extracted. The label correctness does not need to be evaluated. NER performance is 0.9054/0.9134/0.9094 (P/R/F1). For the OIE task, we assess whether all temporal expressions have been correctly extracted and parsed. OIE performance is 0.9608/0.5568/0.7050 (P/R/F1), indicating that while quite some time-related expressions have not been extracted, those found have been correctly parsed. Lastly, we evaluate the normalisation of the found expressions: HeidelTime performance is 0.9736/0.8409/0.9024 (P/R/F1). In all, we deem the quality of the pretrained extraction and normalisation models sufficiently high.

6.3 Experimental Setup

Hyperparameter settings Both c and e_i are tokenised³ and represented using word embeddings (size = 300 (BiLSTM); 768 (DistilRoBERTa)). We experiment with two neural text encoders for encoding c and e_i : a two-layered bidirectional LSTM with skip-connections (dropout = 0.1, hidden size = 128) and a pretrained Sentence-DistilRoBERTa, which is a faster, distilled version of Sentence-RoBERTa (Sanh et al., 2019; Reimers and Gurevych, 2019). For sake of brevity, we continue to refer to this model as RoBERTa. Metadata m is represented as a one-hot vector and encoded by a CNN (filter size = 3, kernel size = 3) with ReLU activation and 1D max pooling. The label scorer consists of two fully-connected layers (hidden size = 100; 50), both with ReLU activation. The evidence scorer is a fully-connected layer (hidden size = 100) with Leaky ReLU activation. All parameters except those of the pretrained RoBERTa model are initialised following a Xavier Uniform distribution. More detailed settings for reproducing the experiments, such as hyperparameter tuning, is provided in Appendix B.

Pretraining and fine-tuning The experiments are conducted on the disjunct, label-stratified train (80%), validation (10%), and test set (10%) provided by Augenstein et al. (2019). We adopt the pretraining and fine-tuning setup of Allein et al. (2021) to ensure transparent comparison. During pretraining, the model is trained on all 26 fact-checking domains where each domain is only presented once in each epoch (batch size = 32 (BiL-

³Huggingface implementation of the DistilRoBERTa tokenizer: [sentence-transformers/all-distilroberta-v1](https://huggingface.co/distilbert/distilbert-v1).

	BiLSTM			RoBERTa		
	Micro F1	Macro F1	Fusion Weights	Micro F1	Macro F1	Fusion Weights
Base	.5520 (.0023)	.3239 (.0064)	-	.6952 (.0195)	.5532 (.0246)	-
DL _{loc}	.5501 (.0095)	.3343 (.0277)	-	.5640 (.0084)	.3357 (.0174)	-
DL _{glob}	.6006 (.0090)	.4271 (.0107)	$\alpha = 0.90$.6973 (.0439)	.5608 (.0488)	$\alpha = 0.75$
CL _{loc}	.6098 (.0028)	.4491 (.0120)	$\gamma = 0.50$.5685 (.0075)	.3601 (.0090)	$\gamma = 0.10$
CL _{glob}	.6089 (.0167)	.4425 (.0167)	$\alpha = 0.25$.6882 (.0208)	.5744 (.0376)	$\alpha = 0.10$
DL+CL _{glob}	.6417 (.0033)	.4743 (.0080)	$\alpha = 0.20$ $\beta = 0.40$.6947 (.0135)	.5739 (.0332)	$\alpha = 0.20$ $\beta = 0.20$

Table 2: Average test results over three (BiLSTM) and two (RoBERTa) runs - with standard deviation in brackets - aggregated over all 26 fact-checking domains. Experiments are conducted for document-level (DL) and content-level (CL) temporal reasoning, where temporal information is integrated before (*loc*) or after (*glob*) encoding.

STM); 16 (RoBERTa)), mitigating model bias towards larger domains. After each epoch, the batch order is randomly shuffled, and Adam with linear scheduler ($\text{lr} = 1e^{-4}$ (BiLSTM)) or RMSprop ($\text{lr} = 2e^{-4}$ (RoBERTa)) optimizes the model parameters using the cross-entropy loss on the prediction output. The best-performing model for each fact-checking domain is selected based on the validation loss. Each domain-specific model is then fine-tuned on only data from that domain and the best-performing model is again selected based on the validation loss.

7 Results

Table 2 reports model performance on the test set, aggregated over all domains, in terms of Micro F1 and Macro F1⁴. The results show that the effect of temporal reasoning depends on (a) the level at which temporal information is integrated in the model (global vs. local), (b) the grounding/reasoning level (document vs. content), and (c) the model architecture (BiLSTM vs. RoBERTa). Regarding the integration level, global integration (*glob*) substantially surpasses local integration (*loc*) for document-level reasoning (both models; .5501/.3343 \rightarrow .6006/.4271 [BiLSTM]; .5640/.3357 \rightarrow .6973/.5608 [RoBERTa]) and content-level reasoning (.5685/.3601 \rightarrow .6882/.5744 [RoBERTa]). Regarding the temporal grounding and reasoning level, the results show that the combination setup where claim and evidence are grounded at both the document and content level (DL+CL) yields the overall highest performance for BiLSTM (.6417/.4743), while marginally improving RoBERTa by 2% Macro F1 (.5739). Lastly, temporal reasoning ap-

pears to impact the prediction performance of the less parameterised BiLSTM model more strongly than that of the Transformer-based RoBERTa model: .5520/.3239 \rightarrow .6417/.4743 [BiLSTM]; .6952/.5532 \rightarrow .6947/.5739 [RoBERTa]. A similar effect was observed by [Allein et al. \(2021\)](#), who explicitly modeled temporal relations between a claim and its evidence by constraining model parameters on evidence rankings following various assumptions on temporal relevance. This could be attributed to the expressive power of large pre-trained Transformers-based language models and the orders of magnitude of their pretraining set size.

Table 3 shows the comparison between our best performing set-up with the baseline from [Augenstein et al. \(2019\)](#) and the model with explicit temporal reasoning from [Allein et al. \(2021\)](#). Overall, our approach outperforms the baseline and the explicit temporal reasoning approach, especially on the Macro F1-score. This demonstrates the appropriateness of our implicit, two-level temporal reasoning method over an approach without temporal reasoning and one that explicitly models temporal relations using only publication dates.

8 Discussion

Weighting text and time We ran experiments with various weight values (α, β, γ) for combining the text features of a claim and its evidence with their temporal information⁵. Table 2 presents the best-performing weight values for each setting based on the validation loss. When reasoning over the document-level temporal relations (DL), the results suggest that higher importance should be attributed to the text of the claim and its evidence

⁴Computed using the [scikit-learn Python package](#).

⁵A full overview of tested values and the tuning approach is provided in [Appendix A](#).

	BiLSTM		Transformer	
	Micro F1	Macro F1	Micro F1	Macro F1
No temporal reasoning (Augenstein et al., 2019)	.5520	.3239	.6952	.5532
Explicit temporal reasoning (Allein et al., 2021)	.6265	.3673	.5921 [†]	.3135 [†]
Implicit temporal reasoning (Ours)	.6417	.4743	.6947	.5739

Table 3: Results of our implicit temporal reasoning approach vs. the baseline results of Augenstein et al. (2019) (our implementation) and the explicit temporal reasoning method of Allein et al. (2021), with a BiLSTM and a Transformer text encoder. [†]: DistilBERT (Sanh et al., 2019) instead of RoBERTa.

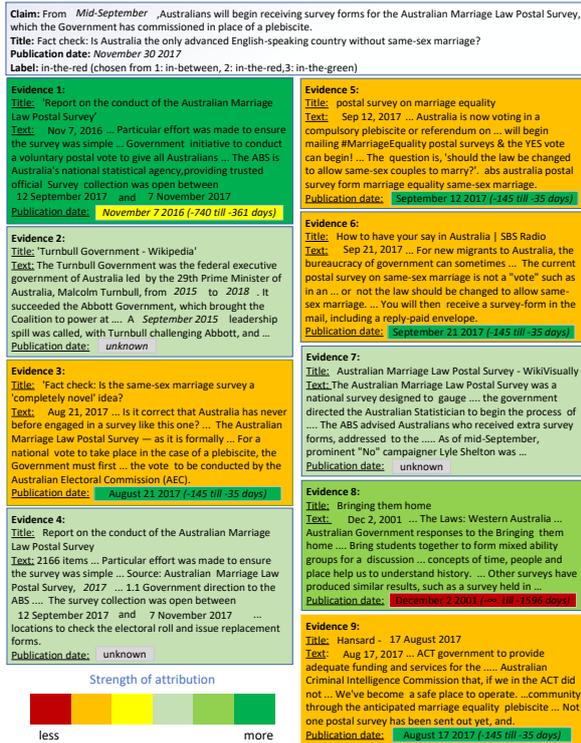
rather than to their temporal information. However, this is the opposite when reasoning at the content level (CL). The combined setup (DL+CL) aligns with (CL) by attributing more importance to time than text. This suggests that specially in-text time expressions carry useful information for fact-checking a claim.

Impact on evidence relevance and label scores

We analyse how and to which extent temporal reasoning influences a model’s assessment of the relevance (o_i) and supporting/refuting nature (q_i) of evidence in relation to a claim. Since the model computes o_i and q_i for each evidence document associated with the claim, a ranking of all evidence can be derived based on either o_i or q_i . We then measure the difference in such rankings between the base and the best-performing temporal models. Following Allein et al. (2021), we rely on the Spearman’s rank correlation r_s , which is a non-parametric, distribution-independent metric for computing the correlation between two rankings. The correlation between the base model and the temporal reasoning models with regard to evidence relevance ranking is very weak, with $0 < |r_s| < 0.19$ for both BiLSTM and RoBERTa. Also between the temporal models, those correlations are generally very weak. Interestingly, the impact of implicit temporal reasoning on a fact-checking model’s estimation of evidence relevance is arguably as strong as when performing explicit temporal reasoning (Allein et al., 2021). The correlations fall within the range of $.17 < |r_s| < .24$. The correlations regarding label scoring (q_i) are comparable to those for evidence ranking, ranging from weak ($0.2 < |r_s| < 0.39$) and to very weak. We can thus conclude that a model’s estimation of the relevance and supporting/refuting nature of evidence documents is strongly influenced not only by the ability to reason over time, but also by the way a claim and its evidence are grounded on a timeline.

Importance of time in final prediction While we have shown that temporal reasoning strongly influences relevance estimations and label scores per evidence document, we now measure how much the time-aware fact-checking models rely on temporal information for their final veracity predictions. For this, we attribute the prediction of the models to the input using integrated gradients (Sundararajan et al., 2017). This attribution technique measures the attribution strength of text and time features on the final prediction. We focus on the base BiLSTM model and its best-performing temporal variants. Given the high dimensionality of text and time embeddings, the attribution strengths across all dimensions are summed to obtain a total attribution value for claim, evidence, and time (c_t and $e_{i,t}$). Figure 4 illustrates the attribution values of a single data entry and presents the ranking of evidence text and time according to their attribution strength. The models typically attributed the prediction to both the claim and evidence, with a stronger emphasis on the collected evidence than on the claim. However, when time information was introduced, the attribution strength of claim and evidence texts strongly decreased, especially when evidence was grounded at the content level (CL/DL+CL). This indicates that time indeed influences model prediction.

Interestingly, the attribution ranking of temporal information was found to be distinct from that of the content, as demonstrated by the example in Figure 4. The publication dates that are closer to that of the claim obtain higher attribution strength than those far from the claim. In line with this, statistical correlation testing between $e_{i,t}$ and label scores q_i - where each label score in q_i for $e_{i,t}$ in the same bucket is compared to the label score in q_i for $e_{i,t}$ in different buckets - show that evidence contained within the same time bucket tend to prefer the same prediction labels as their label rankings strongly correlate ($\rho = 0.7$). We can thus conclude that time



(a) Ranking of evidence by attribution strength in terms of text and publication date (DL reasoning).

	Base	DL	CL	DL+CL
Label distribution	(1) $5.3e^{-4}$ (2) $2.5e^{-3}$ (3) .996	(1) $1.1e^{-7}$ (2) .530 (✓) (3) .470	(1) .076 (2) .172 (3) .752	(1) .174 (2) .266 (3) .560
Claim (text)	16.029	2.688	0.0613	0.0049
Claim (PD)	-	0.994	-	0.0296
Claim (TE)	-	-	0.0899	0.0441
Evidence (text)	5.279	0.4434	0.0007	0.001
Evidence (PD)	-	0.3213	-	0.008
Evidence (TE)	-	-	0.005	0.008

(b) Predicted label distribution and absolute attribution strengths. Note that strengths for evidence are for a single evidence document.

Figure 4: Illustration of BiLSTM (*glob*) attribution strengths for an example taken from MultiFC.

influences both interim and final prediction.

9 Conclusion

Grounding claims and associated evidence documents on a shared timeline and implicitly reasoning over their temporal relations noticeably improves the verification performance of automated fact-checking models. Time plays a dual role in this process, serving both as a source of information for verifying claims, as well as influencing the evaluation of the relevance and supporting or refuting nature of evidence documents. Further research may look into integrating temporal reasoning in claim detection and evidence retrieval processes

or implementing even more sophisticated temporal reasoning during claim verification by examining the temporality of events discussed in a claim and their relation to the evidence.

Limitations

The limitations of this work mainly originate from the data and the use of pretrained models for grounding claims and evidence documents in time. Since the evidence documents were retrieved after the claim had been fact-checked by giving the claim verbatim to a search engine and selecting the first ten search results, their quality and relevance to the claim is not ensured. As a result, evidence-based fact-checking models risk relying on spurious signals in the evidence documents for predicting a claim's veracity. Moreover, the evidence documents are presented as short snippets which only reflect small parts of the full Web documents. This not only affects content representation, but it also limits temporal information extraction since many time expressions may have been omitted from the shortened text. Regarding temporal information extraction and normalisation, we had to rely on pretrained models to obtain temporal representations of claims and its associated evidence documents. This not only introduces noise in the input data, but also requires time-expensive preprocessing.

Ethics Statement

Automated fact-checking technology aims to assist people in distinguishing between verified and unverified content in professional contexts and during their daily information consumption. Nevertheless, the fact-checking models constructed in this paper - like all fact-checking models - should be deployed with caution and its predictions should never be taken as final without further human evaluation. Computational predictions are anything but flawless, and incorrect predictions may unjustly discredit the person or group who uttered the fact-checked statement(s).

Acknowledgements

This work was realised with the collaboration of the European Commission Joint Research Centre under the Collaborative Doctoral Partnership Agreement No 35332. The scientific output expressed does not imply a policy position of the European Commission. Neither the European Commission nor

any person acting on behalf of the Commission is responsible for the use which might be made of this publication. The research leading to this paper also received funding from the European Research Council (ERC) under Grant Agreement No. 788506. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

References

- Talha Mahboob Alam and Mazhar Javed Awan. 2018. Domain analysis of information extraction techniques. *International Journal of Multidisciplinary Science and Engineering*, 9(6).
- Liesbeth Allein, Isabelle Augenstein, and Marie-Francine Moens. 2021. [Time-aware evidence ranking for fact-checking](#). *Journal of Web Semantics*, 71:100663.
- Liesbeth Allein and Marie-Francine Moens. 2020. [Checkworthiness in automatic claim detection models: Definitions and analysis of datasets](#). In *Multidisciplinary International Symposium on Disinformation in open online media*, pages 1–17. Springer.
- Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. [MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.
- Chonghao Chen, Fei Cai, Xuejun Hu, Jianming Zheng, Yanxiang Ling, and Honghui Chen. 2021. [An entity-graph based reasoning method for fact verification](#). *Information Processing & Management*, 58(3):102472.
- European Commission. 2022. [Funded projects in the fight against disinformation](#).
- Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. [A survey on automated fact-checking](#). *Transactions of the Association for Computational Linguistics*, 10:178–206.
- Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. [DeSePtion: Dual sequence prediction and adversarial examples for improved fact-checking](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8593–8606, Online. Association for Computational Linguistics.
- Nan Hu, Zirui Wu, Yuxuan Lai, Xiao Liu, and Yansong Feng. 2022. [Dual-channel evidence fusion for fact verification over texts and tables](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5232–5242, Seattle, United States. Association for Computational Linguistics.
- Nick Jardine and Cornelis Joost van Rijsbergen. 1971. [The use of hierarchic clustering in information retrieval](#). *Information storage and retrieval*, 7(5):217–240.
- Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. [HoVer: A dataset for many-hop fact extraction and claim verification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.
- Yiqiao Jin, Xiting Wang, Ruichao Yang, Yizhou Sun, Wei Wang, Hao Liao, and Xing Xie. 2022. [Towards fine-grained reasoning for fake news detection](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5746–5754.
- Artuur Leeuwenberg and Marie-Francine Moens. 2019. [A survey on temporal reasoning for temporal information extraction from text](#). *Journal of Artificial Intelligence Research*, 66:341–380.
- Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. [Fine-grained fact verification with kernel graph attention network](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online. Association for Computational Linguistics.
- Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. [Natural language inference by tree-based convolution and heuristic matching](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany. Association for Computational Linguistics.
- Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2021. [Multi-hop fact checking of political claims](#). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, pages 3892–3898.
- Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. [Semi-supervised sequence tagging with bidirectional language models](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of the 2019 Conference on Empirical*

- Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pages 3980–3990. Association for Computational Linguistics.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. **Get your vitamin C! robust fact verification with contrastive evidence**. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.
- Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. **That is a known lie: Detecting previously fact-checked claims**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.
- Jiasheng Si, Deyu Zhou, Tongzhe Li, Xingyu Shi, and Yulan He. 2021. **Topic-aware evidence reasoning and stance-aware aggregation for fact verification**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1612–1622, Online. Association for Computational Linguistics.
- Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. **Supervised open information extraction**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.
- Jannik Strötgen and Michael Gertz. 2013. **Multilingual and cross-domain temporal tagging**. *Language Resources and Evaluation*, 47(2):269–298.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. **Axiomatic attribution for deep networks**. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. **FEVER: a large-scale dataset for fact extraction and VERification**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.
- Weizhi Xu, Junfei Wu, Qiang Liu, Shu Wu, and Liang Wang. 2022. **Evidence-aware fake news detection with graph neural networks**. In *Proceedings of the ACM Web Conference 2022*, pages 2501–2510.
- Xia Zeng, Amani S Abumansour, and Arkaitz Zubiaga. 2021. **Automated fact-checking: A survey**. *Language and Linguistics Compass*, 15(10):e12438.
- Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. **Reasoning over semantic-level graph for fact checking**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180, Online. Association for Computational Linguistics.
- Yang Zhou, Tong Zhao, and Meng Jiang. 2020. **A probabilistic model with commonsense constraints for pattern-based temporal fact extraction**. In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, pages 18–25, Online. Association for Computational Linguistics.

A Time Buckets

Table 4 presents an overview of time buckets b_{pub} with their interval bounds used for document-level grounding, while Table 5 presents time buckets b_{text} with their interval bounds used for content-level grounding.

B Reproducibility Settings

This section contains settings for reproducing the experiments in this paper.

Computing infrastructure The BiLSTM models were trained on a Skylake processor type with one compute node, 9 cores per node, one GPU (GPU partition of Skylake) and 5 GB memory per core. The DistilRoBERTa models were trained on a Cascadelake processor type with one compute node with 4 cores per node, one GPU and 5 GB memory per core.

Average runtime Preprocessing, i.e., extraction of timex annotations via Heideltime, open information extraction (where before this a correction of uppercase characters is done via Named Entity Recognition), and construction of the dataset where claims and evidence are already put into buckets and the predicates and timexes are marked in the text of all the data took approximately 150 hours. Training a BiLSTM model for each domain took on average 45 hours, while a DistilRoBERTa model took 72 hours.

Overview of time buckets for document-level grounding and reasoning: b_{pub}		
Start	End	Number of evidence documents
∞ days before the claim	1596 days before the claim	8536
1596 days before the claim	741 days before the claim	8547
740 days before the claim	361 days before the claim	8528
360 days before the claim	146 days before the claim	8517
145 days before the claim	35 days before the claim	8626
34 days before the claim	4 days before the claim	8962
3 days before the claim	1 day before the claim	7549
on the same day as the claim	on the same day as the claim	8963
1 day after the claim	4 days after the claim	8735
5 days after the claim	24 days after the claim	8548
25 days after the claim	85 days after the claim	8345
86 days after the claim	187 days after the claim	8534
188 days after the claim	325 days after the claim	8551
326 days after the claim	498 days after the claim	8515
499 days after the claim	736 days after the claim	8502
737 days after the claim	1061 days after the claim	8533
1062 days after the claim	1436 days after the claim	8529
1437 dagen na de claim	1997 days after the claim	8537
1998 days after the claim	2605 days after the claim	8531
2606 days after the claim	∞ days after the claim	8522

Table 4: Overview of time buckets b_{pub} with their interval bounds.

Overview of time buckets for content-level grounding and reasoning: b_{text}		
Start	End	Number of evidence documents
∞ days before the claim	18172 days before the claim	12853
18171 days before the claim	6295 days before the claim	12851
6294 days before the claim	2928 days before the claim	12856
2927 days before the claim	1678 days before the claim	12862
1677 days before the claim	989 days before the claim	12855
988 days before the claim	569 days before the claim	12863
568 days before the claim	323 days before the claim	12833
322 days before the claim	145 days before the claim	12935
144 days before the claim	42 days before the claim	12771
41 days before the claim	6 days before the claim	13191
5 days before the claim	1 day before the claim	13269
on the same day as the claim	on the same day as the claim	22966
1 day after the claim	8 days after the claim	15135
9 days after the claim	42 days after the claim	12665
43 days after the claim	124 days after the claim	12832
125 days after the claim	241 days after the claim	12739
242 days after the claim	378 days after the claim	12888
379 days after the claim	581 days after the claim	12828
582 days after the claim	834 days after the claim	12852
835 days after the claim	1178 days after the claim	12862
1179 days after the claim	1582 days after the claim	12834
1583 days after the claim	2134 days after the claim	12848
2135 days after the claim	2734 days after the claim	12848
2735 days after the claim	∞ days after the claim	12842

Table 5: Overview of time buckets b_{text} with their interval bounds.

Number of model parameters BiLSTM: 16,129,125 learnable parameters per model; DistilRoBERTa: 82,933,601 learnable parameters per model.

Number of training and evaluation runs Without parameterisation by α , β , and γ : 150 epochs pretraining, 100 epochs fine-tuning (both BiLSTM and DistilRoBERTa). With parameterisation: 600 epochs pretraining, 300 epochs fine-tuning (BiLSTM); 800 epochs pretraining, 300 epochs fine-tuning (DistilRoBERTa).

Hyperparameter bounds We *manually* tested following combinations for α when integrating the time attribution vectors at the global level for document-level (DL) or content-level reasoning: $\alpha \in \{0.10, 0.25, 0.50, 0.75, 0.90\}$. Final α -values: BiLSTM (DL_{glob}): $\alpha = 0.10$; BiLSTM (CL_{glob}): $\alpha = 0.25$; DistilRoBERTa (DL_{glob}): $\alpha = 0.75$ (see Figure 5); DistilRoBERTa (CL_{glob}): $\alpha = 0.10$. We tested following combinations for γ when integrating the time attribution vectors at the local level for content-level reasoning (CL): $\gamma \in \{0.10, 0.25, 0.50, 0.75, 0.90\}$. Final γ -values: BiLSTM (CL_{loc}): $\gamma = 0.50$; DistilRoBERTa (CL_{loc}): $\gamma = 0.10$. We tested following combinations for α and β when grounding the time attribution vectors at both the document and content level (DL+CL): $[(\alpha = 0.20, \beta = 0.20), (\alpha = 0.20, \beta = 0.35), (\alpha = 0.20, \beta = 0.40), (\alpha = 0.20, \beta = 0.55), (\alpha = 0.20, \beta = 0.60), (\alpha = \frac{1}{3}, \beta = \frac{1}{3}), (\alpha = 0.35, \beta = 0.20), (\alpha = 0.35, \beta = 0.55), (\alpha = 0.40, \beta = 0.20), (\alpha = 0.40, \beta = 0.40), (\alpha = 0.55, \beta = 0.20), (\alpha = 0.55, \beta = 0.35), (\alpha = 0.60, \beta = 0.20)]$. Final α - and β -values: BiLSTM (DL+CL_{glob}): $(\alpha = 0.20, \beta = 0.40)$; DistilRoBERTa (DL+CL_{glob}): $(\alpha = 0.20, \beta = 0.20)$. We performed a hyperparameter search trial of 100 epochs pretraining for each combination of hyperparameters. The criteria used to select the final hyperparameter values are the prediction performance (Micro/Macro F1) on the validation loss and the evolution of the validation loss (visualised on a plot, see Figure 5).

Other parameters tested

- Without linear scheduler;
- With linear scheduler with warm up;
- With linear learning scheduler;

- Learning rates: 0.001, 0.005, 0.0002 (only for RMSprop), 0.0001, 0.00001 (for pretraining and fine-tuning);
- Adam, RMSProp (Only BiLSTM), AdamW (only DistilRoBERTa);
- With weight decay: 0.001, 0.0001;
- Without weight decay.

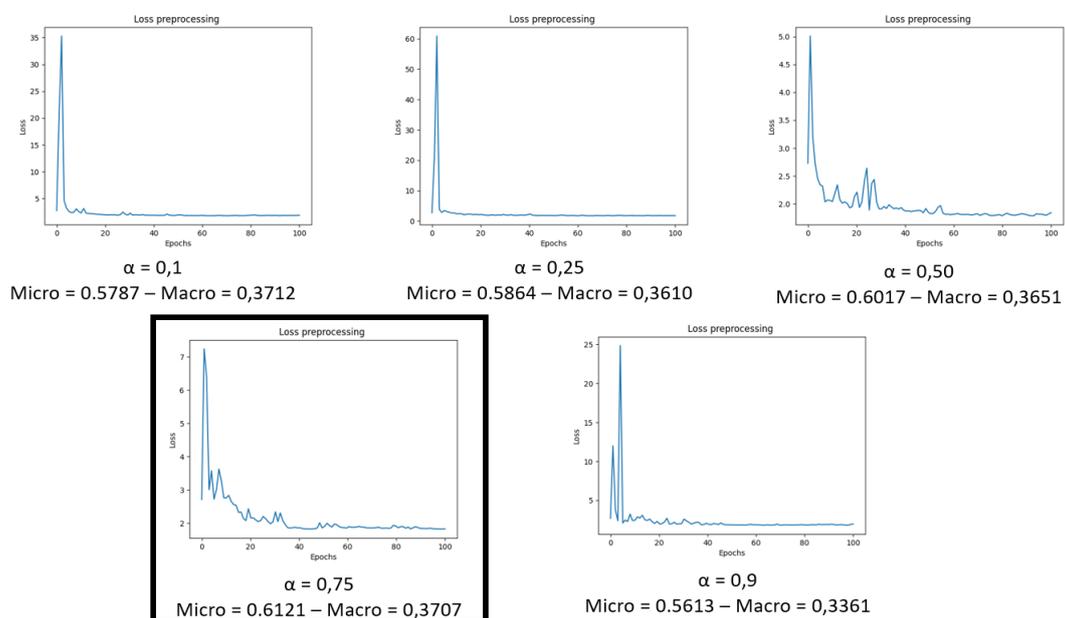


Figure 5: Tuning α for DistilRoBERTa (DL_{glob}) based on the prediction performance on the validation set (metrics: Micro/Macro F1) and the validation loss.

Active PETs: Active Data Annotation Prioritisation for Few-Shot Claim Verification with Pattern Exploiting Training

Xia Zeng, Arkaitz Zubiaga
Queen Mary University of London
{x.zeng, a.zubiaga}@qmul.ac.uk

Abstract

To mitigate the impact of the scarcity of labelled data on fact-checking systems, we focus on few-shot claim verification. Despite recent work on few-shot classification by proposing advanced language models, there is a dearth of research in data annotation prioritisation that improves the selection of the few shots to be labelled for optimal model performance. We propose Active PETs, a novel weighted approach that utilises an ensemble of Pattern Exploiting Training (PET) models based on various language models, to actively select unlabelled data as candidates for annotation. Using Active PETs for few-shot data selection shows consistent improvement over the baseline methods, on two technical fact-checking datasets and using six different pretrained language models. We show further improvement with Active PETs-o, which further integrates an oversampling strategy. Our approach enables effective selection of instances to be labelled where unlabelled data is abundant but resources for labelling are limited, leading to consistently improved few-shot claim verification performance.¹

1 Introduction

As a means to mitigate online misinformation, research in automated fact-checking has experienced a recent surge of interest. Research efforts have resulted in survey papers covering different perspectives (Thorne and Vlachos, 2018; Kotonya and Toni, 2020; Nakov et al., 2021; Zeng et al., 2021; Guo et al., 2022) and novel datasets with enriched features (Augenstein et al., 2019; Chen et al., 2019; Ostrowski et al., 2021; Jiang et al., 2020; Schuster et al., 2021; Aly et al., 2021; Saakyan et al., 2021). Recent work has addressed various challenges, e.g. generating and utilising synthetic data (Atanasova et al., 2020; Pan et al., 2021; Hatua et al., 2021), joint verification over text and tables (Schlichtkrull

et al., 2021; Kotonya et al., 2021), investigating domain adaptation (Liu et al., 2020; Mithun et al., 2021), achieving better evidence representations and selections (Ma et al., 2019; Samarinas et al., 2021; Si et al., 2021; Bekoulis et al., 2021), and performing subtasks jointly (Yin and Roth, 2018; Jiang et al., 2021; Zhang et al., 2021a).

As a core component of a fact-checking system, a claim validation pipeline consists of document retrieval, rationale selection and claim verification (Zeng et al., 2021). Our main focus here is claim verification, the task of assessing claim veracity with retrieved evidence. It is typically treated as a natural language inference (NLI) task: given a claim and an evidence, the aim is to predict the correct veracity label out of “Support”, “Neutral” and “Contradict”. Substantial improvements have been achieved in the performance of claim validation models when a considerable amount of training data is available (Pradeep et al., 2021; Li et al., 2021; Zeng and Zubiaga, 2021; Zhang et al., 2021b; Wadden et al., 2021). However, where new domains needing fact-checking emerge, collecting and annotating new relevant datasets can carry an impractical delay. Availability of unlabelled data can often be abundant, but given the cost and effort of labelling this data, one needs to be selective in labelling a small subset. In these circumstances, rather than randomly sampling this subset, we propose to optimise the selection of candidate instances to be labelled through active learning, such that it leads to overall improved few-shot performance.

To the best of our knowledge, our work represents the first such effort in proposing an approach leveraging an active learning strategy for the claim verification problem, as well as the first in furthering Pattern Exploiting Training (PET) with an active learning strategy. To achieve this, we propose Active PETs, a novel methodology that enables the ability to leverage an active learning strategy

¹Our code is available here: https://github.com/XiaZeng0223/active_pets.

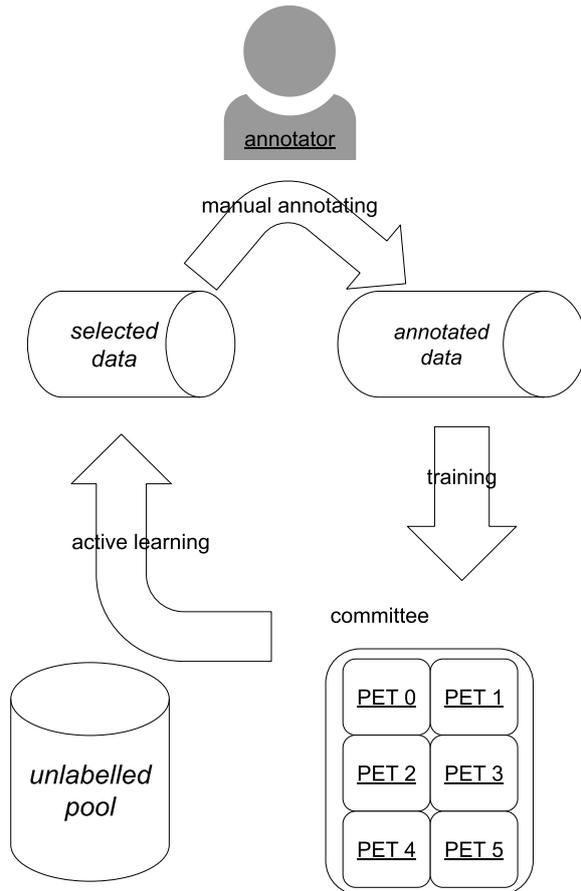


Figure 1: Illustration of the data annotation prioritisation scenario with a committee of 6 PETs. For each iteration, firstly the committee retrieves k new unlabelled samples ($k=10$ in our experiments), secondly the human annotators label them, lastly each of the PET based on different PLMs is trained individually with all of the labelled samples at hand. Our experiments start from 0 labelled samples and end at 300 labelled samples.

through a committee of PETs. Figure 1 illustrates the application of the active learning strategy on data annotation prioritisation.

By exploring effective prioritisation of unlabelled data for annotation and making better use of a small amount of labelled data, we make the following novel contributions:

- we are the first to study data annotation prioritisation through active learning for few-shot claim verification;
- we are the first to study the extensibility of PET to enable active learning, by proposing Active PETs, a novel ensemble-based cold-start active learning strategy that enables multiple pretrained language models (PLMs) to collectively prioritise data instances;
- we further investigate the effect of oversam-

pling on mitigating the impact of imbalanced data selection on few-shot learning, when guided by active learning;

- we conduct further corpus-based analysis on the selected few-shot data instances, which highlights the potential of Active PETs to lead to improved lexical and semantic characteristics that benefit the task.

Our results show consistently improved performance of Active PETs over the baseline active learning strategies on two datasets, SCIFACT (Wadden et al., 2020) and Climate FEVER (Diggelmann et al., 2021). In addition to improved performance over the baselines, our research emphasises the importance of the hitherto unexplored data prioritisation in claim verification, showing remarkable performance improvements where time and budget are limited.

2 Background

2.1 Claim Verification

Claim verification is typically addressed as an NLI problem (Thorne and Vlachos, 2018). Recent progress has enforced a closed-world reliance (Pratapa et al., 2020) and incorporated multiple instance learning (Sathe and Park, 2021). While data scarcity poses a major challenge on automated fact-checking (Zeng et al., 2021), research on few-shot claim verification is limited to date. Lee et al. (2021) investigated a perplexity-based approach that solely relies on perplexity scores from PLMs. Their model was tested on binary claim verification, as opposed to the three-way classification in our work. Zeng and Zubiaga (2022) introduced SEED, a vector-based method that aggregates pairwise semantic differences for claim-evidence pairs to address the task of few-shot claim verification. While their model addresses three-way classification, the experiments are only conducted in ideal scenarios where oracle evidences are available. To the best of our knowledge, however, no work has investigated the use of active learning in the context of claim verification. To further research in this direction, we propose Active PETs, a model that incorporates active learning capabilities into PET (Schick and Schütze, 2021a,b). PET has shown competitive performance in a range of NLP classification tasks, but its adaptation to the context of automated fact-checking and/or active learning settings has not been studied.

2.2 Active Learning

Active Learning (AL) is a paradigm used where labelled data is scarce (Ein-Dor et al., 2020). The key idea is that a strategic selection of training instances to be labelled can lead to improved performance with less training (Settles, 2009). Active learning methods are provided with an unlabelled pool of data, on which a querying step is used to select candidate instances to be annotated with the aim of optimising performance of a model trained on that data. The goal is therefore to optimise performance with as little annotation –and consequently budget– as possible. Traditional active learning query strategies mainly include uncertainty sampling, query-by-committee (QBC) strategy, error/variance reduction strategy and density weighted methods (Settles, 2012). Recent empirical studies have revisited the traditional strategies in the context of PLMs: Ein-Dor et al. (2020) examined various active learning strategies with BERT (Devlin et al., 2019), though limited to binary classification tasks. Schröder et al. (2022) conducted experiments with ELECTRA (Clark et al., 2020), BERT, and DistilRoBERTa (Sanh et al., 2019) respectively, while limiting the scope to uncertainty-based sampling.

Recent efforts on combining active learning with PLMs go into both warm-start and cold-start strategies. Warm-start strategies require a small initial set of labelled data to select additional instances, while cold-start strategies can be used without an initial set of labelled data. Ash et al. (2020) proposed Batch Active learning by Diverse Gradient Embeddings (BADGE) that samples a batch of instances based on diversity in gradient loss. Margatina et al. (2021) proposed Contrastive Active Learning (CAL), the state-of-the-art (SOTA) warm-start strategy that highlights data with similar feature space but maximally different predictions. Furthermore, Active Learning by Processing Surprisal (ALPS) (Yuan et al., 2020), the SOTA cold-start strategy, utilises masked language model (MLM) loss as an indicator of model uncertainty. We use BADGE, CAL and ALPS for baseline comparison, please see detailed descriptions in section 4.3.

To the best of our knowledge, QBC strategies (Seung et al., 1992; Dagan and Engelson, 1995; Freund and Haussler, 1997) that utilise a committee of models remains to be explored with PLMs, as previous studies limit their scope at measuring single model uncertainty. Nowadays various PLMs are publicly available that applying an ensemble-based

query strategy on a downstream task becomes realistic, especially in few-shot settings where the computation required is relatively cheap. Furthermore, previous studies always perform fine-tuning to get classification results from PLMs. Our work presents the first attempt at integrating an active learning strategy into PET, which we investigate in the context of claim verification for fact-checking.

3 Methodology

In this section, we first describe PET, then introduce our model Active PETs, and finally describe the oversampling mechanism we use.

3.1 Pattern Exploiting Training

Pattern Exploiting Training (PET) (Schick and Schütze, 2021a,b) is a semi-supervised training procedure that can reformulate various classification tasks into cloze questions with natural language patterns and has demonstrated competitive performance in various few-shot classification tasks. To predict the label for a given instance x , it is first reformulated into manually designed patterns that have the placeholder $[mask]$. Then, the probability of each candidate token for replacing $[mask]$ is calculated by using a pretrained language model, where each candidate is mapped to a label according to a manually designed verbaliser.

3.2 Proposed method: Active PETs

Having a large pool of unlabelled data, our objective is to design a query strategy that selects suitable candidates to be labelled, such that the labelled pool of instances leads to optimal few-shot performance. Our query strategy is rooted in the intuition that disagreement among different PETs in a committee can capture the uncertainty of a particular instance.

Based on the assumption that performance of different language models is largely dependent on model size (Kaplan et al., 2020), we introduce a weighting mechanism: each PET is first assigned a number of votes V_i that is proportional to its hidden size,² and ultimately all votes are aggregated. Algorithm 1 presents the pseudo-code for executing a single query iteration with Active PETs.

²For example, if we use a committee formed of only base models that have 6 hidden layers and large models that have 12 hidden layers, proportionally each of the base models is allocated one vote and each of the large models is allocated two votes.

Algorithm 1 A Single Query Iteration

Require: The last trained Committee of PETs C , unlabelled data pool U , query size k

```

for  $PET_i \in C$  do
   $v_i \leftarrow Size(PET_i) / \min_{PET_i \in C} Size(PET_i)$ 
end for                                ▷ assign number of votes
for instance  $x \in U$  do
  for  $PET_i \in C$  do
     $V_{x_i} \leftarrow resize(\hat{y}_{x_i}, v_i)$ 
  end for                                ▷ predict label and vote
   $S_x = - \sum_{V_{x_i} \in V_x} \frac{V_{x_i}}{|V|} \log \frac{V_{x_i}}{|V|}$ 
end for                                ▷ calculate entropy scores
return  $Sort(S)[ : k ]$                     ▷ return top k instances

```

We then quantify the disagreement by calculating vote entropy (Dagan and Engelson, 1995):

$$score_x = - \sum_{\hat{y}} \frac{vote(x, \hat{y})}{count(V)} \log \frac{vote(x, \hat{y})}{count(V)} \quad (1)$$

where \hat{y} is the predicted label, x is the instance, $vote(x, \hat{y})$ are the committee votes of \hat{y} for the instance x , and $count(V)$ is the number of total assigned votes. It can be viewed as a QBC generalisation of entropy-based uncertainty sampling that is designed to combine models of different sizes.

3.3 Data Oversampling

One of the risks of the proposed active learning strategy is that the resulting training data may not be adequately balanced, which can impact model performance. An accessible solution is oversampling: resample the instances from the minority class with replacement until balanced. Note that this does not increase the labelling effort as instances are repeated from the labelled pool. Instead of random resampling (Japkowicz, 2000), we propose a novel technique of integrating resampling with the committee’s preference. For each minority class, we start resampling from the instance that has the highest disagreement score to the instance that has the lower disagreement score. In highly imbalanced cases, resampling is repeated from the highest to lowest priority until the overall label distribution is balanced. Algorithm 2 presents the pseudo-code for executing the training loop with the option of conducting oversampling with Active PETs.

Algorithm 2 Training

Require: Labelled and sorted data D , A initial Committee of PETs C

```

if Oversampling then
   $c \leftarrow \max_{class \in D} count(data \in class)$ 
   $D \leftarrow resize_{\forall class \in D}(class, c)$ 
end if                                ▷ oversampling
for  $PET_i \in C$  do
   $PET_i \leftarrow train(PET_i, D)$ 
end for                                ▷ train the committee of PETs
return  $C$                                 ▷ return trained PETs

```

4 Experimental Settings

Here we present the datasets and models used.

4.1 Datasets

SCIFACT			
	‘Support’	‘Neutral’	‘Contradict’
UP	266 (9.31%)	2530 (88.55%)	61 (2.14%)
Test	150 (33.33%)	150 (33.33%)	150 (33.33%)
cFEVER			
	‘Support’	‘Neutral’	‘Contradict’
UP	1789 (24.78%)	4778 (66.19%)	652 (8.66%)
Test	150 (33.33%)	150 (33.33%)	150 (33.33%)

Table 1: Label distribution of SCIFACT and cFEVER. UP = unlabelled pool of training data.

We choose real-world datasets with real claims, SCIFACT and Climate FEVER, known to be challenging, technical and free of synthetic data.³

SCIFACT provides scientific claims with their veracity labels, as well as a collection of scientific paper abstracts, some of which contain rationales to resolve the claims. In addition, it provides the oracle rationales that can be linked to each claim.

For SCIFACT, we perform the pipeline including abstract retrieval and claim verification. For the abstract retrieval step, we use BM25 to retrieve the top 3 abstracts, skipping the more specific rationale selection, as the SOTA system for this dataset suggested (Wadden et al., 2021). We chose BM25 based on high recall results reported in previous work (Pradeep et al., 2021). We merge original SCIFACT train set and dev set and redistribute the data to form a test set that contains 150 instances

³See data samples in Appendix A.

for each class and use the rest in the unlabelled pool. The reformulated data is highly imbalanced as presented in Table 1.

Climate FEVER (cFEVER) is a challenging large-scale dataset that consists of claim and evidence pairs on climate change, along with their veracity labels. As it does not naturally provide options of setting up retrieval modules, we directly use it on the task of claim verification. Similarly we reserve 150 instances for each class to form a test set and leave the rest in the unlabelled pool. Data in the unlabelled pool is heavily skewed, as shown in Table 1.

4.2 Active PETs

Committees of five to fifteen models are common for an ensemble-based active learning strategy (Settles, 2012). Here we form a committee of 6 PETs with 3 types of PLMs: BERT-base, BERT-large (Devlin et al., 2019), RoBERTa-base, RoBERTa-large (Liu et al., 2019), DeBERTa-base and DeBERTa-large (He et al., 2021). Given the commonalities between the NLI and claim verification tasks, we use the PLM checkpoints already fine-tuned on MNLI (Williams et al., 2018).

Despite a line of research in optimising PET patterns and verbalisers (Tam et al., 2021), that is not our main focus. We use the following pattern and verbaliser for PET: `[claim]? [mask], [evidence]`; “Support”：“Yes”, “Contradict”：“No”, “Neutral”：“Maybe”, as they yielded best performance on NLI tasks in our preliminary experiments. Figure 2 provides an example of performing claim verification using PET.

There are two steps in our approach: (1) an ensemble method is used for data annotation prioritisation, after which data is selected and annotated, and (2) with the data instances drawn and annotated, we train a PET model that uses a single PLM to make the predictions. An ensemble method is key in step (1) to support the combined decision-making of choosing instances to annotate, but not in step (2) for the PET model which runs on a single PLM. Hence, results are presented for individual PETs, even if in all cases the ensemble is involved in the underlying prioritisation step. We test two variants: **Active_PETs** with no oversampling, and **Active_PETs-o** with the oversampling described in Section 3.3.

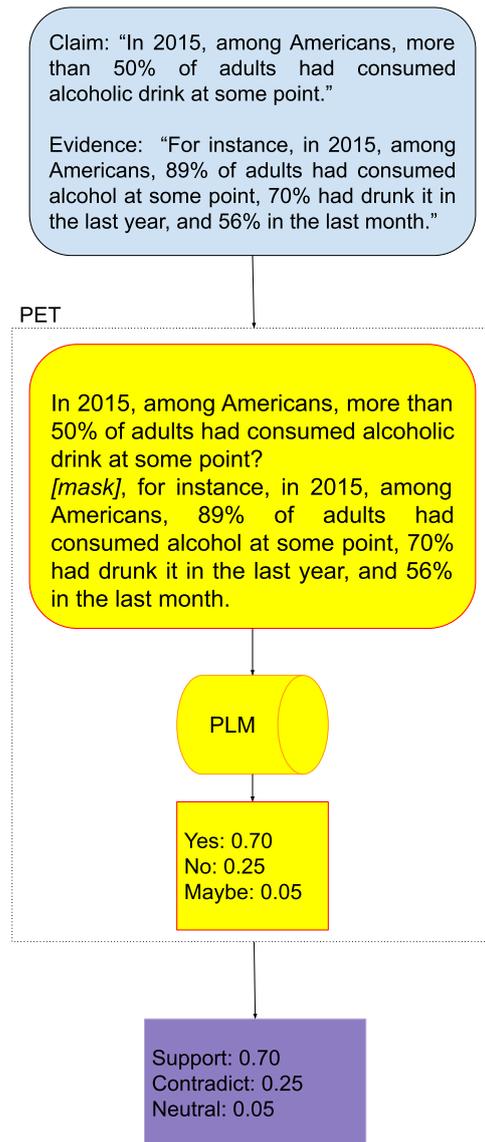


Figure 2: An example of doing claim verification with PET.

4.3 Baselines

We compare our method to four baselines: random sampling, BADGE, CAL and ALPS.

4.3.1 Random sampling

For random sampling, we run each experiment over 10 different sampling seeds ranging from 123 to 132, and present the averaged results.

4.3.2 BADGE

BADGE (Ash et al., 2020) optimises for both uncertainty and diversity. Gradient embeddings g_x are first computed for each data in the unlabelled pool, where g_x is the gradient of the cross entropy loss with respect to the parameters of the model’s last layer. It then applies k-MEANS++ clustering

on the obtained gradient embeddings, and batch selects instances that differ in feature representation and predictive uncertainty.

Though BADGE is proposed as a warm-start method, the required initial set of labelled data is only used for the initial training the model. In our experiments on claim verification, PLMs that are already finetuned on a similar task NLI are used, hence, BADGE can be used for cold-start sampling.

4.3.3 CAL

CAL (Margatina et al., 2021), the SOTA warm-start strategy, highlights contrastive data points: data that has similar model encodings but different model predictions. Unlike BADGE, an initial labelled set of data is essential for CAL. It first calculates the [CLS] embeddings for all of the data and then runs K-Nearest-Neighbours (KNN) to obtain the k closest labelled neighbours for each unlabelled instance. It further calculates predictive probabilities from the model and measures Kullback-Leibler divergence on it. Finally it selects unlabelled instances whose predictive likelihoods diverge the most from their neighbours.

While CAL achieves SOTA performance as a warm-start strategy, its dependence on an initial labelled set of data makes it incompatible in the same few-shot active learning settings without an initial labelled set. However, for comprehensive comparison purposes, we still include it as a baseline starting at 100 labelled instances that are obtained from random sampling with 10 different random seeds.

4.3.4 ALPS

ALPS (Yuan et al., 2020), the SOTA cold-start active learning method, also aims to take both model uncertainty and data diversity into account. It calculates surprisal embeddings to represent model uncertainty. Specifically, for each instance x , it is passed through the masked language modelling head of a PLM and then 15% of the tokens in x are randomly selected to calculate the cross entropy against their target tokens. The surprisal embeddings go through L2-normalisation and then get clustered to select the top samples.

4.4 Training Details

Hyperparameters. As in few-shot settings we lack a development set, we follow previous work (Schick and Schütze, 2021a,b) and use the following hyperparameters for all experiments: $1e^{-5}$ as

learning rate, 16 as batch size, 3 as the number of training epochs, 256 as the max sequence length.⁴

Labelling budget. We set it to a maximum of 300. We experiment with all scenarios ranging from 10 to 300 instances with a step size of 10.

Checkpoints. We always use the PLM checkpoints from the last iteration to perform active learning, but always train the initial PLMs which have never been trained on any fact-checking datasets.

5 Results

We next discuss results for our experiments.

5.1 Results on SCIFACT

Figure 3 presents experimental results on SCIFACT, where the unlabelled pool is large, heavily imbalanced and the domain is technical. Each subfigure shows results for a different PET among the six under consideration.

Data retrieved with Active PETs brings substantial improvements for all of the models, often from the very beginning but consistently as the number of shots increases from around 50 instances. Despite the performance fluctuations, training using data sampled with Active PETs rarely underperforms the baselines for SCIFACT. With Active PETs, Bert-base peaks at 0.352, RoBERTa-base peak at 0.345; DeBERTa-base peaks at 0.385; BERT-large peaks at 0.380; RoBERTa-large peaks at 0.409; DeBERTa-large peaks at 0.541. Generally, Active PETs shows a 10 to 20% increase in F1 scores, compared with various baselines.

Moreover, with Active PETs-o, i.e. when oversampling is further integrated with Active PETs, we observe a significant performance increase. Models tend to learn better from the beginning; the increase trend has less fluctuation; and the overall F1 scores are much higher. In this case, Bert-base peaks at 0.497, RoBERTa-base peak at 0.539; DeBERTa-base peaks at 0.551; BERT-large peaks at 0.548; RoBERTa-large peaks at 0.514; DeBERTa-large peaks at 0.587. This highlights the potential of oversampling, which increases the number of instances without additional labelling budget.

Among the baselines, we observe that training with data retrieved from all baselines failed to lead to any effective outcomes for BERT-base and DeBERTa-base within a labelling budget of 300 instances. While BADGE and CAL lead to some improvements over BERT-large and RoBERTa-large

⁴See further details for reproducibility in Appendix B.

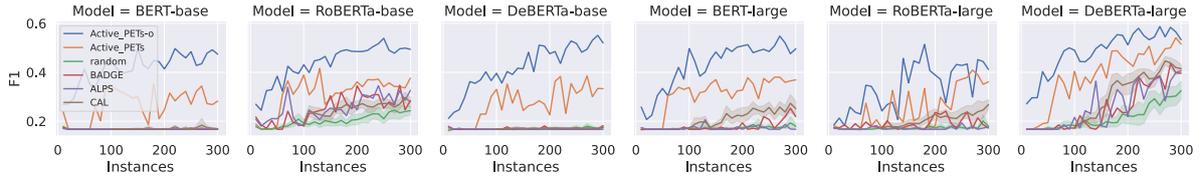


Figure 3: Few-Shot F1 Performance on SCIFACT claim verification.

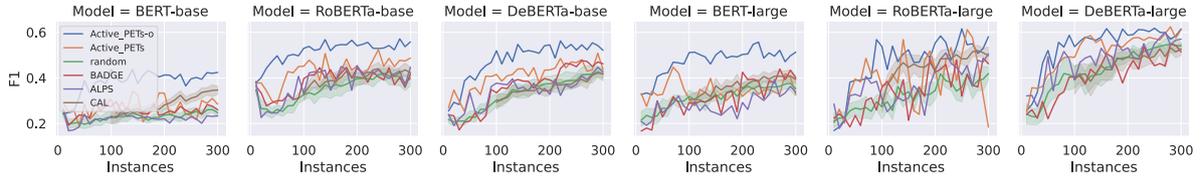


Figure 4: Few-Shot F1 Performance on cFEVER claim verification.

when given over 100 instances, random and ALPS failed to bring any improvements. Baseline results are better with RoBERTa-base and DeBERTa-large, but underperform Active PETS.

5.2 Results on cFEVER

Figure 4 presents F1 scores on cFEVER, where the unlabelled pool is large, imbalanced and the domain is somewhat technical. In this case, models generally achieve higher F1 scores than on SCIFACT. First of all, we observe that Active PETS outperforms random baseline in a more stable manner. It is over 10% higher than random most of the time, although it shows large performance fluctuations on RoBERTa-large. With Active PETS, Bert-base peaks at 0.34, RoBERTa-base peak at 0.524; DeBERTa-base peaks at 0.508; BERT-large peaks at 0.447; RoBERTa-large peaks at 0.612; DeBERTa-large peaks at 0.624. Moreover, Active PETS-o leads to a further performance boost, and more importantly, smooths out the large performance fluctuations. It is about 20% better than the random baseline most of the time. Specifically, Bert-base peaks at 0.438, RoBERTa-base peak at 0.571; DeBERTa-base peaks at 0.562; BERT-large peaks at 0.557; RoBERTa-large peaks at 0.615; DeBERTa-large peaks at 0.618.

When it comes to the baselines, the baselines do not struggle as much in the worst cases. Even if BERT-base’s performance merely increased with most of the baselines, all of the other models managed to improve within the budget. With random sampling, RoBERTa-base, DeBERTa-base, BERT-large and RoBERTa-large all roughly peak at around 0.4, while DeBERTa-large is much better

and peaks at around 0.5. BADGE, CAL and ALPS are in general better than random, but achieves lower F1 scores than Active PETS, especially in few-shot settings when the labelling budget is below 100.

6 Ablation Study

With SCIFACT we designed a slightly different pipeline where we conduct both evidence retrieval and claim verification – a setting that wasn’t provided with cFEVER. To assess the impact of the addition of the evidence retrieval component on SCIFACT, we further perform ablation experiments on SCIFACT with oracle evidence.

With oracle evidence, the number of “Neutral” claim-evidence pairs are significantly reduced, resulting in a more balanced overall label distribution. After reserving 100 instances from each class for the test set, the unlabelled pool has 765 instances in total, where “Support” takes 46.54%, “Neutral” takes 38.43% and “Contradict” takes 15.03%. As shown in Figure 5, overall few-shot performance is much better and active learning demonstrates lesser performance gains. Sampling with baseline active learning strategies in general leads to similar results as random sampling. Surprisingly, coupling Active PETS with oversampling when the labelled pool is reasonably balanced, still maintains performance advantages across models. Under this setting, Bert-base peaks at 0.645, RoBERTa-base peak at 0.655; DeBERTa-base peaks at 0.766; BERT-large peaks at 0.68; RoBERTa-large peaks at 0.657; DeBERTa-large peaks at 0.86.

As demonstrated above, active learning is much

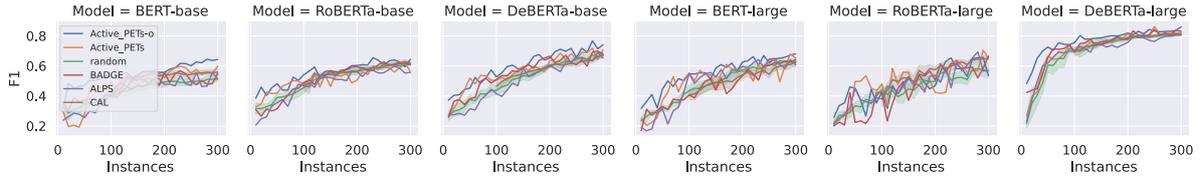


Figure 5: Few-Shot F1 Performance on Oracle SCIFACT claim verification.

more helpful for SCIFACT in a real-world setting than in an oracle setting. We could expect that if this finding generalises to cFEVER, active learning in a real-world setting involving evidence retrieval could possibly lead to larger performance gains.

7 Analysis

To better understand the impact of data prioritisation, we delve into the labelled data. In the interest of focus, we compare Active PETs with the SOTA cold-start method ALPS by analysing the best-performing PLM DeBERTa-large where 300 instances are selected.

7.1 Balancing Effects

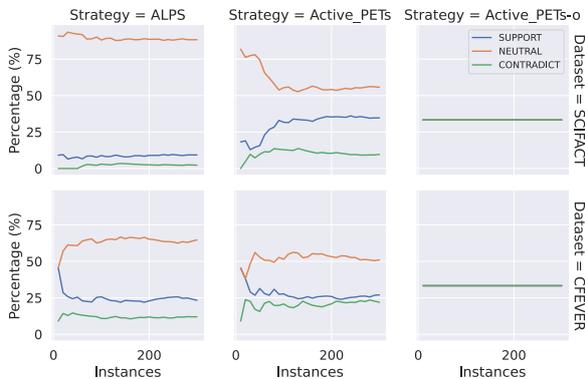


Figure 6: Label Distribution of data obtained with active learning by DeBERTa-large. The upper row is for SCIFACT and the lower row is for cFEVER.

We first look at the distribution of labels for the selected data. Figure 6 shows remarkable difference on label distribution for different active learning strategies. ALPS samples over 80% data from “Neutral”, less than 10% from “Support” and very few from “Contradict” for SCIFACT; over 60% data from “Neutral”, over 20% from “Support” and less than 20% from “Contradict” for cFEVER. They correlate well with original label distribution of each unlabelled pool, as presented in table 1. It suggests that ALPS is not sensitive to label

distribution. However, Active PETs manages to sample a much more balanced distribution out of the extremely skewed original distribution. For SCIFACT, despite the initial few iterations, Active PETs samples less than 60% data from “Neutral”, less than 40% data from “Support”, around 10% data from “Contradict”; for cFEVER, Active PETs samples less than 60% data from “Neutral”, over 20% data from “Support”, around 20% data from “Contradict”. In both datasets, label distribution from Active PETs are significantly more balanced than ALPS. Finally, the strategy with oversampling returns perfectly balanced distribution as expected. We identify a strong correlation between label distribution and classification performance.

7.2 Linguistic Effects

Aiming at providing further insights into data quality, we conduct corpus-based linguistic analysis to investigate lexical richness and semantic similarity.

Lexical Richness			
	ALPS	Active_PETs	Active_PETs-o
SCIFACT	0.0362	0.0387	0.0447
cFEVER	0.0389	0.0413	0.0503
Semantic Similarity			
	ALPS	Active_PETs	Active_PETs-o
SCIFACT	0.7921	0.8031	0.8054
cFEVER	0.7449	0.7744	0.7841

Table 2: Lexical richness is measured with Maas Type-Token Ratio (MTTR) scores and Semantic Similarity is measured by cosine similarity scores on embeddings of claims and evidences.

7.2.1 Lexical Richness

A popular metric for calculating lexical richness is Type-Token Ratio (TTR), where the total number of unique tokens is divided by the total number of tokens. We use Maas Type-Token Ratio (Maas TTR) (Maas, 1972), a logarithmic variant of TTR,

which is demonstrated to be less sensitive to the length of the text (McCarthy and Jarvis, 2007):

$$a^2 = \frac{\log N - \log V}{\log N^2} \quad (2)$$

where N is the number of tokens in the corpus and V is the number of unique tokens in the corpus.

As shown in the upper part of Table 2, data selected by ALPS has the lowest lexical richness, while Active PETs leads to higher lexical richness for both datasets. Even more surprisingly, when integrating Active PETs with oversampling, the corpus has even higher score at lexical richness, despite that there are multiple duplicated instances in the corpus. One possibility is that training data with higher lexical richness may convey more useful information, as a bigger vocabulary enables more precise expressions.

7.2.2 Semantic Similarity

To investigate the overall data diversity, we calculate the average semantic similarity of all possible claim-evidence pairs in the corpus.⁵ We obtain embeddings of claims and evidences with the PLM at interest, namely DeBERTa-large that has been trained on MNLI. For each embedded claim, we calculate its cosine similarity score with all embedded evidences in the corpus. The average of all similarity scores is then obtained. The lower part of Table 2 shows that ALPS leads to lowest overall semantic embedding similarity scores and Active PETs leads to higher scores. Integrated with oversampling, Active PETs leads to even higher similarity scores. It correlates well with the design of the strategies: ALPS explicitly encourages data diversity, while Active PETs focuses on committee uncertainty. One possible explanation is that data diversity is not as beneficial when the unlabelled pool contains less relevant instances: in the case of SCIFACT and cFEVER datasets, the majority of the unlabelled pool belongs to the “Neutral” class where the evidence is not enough to reach a verdict for the claim.

8 Conclusions

We present the first study on data annotation prioritisation for claim verification in automated fact-checking. With our novel method Active PETs, we demonstrate the potential of utilising a committee of PETs to collaboratively select unlabelled

⁵Note that if we only calculate the retrieved pairs, the average similarity scores are approximately 1 for all strategies.

data for annotation, furthering in turn the extensibility of PET to active learning for the first time. Experiments on the SCIFACT and cFEVER datasets demonstrate the effectiveness of our proposed method, particularly in dealing with imbalanced data. Our proposed model consistently outperforms the random, BADGE, CAL and ALPS baselines by a margin. Further integration with an oversampling strategy that does not impact labelling effort leads to consistent performance improvements in all tested settings. Data that is more balanced shows to have higher lexical richness and semantic similarity, leading to better training results. While we have shown its effectiveness for claim verification here, in the future we aim to investigate Active PETs in other downstream tasks.

9 Limitations

We focus on demonstrating the effectiveness of Active PETs in scenarios where the labelling budget is limited and the label distribution is very imbalanced, as they are major challenges for automated fact-checking. Active PETs is shown to be particularly beneficial with low labelling budgets and becomes less so when the labelling budget increases and/or the unlabelled pool is balanced. Furthermore, as Active PETs is built on PET, it inherits the limitations from PET, e.g. a pattern-verbaliser pair (PVP) is required for any classification tasks. Note that a good selection of tested PVPs that cover common NLP tasks are publicly available.

Our experiments are only conducted with PLMs that are of base and large sizes, e.g., BERT-base and BERT-large, due to limited computing resources. Future work may further experiment with giant models like T5-11b and GPT-3. Another interesting direction would be to extend the proposed voting mechanism such that giant models and tiny models can both contribute effectively in the same committee, e.g., GPT-3 and DistillBert. Ideally, despite that GPT-3 is much larger than DistillBert, the extended voting mechanism should still allow DistillBert to contribute effectively.

Acknowledgements

We thank Christopher James Madge and Massimo Poesio from Queen Mary University of London for valuable pointers and comments; Ji-Ung Lee from Technische Universität Darmstadt for insightful discussions. Xia Zeng is funded by China Scholarship Council (CSC). This research utilised Queen

Mary's Apocrita HPC facility, supported by QMUL Research-IT. <http://doi.org/10.5281/zenodo.438045>

References

- Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. **FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information.** *arXiv:2106.05707 [cs]*. ArXiv: 2106.05707.
- J. T. Ash, Chicheng Zhang, A. Krishnamurthy, J. Langford, and Alekh Agarwal. 2020. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. *ICLR*.
- Pepa Atanasova, Dustin Wright, and Isabelle Augenstein. 2020. **Generating Label Cohesive and Well-Formed Adversarial Claims.** In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3168–3177, Online. Association for Computational Linguistics.
- Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. **MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims.** In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.
- Giannis Bekoulis, Christina Papagiannopoulou, and Nikos Deligiannis. 2021. **Understanding the Impact of Evidence-Aware Sentence Selection for Fact Checking.** In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 23–28, Online. Association for Computational Linguistics.
- Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. **Seeing Things from a Different Angle: Discovering Diverse Perspectives about Claims.** In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 542–557, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. **ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators.** *arXiv:2003.10555 [cs]*. ArXiv: 2003.10555.
- Ido Dagan and Sean P. Engelson. 1995. Committee-Based Sampling For Training Probabilistic Classifiers. In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. Morgan Kaufmann.
- J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.** In *NAACL-HLT*.
- Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leppold. 2021. **CLIMATE-FEVER: A Dataset for Verification of Real-World Climate Claims.** *arXiv:2012.00614 [cs]*. ArXiv: 2012.00614.
- Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. **Active Learning for BERT: An Empirical Study.** In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.
- Yoav Freund and David Haussler. 1997. Selective sampling using the query by committee algorithm. In *Machine Learning*, pages 133–168.
- Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. **A Survey on Automated Fact-Checking.** *Transactions of the Association for Computational Linguistics*, 10:178–206.
- Amartya Hatua, Arjun Mukherjee, and Rakesh Verma. 2021. **Claim Verification Using a Multi-GAN Based Model.** In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 494–503, Held Online. INCOMA Ltd.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. **DeBERTa: Decoding-enhanced BERT with Disentangled Attention.** *arXiv:2006.03654 [cs]*. ArXiv: 2006.03654.
- Nathalie Japkowicz. 2000. The Class Imbalance Problem: Significance and Strategies. In *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI)*, pages 111–117.
- Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. **Exploring Listwise Evidence Reasoning with T5 for Fact Verification.** In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410, Online. Association for Computational Linguistics.
- Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. **HoVer: A Dataset for Many-Hop Fact Extraction And Claim Verification.** In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling Laws for Neural Language Models](#). *arXiv:2001.08361 [cs, stat]*. ArXiv: 2001.08361.
- Neema Kotonya, Thomas Spooner, Daniele Magazzeni, and Francesca Toni. 2021. [Graph Reasoning with Context-Aware Linearization for Interpretable Fact Extraction and Verification](#). In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 21–30, Dominican Republic. Association for Computational Linguistics.
- Neema Kotonya and Francesca Toni. 2020. [Explainable Automated Fact-Checking: A Survey](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. [Towards Few-shot Fact-Checking via Perplexity](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981, Online. Association for Computational Linguistics.
- Xiangci Li, Gully Burns, and Nanyun Peng. 2021. [A Paragraph-level Multi-task Learning Model for Scientific Fact-Verification](#). *arXiv:2012.14500 [cs]*. ArXiv: 2012.14500.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A Robustly Optimized BERT Pretraining Approach](#). *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.
- Zhenghao Liu, Chenyan Xiong, Zhuyun Dai, Si Sun, Maosong Sun, and Zhiyuan Liu. 2020. [Adapting Open Domain Fact Extraction and Verification to COVID-FACT through In-Domain Language Modeling](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2395–2400, Online. Association for Computational Linguistics.
- Jing Ma, Wei Gao, Shafiq Joty, and Kam-Fai Wong. 2019. [Sentence-Level Evidence Embedding for Claim Verification with Hierarchical Attention Networks](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2561–2571, Florence, Italy. Association for Computational Linguistics.
- H.D. Maas. 1972. [Über den Zusammenhang zwischen Wortschatzumfang und Länge eines Textes](#). *Springer*, 8:73–96.
- Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. [Active Learning by Acquiring Contrastive Examples](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Philip M. McCarthy and Scott Jarvis. 2007. [voacd: A theoretical and empirical evaluation](#). *Language Testing*, 24(4):459–488. Publisher: SAGE Publications Ltd.
- Mitch Paul Mithun, Sandeep Suntuwal, and Mihai Surdeanu. 2021. [Data and Model Distillation as a Solution for Domain-transferable Fact Verification](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4546–4552, Online. Association for Computational Linguistics.
- Preslav Nakov, D. Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, A. Barr’on-Cedeno, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. [Automated Fact-Checking for Assisting Human Fact-Checkers](#). In *IJCAI*.
- W. Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2021. [Multi-Hop Fact Checking of Political Claims](#). In *IJCAI*.
- Liangming Pan, Wenhui Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. [Zero-shot Fact Verification by Claim Generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 476–483, Online. Association for Computational Linguistics.
- Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. [Scientific Claim Verification with VerT5erini](#). In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, online. Association for Computational Linguistics.
- Adithya Pratapa, Sai Muralidhar Jayanthi, and Kavya Nerella. 2020. [Constrained Fact Verification for FEVER](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7826–7832, Online. Association for Computational Linguistics.
- Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. [COVID-Fact: Fact Extraction and Verification of Real-World Claims on COVID-19 Pandemic](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129, Online. Association for Computational Linguistics.
- Chris Samarinas, Wynne Hsu, and Mong Li Lee. 2021. [Improving Evidence Retrieval for Automated Explainable Fact-Checking](#). In *Proceedings of the 2021 Conference of the North American Chapter of the*

- Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 84–91, Online. Association for Computational Linguistics.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*.
- Aalok Sathe and Joonsuk Park. 2021. [Automatic Fact-Checking with Document-level Annotations using BERT and Multiple Instance Learning](#). In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 101–107, Dominican Republic. Association for Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021a. [Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021b. [It’s Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.
- Michael Sejr Schlichtkrull, Vladimir Karpukhin, Barlas Oguz, Mike Lewis, Wen-tau Yih, and Sebastian Riedel. 2021. [Joint Verification and Reranking for Open Fact Checking Over Tables](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6787–6799, Online. Association for Computational Linguistics.
- Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. [Revisiting Uncertainty-based Query Strategies for Active Learning with Transformers](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.
- Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. [Get Your Vitamin C! Robust Fact Verification with Contrastive Evidence](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.
- Burr Settles. 2009. [Active Learning Literature Survey](#). Technical Report, University of Wisconsin-Madison Department of Computer Sciences. Accepted: 2012-03-15T17:23:56Z.
- Burr Settles. 2012. [Active Learning](#). *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.
- H. S. Seung, M. Opper, and H. Sompolinsky. 1992. [Query by committee](#). In *Proceedings of the fifth annual workshop on Computational learning theory, COLT ’92*, pages 287–294, New York, NY, USA. Association for Computing Machinery.
- Jiasheng Si, Deyu Zhou, Tongzhe Li, Xingyu Shi, and Yulan He. 2021. [Topic-Aware Evidence Reasoning and Stance-Aware Aggregation for Fact Verification](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1612–1622, Online. Association for Computational Linguistics.
- Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. [Improving and Simplifying Pattern Exploiting Training](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- James Thorne and Andreas Vlachos. 2018. [Automated Fact Checking: Task Formulations, Methods and Future Directions](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. [Fact or Fiction: Verifying Scientific Claims](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.
- David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2021. [LongChecker: Improving scientific claim verification by modeling full-abstract context](#). *arXiv:2112.01640 [cs]*. ArXiv: 2112.01640.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- Wenpeng Yin and Dan Roth. 2018. [TwoWingOS: A Two-Wing Optimization Strategy for Evidential Claim Verification](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 105–114, Brussels, Belgium. Association for Computational Linguistics.
- Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. [Cold-start Active Learning through Self-supervised Language Modeling](#). In *Proceedings of the 2020 Conference on Empirical Methods*

in *Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Xia Zeng, Amani S. Abumansour, and Arkaitz Zubiaga. 2021. [Automated fact-checking: A survey](#). *Language and Linguistics Compass*, 15(10):e12438. [eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/lnc3.12438](#).

Xia Zeng and Arkaitz Zubiaga. 2021. [QMUL-SDS at SCIVER: Step-by-Step Binary Classification for Scientific Claim Verification](#). pages 116–123.

Xia Zeng and Arkaitz Zubiaga. 2022. [Aggregating Pairwise Semantic Differences for Few-Shot Claim Veracity Classification](#). ArXiv:2205.05646 [cs].

Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2021a. [Abstract, Rationale, Stance: A Joint Model for Scientific Claim Verification](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3580–3586, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2021b. [Abstract, Rationale, Stance: A Joint Model for Scientific Claim Verification](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3580–3586, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Example Appendix

We present example instances from SCIFACT and cFEVER datasets in this section.

B Reproducibility Appendix

We present further experimental details here for reproducibility.

Number of parameters in each model The number of parameters for BERT-base, BERT-large, RoBERTa-base, RoBERTa-large, DeBERTa-base, DeBERTa-large is 109484547, 335144963, 124647939, 355362819, 139194627 and 406215683 respectively.

Computing infrastructure We use High Performance Compute cluster supported by the university. Each experiment is run with 8 compute cores, 11G RAM per core and a single NVIDIA A100 GPU.

Run time Table 4 reports the average run time of executing a sampling iteration of 150 unlabelled instances and a training iteration with the sampled data over three datasets. It serves as a good indicator for comparing the efficiency among different

active learning methods. As CAL requires an initial labelled set of data, we report the total run time of an iteration of using the random method for 75 instances and an iteration of using CAL method for another 75 instances. Table 5 further reports the total run time of the best method Active PETs-o on different datasets. The actual run time highly correlates with the size of the unlabelled pool for each datasets.

Our key focus has been on resource-efficiency and performance, with a lesser focus on runtime, hence there can be room for optimisation in future work, including: (1) optimising the code e.g. through parallelisation of the ensembled models which are now run sequentially, (2) using DL optimisation libraries such as deepspeed, and (3) using dynamic step sizes to reduce the number of iterations, e.g. increase step size if initial iterations lead to balanced samples. In a real-world, deployed scenario, one would also need to account for the time needed by humans to perform the annotation (in our case simulated).

SCIFACT		
Claim	Evidence	Veracity
“Neutrophil extracellular trap (NET) antigens may contain the targeted autoantigens PR3 and MPO.”	“Netting neutrophils in autoimmune small-vessel vasculitis Small-vessel vasculitis (SVV) is a chronic autoinflammatory condition linked to antineutrophil cytoplasm autoantibodies (AN-CAs). Here we show that chromatin fibers, so-called neutrophil extracellular traps (NETs), are released by ANCA-stimulated neutrophils and contain the targeted autoantigens proteinase-3 (PR3) and myeloperoxidase (MPO). Deposition of NETs in inflamed kidneys and circulating MPO-DNA complexes suggest that NET formation triggers vasculitis and promotes the autoimmune response against neutrophil components in individuals with SVV.”	“Support”
“Cytochrome c is transferred from cytosol to the mitochondrial intermembrane space during apoptosis.”	“At the gates of death. Apoptosis that proceeds via the mitochondrial pathway involves mitochondrial outer membrane permeabilization (MOMP), responsible for the release of cytochrome c and other proteins of the mitochondrial intermembrane space. This essential step is controlled and mediated by proteins of the Bcl-2 family. The proapoptotic proteins Bax and Bak are required for MOMP, while the antiapoptotic Bcl-2 proteins, including Bcl-2, Bcl-xL, Mcl-1, and others, prevent MOMP. Different proapoptotic BH3-only proteins act to interfere with the function of the antiapoptotic Bcl-2 members and/or activate Bax and Bak. Here, we discuss an emerging view, proposed by Certo et al. in this issue of Cancer Cell, on how these interactions result in MOMP and apoptosis.”	“Contradict”
“Incidence of heart failure increased by 10% in women since 1979.”	“Clinical epidemiology of heart failure. The aim of this paper is to review the clinical epidemiology of heart failure. The last paper comprehensively addressing the epidemiology of heart failure in Heart appeared in 2000. Despite an increase in manuscripts describing epidemiological aspects of heart failure since the 1990s, additional information is still needed, as indicated by various editorials.”	“Neutral”
Climate FEVER		
Claim	Evidence	Veracity
“In 2015, among Americans, more than 50% of adults had consumed alcoholic drink at some point.”	“For instance, in 2015, among Americans, 89% of adults had consumed alcohol at some point, 70% had drunk it in the last year, and 56% in the last month.”	“Support”
“Dissociative identity disorder is known only in the United States of America.”	“DID is diagnosed more frequently in North America than in the rest of the world, and is diagnosed three to nine times more often in females than in males.”	“Contradict”
“Freckles induce neuromodulation.”	“Margarita Sharapova (born 15 April 1962) is a Russian novelist and short story writer whose tales often draw on her former experience as an animal trainer in a circus.”	“Neutral”

Table 3: Veracity classification samples from the SCIFACT and Climate FEVER datasets.

	All Six Models	Average Single Model
Random	00:05:50	00:00:58
BADGE	00:07:52	00:01:19
CAL	00:14:59	00:02:30
ALPS	00:07:21	00:01:14
Active PETs	00:08:01	00:01:20
Active PETs-o	00:09:10	00:01:32

Table 4: Average run time for a single iteration for each of the sampling methods. The time format is hours:minutes:seconds.

	CFEVER	SCIFACT	Oracle SCIFACT
Active PETs-o	05:53:08	04:12:33	02:31:27

Table 5: Total run time for running active PETs with oversampling iteratively up to 300 instances on different datasets. The time format is hours:minutes:seconds.

Plan-then-Seam: Towards Efficient Table-to-Text Generation

Liang Li^{1,2}, Ruiying Geng³, Chengyang Fang^{1,2}, Bing Li¹, Can Ma^{1*},
Binhua Li³, Yongbin Li^{3*}

¹Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

²School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

³DAMO Academy, Alibaba Group

{liliang, macan, fangchengyang}@iie.ac.cn

{ruiying.gry, binhua.lbh, shuide.lyb}@alibaba-inc.com

Abstract

Table-to-text generation aims at automatically generating text to help people conveniently obtain salient information in tables. Recent works explicitly decompose the generation process into content planning and surface generation stages, employing two *autoregressive* networks for them respectively. However, they are computationally expensive due to the non-parallelizable nature of autoregressive decoding and the redundant parameters of two networks. In this paper, we propose the first totally *non-autoregressive* table-to-text model (Plan-then-Seam, PTS) that produces its outputs in parallel with one single network. PTS firstly writes and calibrates one plan of the content to be generated with a novel *rethinking* pointer predictor, and then takes the plan as the context for seaming to decode the description. These two steps share parameters and perform iteratively to capture token inter-dependency while keeping parallel decoding. Experiments on two public benchmarks show that PTS achieves 3.0 ~ 5.6 times speedup for inference time, reducing 50% parameters, while maintaining as least comparable performance against strong two-stage table-to-text competitors¹.

1 Introduction

Table-to-text generation (Lebret et al., 2016a; Wiseman et al., 2017) is a long-standing problem that aims to generate natural language descriptions of structured table data. A good table-to-text system can help end users better identify the informative elements and their relations from a table. Therefore, developing table-to-text systems is of tremendous value in a wide range of applications, such as biography generation (Lebret et al., 2016a), basketball news generation (Wiseman et al., 2017), advertising text generation (Shao et al., 2019), and table-based question answering (Yu et al., 2019).

*Corresponding authors: Can Ma, Yongbin Li

¹<https://github.com/liang8qi/Plan-then-Seam>

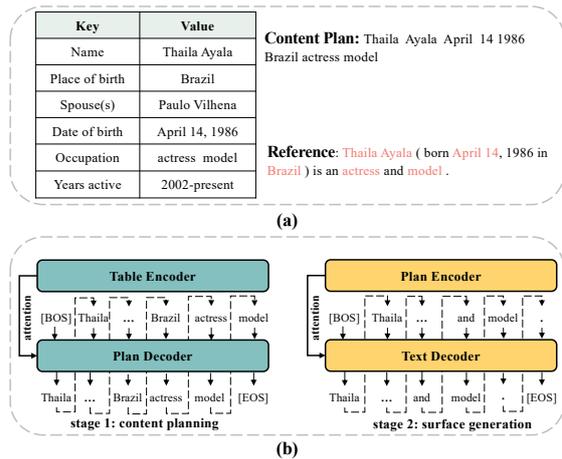


Figure 1: (a): example of table-to-text generation from WikiBio. Tokens from the content planning are colored in red. (b): two-stage model, which disentangles table-to-text generation into two stages: content planning and surface generation.

Recently, neural network-based approaches have made significant progress in this field. The modern neural models for table-to-text generation can be roughly categorized into one-stage models and two-stage models. One-stage models generate natural language descriptions directly from the table by simply relying on representation learning to generate well-organized fluent descriptions. Along this line, some studies propose to modify the model architectures to effectively learn from structured table (Liu et al., 2018; Gong et al., 2019), while some other works introduce auxiliary tasks to help the encoder capture a more accurate semantic representation (Liu et al., 2019; Li et al., 2021). The major drawback of one-stage models is the lack of interpretability and controllability, making the models prone to suffer from unfaithful hallucinations (Wiseman et al., 2017).

To alleviate the aforementioned shortcoming of one-stage models, some researchers propose a two-stage paradigm for table-to-text generation (Puduppully et al., 2019; Su et al., 2021), which explicitly

decomposes the whole generation process into two separate stages: content planning and surface generation (illustrated in Figure 1). The content planning model generates an intermediate sequence that specifies the tokens to be verbalized. The generated plan provides some interpretability and controllability, thus can potentially reduce the risk of hallucinations (Puduppully et al., 2019). The surface generation model then completes the description based on the intermediate plan.

Although two-stage models have some superiority over one-stage models, they are often computationally intensive. The cause of the high computational cost is two-fold. Firstly, most two-stage models use autoregressive (AR) decoders, which is quite time-consuming due to their non-parallelizable nature, especially for long sequences (Gu et al., 2018). Secondly, the two-stage systems often consist of two different models, which usually double the parameter scale (see Section 4.4). The increased parameter scale may introduce more computation overhead and thus slow down the inference speed. These disadvantages limit the deployment of current neural table-to-text systems in practical applications. Recently, non-autoregressive (NAR) generation has attracted much attention because it can significantly accelerate inference speed for text generation (Gu et al., 2018; Stern et al., 2019; Qian et al., 2021a). However, as demonstrated in our preliminary experiments (see also Section 4.4), applying the NAR models directly to table-to-text generation may suffer from lower generation quality because NAR models do not explicitly model the content planning procedure, which can provide good initial input for NAR decoder.

In this work, we propose to reduce the computational cost of two-stage models through a unified NAR framework, which is called *Plan-then-Seam* (PTS). PTS is an iterative NAR table-to-text model. Specifically, PTS first generates the content plan in the first iteration. Then it fills in the other surface tokens in subsequent multiple iterations to seam the intermediate plan tokens. Note that PTS conducts the content planning and surface generation tasks in a single model, thus the model size is smaller than previous two-stage models. Moreover, since PTS is a NAR model, it is more efficient than the AR counterparts. Given that fully NAR content planning may ignore the dependency between planned tokens, we introduce a rethinking pointer predictor, which can better calibrate the planning conditioned

on the generated ones. Our contributions can be summarized as follows:

- We are the first work concerning the computational cost (parameter and inference efficiency) problem in table-to-text. Contrastly, previous works only focus on how to improve the model performance. We hope this can raise more attention to the computational cost problem in table-to-text.
- Regarding methodology, we present the first totally NAR² framework for table-to-text generation, achieving a desired quality–efficiency trade-off. Further, we introduce a rethinking mechanism to improve the NAR planning capability of the model. We demonstrate that initializing the decoder with a good content plan is the key to improving the NAR model.
- Experiments show that, compared with previous strong two-stage competitors, our method can achieve a $3.0 \sim 5.6\times$ speedup with only 50% model parameters without degrading the generation quality.

2 Related Work

Table-to-text generation has long aroused interest in the Natural Language Generation (NLG) community (Kukich, 1983; Reiter and Dale, 1997). Recently, neural models have been the mainstream for this task and made impressive progress. Models for this task can be mainly categorized into two types: one-stage models and two-stage models. One-stage models generate text directly from structured data through a neural encoder-decoder architecture (Sutskever et al., 2014). They simply rely on representation learning to improve the generation. Liu et al. (2018) propose a structure-aware seq2seq architecture, which incorporates the filed information as the additional inputs to the table encoder. Some works design hierarchical table encoder which model table’s representation from the row and column levels (Gong et al., 2019). Liu et al. (2019); Li et al. (2021) introduce auxiliary supervision tasks to help the encoder capture a more accurate semantic representation of the tables. However, one-stage methods are prone to produce unfaithful hallucinations and uncontrollable generation (Wiseman et al., 2017).

As the improvement, neural two-stage models (Ma et al., 2019; Puduppully et al., 2019;

²This means both content planning and surface generation are non-autoregressive.

Moryossef et al., 2019; Puduppully and Lapata, 2021; Su et al., 2021) decompose the table-to-text generation into content planning and surface generation stages. In general, content planning is implemented by Pointer Networks (Vinyals et al., 2015). The explicit content planning mechanism not only decomposes the complex table-to-text generation into two easier tasks but also makes the generation process more interpretable and controllable by generating an intermediate representation. However, the hallucination problem persists in the surface generation stage as it is autoregressive (AR). To address this issue, SANA (Wang et al., 2021) proposes an edit-based non-autoregressive (NAR) surface generation model that generates texts through iterative insertion and deletion operations while maintaining an AR planning stage. Existing two-stage methods solely pay attention to improving the generation quality while ignoring its efficiency. Compared with one-stage models, two-stage methods double the number of parameters. Additionally, the AR generation is slow at inference time. These problems hinder the practical deployment of current neural table-to-text models.

3 Methodology

Given a region of a table as input, table-to-text generation is to produce a natural language description $Y = \{y_1, \dots, y_n\}$ to describe the selected table region. This paper proposes the first totally non-autoregressive table-to-text model, *Plan-then-Seam* (PTS). As depicted in Figure 2, PTS consists of three major components, a table encoder, a non-autoregressive content planning decoder (NAR-P), and a non-autoregressive seaming decoder (NAR-S), which collaborate to generate a description for a source table in an iterative manner. At the first iteration, NAR-P generates content planning sequence in a fully non-autoregressive manner by conditioning on the source table. At the subsequent iterations, NAR-S seams the content planning tokens by inserting connective tokens between them to generate a fluent description. Next, we will describe the proposed PTS in detail.

3.1 Table Encoder

As shown on the left of Figure 2, the source table is a collection of key-value pairs in which each value may contain several tokens. Following LeBret et al. (2016a), we first linearize the source table by flattening all its values to a record se-

quence $T = \{r_1, r_2, \dots, r_K\}$. Each record r_i is represented as a 4-tuple (w_i, k_i, p_i^+, p_i^-) , where w_i is the value token, k_i is its key name. p_i^+ and p_i^- are the relative positions of w_i , where p_i^+ counts from the beginning and p_i^- counts from the end of the sentence. For example, the key-value pair <Name, Thaila Ayala> is represented as two records: (Thaila, Name, 1, 2) and (Ayala, Name, 2, 1). We adopt four trainable embedding matrices to convert each record represented by (w_i, k_i, p_i^+, p_i^-) into dense vectors e_{w_i} , e_{k_i} , $e_{p_i^+}$, and $e_{p_i^-}$. We concatenate these embeddings and use a linear projection to map the four vectors into e_i , which serves as the initial representation of the corresponding record r_i :

$$\mathbf{e}_i = \text{ReLU}(\mathbf{W}_e[e_{w_i}; e_{k_i}; e_{p_i^+}; e_{p_i^-}] + \mathbf{b}_e), \quad (1)$$

where \mathbf{W}_e and \mathbf{b}_e are trainable parameters. $[\cdot; \cdot]$ denotes the vector concatenation operation. Finally, we transform $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_K\}$ into contextual sequence representation $\mathbf{H}^e = \{\mathbf{h}_1^e, \mathbf{h}_2^e, \dots, \mathbf{h}_K^e\}$ with the Transformer encoder (Vaswani et al., 2017).

3.2 Non-autoregressive Content Planning Decoder

We utilize the Transformer decoder layer (Vaswani et al., 2017) as the basic building block of the content planning component. We also remove the causal mask in self-attention modules to realize parallel generation. As shown in Figure 2, given the initial decoder input $y^0 = [\text{BOS}][\text{EOS}]$, non-autoregressive planning decoder (NAR-P) aims to generate the planned sequence (e.g., $y^p = \text{Thaila Ayalia actress model}$). y^p specifies the records that are to be verbalized (*what to say*) in the description and the order in which they are described. To this end, NAR-P consists of three major components: a placeholder predictor π^l , a pointer predictor π^p , and a token deleter π^d . These components work in a serial fashion. First, the placeholder predictor π^l determines the number of plan tokens to be inserted:

$$\pi^l(l|y^0, T) = \text{Softmax}(\mathbf{W}_l[\mathbf{h}_0^{d_1}; \mathbf{h}_1^{d_1}]), \quad (2)$$

where $\mathbf{h}_0^{d_1}$ and $\mathbf{h}_1^{d_1}$ are respectively the decoder states of two symbol tokens in y^0 , $\mathbf{W}_l \in \mathbb{R}^{L \times 2d}$ is the projection matrix and L is the pre-defined maximal placeholder number. $\pi^l(l) \in \mathbb{R}^L$ denotes the probability distribution of possible placeholder numbers, and we choose the one l with the

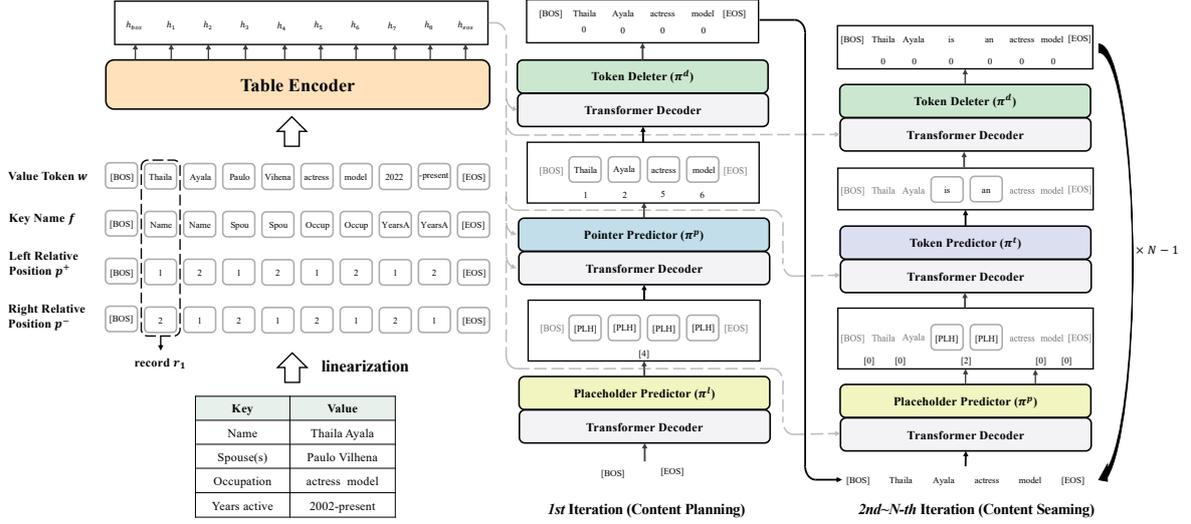


Figure 2: An overview of our Plan-then-Seam non-autoregressive table-to-text model. The modules with the same color share the parameters. The left part contains an example of how to linearize a table, where Spou, Occup, and YearsA denote Spouse(s), Occupation, and Years active, respectively.

highest probability. We insert l placeholders [PLH] between [BOS] and [EOS] to obtain the placeholder sequence y^l .

Then, we need to replace each [PLH] in y^l with an actual token. y^l is firstly passed into the Transformer decoder layer to generate the decoder state $\mathbf{h}_i^{d_2}$ of each placeholder. As all plan tokens are from the source table, we then introduce a pointer predictor π^p that selects tokens from T to reduce hallucinations. Specifically, for the i -th placeholder, we calculate the confidence scores α_j^i of copying the j -th record in T as a plan token by:

$$\pi^p(\alpha_j^i | y^l, T) = \text{Softmax}(\mathbf{W}_p[\mathbf{h}_i^{d_2}; \mathbf{h}_j^e]). \quad (3)$$

Then we replace each placeholder with the most possible record to get y^r .

Last, considering the predictor π^p may copy incorrect or repetitive tokens, we build a token deleter π^d to remove these false plan tokens. For the i -th token in y^r , π^d is employed to decide whether it is required to be deleted or not:

$$\pi^d(d_i | y^r, T) = \text{Softmax}(\mathbf{W}_d \mathbf{h}_i^{d_2}), \quad (4)$$

where $\mathbf{h}_i^{d_2}$ is the representation generated by transformer decoder. $\pi^d(d_i) \in [0, 1]$ is the predicted probability of the deletion operation. The token with $\pi^d(d_i) > 0.5$ is deleted, which yields the final content plan y^p .

Rethinking Pointer Predictor Although non-autoregressive models can accelerate the generation process, they are based on the assumption

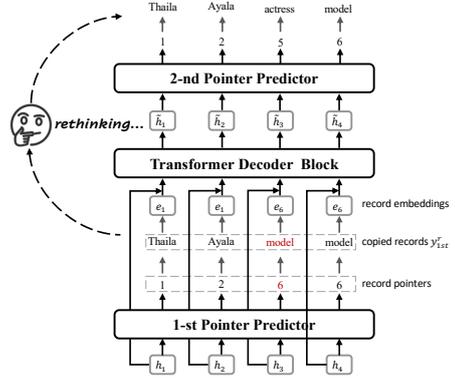


Figure 3: Architecture of the proposed rethinking pointer predictor. Our basic idea is to augment the non-autoregressive predictor with inter-token dependencies.

that the generated tokens are conditionally independent with each other (Gu et al., 2018). As a result, the pointer predictor may suffer from incoherence or repetition (Qian et al., 2021a). We believe that the pointer predictor would produce a better plan by partially observing generated plan tokens. Motivated by this intuition, we adapt the naive pointer predictor and propose its variant *rethinking* pointer predictor. As illustrated in Figure 3, the rethinking pointer predictor first employs a naive pointer predictor to generate a primary record plan $y_{1st}^r = \{r_1, r_2, r_6, r_6\}$, then calibrate it with another pointer predictor. Particularly, for each record r_i in y_{1st}^r , we concatenate its embedding e_i with the transformer hidden state h_i and further process it by a linear layer $\hat{h}_i = W_f[h_i; e_i]$. \hat{h}_i is fed into a transformer decoder block (removing

causal attention) to rebuild representation for the token at i -th position in y_{1st}^r by conditioning on the source table T and the generated plan y_{1st}^r . By looking at the tokens in other positions, the 2-nd pointer predictor can determine if the token at the i -th position is incorrect or repeated with others. The *rethinking* process helps the model adjust the incorrect tokens in y_{1st}^r to generate a precise plan. Moreover, to improve the confidence of y_{1st}^r , both pointer predictors are supervised by the ground truth plan when training.

3.3 Non-autoregressive Content Seaming

Decoder

After obtaining the content plan y^p , at the seaming stage, the non-autoregressive seaming decoder (NAR-S) constructs a fluent description y by iteratively inserting the connective tokens into y^p . Specifically, we employ y^p as the initial input of NAR-S. In each iteration, similar to NAR-P, NAR-S also firstly predicts the number of placeholders should be inserted into at every consecutive position pairs in the generated sentence of the previous iteration, then replace placeholders with actual tokens, and finally delete abundant tokens. NAR-S share parameters in placeholder predictor π^l and token deleter π^d with NAR-P. The difference is that NAR-S replaces the pointer predictor π^p with a token predictor π^t that replaces the placeholders with actual tokens from the predefined vocabulary rather than copies from the table. For example, as shown on the right of Figure 2, given the content plan *Thaila Ayalia actress model*, NAR-S inserts two tokens, *is* and *an*, between *Ayalia* and *actress* in the first iteration.

3.4 Training

We joint train the content planning and seaming tasks and the final learning objective is:

$$\mathcal{L} = \lambda \mathcal{L}_{plan} + \mathcal{L}_{seam}, \quad (5)$$

where λ is the hyper parameter. Next, we describe them in detail.

The content planning learning objective consists of three sub-goals: $\mathcal{L}_{plan} = \mathcal{L}_l^p + \mathcal{L}_p^p + \mathcal{L}_d^p$. Given the source table T , the ground truth content plan y^{p*} as well as its pointers $y^{idx} = \{y_i^{idx}\}_{i=1}^{|y^{p*}|}$ with each entry pointing to an input record in T , the placeholder predictor learning objective \mathcal{L}_l^p is computed as follows:

$$\mathcal{L}_l^p = -\log \pi^l(l_0^* | y^0, T), \quad (6)$$

where l_0^* is the length of ground truth plan y^{p*} . And then, we replace all the tokens in y^{p*} with [PLH] to get y^l , which is utilized to train the pointer predictor:

$$\mathcal{L}_p^p = -\sum_{i=1}^{|y^{p*}|} \log \pi^p(\alpha_{y_i^{idx}}^i | y^l, T). \quad (7)$$

To train the deletion predictor, we apply the pointer predictor π^p to y^l to yield y^r . The loss for deletion predictor is calculated as:

$$\mathcal{L}_d^p = -\sum_{i=1}^{|y^r|} \log \pi^d(d_i | y^r, T), \quad (8)$$

where d_i is the golden deletion operation at the i -th position, and is set as 1 if y_i^r is same with y_i^{p*} , otherwise 0.

The seaming loss also consists of three parts: $\mathcal{L}_{seam} = \mathcal{L}_l^s + \mathcal{L}_p^s + \mathcal{L}_d^s$. Its training process is very similar to content planning task. The biggest difference between them is the initial input. Given a source table, a plan and a reference (T, y^{p*}, y^*), we follow previous works (Gu et al., 2019; Wang et al., 2021) to construct an intermediate sequence y^m as the initial input to NAR-S. Specially, we first calculate the longest common subsequence \hat{y} between y^{p*} and y^* . And then we apply random deletion on y^* except the part of \hat{y} to obtain y^m . Last, the three subgoals are calculated as following:

$$\mathcal{L}_l^s = -\sum_{i=1}^{|y^m|} \log \pi^l(l_i^* | y^m, T), \quad (9)$$

$$\mathcal{L}_p^s = -\sum_{i=1}^{|y^*|} \log \pi^t(y_i^* | y^l, T), \quad (10)$$

$$\mathcal{L}_d^s = -\sum_{i=1}^{|y^*|} \log \pi^d(d_i | y^t, T), \quad (11)$$

where l_i^* is the number of placeholder that should be inserted between y_i^m and y_{i+1}^m . l_i^* is obtained by calculating the Levenshtein distance (Levenshtein et al., 1966) between y^m and y^* . y^t is yielded by applying π^t on y^l .

3.5 Inference

As mentioned above, PTS is an iterative NAR model. Different from the previous iterative NAR model, at the first iteration, PTS first utilizes NAR-P to generate the content plan in a fully NAR manner, where NAR-P alternately performs placeholder

prediction, pointer prediction, and deletion operation. In the subsequent iterations, it uses the generated plan as the initial decoder input for NAR-S, which iteratively fills in the other surface tokens between the content planning tokens. We stop the seaming process when the current text does not change or the predefined maximum iteration has been reached.

4 Experiment

4.1 Datasets and Evaluation Metrics

Following Wang et al., we conduct experiments on two datasets, WikiBio (Lebret et al., 2016b) and WikiPerson (Wang et al., 2018). Both datasets are designed to generate descriptions from a Wikipedia table. Specifically, WikiBio aims to generate the first sentence of a biography. The average length of the description is 26.1 tokens. Different from Wikibio, the reference of WikiPerson contains multiple sentences to cover as many factors in the source table as possible. The average length of the description in WikiPerson is 70.6. We use the official training, development, and test splits for both datasets, which are 582,657/72,831/72,831 in WikiBio and 250,186/30,487/29,982 in WikiPerson. We use these two datasets for two considerations. First, this paper focuses on the inference speed and generation quality of models with similar frameworks. The similar input structures allow us to use the same encoder architecture and prevent us from designing an additional one. Second, the different output length distributions of the two datasets facilitate us to compare the models’ performance and efficiency.

We use BLEU (Papineni et al., 2002) and ROUGE-L to evaluate the fluency, and PARENT (Dhingra et al., 2019) to examine the faithfulness. We also employ inference latency to evaluate the inference speed of the involved approaches. Specifically, Latency is the average time to run an epoch on the test dataset while the batch size is set to 32 with one NVIDIA Tesla V100 GPU.

4.2 Baselines

To rule out the effect of model architecture on the inference speed, we only compare our method to some representative baselines built on the Transformer (Vaswani et al., 2017) model:

- TABLETRANSFORMER is a transformer-based model that replaces the naive transformer encoder with the table encoder.

- LEVT (Gu et al., 2019) is an iterative NAR model. In the first iteration, the decoder input is initialized by “[BOS][EOS]”.
- CONTENT-PLAN (Pudupully et al., 2019) is a representative two-stage method that firstly uses a pointer network to generate the content plan and then uses a pointer generator to complete the remaining text. To make a fair comparison, we reimplement it using Transformer. See Appendix B.2 for more details.
- SANA (Wang et al., 2021) is also a two-stage method. The major difference between SANA and CONTENT-PLAN lies in that SANA uses a LEVT for surface token generation. Additionally, SANA integrates hard constraints by forbidding the LEVT from deleting planned tokens.

4.3 Implementation Details

Our method is implemented by fairseq (Ott et al., 2019). For fair comparison, all the involved systems use a similar configuration. Specifically, the vocabulary sizes on WikiBio and WikiPerson are 30K and 50K, respectively. The dimensions of token embedding, key embedding and position embedding are set to 420, 80, and 50, respectively. All Transformer components used in our methods adopt the base Transformer setting with $d_{model} = 512$, $d_{hidden} = 2048$, and $n_{head} = 8$. The depth is 6 for both the encoder and the decoder. Please refer to Appendix B.1 for more details about training setting. During inference, the maximum iterations of the NAR model is 10 and 40 in WikiBio and WikiPerson, respectively. We conduct experiments over 4 different random seeds and report the average scores.

4.4 Main Results

Table 1 shows the performance of our method and the baselines. For WikiBio, the NAR LEVT model are approximately $3\times$ faster than the AR TABLETRANSFORMER model. However, the description quality of LEVT is much lower than TABLETRANSFORMER, regarding both fluency (-1.27 BLEU) and faithfulness (-4.62 PARENT-F1). Moreover, we observe that two-stage approaches can outperform the one-stage ones (e.g., SANA vs. LEVT), indicating the superiority of explicitly content planning. However, they double the parameters scale and increase the inference latency. Surprisingly, our proposed method can combine the advantages

Models	BLEU	ROUGE-L	PARENT (P / R / F1)	#Param	Latency↓	I _{DEC} ↓
<i>One-Stage Systems</i>						
TABLETRANSFORMER	44.32 _{±0.32}	66.75 _{±0.36}	74.09 _{±0.32} / 42.41 _{±0.18} / 51.76 _{±0.31}	76	680	22.10
LEVT	43.05 _{±0.21}	65.61 _{±0.31}	72.22 _{±0.16} / 37.62 _{±0.14} / 47.14 _{±0.11}	74	223	2.48
<i>Two-Stage Systems</i>						
CONTENT-PLAN	43.44 _{±0.00}	66.21 _{±0.31}	74.55 _{±0.29} / 43.45 _{±0.30} / 52.38 _{±0.11}	150	1,381	30.47
SANA †	45.78	-	76.93 / 46.01 / 55.42	-	-	-
w/o hard constraints †	45.31	-	76.32 / 45.26 / 54.64	-	-	-
SANA	45.50 _{±0.13}	67.98 _{±0.15}	77.01 _{±0.25} / 45.52 _{±0.08} / 55.16 _{±0.10}	148	756	11.02
w/o hard constraints	44.94 _{±0.16}	67.72 _{±0.16}	76.89 _{±0.26} / 44.70 _{±0.26} / 54.68 _{±0.59}	148	761	11.03
OURS	45.65 _{±0.13}	68.30 _{±0.38}	77.29 _{±0.24} / 45.80 _{±0.10} / 55.41 _{±0.24}	80	245	3.49
w/o rethinking	45.21 _{±0.07}	67.99 _{±0.16}	76.88 _{±0.39} / 45.29 _{±0.10} / 55.07 _{±0.17}	75	290	3.63

(a) Results on WikiBio

Models	BLEU	ROUGE-L	PARENT (P / R / F1)	#Param	Latency↓	I _{DEC} ↓
<i>One-Stage Systems</i>						
TABLETRANSFORMER	25.11 _{±0.63}	44.06 _{±0.56}	61.13 _{±0.89} / 52.08 _{±0.90} / 54.45 _{±0.41}	92	2,092	62.42
LEVT	22.10 _{±0.36}	43.60 _{±0.57}	61.43 _{±0.44} / 49.58 _{±0.00} / 53.65 _{±0.16}	91	449	3.60
<i>Two-Stage Systems</i>						
CONTENT-PLAN	25.17 _{±0.78}	44.47 _{±0.03}	62.09 _{±0.55} / 53.63 _{±0.44} / 56.68 _{±0.29}	187	2,708	82.61
SANA †	25.23	-	65.69 / 56.88 / 59.96	183	-	-
w/o hard constraints †	24.97	-	64.72 / 56.42 / 59.29	183	-	-
SANA	24.95 _{±0.31}	45.35 _{±0.13}	69.26 _{±0.83} / 58.16 _{±0.06} / 62.39 _{±0.15}	183	1,370	29.20
w/o hard constraints	22.37 _{±0.38}	45.08 _{±0.15}	69.10 _{±0.49} / 56.62 _{±0.38} / 61.38 _{±0.28}	183	1,217	28.92
OURS	25.11 _{±0.35}	45.23 _{±0.31}	69.72 _{±0.63} / 58.12 _{±0.49} / 62.58 _{±0.36}	97	547	4.83
w/o rethinking	24.45 _{±0.28}	44.87 _{±0.21}	68.17 _{±0.74} / 57.45 _{±0.52} / 61.55 _{±0.43}	92	572	4.95

(b) Results on WikiPerson

Table 1: Results on WikiBio and WikiPerson test sets. Results marked with “†” are copied from previous studies while the other results are implemented in this work. Latency and I_{DEC} denote the average inference time and the average number of decoder iterations, respectively. Mean (±s.d.) over 4 seeds.

of both the two kinds of baselines. On both WikiBio and WikiPerson, our approach can achieve comparable description quality with the strong two-stage baseline (i.e., SANA), while maintaining the model size and the inference speed. Compared with SANA, our model does not require external constraints to guarantee the appearance of planned tokens in the final output. The results also demonstrate the effectiveness of the newly proposed rethinking mechanism, confirming that the inter-dependency between different positions is essential for NAR-P, which can provide a better starting point for NAR-S. Additionally, we notice that the latency increase without rethinking. We believe this is because removing this module reduces the content planning capability of the model, which in turn lowers the quality of the initial input to NAR-S, making the model require more iterations to satisfy the termination condition. The results on WikiPerson show a similar trend to WikiBio. An obvious difference is that the inference speed is much slower for all models, since the average description length is longer than WikiBio (70.6 vs. 26.1). When generating longer sentences,

the speedup of our method over the AR baseline is much higher. On both datasets, our method can achieve high description quality and inference efficiency at the same time.

4.5 Analysis and Discussion

Due to the page limit, we have placed more experimental results and analyses in Appendix A.

Content Planning As mentioned above, explicit content planning is important for table-to-text generation. We thus further investigate the content planning performance in Table 2. We compare our method with two baselines: POINTERNETWORK and NAR-P. POINTERNETWORK is a widely used planning method for two-stage models (Puduppully et al., 2019; Wang et al., 2021). The results indicate that our proposed PTS model performs comparable with POINTERNETWORK. NAR-P has a same architecture with PTS, the difference is that NAR-P is totally trained with the content planning task, while PTS is trained to perform both content planning and seaming. The results show that training the model with both content planning and seaming does not significantly affect the planning perfor-

Models	BLEU	#Param	Latency
<i>WikiBio</i>			
POINTERNETWORK	64.97	76	261
w/o beam search	62.03	76	259
NAR-P	64.78	80	54
w/o rethinking	64.36	75	59
PTS-PLAN	64.75	80	64
w/o rethinking	64.43	75	59
<i>WikiPerson</i>			
POINTERNETWORK	52.42	91	851
w/o beam search	43.48	91	926
NAR-P	52.51	97	61
w/o rethinking	52.14	92	63
PTS-PLAN	52.27	97	60
w/o rethinking	51.67	92	58

Table 2: Performance of different content planning models. “NAR-P” is solely trained using the content planning objective, while “PTS-PLAN” is to use the final PTS model to perform content planning.

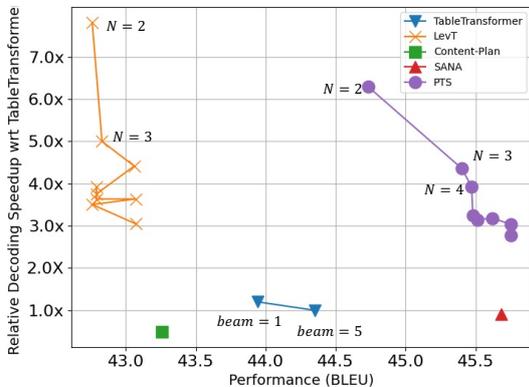


Figure 4: Quality-Speed trade-off on the WikiBio test set. Quality is estimated by BLEU. For clarity, the inference speed is measured by the relative speedup with respect to TABLETRANSFORMER with beam size = 5. $N \in [1, 10]$ denotes the maximum iteration number.

mance, which implies that learning the two tasks with a unified model does not decrease the model’s planning ability.

Quality-Speed Trade-off Since PTS is an iterative NAR model, it is easy to balance the description quality and the inference speed by changing the number of the iterations. As shown in Figure 4, PTS achieves comparable performance with substantially higher speed up than the other involved models. For PTS, increasing the iteration number can improve the description quality while reduce the inference speed. In practice, we can change the iteration number to meet different requirements under various application cases.

Tokens Generated at the Seaming Stage To build a deeper understanding of the proposed PTS model, we investigate the problem of which tokens

Token	Percentage (%)
,	8.01%
.	7.69%
-lrb-	7.28%
-rrb-	7.26%
a	5.21%
is	5.07%
born	4.51%
the	3.12%
an	2.74%
was	2.27%
and	2.41%
in	2.26%
-	2.13%
of	2.12%
who	1.64%
Total	64.18%

Table 3: Top 15 tokens generated by our proposed PTS at the seaming stage on WikiBio test set. -lrb- and -rrb- represent (and), respectively.

are most likely to be generated by PTS at the seaming stage on the WikiBio test set. Specifically, we first remove the words generated at the planning stage from the descriptions generated by PTS to obtain a word set. Then, we count the token frequency for the word in the set. We present the top 15 frequent tokens in Table 3. As we can see, at the seaming stage, PTS is more likely to generate connective tokens, e.g., punctuation, and copulas, than the specific tokens existing in the input table, such as name, time, etc. And the connective tokens are mainly used to link planning words (seaming). The observation is consistent with our motivation and design that PTS first copies content from the input table to construct a plan sequence and then inserts tokens from a pre-defined vocabulary between plan tokens to generate a fluent description. We believe this make our approach more interpretable and controllable.

5 Human Evaluation

To verify whether the system performance is consistent with what the automatic metrics show, we further conduct a human evaluation on the WikiBio test set. We randomly sample 50 instances from each model’s generated outputs. Then, we invite three graduate students, whose English level is very high to understand the text, to score each generated text from 1 to 5 in terms of two criteria: Fluency (is

Models	Fluency	Faithfulness
TABLETRANSFORMER	3.24 \pm 0.77	2.89 \pm 0.86
LEVT	2.75 \pm 0.89	2.57 \pm 0.78
CONTENT-PLAN	3.18 \pm 0.32	3.01 \pm 0.72
SANA	3.31 \pm 0.74	3.26 \pm 0.69
OURS	3.42 \pm 0.58	3.42 \pm 0.59

Table 4: Human evaluation results on WikiBio test set.

the sentence fluency?) and Faithfulness (is the sentence related to the input table?). For each criterion, we average the scores from all annotators as the final score. When evaluating, each annotator is provided the input tables as the references and does not know which model the generated text comes from. The results are summarized in Table 4. As we can see, the overall trend of the human evaluation is similar to the automatic metrics in Table 1. First, the two-stage systems have an advantage over the one-stage ones in generating result fidelity. Second, our method is competitive to these table-to-text baselines regarding generation fluency and faithfulness. Meanwhile, our approach performs better than the LEVT transferred from machine translation. These results indicate that introducing the content planning process in the NAR process and using its results as the initial decoder input can significantly improve the NAR model.

6 Conclusion

We propose a unified non-autoregressive framework, Plan-then-Seam (PTS), for table-to-text generation. Given a source table, PTS first generates the content plan in a fully NAR manner. Then we iteratively fill in the other surface tokens. Experimental results demonstrate that PTS achieves a 3.0 \sim 5.6 speedup with only 50% model parameters compared with previous two-stage table-to-text models, without degrading the description quality. Further analysis reveals that the success of PTS comes from the proposed NAR-P with a rethinking mechanism, whose content planning performance is comparable with AR models. By changing the iteration number, PTS can balance the generation quality and inference efficiency for various practical application requirements.

Limitations

As described in the paper, the content planning ability is important for table-to-text models. However, the planning performance of all the involved methods is still far from satisfactory. We will explore

more advanced methods to improve the content planning performance. Moreover, we train all the models on WikiBio and WikiPerson from scratch, and the training cost is rather expensive: 2.5 days using 4 NVIDIA V100 32G GPUs. Lastly, this paper does not compare the pre-trained language models (PLMs) (Devlin et al., 2019; Raffel et al., 2020), though our approach may also benefit from some pre-trained table encoders, such as TAPAS (Müller et al., 2021). The main reasons why we do not consider PLMs are that PLMs will bring an unfair comparison and bring more variables and may make our work lose focus. See Appendix B.2 for detailed justification. In the future, we will explore how pre-trained models, e.g., pre-trained table encoder TAPAS, can be used to improve our model’s performance and accelerate the training process.

Ethics Statement

We consider our work can make more researchers in table-to-text pay attention to the computational cost problem, which may benefit from saving the cost of the online table-to-text model. We experimented on the public datasets with no discriminatory or insulting sentences.

References

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. [Handling divergent reference texts when evaluating table-to-text generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.
- Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. [Mask-predict: Parallel decoding of conditional masked language models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.
- Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019. [Table-to-text generation with effective hier-](#)

- archical encoder on three dimensions (row, column and time). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3143–3152, Hong Kong, China. Association for Computational Linguistics.
- Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.
- Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.
- Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020. Incorporating bert into parallel sequence decoding with adapters. *Advances in Neural Information Processing Systems*, 33:10843–10854.
- Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
- Karen Kukich. 1983. Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics*, pages 145–150, Cambridge, Massachusetts, USA. Association for Computational Linguistics.
- Rémi Lebret, David Grangier, and Michael Auli. 2016a. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.
- Rémi Lebret, David Grangier, and Michael Auli. 2016b. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.
- Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.
- Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Liang Li, Can Ma, Yinliang Yue, and Dayong Hu. 2021. Improving encoder by auxiliary supervision tasks for table-to-text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5979–5989, Online. Association for Computational Linguistics.
- Tianyu Liu, Fuli Luo, Qiaolin Xia, Shuming Ma, Baobao Chang, and Zhifang Sui. 2019. Hierarchical encoder with auxiliary supervision for neural table-to-text generation: Learning better representation for tables. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6786–6793. AAAI Press.
- Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text generation by structure-aware seq2seq learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4881–4888. AAAI Press.
- Shuming Ma, Pengcheng Yang, Tianyu Liu, Peng Li, Jie Zhou, and Xu Sun. 2019. Key fact as pivot: A two-stage model for low resource table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2047–2057, Florence, Italy. Association for Computational Linguistics.
- Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2267–2277. Association for Computational Linguistics.

- Thomas Müller, Julian Eisenschlos, and Syrine Krichene. 2021. [TAPAS at SemEval-2021 task 9: Reasoning over tables with intermediate pre-training](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 423–430, Online. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [fairseq: A fast, extensible toolkit for sequence modeling](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. [Data-to-text generation with content selection and planning](#). In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6908–6915. AAAI Press.
- Ratish Puduppully and Mirella Lapata. 2021. [Data-to-text generation with macro planning](#). *Transactions of the Association for Computational Linguistics*, 9:510–527.
- Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021a. [Glancing transformer for non-autoregressive neural machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003, Online. Association for Computational Linguistics.
- Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021b. [Glancing transformer for non-autoregressive neural machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003, Online. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Ehud Reiter and Robert Dale. 1997. [Building applied natural language generation systems](#). *Nat. Lang. Eng.*, 3(1):57–87.
- Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. [Non-autoregressive machine translation with latent alignments](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1098–1108, Online. Association for Computational Linguistics.
- Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. [Long and diverse text generation with planning-based hierarchical variational model](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3257–3268, Hong Kong, China. Association for Computational Linguistics.
- Jongyoon Song, Sungwon Kim, and Sungroh Yoon. 2021. [AlignNART: Non-autoregressive neural machine translation by jointly learning to estimate alignment and translate](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1–14, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. [Insertion transformer: Flexible sequence generation via insertion operations](#). In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985. PMLR.
- Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. [Plan-then-generate: Controlled data-to-text generation via planning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to sequence learning with neural networks](#). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. [Pointer networks](#). In *Advances in Neural*

Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 2692–2700.

Peng Wang, Junyang Lin, An Yang, Chang Zhou, Yichang Zhang, Jingren Zhou, and Hongxia Yang. 2021. [Sketch and refine: Towards faithful and informative table-to-text generation](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4831–4843, Online. Association for Computational Linguistics.

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. [Describing a knowledge base](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 10–21, Tilburg University, The Netherlands. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. [Challenges in data-to-document generation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. 2019. [CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1962–1979, Hong Kong, China. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. [Incorporating bert into neural machine translation](#). *arXiv preprint arXiv:2002.06823*.

Models	Token Repetitions (%)	Dist-1	Dist-2
Gold	12.58	5.6	22.83
TableTransformer	9.29	1.5	10.87
LevT	14.59	2.1	13.97
Content-Plan	9.81	4.7	15.11
SANA	9.93	5.0	19.32
Ours	8.22	4.9	18.15

Table 5: The token repetitions and diversity on WikiBio test dataset. Dist-1 and Dist-2 denote Distinct-1 and Distinct-2, respectively.

Models	BLEU	PARENT(P / R / F1)
CONTENT-PLAN	43.72	76.55 / 38.79 / 49.02
+ Golden Plan	51.16	76.32 / 47.33 / 56.45
SANA	45.68	76.79 / 45.64 / 55.12
+ Golden Plan	54.30	80.03 / 51.02 / 61.01
OURS	45.75	77.34 / 45.91 / 55.48
+ Golden Plan	55.50	79.59 / 51.92 / 61.14

Table 6: Effect of golden plan on two-stage methods.

A More Experimental Results

A.1 Token Repetitions and Diversity

Previous works manifest NAR model tends to predict the same token with high confidence, but at different positions, which is caused by the multimodality problem. Therefore, we doubt whether the NAR table-to-text generation has any preference towards token repetitions and diversity. We measure the percentage of repetitive tokens in the generated sent as a proxy metric for the multimodality problem (Gu et al., 2018). Additionally, we utilize Distinct-1 and Distinct-2 (Li et al., 2016) to evaluate the diversity of the output text. All results are summarized in Table 5. We observe that the AR model TABLETRANSFORMER significantly reduces the lexical diversity. Therefore, to better train the non-autoregressive model, AR model is usually used as a teacher model to reduce the complexity of the training corpus (Knowledge Distillation) (Gu et al., 2018). And then, LEVT tends to generate repetitive tokens. We can see that, when explicitly modelling the content planning, two-stage methods can increase the tokens diversity. Especially, the content planning can substantially reduce the tokens repetitions for NAR models (e.g., LEVT vs. SANA and OURS).

A.2 Performance Bottleneck of Two-stage Model

We provide the ground-truth content plan to the models at the second stage, and the results are summarized in Table 6. When fed with the golden plan, all the two-stage models achieves better fluency

and faithfulness. The results indicate that the quality of content planning is a important bottleneck for two-stage table-to-text approaches.

A.3 Case Study

Table 7 shows the descriptions generate by PTS from the test set of WikiBio. First, we observe that when the number of tokens in the generated plan is relatively small, the 1-st pointer predictor can generate a precise content plan. However, when the number of planning tokens increases, it tends to produce repetitive and incorrect ones. We consider this is because the fully NAR generation cannot accurately model the dependencies between planning tokens. After introducing the *rethinking* mechanism, the 2-nd pointer predictor can determine if the token is incorrect or repeated with others and calibrate it by looking at the tokens in other positions. Therefore, the model can generate a more precise plan.

B More Implementation Details

B.1 Training and Hyper-parameter Settings

All models are optimized by Adam (Kingma and Ba, 2015). We use the same learning rate schedule as presented in Vaswani et al. (2017). The maximum value of the learning rate is $5e-4$ and the warmup step is set to 10K. The maximum training step is set to 300K. We use the validation BLEU for early stopping and explore $\lambda = [0.05, 0.08]$. During inference, we use beam search with a beam size 5 for the autoregressive models and the maximum decoding lengths are set to 80 and 160 in WikiBio and Wikiperson. For non-autoregressive models, we set the maximum iterations as 10 and 40 in WikiBio and WikiPerson, respectively.

B.2 Experimental Setting Details

To rule out the effect of model architecture on the inference speed and make a fair comparison, we only compare our method to some table-to-text models built on the Transformer model. For the one-stage models, we chose the autoregressive TableTrasforme and non-autoregressive LevT. For the two-stage methods, we compare with Content-Plan and SANA. Both planning generation and tableau generation of the former are autoregressive, while the second stage of the latter is a non-autoregressive process. All these baselines employ the same table encoder as ours. Additionally, the original Content-Plan is implemented by LSTM.

To make a fair comparison, we re-implemented Content-Plan by replacing its LSTM-based encoder and decoder with Transformer-based ones. And the transformer setting is the same as our model.

We do not consider the pre-trained models (PLMs) in this paper, though our model’s performance may be significantly improved by initializing our table encoder with the pre-trained one, such as TAPAS (Herzig et al., 2020). The reasons why we do not consider PLMs are as follows:

- We consider PLMs will bring an unfair comparison. Because most PLMs (e.g., TAPAS, T5 (Raffel et al., 2020)) are pre-trained on Wikipedia data, and WikiBio and WikiPerson are built from Wikipedia. It may lead to data leakage. Moreover, to our best knowledge, most of the works in the NAR machine translation (please refer to Appendix C) do not compare with PLMs.
- This paper focuses on comparing inference speed and quality under a similar model architecture rather than improving the model performance. And our experimental setting is fair, and all the baselines employ a similar setting as our model. Additionally, in related domains such as neural machine translation, previous work (Zhu et al., 2020) indicates that simply initializing the encoders of sequence-to-sequence models with the pre-trained BERT (Devlin et al., 2019) will actually hurt the performance. And directly fine-tuning NAR sequence-to-sequence models initialized by BERT is very unstable and sensitive to the learning rate (Guo et al., 2020). Therefore, though pre-trained checkpoint may benefit our model, it will bring more variables and may make our work lose focus. We leave this for future work.

B.3 Content Plan Annotation

We follow previous work (Wang et al., 2021) to employ the heuristic method to obtain the content plan annotation for WikiBio and WikiPerson. Specifically, we start by counting the tokens that appear both in the table and in the corresponding description. Then we remove the stop tokens in the tokens collection and sort the rest of the tokens by their positions in the description in ascending order. The sorted sequence is regarded as the content planning sequence. We refer the readers to Wang et al. (2021)’s paper for more details.

C Non-autoregressive Neural Machine Translation

Recently, autoregressive (AR) models have achieved outstanding performances in natural language generation tasks (Raffel et al., 2020). However, AR is quite time-consuming when generating target sentences, especially for long sentences. To overcome this problem and accelerate decoding, Gu et al. (2018) first propose the non-autoregressive generation (NAR) for machine translation, which generates all the target tokens in parallel and hugely increases the inference speed. Therefore, much attention has been attracted to NAR with impressive progress (Stern et al., 2019; Qian et al., 2021a; Song et al., 2021; Qian et al., 2021b). However, compared with AR models, the generation quality is sacrificed because NAR breaks the conditional dependence assumption that prevents a model from properly capturing the highly multi-modal distribution of target translations, which is called the "multi-modality" problem (Gu et al., 2018). To mitigate the problem, some studies (Lee et al., 2018; Stern et al., 2019; Ghazvininejad et al., 2019; Gu et al., 2019; Saharia et al., 2020) propose the iterative NAR models which need N iterations for inference and keep the non-autoregressive property in every iteration step. More specifically, the generated results of the previous iteration will be fed into the decoder again for refinements. In this way, partial target information is provided in each iteration step.

First Example			
Source Table	< Name: sean macias>, < Birth Place: california>, < Known For: litigation>, < Occupation: lawyer>, < Nationality: american>, < Article Title: sean macias>.		
Flatten Table	(Name, sean, 1, 2), (Name, macias, 2, 1), (Birth Place, california, 1, 1), (Known For, litigation, 1, 1), (Occupation, lawyer, 1, 1), (Nationality, american, 1, 1), (Article Title, sean, 1, 2), (Article Title, macias, 2, 1).		
Rerence	sean ernesto macias -lrb- born 31 october 1972 -rrb- is a pasadena-based litigation lawyer known for handling high-profile cases .		
PTS	1st iteration	1st pointer predictor	sean macias american litigation lawyer
	2nd Iteration	2nd pointer predictor	sean macias american litigation lawyer
Second Example			
Source Table	< Name: dave green>, < Poosition: punter placekicker >, < Number: 4>, < Birth Date: 21 september 1949>, < Debutyear: 1973>, < Finalyear: 1978>, < Draftyear: 1972>, < Draftround: 17>, < Draftpick: 418>, < College: ohio university>, < Statlabel: punts punting yards punting avg 446 17,883 40.1>, < Nfl: re162282>, < Brith Place: mason city iowa> ,< Article Title: dave green -lrb- american football -rrb->.		
Flatten Table	(Name, dave, 1, 2), (Name, green, 2, 1), (Poosition, punter, 1, 2), (Poosition, placekicker, 2, 1), (Number, 4, 1, 1), (Birth Date, 21, 1, 3), (Birth Date, september, 2, 2), (Birth Date, 1949, 3, 1), (Debutyear, 1973, 1, 1), (Finalyear, 1978, 1, 1), (Draftyear, 1972, 1, 1), ..., (Birth Place, mason, 1, 3), (Birth Place, city, 2, 2), (Birth Place, iowa, 3, 1), (Article Title, dave, 1, 6), (Article Title, green, 2, 5), (Article Title, -lrb-, 3, 4), (Article Title, american, 4, 3), (Article Title, football, 5, 2), (Article Title, -rrb-, 6, 1).		
Rerence	dave green -lrb- born september 21 , 1949 in mason city , iowa -rrb- is a former punter and placekicker in the national football league .		
PTS	1st iteration	1st pointer predictor	dave dave september 21 1949 american city iowa american football punter placekicker football
		2nd pointer predictor	dave green september 21 1949 mason city iowa american football punter placekicker football
	2nd Iteration	dave alan green -lrb- born september 21 , 1949 in mason city , iowa -rrb- is a former american football punter and placekicker in the national football league .	
	3rd Iteration	dave green -lrb- born september 21 , 1949 in mason city , iowa -rrb- is a former american football punter and placekicker in the national football league .	

Table 7: Two examples from the WikiBio test set that illustrates how PTS generates a description for a source table by planning and then seaming. The incorrect and repetitive planning tokens are in red.

A corpus of metaphors as register markers

Markus Egg and Valia Kordoni
Humboldt-Universität zu Berlin
Unter den Linden 6, 10117 Berlin
{markus.egg, evangelia.kordoni}@hu-berlin.de

Abstract

The paper presents our work on corpus annotation for metaphor in German. Metaphors denote entities that are similar to their literal referent, e.g., when *Licht* ‘light’ is used in the sense of ‘hope’. We are interested in the relation between metaphor and register, hence, the corpus includes material from different registers.

We focussed on metaphors that can serve as register markers and can also be reliably identified for annotation. Our results show huge differences between registers in metaphor usage, which we interpret in terms of specific properties of the registers.

1 Introduction

This paper presents ongoing work on annotating a German corpus for metaphor. We are interested in metaphors as register markers, therefore, the corpus includes material from a number of different registers. We annotate all the metaphors in the corpus but nevertheless put emphasis on a subgroup of metaphors which we believe can function as register markers.

The paper is structured as follows. After outlining the underlying theoretical concepts of metaphor and register and reviewing previous work, we introduce the corpus. Then we present the annotation results, which show huge differences in metaphor usage between the different registers in the corpus. These differences are then correlated with specific properties of registers.

2 Theoretical background

In this section, we introduce the two phenomena of metaphor and register, and the way in which they are related.

2.1 Metaphor

Metaphors involve a semantic shift of an expression in context. They refer to an entity that is similar

to the referent of the literal interpretation of the metaphor. Theories reconstruct this similarity in different ways (for an overview see e.g. Ritchie, 2013). E.g., *vorbeirasen* ‘rush by’ in the temporal sense in (1) is metaphorical and shares with the literal, spatial interpretation the notion of a very fast development:

- (1) das letzte Jahr ist nur so vorbei gerast
‘the last year has rushed by’

Metaphors can be assigned a degree of conventionalisation, from innovative to fully conventionalised. We distinguish conventionalised and non-conventionalised metaphors (see Section 3 for details), e.g., the metaphor in (1) is non-conventionalised.

A small number of metaphors like *Blumen* ‘flowers’ in (2) is signalled openly by ‘metaphor flags’, among them *wie* ‘like’ or *praktisch* ‘in effect’, but most metaphors are not.

- (2) Wir sind wie Blumen praktisch, geerdet.
‘In effect, we are like flowers, earthed.’

In ‘extended metaphor’ (or ‘metaphor chains’), several metaphors in a discourse are based on the same kind of similarity (Reijnierse et al., 2020). E.g., once the word *Licht* ‘light’ is introduced as a metaphor for hope, other words related to light like *anzünden* ‘enkindle’ or *Kerze* ‘candle’ can emerge as metaphors for hope-related phenomena, too (as ‘introduce hope’ and ‘source of hope’, respectively).

Finally, ‘potential metaphor’ combines tokens of an expression with basic and metaphorical senses in the same discourse. E.g., in one of our texts the term *dunkel* ‘dark’ is used in its basic sense ‘without physical light’ before it is used metaphorically in the sense of ‘bad’. Potential metaphors typically participate in extended metaphor structures.

- (3) in dem Dunkel, in dem Wurzelbereich bei

dem Weizen
'in the dark zone, in the rhizosphere of the
wheat'

- (4) die dunkle Erde elterlicher Übermüdung
'the dark soil of parental fatigue'

2.2 Register

Register refers to the influence of situational and functional context on intra-individual linguistic variation (Biber and Conrad, 2009). Systemic-Functional Linguistics (SFL) decomposes register into field, tenor, and mode (Halliday and Hasan, 1985). 'Field' refers to the nature of a linguistic interaction, including its subject matter and its purpose. 'Tenor' targets the participants, in particular, their statuses and social relationships. 'Mode' is about the role of language in the interaction, e.g., whether it is oral or literal, or a monologue or a dialogue.

When metaphors are alternatives to reference via literal expressions, they are optional ways of referring to an entity. This allows intra-individual variation in establishing reference to be influenced by and to influence the situational and functional embedding of a discourse, viz., register. Thus, metaphors can contribute to establishing a specific register or indicate compliance with it.

This relation of metaphors and registers is due to the fact that the function of a metaphor depends on the discourse it is part of (Goatly, 2011). For instance, the function of metaphors can be influenced by the relations between the interlocutors, in that peers strive to build and maintain rapport, whereas experts want to offer explanations to non-experts. Such differences can result in different realisations of the metaphors. For example, Deignan et al. (2013) report that metaphors in the form of a simile ('A is like B') are more likely in expert-non-expert communication than in exchanges between peers.

2.3 Which metaphors for register?

The perspective on metaphor as register marker (or as a marker for other phenomena) raises issues with the state of the art in metaphor annotation, as it was established by Steen et al. (2010) and introduced into computational approaches to metaphor by Shutova and Teufel (2010) and Shutova et al. (2013): In these approaches, all metaphorical expressions are annotated, irrespective of their degree of conventionalisation.

To be able to function as a marker for register, however, metaphors must be free choices in the linguistic system, whose optional use can then be reused to mark a specific register. In contrast, any metaphor whose use is necessitated by the language system cannot be employed for the purpose of register.

For example, in the description of temporal constellations it is often not possible not to use highly conventionalised spatial metaphors, e.g., to express the fact that one time span is located *before* or *inside* another one. I.e., these interpretations of prepositions belong to the lexicon as parts of polysemous sense structures. Since they are not created by a productive metaphorical interpretation and are obligatory irrespective of register, they cannot function as metaphorical register markers.

Steen (2015) comes to similar conclusions about highly conventionalised metaphors and focuses on 'deliberate' metaphors, i.e., those that are intended to be recognised as such by the recipient. We believe that this is the group of metaphors that is also relevant for the relation between metaphor and register.

However, deliberate metaphor is hard to define in formal terms (see e.g. Krennmayr, 2011 and Reijnierse et al., 2018), which raises doubts as to whether it can be annotated with sufficient accuracy. Therefore we based our conclusions predominantly on deliberate metaphors that are recognisable with high accuracy in our corpus, viz., those with a metaphor flag, and non-conventional, extended, and potential metaphor.

3 Previous work

The interdependence between metaphor and register has been investigated for specific registers, e.g., academic discourse (Littlemore, 2001; Herrmann, 2015; Beger, 2015), fiction (Dorst, 2015), newspapers (Krennmayr, 2011) or educational discourse (Cameron, 2003). Functions of metaphors were correlated with SFL features of metaphors (Goatly, 2011; Steen et al., 2010). E.g., the latter claim that metaphor is used in informational registers (like news, fiction, or academic discourse) to express content to a much larger extent than in conversation. Berber Sardinha (2015) investigates the influence of metaphor-related features on register variation.

The group of Gerald Steen created and annotated the VU Amsterdam Metaphor Corpus (187,000

subcorpus	hierarchical/equal	distant/close	oral/literal	dialogue/monologue
speeches	E	D	L	M
sermons	H	C	L	M
commentaries	H	D	L	M
light fiction	E	C	L	M
debates	E	D	O	D

Table 1: SFL register properties of the subcorpora

words from the British National Corpus) with the four registers academic discourse, newspaper texts, fiction, and conversations (Steen et al., 2010).

Shutova and Teufel (2010) and Shutova et al. (2013) annotated a corpus of 13,700 words according to whether the words were used metaphorically or literally. They report different frequencies of metaphors for specific registers, in particular, a very low frequency of metaphor in spoken language. Bizzoni and Lappin (2018) compiled a corpus of 200 sets of metaphorical sentences and potential paraphrases (rated for their aptness). Zayed et al. (2020) created a corpus of 1,500 metaphorical verbs with a direct object.

Steen et al. (2010) developed detailed guidelines for the annotation of their corpus (later adapted to German in Herrmann et al., 2019). They define the context-based sense of an expression as a metaphor if it differs from another, more ‘basic’ sense of the expression (e.g., one which is more concrete or related to bodily action). These senses must be similar but not subsumable under a common hypernym, like in the case of the contextual temporal sense of *vorbeirasen* ‘rush by’ in (1). Senses are defined by suitable dictionaries; if both senses appear in the dictionaries, the metaphor counts as conventionalised, if only the basic sense does, it is regarded as non-conventional.

4 Our approach

4.1 The corpus

To investigate the relation of metaphor and register, we have compiled a corpus that integrates a wide range of register variation. Its five parts (of eventually 30,000 words each) are parliament speeches from the German Parlamentsreden-Korpus (Blaette, 2017), news commentaries (the Potsdam Commentary Corpus; Stede, 2004), sermons, light fiction (written by amateurs for their peers), and debates from competitions of the organisation ‘Jugend debattiert’ (Kemmann, 2013).

Table 1 shows the distribution of SFL register properties in the corpus. We vary two dimensions of *tenor*, viz., hierarchy vs. equality and distance vs. closeness, and the two *mode* dimensions of dialogue vs. monologue and of spoken vs. written register. Following Koch and Oesterreicher’s (1994) distinction of conceptual literality vs. orality, speeches and sermons are classified as literal (they are prepared and fixed in advance), despite their oral presentation.

subcorpus	reference	persuasion
speeches	+	o
sermons	o	+
commentaries	+	+
light fiction	-	-
debates	o	+

Table 2: Biber dimension properties of the subcorpora

The subcorpora also represent the variation we expected along two important Biber (2009) dimensions (Table 2). For ‘situation-dependent vs. elaborated reference’ (how dependent is reference on the situational context), we expect that commentaries and speeches relate to concrete extralinguistic situations and individuals, whereas debates and sermons are more abstract deliberations, and fiction is highly detached from reality. Thus, the anticipated level of situation dependence for reference is low for fiction, medium for debates and sermons, and high for commentaries and parliamentary speeches. For ‘overt expression of persuasion’, the expected level is high for debates, sermons, and commentaries, moderate for speeches (whose influence on actual decision making in politics is usually quite low), and low for fiction.

4.2 The annotation

For the annotation, we use the INCEption tool (Klie et al., 2018). Metaphors are annotated independently by two annotators. Inter-annotator agreement for the metaphor classification according to

subcorpus	metaphor flags**	conventionalised metaphor***	non-conventionalised metaphor*	extended metaphor***	potential metaphor***
speeches	0%	15.13%	.12%	.01%	0%
sermons	.02%	10.14%	.24%	.29%	.27%
commentaries	.05%	11.40%	.26%	.12%	.01%
light fiction	.04%	4.06%	.14%	.04%	0%
debates	0%	10.38%	.15%	.09%	0%

* = significant at $p < .05$; ** = significant at $p < .01$; *** = significant at $p < .0001$

Table 3: Metaphor counts for the subcorpora

subcorpus	metaphor flags	conventionalised metaphor***	non-conventionalised metaphor*	extended metaphor***
highly persuasive	.03%	10.88%	.23%	15%
medium or not persuasive	.02%	9.91%	.14%	.02%

Table 4: Metaphor counts for highly persuasive subcorpora

Krippendorff’s (2011) alpha emerged as .89. The annotation includes a layer of syntactic structure, derived by the Stanza package (Qi et al., 2020), to allow the identification of syntactic constellations for analyses of their metaphorical potential in future work.

To distinguish degrees of conventionalisation of the metaphors, we also fell back on suitable lexical resources, in our case, the *Duden* dictionary and the *Digitales Wörterbuch der deutschen Sprache*¹: When the context-based sense of the expression qualifies as metaphorical according to definition of Steen et al. (2010) (see Section 3), we check if it is listed in at least one of the lexical resources along with the basic sense of the expression. If yes, the metaphor is classified as conventionalised, otherwise, we assume that it is non-conventionalised, like (1).

We created guidelines for the annotation, starting out from the guidelines of Steen et al. (2010) and Herrmann et al. (2019). The corpus and the guidelines will be made available to the research community after their finalisation. See Egg and Kordoni (2022) for a more detailed description of the guidelines.

5 Results

The results of our annotation are summarised in Table 3 (percentages are calculated for word tokens²),

¹www.duden.de and www.dwds.de

²Extended and potential metaphors as a whole are counted only once. The participating metaphors are then counted separately as conventionalised or non-conventionalised metaphors.

showing clear differences in metaphor usage between the different registers. First, the level of conventionalised metaphors is high for speeches, medium for sermons, commentaries, and debates, and low for fiction. Also, metaphor flags are extremely rare in general, which parallels the results of Steen et al. (2010).

Potential metaphor is restricted almost exclusively to sermons. As soon as we omit sermons from consideration in the evaluation of our corpus, potential metaphor does not exhibit a correlation to register anymore ($p = .33$). Consequently, we omit it from further investigations into interdependencies between metaphor types and register properties.

Non-conventionalised and extended metaphors pattern similarly, occurring mostly in sermons and commentaries. We argue that this is due to the fact that these registers are highly persuasive. This correlation is less visible in the debates, which we put down to the time pressure of oral discourse, a conflicting factor impeding the creation of these types of metaphor. Table 4 summarises the counts of highly persuasive against the other registers and shows that the correlations are significant for these two types of metaphor.

Next, we investigated a potential interdependence between metaphoricality and the distinction in oral and literal discourse (summarised in Table 5). Our results first show that oral and literal discourse do not differ significantly for conventionalised and non-conventionalised metaphor. What is more, our oral register did not exhibit a significantly lower

subcorpus	metaphor flags*	conventionalised metaphor	non-conventionalised metaphor	extended metaphor
literal	.033%	10.62%	.21%	.12%
oral	.000%	10.38%	.17%	.09%

Table 5: Metaphor counts for literal and oral subcorpora

subcorpus	metaphor flags*	conventionalised metaphor***	non-conventionalised metaphor**	extended metaphor***
hierarchical	.04%	11.01%	.25%	.17%
equal	.01%	10.07%	.15%	.03%

Table 6: A competing hypothesis: hierarchy

score for extended metaphor as well, even though this kind of metaphor is non-local in that it is based on more than one expression in the discourse, as two or more expressions have to share the same kind of metaphorical similarity. In contrast, conventionalised and non-conventionalised metaphor are local in that they are based on single expressions.

There is a difference between the extended metaphors in debates and in the other registers, however, which has to do with the fact that debates are dialogues that consist of comparatively short turns of different speakers: We found that many extended metaphors are the result of the collaboration of different speakers, in that one speaker introduces a metaphor with a specific kind of similarity and other speakers subsequently pick up this metaphor or use metaphors that exhibit the same kind of similarity.

To sum up, our results for the oral register of debates thus suggest that previous very low metaphoricality scores for oral discourse as in [Steen et al. \(2010\)](#) might not be related to orality in general but to the conversational nature of their data, which calls for further investigation of differences within oral registers.

As for individual registers, our data first suggest a mixed pattern for fiction, like in [Reijnierse et al. \(2019\)](#) in that it is low on conventionalised metaphors but occupies a middle position w.r.t. non-conventionalised and extended metaphor. At the same time, the register in the corpus that conveys the highest degree of register marking are sermons: they exhibit a high degree of non-conventional metaphors, also, extended and potential metaphor emerge as clear register markers for sermons.

6 Conclusion

We presented current work on a German corpus with different registers, which is annotated for metaphors. Future work will use the corpus to investigate the metaphoric potential of specific syntactic constellations (like verb-object and adjective-noun) and include metonymy as another register-sensitive phenomenon ([Deignan et al., 2013](#); [Littlemore, 2015](#)).

Also, our results suggest that further research on oral registers is called for to delimit the actual interdependence between metaphor and the distinction between oral and literal registers. As a first step in this direction, we will include TEDx talks into our corpus, which complement the debates in that they are also oral but at the same time monologic and not persuasive. Other registers we plan to look into are sales talks and classroom interactions.

Limitations

In our study, we have argued for a correlation between forms of metaphor (non-conventional, extended, and potential) and persuasiveness. However, we are at this stage not yet in a position to rule out the competing hypothesis that there is a relation between metaphor and tenor in that metaphor correlates with a hierarchical difference between the interlocutors. Table 6 shows that this hypothesis would be significant for the corpus in its present form, which shows that the inclusion of further registers into the corpus is needed in order to distinguish between the competing hypotheses.

Ethics Statement

We took great care in the compilation of the corpus to include only material that can be published in

this form in order to be able to make the corpus available to the scientific community.

The debates consist of material produced by minors (16-18 years). In the corpus, we anonymised the names of the debaters throughout as ‘Speaker 1-4’. At the same time, we transcribed only debates that had already been made public on the Youtube canal of ‘Jugend debattiert’ (URL) in order to include only material whose publication had already been accepted by the respective speakers. At the same time, we contacted the spokesperson of ‘Jugend debattiert’ and got his consent on our activities as far as they include the debates.

The other material is either taken from already licenced corpora (for the parliament speeches and the commentaries) or has an appropriate CC license. Still, we contacted the authors to inform them about our project and to confirm their willingness to have their material included in our corpus.

Acknowledgment

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1412, 416591334.

References

- Anke Beger. 2015. Metaphors in psychology genres. Counseling vs. academic lectures. In Berenike Herrmann and Tony Berber Sardinha, editors, *Metaphor in specialist discourse*, pages 53–75. Benjamins, Amsterdam.
- Tony Berber Sardinha. 2015. Metapher and register variation. In Berenike Herrmann and Tony Berber Sardinha, editors, *Metaphor in specialist discourse*, pages 17–51. Benjamins, Amsterdam.
- Douglas Biber. 2009. Multi-dimensional approaches. In Anke Lüdeling and Merja Kytö, editors, *Corpus linguistics. An international handbook*, pages 822–855. Mouton de Gruyter, Berlin.
- Douglas Biber and Susan Conrad. 2009. *Register, genre, and style*. Cambridge University Press.
- Yuri Bizzoni and Shalom Lappin. 2018. Predicting human metaphor paraphrase judgments with deep neural networks. In *Proceedings of the Workshop on Figurative Language Processing*, page 45–55.
- Andreas Blaette. 2017. GermaParl. Corpus of plenary protocols of the German Bundestag. TEI files. Available at: <https://github.com/PolMine/GermaParlTEI>.
- Lynne Cameron. 2003. *Metaphor in educational discourse*. Continuum, London.
- Alice Deignan, Jeannette Littlemore, and Elena Semino. 2013. *Figurative language, genre and register*. Cambridge University Press, Cambridge.
- Aletta Dorst. 2015. More or different metaphors in fiction? A quantitative cross-register comparison. *Language and Literature*, 24:3–22.
- Markus Egg and Valia Kordoni. 2022. Metaphor annotation for German. In *Proceedings of LREC 2022*, pages 2556–2562.
- Andrew Goatly. 2011. *The language of metaphors*, 2nd edition. Routledge.
- Michael Halliday and Ruqaiya Hasan. 1985. *Language, context and text: A social semiotic perspective*. Deakin University Press, Victoria.
- Berenike Herrmann. 2015. High on metaphor, low on simile. An examination of metaphor type in subregisters of academic prose. In Berenike Herrmann and Tony Berber Sardinha, editors, *Metaphor in specialist discourse*, pages 163–190. Benjamins, Amsterdam.
- Berenike Herrmann, Karola Woll, and Aletta Dorst. 2019. Linguistic metaphor identification in German. In Susan Nacey, Aletta Dorst, Tina Krennmayr, and Gudrun Reijnierse, editors, *Metaphor identification in multiple languages. MIPVU around the world*, pages 113–135. Benjamins, Amsterdam.
- Ansgar Kemmann. 2013. Debatte. In Björn Rothstein and Claudia Müller, editors, *Kernbegriffe der Sprachdidaktik Deutsch. Ein Handbuch*, pages 41–43. Schneider, Hohengehren.
- Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The Inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of COLING 2018: system demonstrations*, pages 5–9.
- Peter Koch and Wulf Oesterreicher. 1994. Schriftlichkeit und Sprache. In Hartmut Günther and Otto Ludwig, editors, *Schrift und Schriftlichkeit. Writing and Its Use. Ein interdisziplinäres Handbuch internationaler Forschung. An Interdisciplinary Handbook of International Research*, volume 1, pages 587–604. de Gruyter, Berlin.
- Tina Krennmayr. 2011. *Metaphor in newspapers*. Ph.D. thesis, Vrije Universiteit Amsterdam.
- Klaus Krippendorff. 2011. Computing Krippendorff’s alpha-reliability. Technical report, University of Pennsylvania. Retrieved from https://repository.upenn.edu/asc_papers/43.
- Jeannette Littlemore. 2001. The use of metaphors in university lectures and the problems that it causes for overseas students. *Teaching in Higher Education*, 6:333–349.

- Jeannette Littlemore. 2015. *Metonymy: hidden shortcuts in language, thought and communication*. Cambridge University Press, Cambridge.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of ACL 2020: System Demonstrations*.
- Gudrun Reijniere, Christian Burgers, Tina Krennmayr, and Gerard Steen. 2018. DMIP: a method for identifying potentially deliberate metaphor in language use. *Corpus Pragmatics*, 2:129–147.
- Gudrun Reijniere, Christian Burgers, Tina Krennmayr, and Gerard Steen. 2019. Metaphor in communication: the distribution of potentially deliberate metaphor across register and word class. *Corpora*, 14:301–326.
- Gudrun Reijniere, Christian Burgers, Tina Krennmayr, and Gerard Steen. 2020. The role of co-text in the analysis of potentially deliberate metaphor. In Camilla Di Biase-Dyson and Markus Egg, editors, *Drawing attention to metaphor*, pages 15–38. Benjamins, Amsterdam.
- David Ritchie. 2013. *Metaphor*. Cambridge University Press, Cambridge.
- Ekaterina Shutova and Simone Teufel. 2010. Metaphor corpus annotated for source-target domain mappings. In *Proceedings of LREC 2010*, pages 3255–3261.
- Ekaterina Shutova, Simone Teufel, and Anna Korhonen. 2013. Statistical metaphor processing. *Computational Linguistics*, 39:301–353.
- Manfred Stede. 2004. The Potsdam Commentary Corpus. In *ACL 2004 Workshop on Discourse Annotation*, pages 96–102, Barcelona, Spain.
- Gerard Steen. 2015. Developing, testing and interpreting deliberate metaphor theory. *Journal of Pragmatics*, 90:67–72.
- Gerard Steen, Aletta Dorst, Berenike Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A method for linguistic metaphor identification: from MIP to MIPVU*. Benjamins, Amsterdam.
- Omnia Zayed, John McCrae Philip, and Paul Buitelaar. 2020. Figure me out: a gold standard dataset for metaphor interpretation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5810–5819, Marseille. European Language Resources Association.

Translate First Reorder Later: Leveraging Monotonicity in Semantic Parsing

Francesco Cazzaro* Davide Locatelli* Ariadna Quattoni

Universitat Politècnica de Catalunya
name.lastname@upc.edu

Xavier Carreras

dMetrics
xavier.carreras@dmetrics.com

Abstract

Prior work in semantic parsing has shown that conventional seq2seq models fail at compositional generalization tasks. This limitation led to a resurgence of methods that model alignments between sentences and their corresponding meaning representations, either implicitly through latent variables or explicitly by taking advantage of alignment annotations. We take the second direction and propose TPOL, a two-step approach that first translates input sentences monotonically and then reorders them to obtain the correct output. This is achieved with a modular framework comprising a *Translator* and a *Reorderer* component. We test our approach on two popular semantic parsing datasets. Our experiments show that by means of the monotonic translations, TPOL can learn reliable lexico-logical patterns from aligned data, significantly improving compositional generalization both over conventional seq2seq models, as well as over other approaches that exploit gold alignments. Our code is publicly available at <https://github.com/interact-erc/TPol.git>

1 Introduction

The goal of a semantic parser is to map natural language sentences (NLs) into meaning representations (MRs). Most current semantic parsers are based on deep sequence-to-sequence (seq2seq) approaches and presume that it is unnecessary to model token alignments between NLs and MRs because the attention mechanism can automatically learn the correspondences (Dong and Lapata, 2016; Jia and Liang, 2016). However, recent work has shown that such seq2seq models find compositional generalization challenging, i.e., they struggle to predict unseen structures made up of components observed at training (Lake and Baroni, 2018; Finegan-Dollak et al., 2018).

*Equal contribution

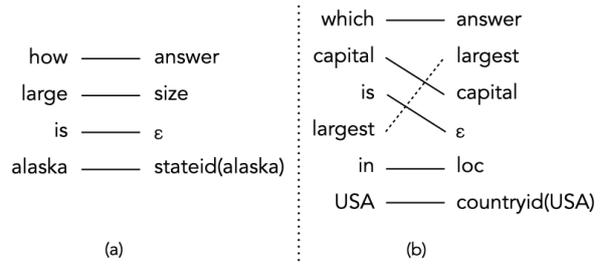


Figure 1: Examples from the GEOALIGNED dataset. (a) is a monotonic alignment, (b) is non-monotonic.

This limitation motivated the resurgence of approaches that model alignments between NL sentences and their corresponding MRs more similarly to classical grammar and translation-based parsers (Herzig and Berant, 2021). Alignments can be modeled either implicitly through latent variables (Wang et al., 2021), or explicitly by leveraging gold alignment annotations (Shi et al., 2020; Liu et al., 2021a). We take the second direction and exploit a recently released multilingual dataset for semantic parsing annotated with word alignments: GEOALIGNED (Locatelli and Quattoni, 2022), which augments the popular GEO benchmark (Zelle and Mooney, 1996).

Figure 1 shows some examples of the annotations provided. One key observation is that a significant percentage of the alignments are monotonic, i.e., they require no reordering of the target MR (Figure 1a), as opposed to non-monotonic alignments (Figure 1b). This suggests that learning reliable lexico-logical translation patterns from aligned data should be possible. If there are simple patterns, shouldn't an ideal model be able to exploit them?

With this in mind, we propose TPOL, a Two-step Parsing approach that leverages monotonic translations. TPOL introduces a modular framework with two components: a *Monotonic Translator* and a *Reorderer*. The Translator is trained from pairs of NLs and MRs, where the MRs have been permuted to be monotonically aligned. Hence,

the Translator’s output will be an MR whose order might not correspond to that of the gold truth. For this reason, the Reorderer is trained to restore the correct order of the original MR.

Our experiments on GEOALIGNED demonstrate that compared to a multilingual BART model (Liu et al., 2020), TPOL achieves similar performance on the random test split but significantly outperforms on the compositional split across all languages. For example, on the query split in English, mBART obtains 69.4% in exact-match accuracy and TPOL obtains 87.8%. This result also improves on the 74.6% obtained by SPANBASED (Herzig and Berant, 2021), another approach that leverages alignment annotations.

Because most semantic parsing datasets do not contain alignment information, we experiment with alignments generated automatically. On GEO, TPOL trained with automatic alignments still outperforms mBART, and in particular on the English query split it improves by almost 10 points. Furthermore, we show competitive results on the popular SCAN dataset (Lake and Baroni, 2018).

In summary, the main contributions of this paper are:

1. We propose TPOL, a modular two-step approach for semantic parsing which explicitly leverages monotonic alignments;
2. Our experiments show that TPOL improves compositional generalization without compromising overall performance;
3. We show that even without gold alignments TPOL can achieve competitive results.

2 Related Work

Recently, the semantic parsing community has raised the question of whether current models can generalize compositionally, along with an effort to test for it (Lake and Baroni, 2018; Finegan-Dollak et al., 2018; Kim and Linzen, 2020). The consensus is that conventional seq2seq models struggle to generalize compositionally (Loula et al., 2018; Keysers et al., 2020). Moreover, large pre-trained language models have been shown not to improve compositional generalization (Oren et al., 2020; Qiu et al., 2022b). This has prompted the community to realize that parsers should be designed intentionally with compositionality in mind (Lake, 2019; Gordon et al., 2020; Weißenhorn et al., 2022).

It has also been pointed out that compositional architectures are often designed for synthetic datasets and that compositionality on non-synthetic data is under-tested (Shaw et al., 2021).

Data augmentation techniques have been proposed to improve compositional generalization (Andreas, 2020; Yang et al., 2022; Qiu et al., 2022a). Another strategy is to exploit some level of word alignments. In general, there has been a resurgent interest in alignments as it has been shown that they can be beneficial to neural models (Shi et al., 2020). It has also been conjectured that the lack of alignment information might hamper progress in semantic parsing (Zhang et al., 2019). As a result, the field has seen some annotation efforts in this regard (Shi et al., 2020; Herzig and Berant, 2021; Locatelli and Quattoni, 2022).

Alignments have been modeled implicitly: Wang et al. (2021) treat alignments as discrete structured latent variables within a neural seq2seq model, employing a framework that first reorders the NL and then decodes the MR. Explicit use of alignment information has also been explored: Herzig and Berant (2021) use alignments and predict a span tree over the NL. Sun et al. (2022) recently proposed an approach to data augmentation via sub-tree substitutions. In text-to-SQL, attention-based models that try to capture alignments have been proposed (Lei et al., 2020; Liu et al., 2021b), as well as attempts that try to leverage them directly (Sun et al., 2022).

Our two-step approach resembles statistical machine translation, which decomposes the translation task into lexical translation and reordering (Chang et al., 2022). Machine translation techniques have previously been applied to semantic parsing. The first attempt was by Wong and Mooney (2006), who argued that a parsing model can be viewed as a syntax-based translation model and used a statistical word alignment algorithm. Later a machine translation approach was used on the GEO dataset, obtaining what was at the time state-of-the-art results (Andreas et al., 2013). More recently, Agarwal et al. (2020) employed machine translation to aid semantic parsing.

3 Preliminaries: Word Alignments

This section briefly explains word alignments, showing the difference between monotonic and non-monotonic alignments, and illustrates the notion of monotonic translations.

Assume that we have a pair of sequences $\mathbf{x} = x_1, \dots, x_n$ and $\mathbf{y} = y_1, \dots, y_m$, where n and m are the respective sequence lengths. A bi-sequence is defined as the tuple (\mathbf{x}, \mathbf{y}) . In our application, \mathbf{x} is a NL sentence, and \mathbf{y} is its corresponding MR. For example:

$\mathbf{x} =$ which city has the highest population density?
 $\mathbf{y} =$ answer(largest(density(city(all))))

A word alignment is a set of bi-symbols \mathcal{A} , where each bi-symbol defines an alignment from a token in the NL to a token in the MR. For instance, the bi-symbol (x_i, y_j) aligns token x_i to token y_j . In our example, the tokens "which" and "answer" could be paired by a bi-symbol (which, answer).

If a token x_i does not align to anything in \mathbf{y} , an ε is introduced in \mathbf{y} : the resulting bi-symbol (x_i, ε) corresponds to a deletion. In our example, the token "has" in the NL can be deleted with a bi-symbol (has, ε). Similarly, if a token y_j is not aligned to a token in \mathbf{x} , an ε is introduced in \mathbf{x} : (ε, y_j) is an insertion. In our example, the token "all" in the MR is inserted with bi-symbol $(\varepsilon, \text{all})$.

The bi-symbols in \mathcal{A} are all one-to-one. Hence, to map a single token to a phrase, i.e., to multiple tokens, it is necessary to choose a head token in the phrase, while the remaining tokens require insertion or deletion. In our example, the token "density" in the MR corresponds to "population density" in the NL, and, if "density" is chosen as the head token in the NL, "population" needs a deletion: the alignment will be given by the bi-symbols (population, ε) and (density, density).¹ Following this strategy, this notation can account for one-to-many and many-to-one alignments with deletion and insertion operations.

Figure 2a shows a possible bi-sequence word alignment for the aforementioned example. Each bi-symbol is conveniently represented by a horizontal line connecting the tokens it aligns.

Alignments can be monotonic or non-monotonic. An alignment is monotonic if it does not involve any *crossing*, i.e., a mapping that does not require reordering tokens. In our example, the alignment is non-monotonic because the bi-symbol (city,city) crosses over others. By permuting the MR, we can obtain a monotonic translation of the NL: Figure

¹Locatelli and Quattoni (2022) showed that annotators are consistent in the way they pick head-tokens, and reported high inter-annotator agreement scores on GEOALIGNED.

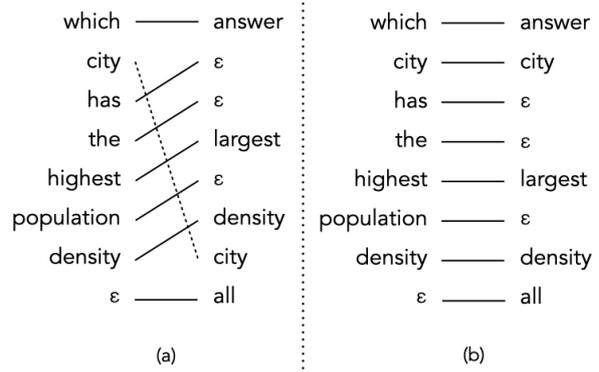


Figure 2: (a) A possible alignment for an NL-MR pair. (b) The corresponding monotonic translation. For simplicity, we removed the brackets and question mark.

2b shows such permutation. The next section illustrates how TPOL can leverage these translations.

4 Translate First Reorder Later

We propose TPOL, a two-step parsing approach with a modular framework made up of two components: a *Monotonic Translator* and a *Reorderer*. Figure 3 shows how our semantic parser takes an input sentence \mathbf{x} and predicts the corresponding MR \mathbf{y} . In the first step, \mathbf{x} is fed to the Translator, which outputs a monotonic translation \mathbf{z} . In other words, \mathbf{z} is the target MR that has been permuted so that it aligns monotonically to the input NL. Then, in a second step, \mathbf{z} is fed to the Reorderer, which is trained to place the MR tokens back into the correct order to produce the final prediction \mathbf{y} .

The main idea behind TPOL is decomposing the task into lexical translation and reordering, to learn more reliable translation patterns. We purport that modeling monotonic alignments eases the learning of novel pattern combinations of seen structures, improving compositional generalization.

An alternative approach would be to permute the NL inputs rather than the MRs monotonically. We do not follow this direction due to the observation that in semantic parsing, multiple NLs can map to the same MR. In other words, the NL domain is larger than that of the MRs, and thus we believe that learning to reorder the MRs is more feasible.

4.1 Monotonic Translator

The Monotonic Translator is responsible for making an initial prediction of the MR sequence, which will contain the correct tokens in monotonic order. To create the training bi-sequences, we use alignment information and permute the gold MR

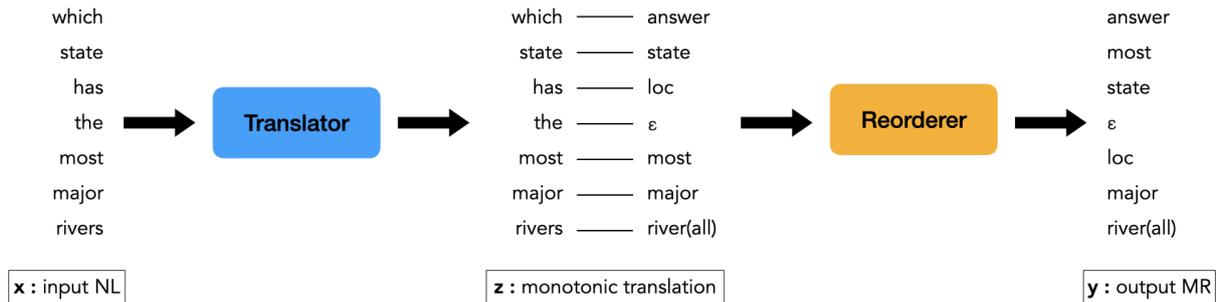


Figure 3: The TPOL parsing approach. An input sentence x is fed to the *Monotonic Translator* that predicts an intermediate monotonic MR z . This is in turn fed to the *Reorderer*, which outputs the final prediction y .

sequences to obtain a monotonic mapping with the NL. As a concrete example, consider the non-monotonic alignment in Figure 2a, and its monotonic translation in Figure 2b.

The translation task can be formulated in various ways. In our implementation, we work with two alternative approaches: a seq2seq Translator, and a tagger Translator. In the seq2seq formulation, x is fed into an encoder network, which produces a hidden vector. The hidden vector is fed to a decoder network which produces the output z , i.e., the monotonically aligned MR. This can be implemented, for example, with a BART model (Lewis et al., 2020), which uses a bidirectional encoder and a left-to-right decoder. In our experiments, we use the multilingual version of BART (Liu et al., 2020). In the tagging formulation, the Translator assigns an MR token to each token in x , obtaining the monotonic translation z by explicitly aligning in a token-by-token fashion. We implement this with a BERT model (Devlin et al., 2019) and we use its classification head as the tagger.

A crucial difference between the seq2seq and the tagger Translator is that the latter needs x and z to be the same length. The seq2seq Translator can learn to perform deletion operations from the raw NL, without needing epsilons in the input to perform insertions. By contrast, the tagger Translator needs insertions to be performed on x before predicting z . In general, NL sequences are significantly longer than the MR sequences, i.e., most epsilons are in the MR sequence. In other words, deletions are more frequent than insertions.

However, for some datasets, some alignments contain epsilons in the NL sequence: at prediction time, we will not know where insertions might occur, and thus we need a way to predict them. For this purpose, for every token followed by an epsilon in the train split, we add an epsilon after it

at test time. We saw that this strategy was sufficient in our experiments. Alternatively, this step could be done by a trained model or with a rule-based system similar to Ribeiro et al. (2018).

4.2 Reorderer

The Reorderer module is responsible for taking the monotonic predictions of the Translator and putting them back into the correct order to obtain the final prediction. This model is trained from pairs of MR sequences (z, y) where the input z is a monotonically permuted MR and the output y is the target MR in its correct order. These training pairs can be generated from the alignment annotations.

Similarly to the Translator module, the Reorderer can be implemented both as a seq2seq model and as a tagger. We use mBART in the seq2seq formulation and BERT as a tagger in our experiments. Note that we do not enforce the output to be a permutation of the input.

5 Experiments

5.1 Datasets

We test TPOL on two semantic parsing datasets, training with gold and automatically generated alignments in multiple languages, on standard IID partitions and the more challenging compositional ones.

5.1.1 GEOALIGNED

GEOALIGNED (Locatelli and Quattoni, 2022) augments the popular GEO semantic parsing benchmark (Zelle and Mooney, 1996) with token alignment annotations. The dataset contains questions about US geography and corresponding meaning representation using the FunQL formalism (Kate et al., 2005). In total, there are 880 examples, all annotated with token alignments. We evaluate on three partitions: question (?), query (Q) and length

(LEN). The question partition is a standard IID split where test and train are sampled from the same distribution. The query partition, introduced by [Finegan-Dollak et al. \(2018\)](#), is designed to be compositional by ensuring that the templates of the MRs in the test set are never seen during training. The length partition, introduced by [Herzig and Berant \(2021\)](#), assigns the longest sequences to the test.

The dataset comes in English, Italian and German: in this way we can test our approach across different languages. In our experiments, we do not anonymize constants: in other words, we keep the original NL and MR sequences which include names of cities, states, etc. We follow [Wang et al. \(2021\)](#) in removing brackets.

5.1.2 SCANSP

SCANSP ([Herzig and Berant, 2021](#)) is a set of navigational commands presented in natural language paired with action sequences. It is based on the SCAN dataset by [Lake and Baroni \(2018\)](#), which does not contain program MRs. [Herzig and Berant \(2021\)](#) translated the sequences into programs to obtain a semantic parsing version of the dataset. Besides the IID split, we test on the compositional partitions based on the "right" (RX) and "around right" (ARX) primitives from [Loula et al. \(2018\)](#). SCANSP has 20,910 commands distributed roughly as 12,000 train, 3,000 validation, and 4,000 test examples.

The SCANSP dataset does not come with alignments. Therefore we employ the IBM models ([Brown et al., 1993](#)) to generate them automatically using the GIZA++ toolkit ([Och and Ney, 2003](#)). We also do this for GEO to compare the performance of TPOL when trained with gold and automatic alignment annotations.

5.2 Models for comparisons

We compare with competitive baselines and state-of-the-art models that do not leverage alignments and competing models that do.

- **LSTM**: a standard seq2seq model with a bi-directional LSTM encoder and an LSTM decoder with attention ([Bahdanau et al., 2015](#)). We use pre-trained GloVe embeddings for the three languages: English ([Pennington et al., 2014](#)), Italian and German ([Ferreira et al., 2016](#)).

- **mBART** ([Liu et al., 2020](#)): a multilingual version of BART ([Lewis et al., 2020](#)), a pre-trained

Transformer-based seq2seq model that has been successfully applied to parsing ([Bevilacqua et al., 2021](#)).

- **mT5** ([Xue et al., 2021](#)): a multilingual version of T5 ([Raffel et al., 2020](#)), pre-trained on the mC4 dataset ([Xue et al., 2021](#)).

- **SPANBASED** ([Herzig and Berant, 2021](#)): a semantic parser that predicts a span tree over an input utterance trained with gold alignment trees. The authors provided annotations for the English version of GEO and SCANSP. For the other languages of GEO we train without gold alignments. We use their model without the lexicon, as that would be unfair with respect to the other models.

- **LEAR** ([Liu et al., 2021a](#)): a model that learns to recombine structures recursively by predicting a latent syntax tree and assigning semantic operations to non-terminal nodes. LEAR explicitly uses alignments using a phrase table.

- **REMOTO** ([Wang et al., 2021](#)): a model that first reorders the tokens in the NL and then predicts the MR. REMOTO is not trained with gold alignments.²

5.3 Evaluation metric

We follow the standard practice of using exact-match accuracy for evaluation: the predicted MR is correct only if it is the same as the gold.

5.4 Main Results

We report the results of our experiments in Table 1. For TPOL, the choice of the modules' architecture is validated on the development set, and we report a performance breakdown in Section 7.

We first consider the results of TPOL trained with gold alignments. On the GEO dataset, the LSTM and mT5 achieve the lowest performance in all the partitions. Looking at the question partition (?), the models show similar performance to mBART and SPANBASED, which is not surprising as the test split does not require compositional generalization. On the query partition (Q), designed to test for compositional generalization, TPOL shows significant improvements over all the other models across all languages. In English it obtains 87.3% outperforming mBART (69.4%), SPANBASED (74.6%) and LEAR (84.1%). In Italian and German, it obtains 81.6% and 69.4% respectively, while mBART 67.4% and 56.3%. On

²For LEAR and REMOTO, we report results directly from the respective papers, noting that in their setting constants, such as names of states, cities, and so on, are anonymized.

Model	GEO											
	EN			IT			DE			SCANSP		
	?	Q	LEN	?	Q	LEN	?	Q	LEN	IID	RX	ARX
LSTM	52.9	24.9	5.0	46.4	18.1	4.3	42.9	17.6	3.2	100	24.4	1.1
mT5	80.0	60.0	19.3	73.2	44.9	20.4	68.2	47.8	18.6	100	41.2	99.8
mBART	87.5	69.4	27.5	86.6	76.4	23.3	75.5	56.3	18.2	100	99.4	100
LEAR	-	84.1	-	-	-	-	-	-	-	-	-	-
SPANBASED	87.7	74.6	55.0	-	-	-	-	-	-	100	100	100
- gold	66.4	51.8	24.6	49.6	37.3	10.4	40.4	21.4	5.0	100	100	100
REMOTO	75.2	43.2	23.2	-	-	-	55.6	22.3	16.6	100	-	-
TPOL	87.3	87.8	41.9	85.9	81.6	31.3	73.3	69.4	22.9	-	-	-
- gold	85.8	79.0	35.6	83.6	75.1	20.2	73.8	60.7	17.5	100	99.4	100

Table 1: Exact-match accuracy of all models on GEOALIGNED and SCANSP datasets. ? stands for question, Q for query and LEN for the length partition. RX stands for right and ARX for around right partitions. LEAR and REMOTO both anonymize constants in GEO, and the results are taken directly from the respective papers.

the length partition (LEN), TPOL does better than all the baselines across all languages, except for SPANBASED, which fares better on LEN(English). This is only the case, however, when gold alignments are provided.

Looking at the results obtained without gold alignments, TPOL shows considerable improvements over REMOTO and SPANBASED. In particular, it improves on the English query partition obtaining 79% against 43.2% and 51.8%, respectively. Furthermore, the accuracy does not drop significantly compared to TPOL trained with gold alignments. We tested using automatic alignments from IBM models 3, 4, and 5 and picked the best out of the three. In general, all lead TPOL to achieve similar performance.

Finally, looking at SCANSP, as expected, the models designed for compositional generalization achieve perfect performance on the dataset. What is surprising is that also mBART can do so, contrary to other deep models. With some internal testing, we have seen that this is not the case for English BART, as opposed to the multilingual version. We hypothesize that model size and pre-training might be a factor of success for mBART.

6 Error Analysis

Table 3 shows a breakdown of performance of TPOL on the English version of GEOALIGNED. The results indicate the exact-match accuracy achieved by the two modules: we can check whether the model struggles more with the trans-

lation or reordering step. To analyze the Translator’s performance, we regard the monotonically aligned MRs as the gold truth. For the Reorderer, we provide it with the monotonically aligned MRs in input. In other words, the evaluation of the Reorderer assumes that the Translator makes a correct prediction.

The performance of the two modules is fairly similar, and, by comparing these results with Table 1, we see that the accuracy of each component is not much higher than the overall accuracy, suggesting that neither component is hampering performance more than the other. The only exception seems to be in the length partition, where the Reorderer does considerably better than the Translator.

Table 2 shows the breakdown of the performance over monotonically and non-monotonically aligned MRs. We can observe that, compared to SPANBASED, TPOL generally presents smaller drops in performance over the non-monotonic sequences. For instance, in the question partition, SPANBASED drops from 94.7% over the monotonic examples to 73.7% over the non-monotonic, while TPOL drops from 89.9% to 81.4%. When we look at the query partition, we see that for both of these models, the drop is much higher than the one on the question partition: SPANBASED goes from 89.6% to 39.8%, while TPOL drops from 93.7% to 69.9%. In other words, both models struggle more with non-monotonic examples when compositional generalization is required. TPOL, however, still performs significantly better.

Model	GEO EN					
	?		Q		LEN	
	MN	NMN	MN	NMN	MN	NMN
mBART	90.4	81.0	67.1	76.5	29.2	25.1
SPANBASED	94.7	73.7	89.6	39.8	68.1	35.9
TPOL	89.9	81.4	93.7	69.9	55.5	23.2

Table 2: Performance breakdown of TPOL over monotonic (MN) and non-monotonic (NMN) sequences in GEOALIGNED English. In Appendix B we report the number of MN and NMN examples.

Module	GEO EN		
	?	Q	LEN
Translator	86.1	86.2	42.5
Reorderer	87.6	87.3	57.1

Table 3: Performance breakdown of TPOL modules.

Surprisingly, mBART shows the opposite trend on the query partition, with the non-monotonic accuracy being higher than the monotonic one. By contrast, most of TPOL’s improved performance comes from better modeling of the monotonic sequences (67.1% \rightarrow 93.7%). TPOL’s results suggest that regular patterns in the non-monotonic sequences can be learned. Its generalization problems can be attributed to the difficulty of learning the more challenging non-regular patterns in a small dataset. On the other hand, mBART appears to have the capacity to model these challenging reorderings better. Still, this comes at the cost of failing on the regular monotonic ones, leading to a lower performance overall.

Interestingly, the SPANBASED approach shows a similar improvement on monotonic sequences compared to mBART (67.1% \rightarrow 89.6%). This suggests that exploiting lexico-logical alignments allows models to capture the simpler patterns that mBART fails to learn. Most of TPOL’s gains over SPANBASED come from better modeling the non-monotonic examples in the query partition (39.8% \rightarrow 69.9%). This shows that the two-step approach offers the best of both worlds: it can capture the simple monotonic patterns while maintaining reasonable performance over the more complex alignments on which SPANBASED fails.

Figure 4 shows the average drop in performance of the different models trained with automatic IBM alignments compared to the same model trained

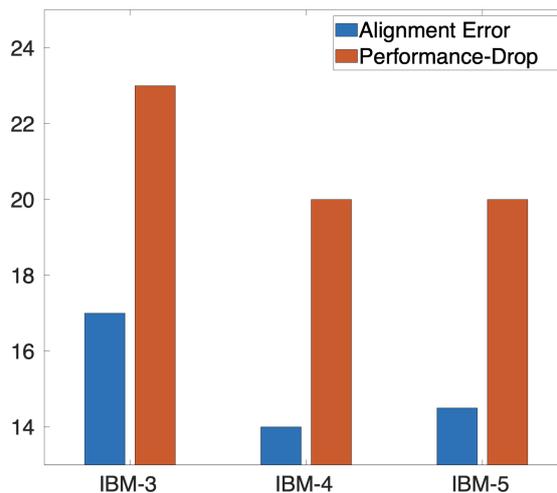


Figure 4: Average error of IBM alignment models over all partitions and languages on GEOALIGNED. We also plot the average drop in performance for each TPOL model trained with IBM alignments with respect to the corresponding one trained with gold alignments.

with gold alignments. We also report the corresponding alignment error, calculated as the percentage of bi-symbols that differ from the gold annotations from GEOALIGNED. We observe that, in general, higher alignment error is associated with a higher drop in performance. This validates the importance of the alignment information and points to improving the unsupervised alignment algorithm as a natural line of future work. We believe that one possible reason for the drop in performance when training with IBM alignments might be because the GEO dataset is of relatively small size, and the IBM models might have difficulty learning good alignments. That would explain why in contrast to GEO, the performance over SCAN is not affected by automatic alignments since SCAN is a much larger dataset.

Model	GEO								
	EN			IT			DE		
	?	Q	LEN	?	Q	LEN	?	Q	LEN
BERT2BERT	74.9	84.6	41.9	74.0	74.2	31.3	62.5	67.5	22.9
BERT2mBART	82.1	87.8	36.4	76.9	81.6	27.5	63.2	68.9	20.0
BERT2mBART SILVER	82.5	87.0	34.6	76.6	81.1	27.0	65.4	69.4	19.9
mBART2BERT	74.5	70.7	24.5	77.7	76.1	24.2	65.7	56.1	17.5
mBART2mBART	87.3	72.2	25.2	85.7	76.3	23.1	73.3	59.2	16.3
mBART2mBART SILVER	86.4	71.5	24.9	85.9	75.8	20.0	73.2	59.5	17.0

Table 4: Performance breakdown of TPOL for different module architectures on GEOALIGNED.

7 Architecture study

We emphasize here that our approach is an abstract, high-level methodology and does not place any constraint on the underlying architectures of the two components. We believe that different architectures, particularly specialized ones for each module, could be beneficial for parsing performance. We encourage further work to be carried out in this regard. To this purpose, we present some architectural studies using BERT (Devlin et al., 2019) and mBART (Liu et al., 2020) as components. We employ them as both Translator and Reorderer, examining all possible combinations. As explained in Section 4.1, mBART is used as a standard seq2seq model, and BERT is employed with a classification head to function as a tagger for every input token.

Additionally, when mBART is used as a Reorderer, we introduce a silver training setting. In the normal setting, the Reorderer is trained by taking the gold alignment annotations and outputting the meaning representation. In the silver setting, we use the predictions of the Translator model as training input. By doing so, the Reorderer trains on inputs that mimic more closely what it will actually receive at test time: this is done straightforwardly for a seq2seq model like mBART, while for our BERT tagger, every token in input needs to be aligned with a token in output, and when the input is corrupt it is not possible to achieve the same training technique.

In Table 4, we present the results for our different architectural components. To distinguish among the different model combinations, we use a [Translator]2[Reorderer] naming convention, meaning that mBART2BERT uses mBART as the Translator and BERT as the Reorderer. We observe that our two-step approach seems to be robust overall.

We can discern trends in different architecture combinations, which can be helpful when choosing an architecture for a specific task. One important observation is that the architectures that use BERT as a Translator are consistently better than the ones using mBART over the compositional partitions. We hypothesize that the BERT Translator can achieve higher compositional generalization because it can better leverage alignment information to predict unseen combinations of observed training patterns. We believe this is because a tagger’s predictions can be more naturally broken into parts that can be recombined. In contrast, encoder-decoder architectures fare better on the IID partition but struggle to generalize to unseen patterns. One possible reason is that these models have a harder time inducing local patterns that can be recombined since they encode and decode complete structures all at once.

8 Conclusion

Seq2seq models have become increasingly popular in semantic parsing. However, they are limited in their abilities to generalize to unobserved structures. Here, we proposed TPOL: a two-step parsing approach that leverages alignment annotations with a modular framework composed of a Translator and a Reorderer.

We showed that TPOL improves compositional generalization over conventional seq2seq models and over competing models that also leverage alignment information. Our results also showed that our approach is robust when trained with automatically generated alignments, demonstrating competitive results on two semantic parsing datasets.

We have experimented with two possibilities for the Translator and Reorderer, but we believe

that different architectural components could further improve performance. The divide-and-conquer strategy of breaking the problem into two simpler sub-tasks is designed to enable further component specialization.

9 Limitations

Regarding the limitations of our approach, our experiments used the standard FunQL meaning representation. Transitioning to a different meaning representation might need some adaptation of the framework. In particular, the alignments between NL and MRs for other meaning representations might require more insertion and deletion operations. We might also expect that other MRs might require more reordering.

A second limitation of our work is training with gold alignments. We partially address this by training TPOL with automatic alignments obtained with the IBM models. Still, we believe there is room for more work to be done so that this approach can be more easily scaled to datasets that do not have alignment annotations.

Despite TPOL's partial improvements on the length test splits, this type of partition remains challenging for all models. Here, models are required to generate predictions of greater length than what they have seen during training. This requires complex compositional productivity skills, i.e., recombining known constituents into larger structures. Further work is needed to address the limitation of the current state-of-the-art on compositional productivity benchmarks.

Acknowledgements

We would like to thank the EACL area chair and the anonymous reviewers for their feedback, as well as the other members of the INTERACT group at the Universitat Politècnica de Catalunya for fruitful discussions on an earlier draft of this work. This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No.853459). This paper reflects the authors' view only, and the funding agencies are not responsible for any use that may be made of the information it contains. The authors gratefully acknowledge the computer resources at Artemisa, funded by the European Union ERDF and Comunitat Valenciana, as well as the technical support provided by the Instituto de Física Corpuscular, IFIC (CSIC-UV).

References

- Oshin Agarwal, Mihir Kale, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. [Machine translation aided bilingual data-to-text generation and semantic parsing](#). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 125–130, Dublin, Ireland (Virtual). Association for Computational Linguistics.
- Jacob Andreas. 2020. [Good-enough compositional data augmentation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.
- Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. [Semantic parsing as machine translation](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 47–52, Sofia, Bulgaria. Association for Computational Linguistics.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. [Neural machine translation by jointly learning to align and translate](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. [One SPRING to Rule Them Both: Symmetric AMR Semantic Parsing and Generation without a Complex Pipeline](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. [The mathematics of statistical machine translation: Parameter estimation](#). *Computational Linguistics*, 19(2):263–311.
- Chih-Chiang Chang, Shun-Po Chuang, and Hung-yi Lee. 2022. [Anticipation-free training for simultaneous machine translation](#). In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 43–61, Dublin, Ireland (in-person and online). Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Li Dong and Mirella Lapata. 2016. [Language to logical form with neural attention](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

- 33–43, Berlin, Germany. Association for Computational Linguistics.
- Daniel C. Ferreira, André F. T. Martins, and Mariana S. C. Almeida. 2016. [Jointly learning to embed and predict with multiple languages](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2019–2028, Berlin, Germany. Association for Computational Linguistics.
- Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. [Improving text-to-SQL evaluation methodology](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.
- Jonathan Gordon, David Lopez-Paz, Marco Baroni, and Diane Bouchacourt. 2020. Permutation equivariant models for compositional generalization in language. In *ICLR*.
- Jonathan Herzig and Jonathan Berant. 2021. [Span-based semantic parsing for compositional generalization](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 908–921, Online. Association for Computational Linguistics.
- Robin Jia and Percy Liang. 2016. [Data recombination for neural semantic parsing](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany. Association for Computational Linguistics.
- Rohit J. Kate, Yuk Wah Wong, and Raymond J. Mooney. 2005. Learning to transform natural to formal languages. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3, AAAI’05*, page 1062–1068. AAAI Press.
- Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. [Measuring compositional generalization: A comprehensive method on realistic data](#). In *International Conference on Learning Representations*.
- Najoung Kim and Tal Linzen. 2020. [COGS: A compositional generalization challenge based on semantic interpretation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.
- Brenden Lake and Marco Baroni. 2018. [Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR.
- Brenden M Lake. 2019. [Compositional generalization through meta sequence-to-sequence learning](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Wenqiang Lei, Weixin Wang, Zhixin Ma, Tian Gan, Wei Lu, Min-Yen Kan, and Tat-Seng Chua. 2020. [Re-examining the role of schema linking in text-to-SQL](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6943–6954, Online. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chenyao Liu, Shengnan An, Zeqi Lin, Qian Liu, Bei Chen, Jian-Guang Lou, Lijie Wen, Nanning Zheng, and Dongmei Zhang. 2021a. [Learning algebraic recombination for compositional generalization](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1129–1144, Online. Association for Computational Linguistics.
- Qian Liu, Dejian Yang, Jiahui Zhang, Jiaqi Guo, Bin Zhou, and Jian-Guang Lou. 2021b. [Awakening latent grounding from pretrained language models for semantic parsing](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1174–1189, Online. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Davide Locatelli and Ariadna Quattoni. 2022. [Measuring alignment bias in neural seq2seq semantic parsers](#). In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 200–207, Seattle, Washington. Association for Computational Linguistics.
- João Loula, Marco Baroni, and Brenden Lake. 2018. [Rearranging the familiar: Testing compositional generalization in recurrent networks](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114, Brussels, Belgium. Association for Computational Linguistics.

- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Inbar Oren, Jonathan Herzig, Nitish Gupta, Matt Gardner, and Jonathan Berant. 2020. [Improving compositional generalization in semantic parsing](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2482–2495, Online. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [Glove: Global vectors for word representation](#). In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. 2022a. [Improving compositional generalization with latent structure and data augmentation](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4341–4362, Seattle, United States. Association for Computational Linguistics.
- Linlu Qiu, Peter Shaw, Panupong Pasupat, Tianze Shi, Jonathan Herzig, Emily Pitler, Fei Sha, and Kristina Toutanova. 2022b. [Evaluating the impact of model scale for compositional generalization in semantic parsing](#).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Joana Ribeiro, Shashi Narayan, Shay B. Cohen, and Xavier Carreras. 2018. [Local string transduction as sequence labeling](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1360–1371, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. [Compositional generalization and natural language variation: Can a semantic parsing approach handle both?](#) In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938, Online. Association for Computational Linguistics.
- Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. 2020. [On the potential of lexico-logical alignments for semantic parsing to SQL queries](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1849–1864, Online. Association for Computational Linguistics.
- Runxin Sun, Shizhu He, Chong Zhu, Yaohan He, Jinlong Li, Jun Zhao, and Kang Liu. 2022. [Leveraging explicit lexico-logical alignments in text-to-SQL parsing](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–289, Dublin, Ireland. Association for Computational Linguistics.
- Bailin Wang, Mirella Lapata, and Ivan Titov. 2021. [Structured reordering for modeling latent alignments in sequence transduction](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 13378–13391. Curran Associates, Inc.
- Pia Weißenhorn, Lucia Donatelli, and Alexander Koller. 2022. [Compositional generalization with a broad-coverage semantic parser](#). In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 44–54, Seattle, Washington. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Yuk Wah Wong and Raymond Mooney. 2006. [Learning for semantic parsing with statistical machine translation](#). In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 439–446, New York City, USA. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Jingfeng Yang, Le Zhang, and Diyi Yang. 2022. [SUBS: Subtree substitution for compositional semantic parsing](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, Seattle, United States. Association for Computational Linguistics.
- John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2, AAAI’96*, page 1050–1055. AAAI Press.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. [AMR parsing as sequence-to-graph transduction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy. Association for Computational Linguistics.

A Experimental details

For our experiments with TPOL we report the average of three runs for every result. We select the hyperparameters with grid search on the development set performance stopping when there is no more improvement. We choose the learning rate among $1e^{-4}$, $1e^{-5}$ and $1e^{-6}$ and the batch size between the bounds of 4 and 32. Usually the best performing models choose a learning rate of $1e^{-5}$ and batch size of 8. An experiment takes about 20 minutes on a single Nvidia V100 GPU. Our BERT (110M parameters) and mBART (680M parameters) implementations are taken from the transformers library (Wolf et al., 2020). We use for English *bert-base-uncased*, for Italian *dbmdz/bert-base-italian-uncased* and for German *dbmdz/bert-base-german-uncased*. For mBART we use *facebook/mbart-large-50*. For mT5 we use the *google/mt5-small* pre-trained checkpoint from the Transformers library.

B GEOALIGNED statistics

Table 5 provides statistics from the English version of GEOALIGNED (Locatelli and Quattoni, 2022). In particular, we report the number of examples that fall in the monotonic (MN) and non-monotonic (NMN) categories.

Category	GEO EN		
	?	Q	LEN
MN	194	154	162
NMN	86	51	118

Table 5: Number of examples that belong to the monotonic (MN) and non-monotonic (NMN) categories in GEOALIGNED English.

PePe: Personalized Post-editing Model utilizing User-generated Post-edits

Jihyeon Lee^{*†}

gina.lee@kakaobrain.com

Taehee Kim^{*}

taehee.kim@letsur.ai

taeheekim@kaist.ac.kr

Yunwon Tae^{*‡}

yunwon.tae@vuno.co

Cheonbok Park

cbok.park@navercorp.com

Jaegul Choo

jchoo@kaist.ac.kr

Abstract

Incorporating personal preference is crucial in advanced machine translation tasks. Despite the recent advancement of machine translation, it remains a demanding task to properly reflect personal style. In this paper, we introduce a personalized automatic post-editing framework to address this challenge, which effectively generates sentences considering distinct personal behaviors. To build this framework, we first collect post-editing data that connotes the user preference from a live machine translation system. Specifically, real-world users enter source sentences for translation and edit the machine-translated outputs according to the user's preferred style. We then propose a model that combines a discriminator module and user-specific parameters on the APE framework. Experimental results show that the proposed method outperforms other baseline models on four different metrics (*i.e.*, BLEU, TER, YiSi-1, and human evaluation).

1 Introduction

Language usage is strongly influenced by the state of the individual, which can be considered by multiple attributes such as age, gender, socioeconomic status, and occupation (Tannen et al., 1991; Pennebaker et al., 2003). Taking these aspects into account in the machine translation task, we need personalized translations to reflect individual characteristics that vary from person to person; thus, the translation system should consider not only fluency and content preservation, but also personal style.

However, most existing neural machine translation (NMT) models ignore personal style (Mirkin et al., 2015). Previous studies attempt to address this problem by personalizing the NMT models, but in these studies the definition of personal style

* Indicates equal contribution.

†Work done at Korea Advanced Institute of Science and Technology, correspondence to jihyeonlee@kaist.ac.kr.

‡Work done at Korea University, correspondence to tj204@korea.ac.kr.

<i>src:</i>	This is a <i>primary contribution</i> of our research.				
<i>mt:</i>	이것은	우리	연구의	주로	공헌이다.
	(This is a)	(of our)	(research)	(<i>primarily</i>)	(contribution)
				grammar error	
<i>pe:</i>	이것은	우리	research의	주된	기여이다.
	(This is a)	(of our)	(<i>research</i>)	(primary)	(<i>contribution</i>)
			src language		synonym

Figure 1: Example of a *personal post-editing* triplet (*i.e.*, source (*src*), machine translation (*mt*), and post-edit (*pe*)) given the source text in English and the translated text in Korean. A post-edited sentence does not only contain error correction of an initial machine translation result but also reflects individual preference. For instance, a human post-editor modifies the word "primarily" to "primary," but also change "공헌" to its synonym "기여" while keeping the rest as it is (*e.g.*, "research").

is often over-simplified (Rabinovich et al., 2017; Sennrich et al., 2016; Si et al., 2019). For example, Rabinovich et al. (2017) and Sennrich et al. (2016) define the personal style as politeness and gender respectively, which is not sufficient to tackle the multifarious character of an individual. Namely, previous works defined the personal style in a constrained form.

In contrast with previous studies, we propose a method based on an APE framework and newly utilize post-editing data to capture diverse personal traits in translation. Originally, the need for post-editing data is to improve the quality of machine-translated sentences in an APE task (Simard et al., 2007; Pal et al., 2016; Correia and Martins, 2019). However, we suggest that the post-editing data can also be adequate references for personalized translation if various users post-edit sentences according to their preferences. In this respect, we collect a **user-generated post-editing** dataset called USP through a live translation system. After the system translates a source sentence (*src*) to a target sentence, *i.e.*, machine-translated sentence (*mt*), each

user edits the translated result according to their purpose or preferences, *i.e.*, post-edited sentence (*pe*). We collect (*src*, *mt*, *pe*) triplets called *personalized post-editing* triplets for each user and an example is depicted in Fig 1.

Along with the personalized post-editing data, we develop a model which utilizes user parameter and a discriminator module. The user-specific parameters allow the model to adapt to each user in that the model can consider inter-personal variations. These parameters are aggregated with the output word probability such that the generation word probability distribution differs by each particular user. Moreover, since the prevalence of pre-trained language models encourages significant performance improvements on various natural language generation tasks (Song et al., 2019; Lewis et al., 2019; Correia and Martins, 2019), we exploit the pre-trained language model (LM) but do not fully lean on it. We assume that not all the features from the pre-trained LM contribute to capturing the distinct taste of users that are departing from the neutral and standardized patterns. Thus, our discriminator module, inspired by adversarial training (Goodfellow et al., 2014), attempts to dismantle the unnecessary features from a pre-trained LM, while tuning the model to incorporate a personal style. The details will discuss in Section 3.

Experiments on our dataset and speaker annotated TED talks dataset (Michel and Neubig, 2018) (SATED) demonstrate that the proposed approach generates diverse translations for different users.

In summary, our contributions include the following:

- To the best of our knowledge, this is the first work that leverages the APE framework to a personalized translation task.
- We propose a personalized post-editing model based on user-generated post-edits, which is able to capture the inter-personal variations that consist of multiple attributes.
- Extensive experimental results show that the proposed method robustly reflects personal traits and consistently outperforms baselines in three different quantitative metrics and human evaluation results.

2 Related Work

Our work is closely related to the recent work on personalized neural machine translation and auto-

matic post-editing.

Personalized neural machine translation.

Standard NMT systems are not able to consider the personal preference in a machine-translated output (Mirkin et al., 2015). Mima et al. (1997) is the early paper that proposes a concept of reflecting an author’s properties, such as gender, dialog domain, and role in the translation. However, including Mima et al. (1997), most studies conduct a limited range of personalized translations, which address only a single attribute (*e.g.*, politeness) (Sennrich et al., 2016; Rabinovich et al., 2017).

Turchi et al. (2017) and Karimova et al. (2018) fine-tune the model on the human post-edits to improve the NMT quality, which can be viewed as a naive approach to handle the personalized translation without attribute labels. Wuebker et al. (2018) extend this approach to adjust only a small number of parameters, but still requires extensive training resources. Meanwhile, Michel and Neubig (2018) and Huan et al. (2021) propose a generalized form of a personalized translation method, which are closely related work with ours. Michel and Neubig (2018) cast this problem as an extreme form of domain adaptation, while Huan et al. (2021) introduce cache module and contrastive learning to increase the diversity on dissimilar users. However, the reference sentences for personalized translation were constructed by a few professional translators, not by a variety of people with diverse characteristics; personal preferences reflected in the dataset are limited. Our user-generated post-edits are edited by a large number of people who provide the original sentences.

Automatic post-editing. Prior to the emergence of the transformer (Vaswani et al., 2017), RNN based APE models (Pal et al., 2016; Junczys-Dowmunt et al., 2016; Junczys-Dowmunt and Grundkiewicz, 2017) are actively studied. Subsequently, as self-attention based models show significant improvements on various downstream tasks, transformer based models also prevail in the APE task. Specifically, a popular approach is to set a separate encoder for the source and machine-translated (MT) output. Separately encoded representations are joined in the following encoder (Pal et al., 2018) or fused in the decoder (Tebbifakhr et al., 2018; Junczys-Dowmunt and Grundkiewicz, 2018). More recently, Correia and Martins (2019) improve the performance of APE tasks by leveraging a

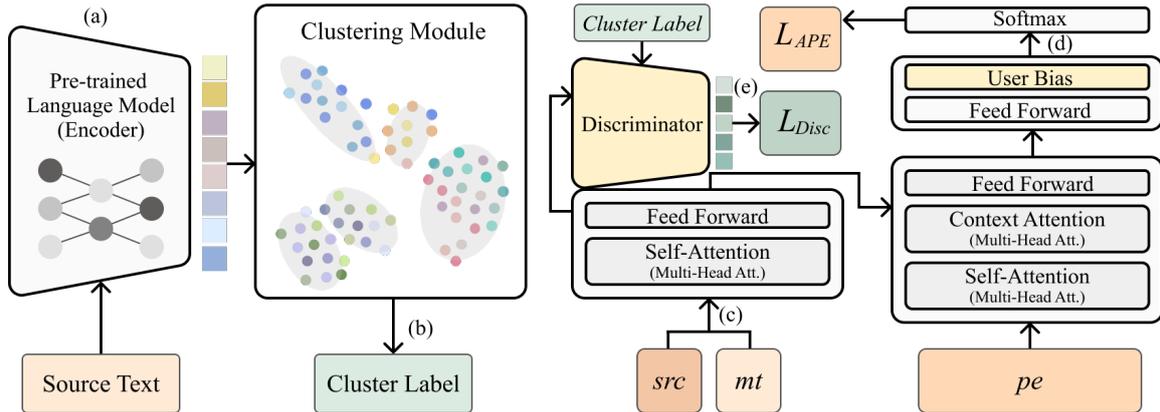


Figure 2: An overview of our proposed method. PePe consists of two parts: 1) Clustering module that relies on pre-trained LM encoder and Gaussian mixture model. 2) APE architecture that includes an auxiliary discriminator and user-specific parameters.

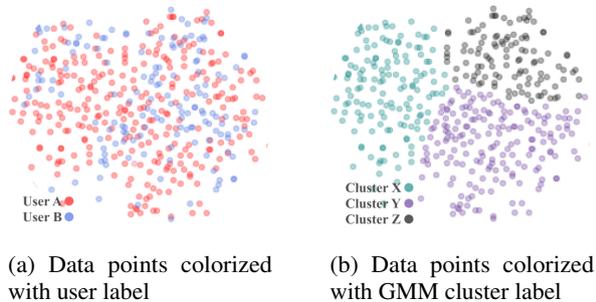


Figure 3: Source sentences of USP embedded with the pre-trained LM. (a) and (b) shows the discrepancy between the user data distribution and the contextual similarity-based data distribution.

pre-trained LM. Compared to these studies, our work is the first attempt to examine the neural network based APE model for personalized translation. There is a study where they use an APE module for domain adaptation (Isabelle et al., 2007), but the explored one is based on a statistical machine translation system.

3 Proposed Method

Overview: It is challenging to generate appropriate translations that impose personal variations. To address such a demanding problem, we take a detour by applying an APE framework. We propose PePe, a personalized post-editing model utilizing user-generated post-edits. PePe includes a discriminator module to allow the model to dismantle the pre-trained LM features. Specifically, we maximize the discriminator loss to encourage the encoder to throw away irrelevant pre-trained LM features, while minimizing the APE loss to guide the model

to utilize the pre-trained LM features that are useful for personalization. In addition, PePe utilizes user-specific parameters to capture the personal style. User-specific parameters are combined at the end of the decoder layer to adjust the prediction of the word probability, i.e., the word choice based on a user preference. Our strategy does not require expensive supervision on the personal style, such as explicit attribute labeling or an attribute-tailored model architecture.

The overall architecture of PePe is illustrated in Fig. 2. The two following subsections will describe the modules shown in Fig. 2-(a), (b), (c), (d), and (e), respectively.

3.1 Contextual Similarity vs. User-specific Style

The pre-trained LM is well known for capturing the contextual similarity that is useful to define the label for in-domain data (e.g., sports, IT, and economy). However, the user-specific style is far from those domains; it does not coincide with contextual similarity yet involves somewhat arbitrary traits (i.e., user preferences). Hence, we argue that some of the features from a pre-trained LM distract personalized translation, which rather requires generating biased results to meet the individuals' needs. Fig. 3 demonstrates the discrepancy between the user data distribution and the contextual similarity-based data distribution.

We map the sampled sentences from USP to the embedding space of the pre-trained LM. Each sentence is encoded with RoBERTa (Liu et al., 2019) and visualized using t-SNE (van der Maaten and Hinton, 2008). The data on both sides show the same embedding representations obtained from the

same set of sentences, but labeling is different. The data items in Fig. 3a are color-coded by the users, whereas those in Fig. 3b are color-coded by the semantic cluster labels obtained from the Gaussian mixture model (GMM) (Rasmussen, 2000), which allocates the similar sentences to the same label based on the RoBERTa embedding of each.

In Fig. 3b, semantically similar points, which are close in embedding space, belong to the same clusters. However, the red and blue points in Fig. 3a, which indicate sentence representations from two different users, are distributed unruly instead of being grouped by semantic similarity. In other words, the fine-grained style differences of each user are somewhat distant from the contextual similarity of the sentences; thus it is hard to distinguish user-specific preferences when the model is highly oriented to learning the contextual similarity.

3.2 Generating Cluster Labels based on Pre-trained LM

Inspired by the finding in Section 3.1, we devise a discriminator module that uses the semantic cluster labels to unlearn the features from the pre-trained LM that are unnecessary to reflect the personal styles. Before introducing the details about PePe, we describe how to generate the semantic cluster labels from a pre-trained LM in an unsupervised manner. We first encode *src* into encoded vectors using a pre-trained LM¹ as shown in Fig 2-(a). Based on these encoded vectors, semantic cluster labels are generated by GMM (Rasmussen, 2000) as illustrated in Fig 2-(b). A Gaussian mixture is a function made up of the k number of Gaussian components, where k is the number of clusters² and is a hyperparameter. Specifically, in GMM, $\sum_{i=1}^k \pi_i p_i(\mathbf{h}|\theta_i)$ represents the distribution of data point \mathbf{h} , where \mathbf{h} is an encoded vector of the first token of *src*, *i.e.*, [CLS] token, π_i is the probability of each Gaussian fitting the data, and p_i is the Gaussian density function parameterized by θ_i . We assign each sentence to a Gaussian that best describes the data, and the Gaussian corresponds to the semantic cluster label. The label, *i.e.*, $T = t_1, \dots, t_k$, is then used as a classification label for our discriminator, which is described in the following subsection.

¹Though we use RoBERTa as a pre-trained LM to generate cluster labels, other pre-trained LMs can also be used in our approach.

²Ten clusters are used for all the experiments in the main paper.

3.3 PePe: Personalized Post-editing Model utilizing User-generated Post-edits

We adopt BERT-based Encoder-Decoder APE model (Correia and Martins, 2019) called Dual-Source BERT (DS-BERT) as our backbone, which is based on transformer (Vaswani et al., 2017) with pre-trained multilingual BERT (Devlin et al., 2019). DS-BERT uses a single encoder which is used to encode both the *src* and the *mt* by concatenating them with the specialized token [SEP] as described in Fig 2-(c).

Our model also learns to generate $y = [y_1, \dots, y_n]$, *i.e.*, *pe*, from x , *i.e.*, *src*, and $\tilde{y} = [\tilde{y}_1, \dots, \tilde{y}_m]$, *i.e.*, *mt*, by maximizing the likelihood,

$$P(y|x, \tilde{y}; \theta_{APE}) = \prod_{i=1}^n P(y_i|x, \tilde{y}, y_{<i}; \theta_{APE}),$$

where y_i is the i -th target word and $y_{<i} = y_1 \dots y_{i-1}$ is the partial translation result. θ_{APE} represents the parameters for translating source sentence into post-edited sentence with machine-translated result \tilde{y} .

In order to adapt user-specific linguistic styles, we add user-specific parameters before the softmax layer in the decoder as shown in Fig 2-(d), *i.e.*,

$$P(y_i|x, \tilde{y}, y_{<i}; \theta_{APE}, \theta_{user}) = f(FFN(o_i) + \theta_{user}),$$

where FFN and f are a feed-forward network and softmax function, respectively. o_i is the output for the i -th target word from the decoder. $\theta_{user} \in \mathbb{R}^V$ is a user-specific embedding vector from a set of trainable user embedding matrix $U \in \mathbb{R}^{N \times V}$ where N is the number of users and V is the size of vocabulary.

The model is then optimized by minimizing \mathcal{L}_{APE} defined as

$$\mathcal{L}_{APE} = - \sum_{i=1}^n \log P(y_i|x, \tilde{y}; \theta_{APE}, \theta_{user}).$$

Furthermore, as shown in Fig 2-(e), we introduce a discriminator module to unlearn the contextual similarity feature learned from a pre-trained LM. To train the discriminator, we compute the discriminator loss \mathcal{L}_{Disc} defined as

$$\mathcal{L}_{Disc} = \sum_i^k t_i \log(\tilde{t}_i),$$

where k is the number of classes (*i.e.*, the number of Gaussians we pre-defined) and t_i is the ground-truth label of the semantic cluster. \tilde{t}_i represents the

output from the discriminator which is a single-layer feed-forward network for the classification of semantic cluster labels. We use the first token of a source sentence to extract a sentence representation from the encoder and pass it to the discriminator as an input. Note that we use the gradient ascent method to prevent the encoder from classifying the clusters. In this way, we diminish the unnecessary feature from pre-trained LM, while our APE loss function incorporated with user-specific parameters leads the model to capture the user-specific style.

Finally, PePe optimizes a combination of two losses, \mathcal{L}_{Disc} and \mathcal{L}_{APE} , with an adjustment rate β , *i.e.*,

$$\mathcal{L}_{Train} = \beta \cdot \mathcal{L}_{Disc} + (1 - \beta) \cdot \mathcal{L}_{APE}.$$

4 Experiments

In this section, we qualitatively and quantitatively demonstrate the effectiveness of our proposed method. We validate PePe, described in Section 3, against other baseline methods using a real-world user dataset USP. We also provide a detailed explanation for the dataset. Moreover, through extensive experiments and analyses, we show that PePe can incorporate inter-personal variations into a target sentence. We provide training details in Appendix A.

4.1 Dataset

We collect the user-generated post-editing dataset, USP, from a publicly available online translation system³ (*e.g.*, Google Translate). Fig 4 illustrates the user experience flow. The users enter the sentences they want to translate, and the system provides the corresponding outputs that are generated by the high-quality commercialized machine translator. From the machine-translated outputs, users can start to edit the translated sentences according to their preference by clicking the “post-edit” button. Consequently, when the users click the “Finish” button after completing the changes, a triplet of the source sentence, machine-translated output, and personalized post-edit is sent to our database. Note that the origin of post-edited sentences is each particular user, which makes USP contains inter-personal variation, unlike existing APE datasets.

³We collected data only from users who consent to the data collection for research purposes. In addition, there is no privacy issue because de-identification had taken for the collected data.

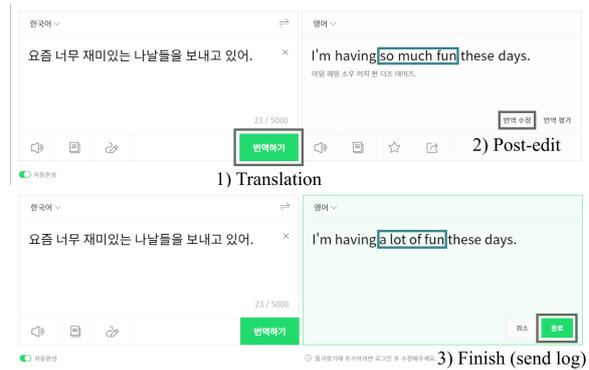


Figure 4: Illustration of the user experience flow for post-edit log generation.

Since we collect USP from the real-world users’ inputs that contain various noises (*e.g.*, unedited, duplicated, or meaningless examples), we preprocess the data to eliminate these noises. Furthermore, most users only edited few examples, which are not sufficient to represent their style. Therefore, we select the users who left more than 100 samples, *i.e.*, 30 users with 7K sentences and 70 users with 9K sentences for $en \rightarrow ko$ and $ko \rightarrow en$ USP dataset, respectively. For users who left less than 100 samples, we aggregate the samples (*i.e.*, 0.12M sentences) and utilize them as training data for the task-adaptive pre-training (Gururangan et al., 2020). The discriminator module and user-specific parameters are not used in the task-adaptive pre-training and only the parameters for DS-BERT are utilized for the pre-training stage. Details of data preprocessing are in Appendix A.

Additionally, we adopt a Speaker Annotated TED (SATED) dataset (Michel and Neubig, 2018) containing more than 2,000 sets of speaker style-contained source sentences, which is publicly available. We select the dataset to show the robustness of our model regarding different datasets and languages.

4.2 Experimental Setup

Evaluation metric. We use three different metrics to evaluate how well our proposed model preserves the content and incorporates the personal preferences. BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) scores are considered to assess the translated sentence where the ground-truth sentence is a *pe* sentence. We also leverage YiSi-1 (Lo, 2019) that computes the semantic similarity of phrases between the model output and *pe*, which can be sensitive to detailed styles. We

Methods	<i>en</i> → <i>ko</i>			<i>ko</i> → <i>en</i>		
	BLEU↑	TER↓	YiSi-1↑	BLEU↑	TER↓	YiSi-1↑
(1) Uncorrected	64.9 (-5.6)	21.1 (+1.2)	87.3 (-1.1)	75.1 (-3.5)	17.7 (+1.4)	88.9 (-0.8)
(2) DS-BERT	68.4 (-2.1)	21.1 (+1.2)	87.6 (-0.8)	77.1 (-1.5)	17.6 (+1.3)	89.1 (-0.6)
(3) DS-BERT + Full Bias	68.6 (-1.9)	20.9 (+1.0)	88.0 (-0.4)	78.0 (-0.6)	16.9 (+0.6)	89.6* (-0.1)
(4) DS-BERT + Factor Cell	67.5 (-3.0)	22.1 (+2.2)	88.0 (-0.4)	76.5 (-2.1)	18.4 (+2.1)	89.2 (-0.5)
(5) DS-BERT + User CLS	69.0 (-1.5)	20.9 (+1.0)	87.1 (-1.3)	78.1 (-0.5)	16.5 (+0.2)	89.4 (-0.3)
(6) DS-BERT + User Token	68.8 (-1.7)	20.5 (+0.6)	87.0 (-1.4)	74.3 (-4.3)	21.6 (+5.3)	88.5 (-1.2)
(7) PePe	70.5	19.9	88.4	78.6	16.3	89.7
(8) -discriminator	68.6 (-1.9)	20.9 (+1.0)	88.0 (-0.4)	78.0 (-0.6)	16.9 (+0.6)	89.6* (-0.1)
(9) -(8) & user bias	68.4 (-2.1)	21.1 (+1.2)	87.6 (-0.8)	77.1 (-1.5)	17.6 (+1.3)	89.1 (-0.6)
(10) -(9) & pre-training	60.2 (-10.3)	31.9 (+12.0)	86.3 (-2.1)	67.6 (-11.0)	28.7 (+12.4)	87.6 (-2.1)

Table 1: Quantitative comparison with the baselines on the USP dataset that contains *en*→*ko* language pairs and vice versa. (8) to (10) denotes the ablation results. The ablation study is designed to verify each module in PePe. User bias in (9) denotes the user-specific parameters located at the end of the decoder, and pre-training in (10) denotes the task-adaptive pre-training stage. The bold represents the significant difference ($p < 0.05$) against other baselines. We conduct the t-test with five runs and report the average score of it. * means that there is no significant difference in the scores between the model and PePe.

also conduct a human evaluation, which will be described in the following section.

Baseline Methods. We compare the performance of our method with the following baselines. Since this is the first attempt to personalize the translation using post-edits, we newly adjust existing personalized translation methods onto the APE framework for comparisons.

1) Uncorrected is the same as *mt* in personalized post-editing data, which is generated from the online MT system. No correction was made on it. **2) DS-BERT** is a transformer based post-editing model (Correia and Martins, 2019) that we adopt as our backbone in the method section. DS-BERT is a general approach in the recent APE task. To our knowledge, the recently proposed state-of-the-art APE models (Yang et al., 2020; Oh et al., 2021) are either based on the Dual-Source Transformer (Junczys-Dowmunt and Grundkiewicz, 2018) or DS-BERT. We believe that demonstrating the feasibility of personalized post-editing using a fundamental APE model is more suitable than models that use APE task-specific techniques. **3) DS-BERT + Full bias** (Michel and Neubig, 2018) utilizes additional user bias vectors on the decoder’s output. **4) DS-BERT + Factor bias** (Michel and Neubig, 2018) uses factorized user bias on the output of the decoder. User-independent biases are shared with all users. However, the user-specific vector can adjust each user-independent vector’s magnitude. **5) DS-BERT +**

User CLS is a multi-task composed of a user classification and APE task. The first token of an encoder input is used to stand for user identity. The corresponding output vector is used to classify a ground-truth user label. A single layer of a feed-forward neural network is used for the classifier. **6) DS-BERT + User Token** (Sennrich et al., 2016) adds a token at the start of each post-edited sentence to indicate the user for each sentence. We train the model in a teacher-forcing manner.

4.3 Quantitative Evaluation

Results using automatic metrics and human evaluation are presented in this section. PePe consistently outperforms the baselines on all datasets we considered. We also show the robustness of PePe regarding the different number of users, data distributions, and language pairs.

Performance of PePe against other baselines.

(1) to (7) in Table 1 shows the personalized translation results of varied baselines. Our proposed method outperforms the six baselines with the non-trivial margin both on *en*→*ko* and *ko*→*en* USP dataset. For instance, BLEU score increased in the range of 1.7 to 5.6, YiSi-1 increased in the range of 0.4 to 1.4, and TER decreased in the range of 0.6 to 2.2 over baselines, in *en*→*ko* dataset. Consistent results from these three different metrics verify that PePe easily figure out distinct taste of users while preserving source contents. Especially, experiments in *en*→*ko* dataset show the most out-

Metrics	PePe	DS-BERT	Uncorr.
Style - 1st	59.6	18.1	22.2
Style - 2nd	21.0	39.1	39.9
Style - 3rd	19.5	42.6	37.9
Non-Style	3.94(1.08)	3.60(1.19)	3.82(1.16)

Table 2: Human evaluation on $en \rightarrow ko$ USP dataset. Style and non-style factors are both surveyed. For the style factor, each score represents the proportion. For instance, 59.6% of evaluators choose PePe as the first place among other models. For the non-style factor, a Likert scale from 1 to 5 evaluates fluency and source contents preservation. We report the average score and the standard deviation.

Model	$en \rightarrow de$	$en \rightarrow fr$
	BLEU \uparrow	BLEU \uparrow
Michel and Neubig (2018)	27.2	38.5
DS-BERT	30.4	42.2
PePe	31.2	43.7

Table 3: Experiments on the SATED dataset. PePe outperforms DS-BERT on different language pairs even for a synthetic post-editing dataset. The bold represents the best score among the baselines and significantly ($p < 0.05$) outperforms DS-BERT.

standing performance gains since the data mostly come from the users whose first language is Korean; the users can reflect the linguistic preference more naturally on this dataset.

Ablation study. The comparison between PePe and (8) in Table 1 shows the importance of the discriminator module. When we exclude the discriminator module, the BLEU and TER scores are decreased on both $en \rightarrow ko$ and $ko \rightarrow en$. The results of the vanilla APE model (*i.e.*, (9) in Table 1) show that the user-specific parameters are also significant for personalized translation. Moreover, when we do not adopt the APE task-adaptive pre-training (*i.e.*, (10) in Table 1), the performance of the model drops even further. Overall, our ablation study demonstrates that each component is essential for the task.

Human evaluation. To validate the advantage of our approach, we conduct human evaluations. Human evaluation can be a reasonable measurement choice to evaluate the personalization task because even sophisticated evaluation metrics can fail to capture the abstract (*i.e.*, high-level) user behavior

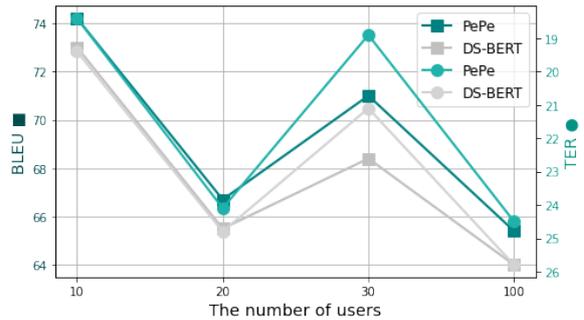


Figure 5: Robustness on the number of users. The dark squares denote the BLEU score, and the light circles denote the TER score. The number of clusters is equally adopted as ten for all cases. The users are randomly selected from the USP dataset.

reflected in the pe sentence. We hired 20 Korean-English who are bilingual and engaged in the fields of linguistics and machine learning for human evaluation. We randomly select 30 source sentences and generate corresponding target sentences from Uncorrected, DS-BERT, and PePe before carrying out two types of questions to compare different metrics. 1) We ask participants to annotate generated sentences along with fluency and content preservation. Sentences are measured on a Likert scale from 1 to 5. 2) We take three sentences generated from three different models. Participants rank these sentences from first to third, *i.e.*, asking which sentence is most similar to the ground-truth pe that contains distinct writing styles.

As reported in Table 2, PePe is ranked 1st by most evaluators. PePe not only achieved the best score on style evaluation but also on non-style factors (*i.e.*, fluency and contents preservation), which is essential for the translation task. DS-BERT achieves the lowest score on both measures, indicating that the ambiguous reflection of style is worse than none. The human evaluation results are consistent with our quantitative results measured by automatic metrics.

Robustness of our model. Table 3 shows the personalized translation results on $en \rightarrow de$ and $en \rightarrow fr$ SATED dataset. Since the dataset is initially constructed for the machine translation task where post-edited sentences do not exist, we utilize target sentence (*i.e.*, mt) in the place of pe and independently generate mt from a particular translation model (*i.e.*, pre-trained transformer based NMT model). Regardless of the language, the results demonstrate that PePe and DS-BERT, which leverages triplets (src , mt , pe), outperform Michel and Neubig (2018)

<i>en</i> → <i>ko</i>	
<i>src</i>	<u>Immediately</u> provide non-monetary <u>benefits</u> as required.
<i>mt</i>	필요에 따라 즉시 비화폐성 편익을 제공하십시오.
PePe	요구된대로 비화폐성 혜택을 즉시 제공한다.
<i>pe</i>	요구되어진 비화폐성 혜택을 즉시 제공한다.
<i>src</i>	Choose this option to make the current preset load whenever a new multi Instrument is created.
<i>mt</i>	새 멀티 계측기가 생성될 때마다 현재 사전 설정된 로드를 만들려면 이 옵션을 선택하십시오.
PePe	새 multi instrument 가 생성될 때마다 현재 preset 을 만들려면 이 옵션을 선택하십시오.
<i>pe</i>	새 multi instrument 가 생성될 때마다 현재 preset load 를 만들려면 이 옵션을 선택하십시오.
<i>ko</i> → <i>en</i>	
<i>src</i>	겨울왕국 2 속에 나오는 장면이 있다.
<i>mt</i>	there is a scene in winter kingdom 2 .
PePe	there is a scene in frozen 2 .
<i>pe</i>	there is a scene in frozen 2 .
<i>src</i>	관사 아래에 있는 모음코드가 이렇게 바뀌어진다.
<i>mt</i>	the vowel code under the official building changes like this.
PePe	the vowel code under the article changes like this.
<i>pe</i>	the vowel code under the article changes like this.

Table 4: Qualitative examples of post-edited sentences generated from PePe. User-specific parts in *pe* and corresponding parts in *mt* are colored. We highlight the post-edited words in PePe with bold if the words are identical to *pe*. Corresponding parts in *src* are underlined. Our model finds an appropriate combination of attributes in accordance with sentences and users.

that relies on paired sentences (*src*, *mt*). In addition, the results also show that even if *pe* is not edited from the *mt*, PePe translates the source sentence close to the ground-truth target sentence that connotes the speaker’s characteristics.

Fig. 5 shows that our model works well regardless of the number of users. Grey-colored lines are the performance of the baseline model, and colored lines are the performance of PePe. TER axis is reversed on the graph to make consistency with the BLEU score. Note that the higher points denote a better score than the lower points.

Furthermore, we conduct additional experiments that show the robustness of our approach regarding the number of clusters and adjustment rates, which are hyperparameters. We represent the results in Appendix B.

4.4 Qualitative Analysis

To understand how user-specific preferences are incorporated into the sentences, we qualitatively analyze the post-edited results of our model as shown in Table 4. A typical example of the multi-attribute correction appears in the first row, which changes the sentence structure and the preferred word choices. Our model tends to retain the overall meaning of the source sentence while precisely treating an abstractive personal behavior. The output of PePe in the second row tends to keep loanwords in English instead of translating them into Korean (*i.e.*, “*multi instrument*”, “*preset load*”),

while *mt* suffers from generating sentences that consider those preferences. An example of changing a homonym to a suitable word is shown in the last row. Since “*official building*” and “*article*” are homonyms in Korean, PePe chooses the word that is appropriate for the semantic meaning of the sentence. We further provide several examples that consider the multidimensional attributes in Appendix C. In either a single attribute or a multi attributes case, our model properly reflects distinct preferences.

5 Conclusion

In this work, we propose a personalized post-editing method, PePe, utilizing user-generated post-edits. Based on the APE framework, PePe leverages two modules, 1) user-specific parameters and 2) a semantic cluster-based discriminator module. These modules lead to reflect the multifarious interpersonal variations, where the former allows the model to learn user-dependent probabilities for each word while the latter unlearns the detrimental features in a pre-trained language model and maintains advantageous effects of the transfer learning. We empirically demonstrate that PePe reflects fine-grained user preference in a variety of settings. To the best of our knowledge, this work is the first attempt to utilize the APE framework with the user-generated post-edits for personalized translation. We believe that our work can draw more attention toward personalized translation, which is the ul-

timate direction that the neural translation model should go forward.

6 Limitations

Promising future work is analyzing the pattern of personalization depending on language pairs. Depending on the nationality of users, the pattern of personalization may appear differently due to cultural differences, and extensive experiments on various language pairs are required to analyze this. In addition, if anyone can access the personalized model, there is a potential risk that the model can be abused to disguise itself as a specific individual. Therefore, there is a need for a strategy of limiting the authority to access the personalized model or verifying a person who uses the personalized model.

Acknowledgement

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government(MSIT) (No. 2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)). This work was supported by Papago, NAVER Corp. The authors appreciate Hyung-Gyu Lee, Eunjeong Lucy Park, and Papago MT researchers in NAVER. We also thank the anonymous reviewers for their valuable feedback.

References

- Gonalo M. Correia and Andr  F. T. Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3050–3056.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.
- Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680.
- Suchin Gururangan, Ana Marasovi , Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don’t stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.
- Lin Huan, Yao Liang, Yang Baosong, Liu Dayiheng, Zhang Haibo, Luo Weihua, Huang Degen, and Su Jin-song. 2021. Towards user-driven neural machine translation. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- P. Isabelle, Cyril Goutte, and Michel Simard. 2007. Domain adaptation of mt systems through automatic postediting. *Proc. of Machine Translation Summit*.
- Marcin Junczys-Dowmunt, Tomasz Dwojak, and Rico Sennrich. 2016. The amu-uedin submission to the wmt16 news translation task: Attention-based nmt models as feature functions in phrase-based smt. In *Proc. of the Conference on Machine Translation (WMT)*, pages 319–325.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. In *Proc. of the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 120–129.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Ms-uedin submission to the wmt2018 ape shared task: Dual-source transformer for automatic post-editing. In *Proc. of the Conference on Machine Translation (WMT)*, pages 822–826.
- Sariya Karimova, Patrick Simianer, and Stefan Riezler. 2018. A user-study on online adaptation of neural machine translation to human post-edits. *Machine Translation*, 32:309–324.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. the International Conference on Learning Representations (ICLR)*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Computing Research Repository (CoRR)*, abs/1907.11692.
- Chi-kiu Lo. 2019. Yisi-a unified semantic mt quality evaluation and estimation metric for languages with different levels of available resources. In *Proc. of the Conference on Machine Translation (WMT)*, pages 507–513.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proc. the International Conference on Learning Representations (ICLR)*.

- Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 312–318.
- Hideki Mima, Osamu Furuse, and Hitoshi Iida. 1997. Improving performance of transfer-driven machine translation with extra-linguistic information from context, situation and environment. In *Proc. the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 983–989.
- Shachar Mirkin, Scott Nowson, Caroline Brun, and Julien Perez. 2015. Motivating personality-aware machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1102–1108.
- Shinhyeok Oh, Sion Jang, Hu Xu, Shounan An, and Insoo Oh. 2021. Netmarble ai center’s wmt21 automatic post-editing shared task submission. *arXiv preprint arXiv:2109.06515*.
- Santanu Pal, Nico Herbig, Antonio Krüger, and Josef van Genabith. 2018. A transformer-based multi-source automatic post-editing system. In *Proc. of the Conference on Machine Translation (WMT)*, pages 827–835.
- Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. 2016. A neural network based approach to automatic post-editing. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 281–286.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.
- James W Pennebaker, Matthias R Mehl, and Kate G Niederhoffer. 2003. Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577.
- Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. 2017. Personalized machine translation: Preserving original author traits. In *Proc. of the Annual Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1074–1084.
- Carl Edward Rasmussen. 2000. The infinite gaussian mixture model. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, pages 554–560.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proc. of The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 35–40.
- Chenglei Si, Kui Wu, Ai Ti Aw, and Min-Yen Kan. 2019. Sentiment aware neural machine translation. In *Proceedings of the 6th Workshop on Asian Translation*.
- Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007. Rule-based translation with statistical phrase-based post-editing. In *Proc. of the Workshop on Statistical Machine Translation (WMT)*, pages 203–206.
- Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of the association for machine translation in the Americas (AMTA)*, pages 223–231.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.
- Deborah Tannen et al. 1991. *You just don’t understand: Women and men in conversation*. Virago London.
- Amirhossein Tebbifakhr, Ruchit Agrawal, Matteo Negri, and Marco Turchi. 2018. Multi-source transformer with combined losses for automatic post editing. In *Proc. of the Conference on Machine Translation (WMT)*, pages 846–852.
- Marco Turchi, Matteo Negri, M Amin Farajian, and Marcello Federico. 2017. Continuous learning from human post-edits for neural machine translation. *The Prague Bulletin of Mathematical Linguistics (PBML)*, 108:233–244.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9:2579–2605.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.
- Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 881–886.
- Hao Yang, Minghan Wang, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, and Yimeng Chen. 2020. Hw-tsc’s participation at wmt 2020 automatic post editing shared task. In *Proc. of the Conference on Machine Translation (WMT)*.

Supplementary Material

This material complements our paper with additional experimental results and miscellaneous details. Section A provides the implementation details. Section B addresses the additional experiments that show the robustness of our model against a varied number of clusters and adjustment rates. In Section C, we demonstrate the variety of qualitative examples of post-edited sentences generated from PePe.

A Training details

Data Preprocessing. For the data preprocessing, we first filter out the duplicate lines and normalize the data such that each line represents a single sentence. Also, we exclude sentence that is longer than 100 words. Then, we utilize term frequency inverse document frequency (TF-IDF) to compute the user similarity score and filter out the noisy users. To be specific, we form a document for each user by aggregating *src*. If a particular user has a lower than 0.1 similarity score, we exclude those users. We assume that if a user has a lower similarity score with others, then those users may contain noisy sentences. After preprocessing noisy data for USP, we divide the dataset into train/valid/test, which results in 5,207, 1,001, and 1,125 samples for Korean to English language pair, and 6,330, 1,360, and 1,357 samples for English to Korean. Since we split into train/valid/test for each user, the user appearing in the train dataset guarantees to appear in the test dataset.

Training and Inference Procedures. The main difference between training and inference procedures is the existence of a discriminator module. In other words, the clustering module and the discriminator are not utilized during the inference procedure. However, similar to the training procedure, we utilize the trained user-specific bias vector that corresponds to the user ID of each input sentence while generating a post-edited sentence.

Evaluation and HyperParameter Details. We evaluate all experiments based on SacreBLEU⁴, TER⁵, and YiSi-1⁶ scores. Since YiSi-1 requires pre-trained word embedding vectors, we utilize fastText⁷ to pretrain word embeddings. For the

⁴<https://github.com/mjpost/sacrebleu>

⁵<https://www.statmt.org/wmt18/ape-task.html>

⁶<https://github.com/chikiulo/yisi>

⁷<https://github.com/facebookresearch/fastText>

Hyperparameters	Value
Pre-trained LM	BERT-base-multilingual
Learning rate	0.00005
Batch size	512
Accumulation step	2
Optimizer	AdamW
Dropout	0.1
Label smoothing	0.1
Random seed	42, 101, 1215, 1129, 909
Decoding strategy	Beam search
Beam size	3

Table 5: Hyperparameter settings. AdamW (Loshchilov and Hutter, 2019) is the Adam (Kingma and Ba, 2015) optimizer with weight decay.

hyper-parameter settings, we use 10 clusters with 0.3 adjustment rates for all the experiments in the main paper. We select the combination of hyperparameters by manual tuning, which achieves the highest performance in the validation set based on the TER metric. Conditions for early-stopping and decoding are equally applied to the baselines. We follow the settings of hyperparameters in Correia and Martins (2019) except sharing the weight of the encoder and the decoder. We conducted all the experiments five times, and the random seeds used were 42, 1215, 101, 909, and 1129. We selected the highest performance learning rate value between 0.00005 and 0.0001. We report the configuration of our best model in Table 5.

Environment Details. All experiments in Table 1 is examined with CentOS Linux release 7.8.2003, Tesla P40 GPU, and Intel Xeon CPU E5-2630. Results in Table 3 are examined with Ubuntu 16.04.6, Intel Xeon processor, and Tesla V100-PCIE-32GB GPU. The versions of the libraries we used in all experiments are 3.7.6 for Python and 1.4.0 for Pytorch.

B Robustness to the number of cluster and hyperparameter

In the main paper, we conduct all experiments with 10 cluster labels. However, to be useful for the varied settings, it is crucial to demonstrate the model’s robustness to the number of clusters and adjustment rate. Here we provide the results trained on 30 cluster labels with various adjustment rates from 0.1 to 0.5. Identical with Table 1, we utilize *en*→*ko* dataset of 30 users. Table 6 indicates that PePe

Models	BLEU \uparrow	TER \downarrow
Uncorrected	64.9	21.1
DS-BERT	68.5	21.1
PePe (k30, m0.1)	70.2	20.2
PePe (k30, m0.2)	69.7	20.3
PePe (k30, m0.3)	69.9	19.9
PePe (k30, m0.4)	69.0	20.8
PePe (k30, m0.5)	70.2	20.3

Table 6: Experiments on various hyperparameter settings on a USP dataset. k denotes the number of clusters and m denotes the adjustment rate.

consistently generates high-quality sentences, regardless of the hyperparameters.

C Additional qualitative examples

This section provides additional qualitative examples from PePe. We choose the samples from the inference results of USP dataset, and both $ko \rightarrow en$ and $en \rightarrow ko$ language pairs are reported. As shown in Table 7, Table 8, Table 9, Table 10, Table 11, and Table 12, the tables are organized according to the typical personalization cases (*i.e.*, error correction, word choice, politeness, and multiple attributes). **Red color** represents error correction case, **Yellow color** represents word choice case, and **Green color** represents politeness case. Each case also accompanies the insertion and deletion of the words (*i.e.*, tokens). Sentences inferred from PePe show that it well reflects the personal traits of each user and the characteristics of each language.

D Importance of personalized translation

The importance of stylized translation can stand out in certain scenarios, such as the translation of everyday conversations. For example, when an English speaker uses a translator to talk to a German speaker, he or she may wish to communicate with a translation result that includes an individual’s personality rather than a normal translation of a neutral tone (e.g., replacing with a word that the user likes to use). At this time, our personalized translation methodology can be used to deliver translation results containing the user’s personality to the German speaker without post-processing. We believe that in order for a translation model to be utilized in the everyday conversation of various users, it is ultimately important to consider the individuality of each user beyond fluency.

Error Correction (en→ko)	
src	Begin the stroke by moving the hand , while the elbow remains still and high.
mt	팔꿈치가 가만히 있고 높게 유지되는 동안 손을 움직이면서 뇌졸중을 시작한다.
PePe	팔꿈치가 가만히 있고 높게 유지되는 동안 손을 움직이면서 팔동작을 시작한다.
pe	팔꿈치가 가만히 있고 높게 유지되는 동안 손을 움직이면서 팔동작을 시작한다.
src	Periodically check on her progress.
mt	정기적으로 진행 상황을 확인합니다.
PePe	정기적으로 그녀의 진행 상황을 확인합니다.
pe	정기적으로 그녀의 진행 상황을 확인합니다.
src	All manual checks unclaimed for more than 6 months shall be canceled.
mt	6개월 이상 청구되지 않은 모든 수동 점검은 취소된다.
PePe	6개월 이상 청구되지 않은 모든 수동 수표는 취소된다.
pe	6개월 이상 청구되지 않은 모든 수동 수표는 취소된다.
src	If a signal has finite power its energy will be infinite.
mt	만약 어떤 신호가 한정된 힘을 가지고 있다면 그 에너지는 무한할 것이다.
PePe	만약 어떤 신호가 한정된 전력을 가지고 있다면 그 에너지는 무한할 것이다.
pe	만약 어떤 신호가 한정된 전력을 가지고 있다면 그 에너지는 무한할 것이다.
src	The historical cost of the intangible fixed assets transferred shall be the historical cost recorded in the accounting records of the receiver.
mt	이전하는 무형고정자산의 역사적원가는 수취인의 회계기록에 기록된 역사적원가를 말한다.
PePe	이전하는 무형고정자산의 취득원가는 수취인의 회계기록에 기록된 취득원가를 말한다.
pe	이전하는 무형고정자산의 취득원가는 수취인의 회계기록에 기록된 취득원가를 말한다.
src	Emotional exhaustion is the central quality and the most obvious manifestation of burnout.
mt	감정 기진맥진은 중심적인 질이고 가장 명백한 소진 증상이다.
PePe	정서적 소진은 중심적인 질이고 가장 명백한 번아웃 증상이다.
pe	정서적 소진은 번아웃의 중심특성이자 가장 명백한 징후이다.

Table 7: PePe generates post-edited sentences that corrects the grammar errors from the machine-translated outputs.

Word Choice (en→ko)	
src	Hide the layer containing the cutting lines.
mt	절단선이 들어 있는 레이어를 숨긴다.
PePe	커팅 라인이 들어 있는 레이어를 숨긴다.
pe	커팅 라인이 들어 있는 레이어를 숨긴다.
src	The worker explores cultural diversity factors that may be a part of the problem or situation.
mt	노동자는 문제나 상황의 일부일 수도 있는 문화적 다양성 요소를 탐구한다.
PePe	사회복지사는 문제나 상황의 일부일 수도 있는 문화적 다양성 요소를 탐구한다.
pe	사회복지사는 문제나 상황의 일부일 수도 있는 문화적 다양성 요소를 탐구한다.
src	Exemestane is one of the most potent aromatase inhibitors presently available.
mt	exemestane은 현재 사용 가능한 가장 강력한 방향족 억제제 중 하나이다.
PePe	exemestane은 현재 사용 가능한 가장 강력한 aromatase 억제제 중 하나이다.
pe	exemestane은 현재 사용 가능한 가장 강력한 aromatase 억제제 중 하나이다.
src	You do not want them drunk and lazy.
mt	너는 그들이 술에 취해서 게을러지는 것을 원하지 않는다.
PePe	당신은 그들이 술에 취해서 게을러지는 것을 원하지 않는다.
pe	당신은 그들이 술에 취해서 게을러지는 것을 원하지 않는다.
src	By combining the two outputs without the external phase shift, a sum signal is provided for range tracking.
mt	외부 위상 이동 없이 두 출력을 결합함으로써 범위 추적을 위한 합계 신호가 제공된다.
PePe	외부 위상 이동 없이 두 출력을 결합함으로써 거리 추적을 위한 합계 신호가 제공된다.
pe	외부 위상 이동 없이 두 출력을 결합함으로써 거리 추적을 위한 합계 신호가 제공된다.
src	What are my needs for developing my capacity and potentiality?
mt	내 능력과 잠재력을 개발하기 위한 나의 필요성은 무엇인가?
PePe	내 능력과 잠재력을 개발하기 위한 나의 needs는 무엇인가?
pe	내 능력과 잠재력을 개발하기 위한 나의 needs는 무엇인가?

Table 8: PePe changes the words that are not suitable for personal style but are grammatically correct to other candidates, such as synonyms and loanwords.

Politeness (en→ko)	
src	Spend some time looking over the meeting agenda in advance and think about some of the key topics.
mt	회의 안건을 미리 살펴보고 몇 가지 주요 주제에 대해 생각해 보십시오.
PePe	회의 안건을 미리 살펴보고 몇 가지 주요 주제에 대해 생각해 보라.
pe	회의 안건을 미리 살펴보고, 몇 가지 주요 주제에 대해 생각해 보라.
src	She wants the assignment.
mt	그녀는 그 과제를 원한다.
PePe	그녀는 그 과제를 원합니다.
pe	그녀는 그 과제를 원합니다.
src	<u>Watch</u> this video for directions on how to complete the S1 Conversations challenge.
mt	대화 과제를 완료하는 방법은 이 비디오를 참조하십시오.
PePe	대화 과제를 완료하는 방법은 이 비디오를 참조하세요.
pe	대화 과제를 완료하는 방법은 이 비디오를 참조하세요.
src	According to the U.S. Bureau of Census, there are approximately 90 million households in the United States.
mt	미국 인구조사국에 따르면, 미국에는 약 9천만 가구가 살고 있다고 합니다.
PePe	미국 인구조사국에 따르면, 미국에는 약 9천만 가구가 살고 있다고 한다.
pe	미국 인구 조사국에 따르면, 미국에는 약 9천만 가정이 살고 있다고 한다.
src	The store is located inside the Terminal 1.
mt	그 상점은 터미널 1 안에 있다.
PePe	지점은 터미널 1 안에 있습니다.
pe	지점은 터미널 1 내에 있습니다.
src	Maps are also available that show the tract boundaries, making the data readily discernible.
mt	트랙 경계가 표시된 지도도 사용할 수 있어 데이터를 쉽게 식별할 수 있습니다.
PePe	트랙 경계가 표시된 지도도 사용할 수 있어 데이터를 쉽게 식별할 수 있다.
pe	통로 경계를 보여주는지도도 제공되므로 데이터를 쉽게 식별 할 수 있다.

Table 9: PePe controls the level of politeness. The usage of the honorifics varies from language to language.

Multiple Attributes (en→ko)	
src	Our staff will send you back to the airport.
mt	우리 직원이 너를 공항으로 돌려보낼 것이다.
PePe	저희 직원이 공항으로 돌려보낼 것입니다.
pe	저희 직원이 고객님 을 공항으로 데려다 줄 것입니다.
src	When the <u>machine</u> receives the data, it automatically reads the crop marks using a sensor, and then starts cutting .
mt	기계가 데이터를 받으면 자동으로 센서를 이용해 자르기 표시 를 읽은 뒤 절단 을 시작한다.
PePe	장비 가 데이터를 받으면 자동으로 센서를 이용해 crop mark 를 읽은 뒤 커팅 을 시작한다.
pe	장비 가 데이터를 받으면 자동으로 센서를 이용해 crop mark 를 읽은 뒤 커팅 을 시작한다.
src	When transferring major repairs of fixed assets for non-business activities, <u>the following accounts shall be recorded</u> .
mt	비사업활동용 고정자산의 주요수리를 이전할 때에는 다음 사항을 기재하여야 한다.
PePe	비영리활동용 고정자산의 주요수리를 이전할 때에는 다음과 같이 회계처리 하여야 한다.
pe	비영리활동용 고정자산의 주요수리를 이전할 때에는 다음과 같이 회계처리 하여야 한다.
src	The free shuttle bus will come to pick you up around 10 minutes.
mt	무료 셔틀버스가 약 10분 정도 당신 을 데리러 올 것이다.
PePe	무료 셔틀버스가 약 10분 정도 고객님 을 데리러 올 것입니다.
pe	무료 셔틀버스가 약 10분 정도 고객님 을 데리러 올 것입니다.
src	If using 3 crop marks, select 3-point start.
mt	3개의 자르기 표시 를 사용하는 경우 3-point start를 선택하십시오.
PePe	3개의 crop mark 를 사용하는 경우 3-point start를 선택한다.
pe	3개의 crop mark 를 사용하는 경우 3-point start를 선택한다.
src	The following parameters control the display of points-clouds (right).
mt	다음 매개 변수 는 점 구름 (오른쪽) 의 표시를 제어합니다.
PePe	다음 파라미터 는 포인트 클라우드 (오른쪽) 의 표시를 제어합니다.
pe	다음 파라미터 는 포인트 클라우드 (오른쪽) 의 표시를 제어합니다.

Table 10: PePe not only tackles a single attribute but also generates high-quality sentences with multiple attributes revised. Each attribute is colored with a corresponding color.

Error Correction (<i>ko</i> → <i>en</i>)	
<i>src</i>	그래서 전치사 ‘reo’는 ‘to’와 ‘for’의 의미가 있다.
<i>mt</i>	so the prepositions ‘reo’ have the meaning of ‘to’ and ‘for’.
PePe	so the preposition ‘reo’ has the meaning of ‘to’ and ‘for’.
<i>pe</i>	so the preposition ‘reo’ has the meaning of ‘to’ and ‘for’.
<i>src</i>	관사 아래에 있는 모음코드가 이렇게 바뀌어진다.
<i>mt</i>	the vowel code under the official building changes like this.
PePe	the vowel code under the article changes like this.
<i>pe</i>	the vowel code under the article changes like this.
<i>src</i>	professor는 단일한 관리통제 기구보다 상업적 차원, 정부 차원 등 다차원적 모델의 시도를 결합하는 노력이 필요하다고 지적한다.
<i>mt</i>	the processor points out that efforts need to be made to combine attempts by multi-dimensional models such as commercial and government levels rather than single management and control organizations.
PePe	the professor points out that efforts need to be made to combine attempts by multi-dimensional models such as commercial and government levels rather than single management and control organizations.
<i>pe</i>	the professor gasser points out that efforts need to be made to combine attempts by multi-dimensional models such as commercial and government levels rather than single management and control organizations.
<i>src</i>	‘dagesh’가 놓일 수 없다.
<i>mt</i>	‘dagesh’ can’t be let go .
PePe	‘dagesh’ can’t be placed .
<i>pe</i>	‘dagesh’ can’t be placed .

Table 11: PePe generates post-edited sentences that corrects the grammar errors from the machine-translated outputs.

Word Choice (<i>ko</i> → <i>en</i>)	
<i>src</i>	내부배선의 색상은 아래와 같이 구분하여 사용하여야 한다.
<i>mt</i>	the colour of the inner wiring shall be used separately as follows.
PePe	the color of the inner wiring shall be used separately as follows.
<i>pe</i>	the color of the inner wiring shall be used separately as follows.
<i>src</i>	추정 공시가격이 올해 거래된 urgent sale price를 앞서고 있다.
<i>mt</i>	the estimated official price is ahead of the current sales price traded this year.
PePe	the estimated official price is ahead of the urgent sale price traded this year.
<i>pe</i>	the estimated official price is ahead of the urgent sales price traded this year.
<i>src</i>	12월 싱가포르, 말레이시아지역 패키지 상품 판매 확대
<i>mt</i>	expanding sales of package products in singapore and malaysia in december.
PePe	expanding sales of pkg products in singapore and malaysia in december.
<i>pe</i>	expanding sales of pkg products in singapore and malaysia in december.
<i>src</i>	한해 전에 쓰고 남은 돈이 1억2천만원 정도였다.
<i>mt</i>	the remaining money was about 120 million won a year ago.
PePe	the remaining money was about krw 120 million a year ago.
<i>pe</i>	the remaining money was about krw 120 million a year ago.

Table 12: PePe changes the words that are not suitable for personal style but are grammatically correct to other candidates, such as synonyms and loanwords.

Infusing Context and Knowledge Awareness in Multi-turn Dialog Understanding

Ting-Wei Wu

Georgia Institute of Technology
Electrical & Computer Engineering
waynewu@gatech.edu

Biing-Hwang Juang

Georgia Institute of Technology
Electrical & Computer Engineering
juang@ece.gatech.edu

Abstract

In multi-turn dialog understanding, semantic frames are constructed by detecting intents and slots within each user utterance. However, recent works lack the capability of modeling multi-turn dynamics within a dialog in natural language understanding (NLU), instead leaving them for updating dialog states only. Moreover, humans usually associate relevant background knowledge with the current dialog contexts to better illustrate slot semantics revealed from word connotations, where previous works have explored such possibility mostly in knowledge-grounded response generation. In this paper, we propose to amend the research gap by equipping a BERT-based NLU framework with knowledge and context awareness. We first encode dialog contexts with a unidirectional context-aware transformer encoder and select relevant inter-word knowledge with the current word and previous history based on a knowledge attention mechanism. Experimental results in two complicated multi-turn dialog datasets have demonstrated significant improvements of our proposed framework. Attention visualization also demonstrates how our modules leverage knowledge across the utterance.

1 Introduction

In conventional task oriented dialog systems, natural language understanding (NLU) modules aim to transform utterances into meaningful semantic representations for dialog management (Weld et al., 2021; Zhang et al., 2020). It mainly detects associated dialog acts or intents and extracts key slot information as so-called ‘semantic frames’ (Abbeduto, 1983), shown in Table 1. Humans usually associate relevant knowledge and previous contexts with current utterance’s entities to understand an utterance. Similarly, models’ prediction of overall intent semantics and slot values can benefit from act relations such as ‘Inform’ may follow ‘Request’ acts, and background knowledge which is usually

Speaker	Utterance
1. User	Is there something that’s maybe a good intelligent comedy ?
Act & Slots:	<i>Request (genre: comedy)</i> <i>(intelligent; related to; well_informed)</i> <i>(comedy; related to; comic)</i> <i>(comedy; is a; drama)</i>
2. System	Whiskey Tango Foxtrot is the only Adult comedy I see playing in your area . Would you like to try that?
Act & Slots:	<i>Inform (movie: Whiskey Tango Foxtrot)</i> <i>Inform (genre: Adult comedy)</i> <i>Inform (distance limits: in your area)</i> <i>Confirm_question</i> <i>(foxtrot; related to; dance)</i> <i>(foxtrot; related to; rhythm)</i> <i>(adult; capable of; work)</i> <i>(area; is a; region)</i>

Table 1: Excerpt of a single turn within a dialog with corresponding dialog acts, slots and knowledge samples that are related to **keywords** in the utterance.

represented as triples in knowledge graphs (Wang et al., 2021a).

However such intuition has not been emphasized when automating NLU tasks. In early attempts of NLU systems, utterances were isolated and analyzed separately for user intents and semantic slots (Raymond and Riccardi, 2007; Liu et al., 2017). Models that maximize the joint distribution likelihood were proposed to allow transitions between two tasks (Liu and Lane, 2016; Wang et al., 2018; Wu et al., 2021a; Li et al., 2018a). While driven by large pretrained corpora, these methods still fall short of employing complete dynamic interactions within dialogs, especially in multiple intent cases (Qin et al., 2019; Rashmi Gangadharaiah, 2019; Qin et al., 2020). Some works have then integrated dialog contexts for more robust NLU (Wang et al., 2019; Gupta et al., 2019; Su et al., 2021; Wu et al., 2021c). However, many of them could not capture dialog flows well with RNN encoders or explain how contexts should affect the slot filling task.

Publicly available models like BERT or XLNet

provide universal contextualized representations that could be adapted for learning task-oriented contexts. However, it may not give full play to its value when tagging some rare words like *Foxtrot* together with *Tango* as *Movie* in Table 1 that may appear in a domain-specific dataset. One can pretrain these models beforehand emphasizing such phrase relationship which nevertheless tends to be time-consuming and computationally expensive. Therefore, directly integrating external knowledge like a knowledge graph (KG) becomes a more tractable solution (Liu et al., 2019; Zhang et al., 2019b; Wu and Juang, 2022b).

However, there are mainly three challenges lying in the way of such integration: (1) **Heterogeneous information fusion**: the vector space of KG entities is inconsistent with that of the pre-trained models. (2) **Knowledge noise**: overwhelming knowledge for models may adversely cause redundant noises for more ambiguity. Many works in knowledge grounded dialog generation has applied term-level denoising (Zheng et al., 2021) or filtering techniques (Wang et al., 2021b) to refine the adopted knowledge for better semantic considerations. (3) **Inter-token knowledge sharing**: Wang et al. (2019) predicts a slot for a given word along with its own associated knowledge. However, real sentences may contain phrases where knowledge between words should be shared to probably enrich the entire utterance semantics. To overcome these challenges and ground knowledge in contextual NLU, which is less explored in the research community, we propose a **Context and Knowledge Awareness NLU Framework (CKA-NLU)** to effectively incorporate relevant knowledge and dialog history in dialog understanding.

The key ingredients lie in how we can efficiently integrate relevant knowledge and previous history for understanding. We first introduce a context attention module to retrieve context-aware representations. Different from previous works of determining a given word’s slot based on its own knowledge, our objectives require models to aggregate both previous dialog contexts and all intra-sentence knowledge facts together to formulate context-attended knowledge vectors in the same space. Such vectors are a weighted combination of all knowledge facts based on the aggregated information until the current turn. We use attention masks and filtering to remove adversarial effects from redundant knowledge noises. Finally

we adopt these context-attended vectors for NLU tasks with RNN decoders. Experiment results have shown superior performances of our methods that beat all competitive baselines.

Our contributions are as follows:

1. We propose a novel CKA-NLU framework that incorporates inter-word knowledge with inter-sentence contexts to fill the void of relevant knowledge exploration for important NLU tasks.
2. We demonstrate the benefits of adopting knowledge for token-level slot filling and dialog history for sentence-level intent detection.
3. Experimental and attention visualization results show that our model achieves superior performances over several competitive baselines and demonstrates how our model adopts the knowledge.

2 Problem Formulation

For each utterance $x_n = \{w_1^n, w_2^n, \dots, w_T^n\}$ in a task-oriented dialog \mathbf{X} with N utterances, given the domain ontology of a dialog act set \mathbf{A} and a slot set \mathbf{S} , we aim to find one or more acts $\{a_i^n\}$ ¹ and a sequence of slot tags $\{s_1^n, s_2^n, \dots, s_T^n\}$ to construct a semantic frame. Namely, we hope to maximize the joint log likelihood of \mathbf{A} and \mathbf{S} in Eq 1 given a parametrized model θ , its context $\mathbf{C}_n = \{x_1, \dots, x_{n-1}\}$ and associated knowledge $\mathbf{K}_n = \phi(K_G, x_n)$ for the current utterance x_n . We deem K_G as an external large knowledge base with knowledge represented as triples (head h , relation r , tail t) and $\phi(\cdot)$ helps to extract related knowledge pairs for x_n (§3.2.1). It will be critical to match correct knowledge based on current dialog history and the utterance for better dialog understanding.

$$\mathcal{L}(\mathbf{A}, \mathbf{S}) \triangleq \sum_n \log P(A_n, S_n | x_n, \mathbf{C}_n, \mathbf{K}_n; \theta) \quad (1)$$

3 Methodology

3.1 Context Attention

Our overall framework is illustrated in Figure 1. To allow information flow across the dialog, we first encode the entire dialog with a token-level BERT (Devlin et al., 2019) encoder and a turn-level context-aware transformer encoder. Instead of concatenating all sentences which may cause an extreme sequence length, we first generate the token-level representations $\mathbf{H} = \{h_1, h_2, \dots, h_N\}$

¹Dialog acts and intents are equivalent and interchangeably used in this paper.

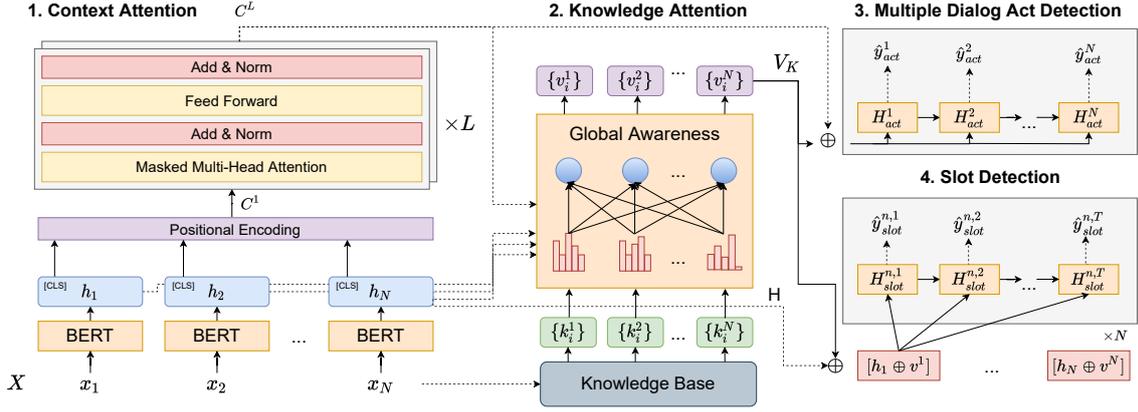


Figure 1: Illustration of our proposed framework for joint dialog act detection and slot filling in multi-turn dialogs. It consists of context and knowledge attention modules, and two LSTM-based decoders. The utterance-level representations will be encoded with the context attention module and token-level representations will interact with their corresponding knowledge in three proposed awareness submodules.

for each utterance x_n in a dialog X by taking vectors from each [CLS] token. During testing at turn n , we may directly reuse these calculated representations $\{h_1, h_2, \dots, h_{n-1}\}$ until turn $n - 1$.

In contrast with other contextual NLU (Wang et al., 2019; Gupta et al., 2019) with hierarchical components, we introduce a GPT-like unidirectional transformer encoder with the hidden size H_a to encode $\mathbf{H} \in \mathbb{R}^{N \times H_b}$. It consists L layers of masked multi-head self-attention (MHA), point-wise feed forward network (FFN), residual sublayer and layer normalization. The future time steps are masked for training since we will not have access to future utterances during testing. We will send \mathbf{H} as the first layer input \mathbf{C}^1 and iteratively encode it with two sublayers in Eq 2. Each head $\mathbf{C}_i \in \mathbb{R}^{N \times (H_a/h)}$ will be first mapped into a query \mathbf{C}^Q , a key \mathbf{C}^K and a value \mathbf{C}^V which participate in the multi-head self-attention. Here $f(\cdot)$ is softmax function. Finally, we will obtain the final contextual dialog representations \mathbf{C}^L .

$$\mathbf{C}^1 = \text{FFN}(\text{MHA}(\mathbf{C}^{1-1}, \mathbf{C}^{1-1}, \mathbf{C}^{1-1})) \quad (2)$$

$$\text{MHA}(\mathbf{C}_i^Q, \mathbf{C}_i^K, \mathbf{C}_i^V) = f\left(\frac{\mathbf{C}_i^Q (\mathbf{C}_i^K)^T}{\sqrt{H_b}}\right) \mathbf{C}_i^V \quad (3)$$

$$\text{FFN}(x) = \max(0, x \mathbf{W}_1 + b_1) \mathbf{W}_2 + b_2 \quad (4)$$

3.2 Knowledge Attention

Humans could naturally associate contexts with relevant knowledge to predict semantics. Here we elaborate on how we can leverage current contexts

$\mathbf{C}^L = \{c_n^L\}$ and a relevant knowledge base K_G to induce the intents and slots for each utterance x_n .

3.2.1 Knowledge extraction

The first step is to gather all necessary knowledge triples $\gamma = \{h, r, t\}$, which are head h and tail t entities with their relation r , related to the current utterance $x_n = \{w_1^n, w_2^n, \dots, w_T^n\}$. For each word w_i^n , we first retrieve a list of triples with the exactly same head entity being w_i^n from a knowledge base K_G . If no head entities are matched, we instead seek entities that has a substring of w_i^n . Each triple in the pretrained K_G (Bordes et al., 2013) has a pre-given relation weight $w_r \in [0, 1]$. For each w_i^n , we select $|K|$ triples that have the largest $|K|$ weights as the final word-level knowledge k_i^n . We will finally obtain a T length knowledge sequence $\mathbf{K}_n = \{k_1^n, k_2^n, \dots, k_T^n\}$ gathered from each word w_i^n . In case of non-alphabetic or out-of-vocabulary (OOV) words with no match in K_G , we instead replace their \mathbf{K}_n as zero vectors to represent agnosticism of knowledge.

3.2.2 Global awareness

To improve the heterogeneous information fusion between contexts and knowledge, after obtaining the knowledge sequence $\mathbf{K}_n = \{k_i^n\}$ (i.e. total $T \times |K|$ triples $\gamma = \{h, r, t\}$), we aim to obtain the context-attended knowledge sequence $\mathbf{V}_K = \{v_i^n\}$ by selecting the most appropriate knowledge (i.e., removing redundant knowledge noise) within the entire sentence, given each word w_i^n and its previous dialog history c_n^L . Different from the term-level denoising like Zheng et al. (2021) and Wang et al. (2019), to allow phrase-level knowledge sharing,

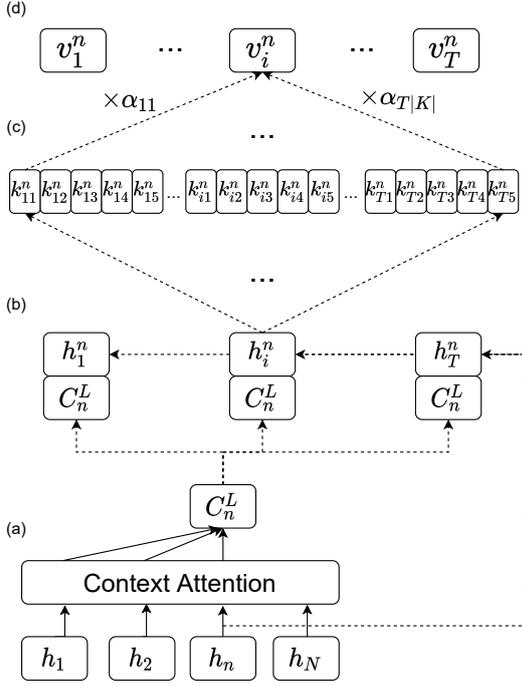


Figure 2: Knowledge Attention Diagram. (a) Context Attention module will first process the dialog history and produce context-aware vectors for each utterance. (b) Token-level representation will be concatenated with the context-aware vector. (c) The fused vector will be used to calculate the attention weights for every knowledge vector in the utterance. (d) The final context-attended knowledge vector will be the weighted combination of all knowledge vectors.

for each word, we aim to globally select all related knowledge in the sentence after seeing previous turns C_n^L . This will allow us to possibly consider knowledge of words in the same phrase.

Shown in Figure 2, we calculate the vector v_i^n where r_{ij}^n , t_{ij}^n are j -th relation and j -th tail entity vectors for the word w_i^n . \mathbf{W}^H , \mathbf{W}^R , \mathbf{W}^T are learnable matrices during training. $[\cdot]$ is the concatenation of two vectors:

$$v_i^n = \sum_{i=1}^T \sum_{j=1}^{|K|} \alpha_{ij} [r_{ij}^n; t_{ij}^n] \quad (5)$$

$$\alpha_{ij} = \exp(\beta_{ij}) / \sum_{i'=1}^T \sum_{j'=1}^{|K|} \exp(\beta_{i'j'}) \quad (6)$$

$$\beta_{ij} = (\tilde{h}_i^n \mathbf{W}^H) (\tanh(r_{ij}^n \mathbf{W}^R + t_{ij}^n \mathbf{W}^T))^T \quad (7)$$

$$\tilde{h}_i^n = [h_i^n; c_n^L] \quad (8)$$

We first concatenate the token-level representations for each word h_i^n in the utterance x_n with its context vector c_n^L , which entails the embedded information from previous turns (Eq 8). Then we use \tilde{h}_i^n

to calculate the attention weight α_{ij} with any of the knowledge (r_{ij}^n, t_{ij}^n) related to this utterance (Eq 6, 7). Eventually, we linearly combine all knowledge vectors together to formalize the context-attended knowledge vector v_i^n (Eq 5). Additionally, to avert the noise from zero-vectors of non-alphabetic word knowledge, we introduce an attention mask to calculate α_{ij} only on the non-zero knowledge vectors.

3.3 Semantic Decoder

After obtaining the context-attended knowledge $\mathbf{V}_K = \{v_i^n\}$, context vectors \mathbf{C}^L and initial token-level vectors \mathbf{H} , we adopt two BiLSTMs to predict multiple dialog acts and slots which exhibit the sequential information in BIO scheme.

$$\mathbf{H}_{\text{act}} = \text{BiLSTM}([\tilde{\mathbf{H}}; \mathbf{V}_K]) \quad (9)$$

$$\mathbf{H}_{\text{slot}} = \text{BiLSTM}([\mathbf{H}; \mathbf{V}_K]) \quad (10)$$

For dialog act detection, we concatenate \mathbf{V}_K with the fused context $\tilde{\mathbf{H}} = ([\mathbf{H}; \mathbf{C}^L]) \mathbf{W}^H$ from the attention mechanism and serve as the inputs of BiLSTM. For slot filling, since the task focuses more on token-level information for decision, we only concatenate raw token-level representations and \mathbf{V}_K to be inputs of another BiLSTM, which empirically works better. Finally, we can generate logits $\hat{y}_{act} = \sigma(\mathbf{H}_{act} \mathbf{W}_{act})$ by transforming \mathbf{H}_{act} with $\mathbf{W}_{act} \in \mathbb{R}^{H_L \times |\mathcal{Y}^a|}$ and a sigmoid function σ . H_L is LSTM hidden size and $|\mathcal{Y}^a|$ is the size of dialog act set. Likewise, we compute $\hat{y}_{slot} = \text{softmax}(\mathbf{H}_{slot} \mathbf{W}_{slot})$. Total loss will be the combination between the binary cross entropy loss based on \hat{y}_{act} and the cross entropy loss based on \hat{y}_{slot} as shown in Eq 11, 12. Finally, the joint objective is formulated as the sum of \mathcal{L}_a and \mathcal{L}_s .

$$\mathcal{L}_a \triangleq - \sum_{n=1}^N \sum_{a=1}^{|\mathcal{Y}^a|} (y_a^n \log(\hat{y}_a^n) + (1 - y_a^n) \log(1 - (\hat{y}_a^n))) \quad (11)$$

$$\mathcal{L}_s \triangleq - \sum_{n=1}^N \sum_{t=1}^T \sum_{s=1}^{|\mathcal{Y}^s|} (y_s^{(n,t)} \log(\hat{y}_s^{(n,t)})) \quad (12)$$

4 Experiment Setting

4.1 Experimental setup

We evaluate our proposed framework on two large-scale dialog datasets, i.e. Microsoft Dialog Challenge dataset (MDC) (Li et al., 2018b) and Schema-Guided Dialog dataset (SGD) (Rastogi et al., 2019). MDC contains human-annotated conversations in

three task-completion domains (movie, restaurant, taxi) with total 11 dialog acts and 50 slots. **SGD** entails large-scale task-oriented dialogs over 20 domains ranging from travel, weather to banks, etc. It has total 18 dialog acts and 89 slots. To compare the relevant knowledge usage in different domains and save computational resources, we randomly select 1k dialogs for each domain in MDC and two restaurant and flights domains from SGD for total 5k dialogs in 6:1:3 train, validation, test ratio. For SGD, Restaurant domain is chosen to compare with that of MDC and Flights domain is the one not existing in MDC. Each utterance is labeled with one or more dialog acts and several slots.

4.2 Baselines

We compare our models with several competitive baselines which sequentially include more features:

- **MID-SF** (Rashmi Gangadharaiyah, 2019) considers joint multi-intent and slot detection in use of BiLSTMs.
- **ECA** (Chauhan A., 2020) encodes the dialog context with LSTM for joint tasks.
- **KANLUM** (Wang et al., 2019) extracts knowledge from the knowledge base and incorporates dialog history for joint tasks.
- **ERNIE** (Zhang et al., 2019b): We take ERNIE backbone to integrate knowledge entities and take the token and entity outputs for intent detection and slot filling directly.
- **LABAN** (Wu et al., 2021b) leverages label information to construct a latent semantic space for utterance projection. It is mainly for the multiple intent detection task only.
- **CASA-BERT** (Gupta et al., 2019) encodes the context with sentence2token and DiSAN which we replace with BERT for fair comparison with other BERT-based models.

We also perform several variations of our proposed framework to conduct the ablation study with the following detailed descriptions.

- **Less-Relevant knowledge triples (LR-KA)**: We replace the top $|K|$ knowledge triples with the less related knowledge triples ranked from $|K| \sim 2|K|$ (from relation weights in K_G) to perform sensitivity analysis on the quality of knowledge.
- **Word-Level knowledge attention (WL-KA)**: We use the attention-based filter (AF) (Wang et al., 2021b) to perform token-level knowledge

attention instead of sentence-level attention in our framework.

- **Transformer decoder (Trans)**: We replace the semantic decoder (§ 3.3) with a transformer decoder to both predict dialog acts and slots.

4.3 Implementation details

We adopt the pretrained **BERT_{base}** (Devlin et al., 2019) as our utterance encoder. Context attention transformer has $L = 6$ -layer attention blocks with 768 head size and 4 attention heads. The max sequence length is 60. We use ConceptNet knowledge base (Speer et al., 2018) to obtain relevant knowledge for attention. It involves many crowd-sourced and expert-created resources like DBpedia, OpenCyc and WordNet with 1.5M word entities connected with weighted edges (relation). Each word or relation is represented as a dense 100-dim vectors by adopting TransE (Bordes et al., 2013) learning mode. Each knowledge also contains an ExternalURL to represent the external source. We retrieve $|K| = 5$ most related knowledge from each word based on weights assigned on the edges. Both LSTMs have 256 hidden units. We use the batch size of 2 dialogs for MDC and 1 for SGD. In all training, we use Adam optimizer with learning rate as $5e-5$. The best performance on validation set is obtained after training 30 epochs on each model. For metrics, we report the dialog act accuracy (exact match) and slot filling F1 score. Here we only consider a true positive when all BIO values for a slot is correct and forfeit ‘O’ tags.

5 Main Results

5.1 Main results

Table 2 shows our main results on the joint task performance. MID-SF with only LSTMs has relatively inferior performance on both datasets especially in SGD. ECA by taking dialog contexts into consideration has much greater increase in SGD than in MDC. ERNIE and KANLUM have better slot filling performance which suggests the importance of further knowledge induction. Leveraging BERT-based encoder seems to substantially increase semantic visibility in ERNIE, CASA-BERT and our proposed framework, while introducing dialog contexts additionally gives better dialog act detection performance in CASA-BERT and our model. Eventually, our proposed framework beats all baselines both in MDC and substantially in SGD, by more

Dataset	MDC						SGD			
Domain	Movie		Restaurant		Taxi		Restaurant		Flights	
Model	MDA	SL	MDA	SL	MDA	SL	MDA	SL	MDA	SL
MID-SF (Rashmi Gangadharaiah, 2019)	76.56	67.56	77.35	65.77	85.03	70.03	74.26	81.38	84.74	84.48
ECA (Chauhan A., 2020)	77.10	69.72	77.56	66.85	86.61	71.28	87.98	84.87	95.16	87.91
KANLUM (Wang et al., 2019)	81.86	73.32	80.76	68.36	88.31	74.07	86.81	87.82	92.87	90.05
ERNIE (Zhang et al., 2019b)	81.52	79.18	80.60	74.68	87.72	76.85	88.53	91.37	89.33	90.50
LABAN (Wu et al., 2021b)	82.05	-	82.28	-	88.19	-	90.51	-	94.23	-
CASA-BERT (Gupta et al., 2019)	84.22	79.59	83.17	74.89	90.00	78.54	92.54	94.20	95.00	91.79
CKA-NLU	86.09[†]	80.58[†]	84.01[†]	75.27[†]	90.80[†]	79.60[†]	98.47[†]	94.86	99.22[†]	92.67[†]

Table 2: Experimental Results on several NLU models including our proposed frameworks which are specified in percentage (%). MDA indicates the dialog act detection accuracy by counting corrects when all acts are predicted correctly. SL indicates the slot filling F1 score. † indicates the significant improvement of p-value < 0.05, compared with CASA-BERT.

Dataset	MDC						SGD			
Domain	Movie		Restaurant		Taxi		Restaurant		Flights	
Model	MDA	SL								
CKA-NLU	86.09	80.58	84.01	75.27	90.80	79.60	98.47	94.86	99.22	92.67
w/ LR-KA	85.63	80.26	83.43	75.76	89.77	80.03	98.38	94.31	98.93	91.99
w/ WL-KA	85.25	79.46	83.27	74.89	90.05	79.59	96.84	94.61	97.17	91.14
w/ Trans	85.98	79.94	83.27	75.19	90.40	78.33	97.35	94.34	98.20	91.95
w/o KG	86.01	79.92	83.53	74.76	90.56	78.29	97.53	94.83	97.73	92.23
w/o CA	84.87	79.79	81.33	74.68	89.00	78.50	95.88	94.36	97.17	91.94
w/o LSTM	84.57	79.14	82.70	74.35	89.65	79.00	90.96	93.64	94.80	91.33

Table 3: Ablation Results of joint tasks (%) by removing some key components of our proposed model: CKA-NLU.

efficiently incorporating external knowledge and dialog contexts with the proposed global awareness attention mechanism.

5.2 Ablation analysis

To better estimate the effectiveness of each module of our best model, we conduct ablation experiments in Table 3. We ablate or replace each component from CKA to observe the performance drops. First, we could see knowledge quality may affect the performance of joint tasks where most performance drops are observed with LR-KA, while we found that slot accuracy may increase if the overall extracted knowledge is less relevant to utterances. To note, the word matching accuracies in the knowledge base are 78.12% (MDC) and 80.97% (SGD), which indicates that there is still about 20% of zero vectors introduced as redundant noises. Second, considering global knowledge across the entire sentence has overall better performance than only word-level knowledge, where knowledge of some phrases should be treated jointly. Finally, we see a single transformer decoder may still entangle the act and slot information by updating gradients simultaneously with poorer performance.

By removing the entire knowledge attention module, we could see a larger accuracy decrease in slot filling tasks, denoting the necessity of external knowledge in enriching the current word representations. By substituting a LSTM on top of BERT

for our context attention module (CA), we obtain poorer performance in dialog act detection. By replacing two LSTMs with fully connected layers after knowledge attention, the performance drops especially in SGD. Overall, we observe dialog act detection relies more on contexts while slot filling tasks may concentrate on inter-utterance relations where external knowledge benefits more instead.

5.3 Further Discussion

Could knowledge amend the data scarcity? We also study how knowledge could contribute to the joint tasks when resources are scarce. Figure 3 shows the performance changes with different numbers of training data. We found that inducing the knowledge will have the positive effect on both tasks. In the few-shot setting, we see the performance difference enlarges where knowledge becomes beneficial to enrich the external information aside from data itself. However, knowledge becomes less useful when we have extreme low dataset particularly for slot detection in MDC. Introducing more MDC data at a certain point may contradict with the external knowledge data base that possibly makes models hard to generalize, while it helps dialog act detection that amends the training instability from data scarcity.

Does global knowledge help non-alphabetic slots? We are interested if knowledge for other words would also help with the slot prediction of

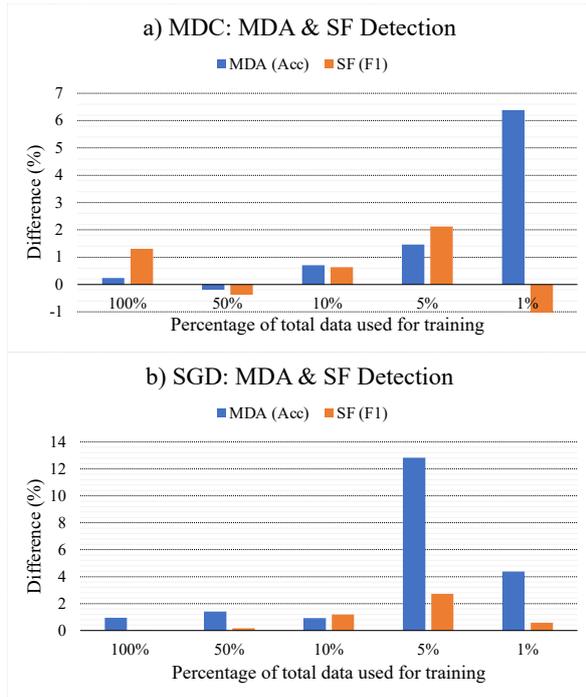


Figure 3: NLU performance gain by using knowledge in CKA-NLU with a subsample (%) of the original training data of two datasets: MDC and SGD.

non-alphabetic words. Table 4 shows the results for each non-alphabetic slot for our global and local attention models. Since there is no knowledge for the non-alphabetic words, we observe an overall 2% increase by inducing global attention. Contexts are beneficial especially for slots associated with rating, money and address, which should be likely inferred by other keywords near them. However, introducing more knowledge noises may not help to predict time and zip code since they are rather independent to contexts.

5.4 Knowledge Attention

In Figure 4, we visualize the attention heatmap of tokens with their slot labels vs. all knowledge triples from each token. First, we focus on the rows of the heat map. Without attached knowledge for the words like numbers or punctuations, their attention weights are perceived blank across all tokens in the utterance. Second, for valid attention weights, we found the knowledge corresponding to keywords like ‘you’, ‘with’, ‘restaurant’ and ‘antioch’ are most adopted for overall knowledge representations across all the utterance. It elucidates that the model will mostly grasp knowledge in words especially tagged as valued slots (non-O tag) for overall semantic understanding. Interest-

Slot	CKA-NLU (%)	WL-KA (%)	Δ (%)
address	17.39	0.00	+17.39
price	66.67	50.00	+16.67
critic_rating	34.48	23.08	+11.41
dress_code	50.00	44.44	+5.56
rating	52.17	49.32	+2.86
cost	95.54	95.29	+0.26
numberofpeople	95.63	95.51	+0.12
date	86.96	86.99	-0.02
pricing	42.55	43.14	-0.58
starttime	76.80	77.68	-0.88
numberofkids	73.68	77.78	-4.09
mpaa_rating	76.92	83.33	-6.41
zip_code	77.65	84.44	-6.80
pickup_time	75.19	82.29	-7.09
total	65.83	63.80	+2.03

Table 4: F1 scores of non-alphabetic slots in overall SGD dataset when using all (CKA-NLU) or word-level (WL-KA) knowledge.

ingly, this collection of knowledge is more emphasized on predicting a word to be non-valued than those words with valued slots. For the columns, we could see for non-valued words, they will rely on knowledge of valued words like ‘restaurant’ and ‘antioch’, than the knowledge related to itself. It substantiates the belief that the overall semantics of the utterance may be driven by these valued words.

In Table 5, we further show an utterance example with some highlighted words including ‘you’, ‘restaurant’ and ‘Antioch’ with their extracted knowledge and weights for semantic detection. We take the average of all attention weights across all tokens for that knowledge triple; then normalized across the knowledge triples in the same word (head). We could see ‘you’ as an object is most adopted to clarify the user being offered and informed counts. Then we observe that the knowledge triple (*restaurant, atl, city*) where *restaurant is at a location of the city* is most recognized to illustrate the relations of restaurant and city tags. Finally, knowledge for ‘Antioch’ keyword is mostly relevant to a country which is conducive when the system seldom sees this word during training. But without further contexts, our model believes ‘Antioch’ is more of a part of Turkey.

6 Related Work

Intent detection and slot filling are two main NLU tasks (Weld et al., 2021). Many classification or clustering approaches (Sarıkaya et al., 2011; Raymond and Riccardi, 2007; Liu et al., 2017; Wu and Juang, 2022a) had been proposed for single intent detection. However, treating two tasks separately may experience error propagation. Liu and Lane

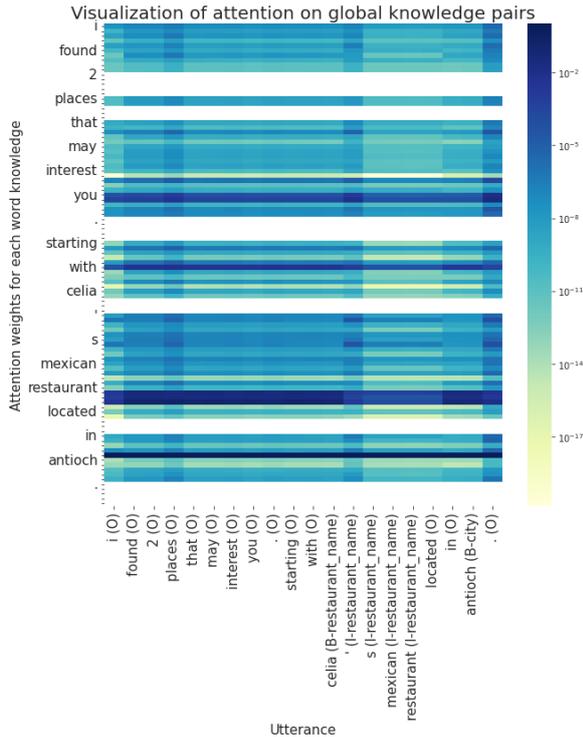


Figure 4: Attention visualization of a single utterance example with respect to all knowledge related to each word. We denote an utterance with tokens followed by their predicted tag in x-axis. For y-axis, each word will have five knowledge triples with each as a single tick. The blank area is where attention weights are zero.

(2016) first proposed an attention-based LSTM network to model the correlations between intents and slots. Li et al. (2018a) proposed the gating mechanism for better self-attention on joint tasks, which is not scalable for longer sequences. Wang et al. (2018) instead proposed the bi-model to directly model the cross impacts and Zhang et al. (2019a) utilized capsule neural networks. Memory networks are also popular choices to model long-range dependency (Wu et al., 2021a). However, a single utterance may have many intents. Qin et al. (2019) proposed a stack-propagation networks to predict intents on each token. Rashmi Gangadhariah (2019) and Qin et al. (2020) considered the dynamic interactions between two tasks by jointly detecting multiple intents. Wu et al. (2021b) extended the multiple intent scenario with zero-shot cases. These methods nevertheless restrict their resources to current utterances for prediction where we consider the multi-turn dialogs jointly where dialog acts could be context-sensitive (Bothe et al., 2018).

Contexts and knowledge Contexts are also crit-

Utterance Example in Figure 4	
Utterance	I found 2 places that may interest you. Starting with Celia's Mexican restaurant located in Antioch.
Dialog acts	Offer, Inform Count
Slots	O O O O O O O O O B-res I-res I-res O O B-city
Keyword	Knowledge
you	(hc, noun) (0.29), (hc, object) (0.7) (rel, guys) (6e-4)
restaurant	(isa, establishment) (8e-9), (atl, hotel) (0.2) (atl, town) (0.14), (atl, city) (0.65)
Antioch	(rel, orontes) (4e-5), (rel, swiss) (2e-2) (rel, usa) (5e-2), (ptof, turkey) (0.9)

Table 5: The utterance example in Figure 4 for joint task prediction. Knowledge (Relation, Tail) related to three keywords as head are presented with their attention weights (number after the knowledge). We only show the top four knowledge adopted for each keyword based on the attention weights. ‘hc’ represents ‘has context’, ‘rel’ represents ‘related to’, ‘atl’ represents ‘at location’ and ‘ptof’ represents ‘part of’.

ical for dialog understanding. Bertomeu et al. (2006) first studied the contextual phenomena in words. Bhargava et al. (2013) and Shi et al. (2015) then introduced contextual signals to the joint intent-slot tasks. Advanced hierarchical structures are also emphasized to encode multi-turn dialog contexts efficiently (Chauhan A., 2020; Wang et al., 2019; Gupta et al., 2019; Wu et al., 2021c). Knowledge is also another important resource to induce commonsense for understanding. It is widely adopted for knowledge-enhanced pretraining to enrich representations (Liu et al., 2019; Zhang et al., 2019b). In task-oriented dialogs, main emphasis lies in the interaction with task-related knowledge bases (Madotto et al., 2020; Yang et al., 2020). Most of works also focus on open-domain dialog response generation (Zhao et al., 2020; Wang et al., 2021b; Rashkin et al., 2021; Zheng et al., 2021) or task-specific responses (Wang et al., 2021a). However, commonsense knowledge is seldom adopted in NLU. Wang et al. (2019) tried to apply knowledge in NLU but it is not suitable for complex dialog modeling. To amend the gap in modeling such knowledge and context interactions, we follow these previous works’ paradigms and explore the mechanisms of characterizing their mutual effects.

7 Conclusion

In this paper, we propose a novel BERT-based knowledge-augmented network to effectively incorporate dialog history and external knowledge in the joint NLU tasks. Compared to recent works

which consider only intra-word knowledge, we instead raise the knowledge awareness by selecting all relevant knowledge triples in an utterance with the current dialog contexts. We found that our framework is verified to be effective in two complex multi-turn dialog datasets where contexts and knowledge are crucial in dialog act detection and slot filling respectively. The visualization shows that our models adopt some key knowledge in particular words and learn to grasp useful information for better interpretability. These context-attended knowledge vectors could be easily applied to downstream dialog state tracking or management tasks.

Limitations

The possible limitations for our works are two-folds. First, the scalability of our method is subject to the size of the knowledge base and the number of incorporated knowledge since selecting from larger knowledge candidates may require more computational memory and training time but with higher performance. Exact string matching between context words and knowledge entities is relatively simple and could be replaced with more advanced semantic matching techniques, which nevertheless may increase model complexity. Second, depending on the domains of datasets to apply, too many out-of-vocabulary words (OOV) with no match in the knowledge base may affect the model performance and our future works will investigate a better solution to replace zero-vectors that are associated with non-alphabetic words.

References

- Leonard Abbeduto. 1983. *Linguistic communication and speech acts*. kent bach, robert m. harnish. cambridge: M.i.t. press, 1979, pp. xvii 327. *Applied Psycholinguistics*, 4(4):397–407.
- Núria Bertomeu, Hans Uszkoreit, Anette Frank, Hans-Ulrich Krieger, and Brigitte Jörg. 2006. *Contextual phenomena and thematic relations in database QA dialogues: results from a Wizard-of-Oz experiment*. In *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 1–8, New York, NY, USA. Association for Computational Linguistics.
- A. Bhargava, A. Celikyilmaz, D. Hakkani-Tür, and R. Sarikaya. 2013. *Easy contextual intent prediction and slot detection*. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8337–8341.
- A Bordes, N Usunier, A Garcia-Duran, J Weston, and O Yakhnenko. 2013. *Translating embeddings for modeling multi-relational data*. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.
- Chandrakant Bothe, Cornelius Weber, Sven Magg, and Stefan Wermter. 2018. *A context-based approach for dialogue act recognition using simple recurrent neural networks*. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Singh A. Arora J. Shukla S. Chauhan A., Malhotra A. 2020. *Encoding context in task-oriented dialogue systems using intent, dialogue acts, and slots*. In *Saini H., Sayal R., Buyya R., Aliseri G. (eds) Innovations in Computer Science and Engineering. Lecture Notes in Networks and Systems, vol 103. Springer, Singapore*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *Bert: Pre-training of deep bidirectional transformers for language understanding*.
- Arshit Gupta, Peng Zhang, Garima Lalwani, and Mona Diab. 2019. *Casa-nlu: Context-aware self-attentive natural language understanding for task-oriented chatbots*.
- Changliang Li, Liang Li, and Ji Qi. 2018a. *A self-attentive model with gate mechanism for spoken language understanding*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3824–3833, Brussels, Belgium. Association for Computational Linguistics.
- Xiujun Li, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018b. *Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems*. *arXiv preprint arXiv:1807.11125*.
- Bing Liu and Ian Lane. 2016. *Attention-based recurrent neural network models for joint intent detection and slot filling*.
- Ting Liu, Xiao DING, Yue QIAN, and Yiheng CHEN. 2017. *Identification method of user’s travel consumption intention in chatting robot*. *SCIENTIA SINICA Informationis*, 47:997.
- Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019. *K-bert: Enabling language representation with knowledge graph*.
- Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. *Learning knowledge bases with parameters for task-oriented dialogue systems*.
- Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. *A stack-propagation framework with token-level intent detection for spoken language understanding*.

- Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. [Agif: An adaptive graph-interactive framework for joint multiple intent detection and slot filling.](#)
- Hannah Rashkin, David Reitter, Gaurav Singh Tomar, and Dipanjan Das. 2021. [Increasing faithfulness in knowledge-grounded dialogue with controllable features.](#)
- Balakrishnan Rashmi Gangadharaiah. 2019. [Joint multiple intent detection and slot labeling for goal-oriented dialog.](#) Proc. of NAACL.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.
- Christian Raymond and Giuseppe Riccardi. 2007. [Generative and discriminative algorithms for spoken language understanding.](#) In *Proc. Interspeech 2007*, pages 1605–1608.
- Ruhi Sarikaya, Geoffrey E. Hinton, and Bhuvana Ramabhadran. 2011. [Deep belief nets for natural language call-routing.](#) In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5680–5683.
- Yangyang Shi, Kaisheng Yao, Hu Chen, Yi-Cheng Pan, Mei-Yuh Hwang, and Baolin Peng. 2015. Contextual spoken language understanding using recurrent neural networks.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2018. [Conceptnet 5.5: An open multilingual graph of general knowledge.](#)
- Ruolin Su, Ting-Wei Wu, and Biing-Hwang Juang. 2021. [Act-Aware Slot-Value Predicting in Multi-Domain Dialogue State Tracking.](#) In *Proc. Interspeech 2021*, pages 236–240.
- Qingyue Wang, Yanan Cao, Junyan Jiang, Yafang Wang, Lingling Tong, and Li Guo. 2021a. [Incorporating specific knowledge into end-to-end task-oriented dialogue systems.](#) In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.
- Yanmeng Wang, Ye Wang, Xingyu Lou, Wenge Rong, Zhenghong Hao, and Shaojun Wang. 2021b. [Improving dialogue response generation via knowledge graph filter.](#) In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7423–7427.
- Yu Wang, Yilin Shen, and Hongxia Jin. 2018. [A bi-model based rnn semantic frame parsing model for intent detection and slot filling.](#)
- Yufan Wang, Tingting He, Rui Fan, Wenji Zhou, and Xinhui Tu. 2019. [Effective utilization of external knowledge and history context in multi-turn spoken language understanding model.](#) In *2019 IEEE International Conference on Big Data (Big Data)*, pages 960–967.
- H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han. 2021. [A survey of joint intent detection and slot-filling models in natural language understanding.](#)
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing.](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Jie Wu, Ian Harris, and Hongzhi Zhao. 2021a. [Spoken language understanding for task-oriented dialogue systems with augmented memory networks.](#) In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 797–806, Online. Association for Computational Linguistics.
- Ting-Wei Wu and Biing Juang. 2022a. [Induce Spoken Dialog Intents via Deep Unsupervised Context Contrastive Clustering.](#) In *Proc. Interspeech 2022*, pages 1081–1085.
- Ting-Wei Wu and Biing-Hwang Juang. 2022b. [Knowledge augmented bert mutual network in multi-turn spoken dialogues.](#) In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7487–7491.
- Ting-Wei Wu, Ruolin Su, and Biing Juang. 2021b. [A label-aware BERT attention network for zero-shot multi-intent detection in spoken language understanding.](#) In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4884–4896, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ting-Wei Wu, Ruolin Su, and Biing-Hwang Juang. 2021c. [A Context-Aware Hierarchical BERT Fusion Network for Multi-Turn Dialog Act Detection.](#) In *Proc. Interspeech 2021*, pages 1239–1243.
- Shiquan Yang, Rui Zhang, and Sarah Erfani. 2020. [GraphDialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems.](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1878–1888, Online. Association for Computational Linguistics.
- Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S. Yu. 2019a. [Joint slot filling and intent detection via capsule neural networks.](#)
- Zheng Zhang, Ryuichi Takanobu, Qi Zhu, Minlie Huang, and Xiaoyan Zhu. 2020. [Recent advances and challenges in task-oriented dialog system.](#)

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. [Ernie: Enhanced language representation with informative entities](#).

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. [Knowledge-grounded dialogue generation with pre-trained language models](#).

Wen Zheng, Natasa Milic-Frayling, and Ke Zhou. 2021. [Knowledge-grounded dialogue generation with term-level de-noising](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2972–2983, Online. Association for Computational Linguistics.

A Additional Experimental Setting

We use huggingface transformers (Wolf et al., 2020) to implement our framework and we use two Nvidia 2080Ti GPUs for all model training. The number of model parameters is around 146M. It takes 30 minutes to train 30 epochs for a single model.

MCoNaLa: A Benchmark for Code Generation from Multiple Natural Languages

Zhiruo Wang^{*♦} Grace Cuenca^{*♦} Shuyan Zhou[♦] Frank F. Xu[♦] Graham Neubig^{♦♦}

[♦]Carnegie Mellon University [♦]Princeton University [♦]Inspired Cognition
{zhiruow, shuyanzh, fangzhex, gneubig}@cs.cmu.edu, gcuenca@princeton.edu

Abstract

While there has been a recent burgeoning of applications at the intersection of natural and programming languages, such as code generation and code summarization, these applications are usually English-centric. This creates a barrier for program developers who are not proficient in English. To mitigate this gap in technology development across languages, we propose a multilingual dataset, MCoNaLa, to benchmark code generation from natural language commands extending beyond English. Modeled off of the methodology from the English Code/Natural Language Challenge (CoNaLa) dataset, we annotated a total of 896 NL-Code pairs in three languages: Spanish, Japanese, and Russian. We present a systematic evaluation on MCoNaLa by testing state-of-the-art code generation systems. Although the difficulties vary across three languages, all systems lag significantly behind their English counterparts, revealing the challenges in adapting code generation to new languages.¹

1 Introduction

There are an increasing number of applications related to “code intelligence”, such as code summarization (Allamanis et al., 2016; Hu et al., 2018; Ahmad et al., 2020) and natural language (NL) to code generation (Ling et al., 2016; Rabinovich et al., 2017; Yin et al., 2018a; Xu et al., 2020; Norouzi et al., 2021; Wang et al., 2021), accompanied by code-specific tasks and benchmarks (Oda et al., 2015; Zhong et al., 2017; Yin et al., 2018b; Lu et al., 2021). However, in the cases where these benchmarks include natural language, that language is almost invariably English.

There are a few exceptions, but most of them either focus on languages of specific domains (Sherborne and Lapata, 2021; Sherborne et al., 2020;

^{*}Equal contribution.

¹Code and data are available at <https://github.com/zorazrw/multilingual-conala>

Spanish	¿Cómo sumar el campo 'precio' de todos los elementos del modelo 'Precompra' en Django? (How to sum the 'precio' field of all the elements of the 'Precompra' model in Django?) <code>totaldos = Precompra.objects.aggregate(Sum(precio)).values()[0]</code>
Japanese	2次元配列arrの要素となっている1次元配列から先頭の値のみを抜き出す (Extract only the first value from the 1D array that is the element of the 2D array 'arr') <code>arr[0]</code>
Russian	Установить кодировку 'my_encode' для переменных окружения пользователя 'username' (Set 'my_encode' encoding for 'username' environment variables) <code>os.environ('username').decode(my_encode)</code>

Figure 1: Examples in the MCoNaLa dataset, that aim to generate general-purpose Python code snippets from source intent of multiple natural languages.

Moradshahi et al., 2020) or types of code (Oda et al., 2015; Liang et al., 2021), or contain NL intents collected via automatic translation (Li et al., 2021) (Appendix A). However, similarly to how Kwiatkowski et al. (2019) argue that “natural questions” are necessary to appropriately benchmark QA systems, we argue that ensuring the naturalness and coverage of questions is essential for benchmarking code generation systems as well.

A dataset for English code generation based on natural programming questions is the CoNaLa dataset (Yin et al., 2018a). It is based on natural developer questions harvested from the Stack Overflow (SO) question answering forum. In fact, in addition to English, SO also supports four other languages (Spanish, Portuguese, Japanese, and Russian) that have strong developer communities and engage in non-English programming environments. In this work, we utilize this resource to construct the MCoNaLa dataset, consisting of 341, 210, and 345 manually curated parallel samples with natural intents in Spanish, Japanese, and Russian, along with corresponding Python code snippets. Like CoNaLa, these snippets are collected from language-specific SO sites and annotated by na-

tive speakers who are also proficient in the Python programming language.

To provide insights in the state of code generation on this new resource, we conduct comprehensive experiments with three state-of-the-art text generation models in the context of cross-lingual transfer, by unifying training and testing NL via translation (Ruder and Sil, 2021; Shi et al., 2021; Shima and Mitamura, 2010; Hartrumpf et al., 2008), or utilizing a multilingual NL encoder such as MBART (Liu et al., 2020). Our results suggest that cross-lingual NL-to-Code generation is challenging. Among all languages and experiment settings, the highest average BLEU score is 7.28, far behind that of English, which achieves 33.41, presumably because English resembles Python more than other NLs. In addition, we find models with task-specific modules and training outperform generic seq2seq models, yet the discrepancy between languages are consistent across all baseline models. In all, our corpus and experiments demonstrate the varied difficulty of the NL-to-Code generation task under different languages, emphasizing the need to develop a language-comprehensive approach to code intelligence.

2 The MCoNaLa Dataset

2.1 Task Definition

Concerning the task of answering natural language questions with machine-executable programs, our focus is to build a benchmark dataset to evaluate models for their ability to encode NL *intents* in multiple languages and generate code *snippets*. For each example in Figure 1, the *intent* above asks how to achieve a particular goal, and the *snippet* below responds with a piece of Python code.

2.2 Annotation Workflow

Our goal is to collect *intent-snippet* parallel data in multiple natural languages. In this section, we outline the main workflow for data annotation: (1) language source selection, (2) valid SO post identification, and (3) parallel sample annotation.

Language source and selection Besides the English version, Stack Overflow also has forums available in four other languages: Spanish, Portuguese, Japanese, and Russian. Data annotation in each language requires a native speaker of that language, who should also be proficient in both English and Python. Due to the high cost and difficulty of hiring

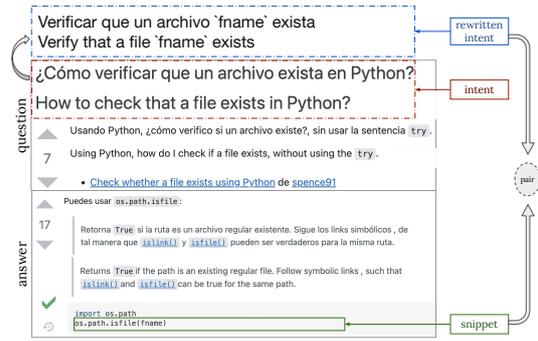


Figure 2: Illustration of the annotation process.

reliable annotators with such a specialized skill set, we only employ one Upwork annotator for each of Spanish, Japanese, and Russian. From the official SO data dump² dated March 2021, we obtained all posts in these languages. However, we were unsuccessful in finding a Portuguese-speaking annotator at the time of corpus collection.

Identifying how-to questions Following Yin et al. (2018a), annotators are first asked to identify valid posts that contain how-to type questions, which are imperative utterances seeking particular goals achievable by code. They are often in the post title or description, such as the example in Figure 2.

Posts are sent in 100-sample batches, and then categorized by annotators. To improve annotation efficiency, we bootstrapped a MBART how-to question classifier, with English examples, then iteratively multilingual samples. It achieves an accuracy of 72.50%. We then automatically filter the probable invalid posts using this classifier and designate the rest for manual annotation. We collect all valid posts and extract questions as raw intents, for subsequent parallel data annotation.

Collecting intent-snippet pairs For each post, we ask the annotators to find at most three snippets of Python code that correctly answer the extracted question. However, questions from post title or description are often ambiguous, especially in respective context of answer snippet, such as the example in Figure 2, that the question does not specify the names of “data” and “list” variables to allow precise code implementation. To disambiguate the intent and align it with a snippet, we ask annotators to rewrite the intent by: (1) specifying variable names appearing in the answer snippet, and (2) clarifying commands with reference question descriptions. Concretely, variable names and data types in the rewritten intent

²<https://archive.org/details/stackexchange>

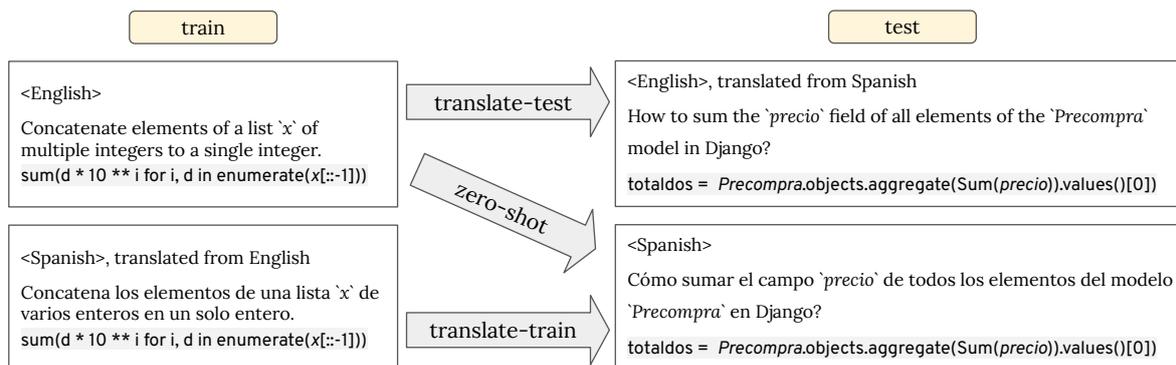


Figure 3: Example usage on the original English and Multilingual samples in three settings.

need to be surrounded by the ASCII grave accent marks (e.g., ``data``), string literals or file paths should use singular typographic quotation marks (e.g., `'file1.txt'`, `'https://www.abc.com/'`).

The final MCoNaLa dataset consists of 341, 210, and 345 intent-snippet pairs in Spanish, Japanese, and Russian. It is noteworthy that the goal of MCoNaLa is to benchmark cross-lingual NL-to-Code generation task and mainly for testing purposes, instead of curating large-scale dataset for training. While its size is relatively small given the collection difficulty, we show that it can reliably inform significant method improvements (§ 3.3). We believe it is important for our dataset to be representative of the naturally occurring questions in respective language environments.

2.3 Quality Analysis

To ensure high data quality as intended, we checked 15 random samples from each language subset. Each rater score NL intents and code snippets from 1 to 5 based on their correctness and specificity.

The results demonstrate the high quality of our dataset, achieving 4.78, 4.65, 4.78 points on Spanish, Japanese, and Russian intents; and 4.84, 4.89, 4.78 points on their corresponding code snippets. Meanwhile, three raters present high agreement – the Fleiss’ Kappa measure is 64.29 for NL intents and 69.49 for code snippets – both numbers indicate substantial agreement among the raters.

3 Method

To provide insights about evaluating on MCoNaLa, we demonstrate potential dataset usage in three train-test settings (§ 3.1), and propose to adapt three baseline models from either multilingual (NL) or code understanding to achieve both ends (§ 3.2).

Because the size of MCoNaLa allows only testing purposes, we resort to its larger English counter-

part, CoNaLa (Yin et al., 2018a), to allow training. CoNaLa contains 2,879 manually annotated samples and 600k samples extracted from English SO forum and API documents, which can serve as a sufficient source for training. Given this usage, we denote the three test languages as *target* languages.

3.1 Train-Test Settings

We adopt three settings from two paradigms (Hu et al., 2020) as illustrated in Figure 3: (1) translating intents in train (*translate-train*) or test (*translate-test*) sets to bridge the language gap, (2) using multilingual encoder to transfer from English to target languages (*zero-shot*).

For each target language, we can align the languages of training and testing intents and use a monolingual encoder. The *translate-train* setting translates English intents in CoNaLa to each target language for training and then tests with MCoNaLa samples. *translate-test* translates MCoNaLa intents in three target languages into English. Because it is not feasible to manually translate 600K+ intents, we use existing multilingual machine translation (MMT) models to automate translation. We benchmarked several open-source options, as elaborated in § 4.2, and settled on the M2M-124 model used on the FLORES-101 dataset (Goyal et al., 2022).

Also, we can train models on English samples and directly evaluate on MCoNaLa samples in target languages *zero-shot*, requiring models to encode multiple NLs, further, transfer the code generation ability from English context to target ones.

3.2 Baseline Models

We introduce three baseline methods targeting the above train-test settings. We encourage readers to refer to the original papers for more details.

In a monolingual context, models should function in target languages for *translate-train* and En-

glish for *translate-test*. TRANX (Yin and Neubig, 2018) is a BiLSTM-based encoder-decoder model that uses a transition-based abstract syntax parser to map NLs into formal meaning representations (MR) such as Python programs. It is agnostic to input languages and hence can be evaluated on both translated settings. TAE (Norouzi et al., 2021) is the state-of-the-art method on CoNaLa by training a generic transformer with an added target autoencoder (TAE) objective. However, it is built with (English-)BERT and is intended for English scenarios, therefore only tested on *translate-test*.

As is required by *zero-shot* evaluation, we adopt a multilingual model, MBART (Liu et al., 2020), which is a seq2seq model pre-trained on 25 natural languages including our target ones. Note that MBART can also function in monolingual contexts, for both *translate-train* and *translate-test* settings.

3.3 Experiment

We train baseline models in their available settings, then tokenize the generated and reference code snippets following Yin and Neubig (2018) to evaluate the BLEU-4 scores. We report the average scores of five rounds using different random seeds.

Model	Setting	Language				
		en	es	ja	ru	avg.
MBART	translate-test		2.38	3.07	2.04	2.50
	translate-train	25.20	2.64	3.45	2.65	2.91
	zero-shot		2.49	1.83	2.28	2.20
TRANX	translate-test	32.26	2.46	8.34	8.12	6.31
	translate-train		2.44	6.11	6.02	4.86
TAE	translate-test	33.41	2.39	9.90	9.56	7.28

Table 1: BLEU scores of baselines for various train-test settings in English (en) and target languages (es, ja, ru).

In Table 1, first, scores on target languages average to at most 7.28, much lower than 33.41 on English, revealing the similarity of English and Python, and the difficulty of generating code from other languages. Second, models with code-specific designs and training (TRANX and TAE) performs better in general. The lower scores of MBART potentially suggest a certain representation gap between NL and PL. Third, results on two code-specific models show consistent variations across languages: scores are lower for Spanish, but rise similarly on Japanese and Russian. As we will discuss in § 4.1, this is possibly due to the distributional gap between languages with varied complexity.

3.4 Significance Test

To verify the effectiveness of MCoNaLa, we perform significance tests (Dror et al., 2018) to show its capability of showing significant differences between systems. We conduct paired bootstrap resampling tests with each pair of models in their available settings, using a sample rate of 0.5 and a sample size of 10,000.

Setting	Language	Win Rate (%)			Tie	p-value
		MBART	TRANX	TAE		
translate-test	es	0.532	0.402	-	0.066	0.468
		0.522	-	0.396	0.102	0.478
		-	0.508	0.448	0.044	0.492
	ja	0.000	1.000	-	0.000	0.000
		0.000	-	1.000	0.000	0.000
		-	0.002	0.998	0.000	0.002
	ru	0.000	1.000	-	0.000	0.000
		0.000	-	1.000	0.000	0.000
		-	0.001	0.998	0.001	0.002
translate-train	es	0.592	0.408	-	0.000	0.408
	ja	0.000	1.000	-	0.000	0.000
	ru	0.000	1.000	-	0.000	0.000

Table 2: Significance testing results between each pair of baseline models. ‘-’ marks the model not in the pair.

In both *translate-test* and *translate-train* settings of Table 2, code-specific systems (TRANX and TAE) significantly outperform MBART on Japanese and Russian. However, no significant differences are shown in Spanish, as expected given its relative difficulty. With significance testing, one can obtain reliable results even on a small dataset. While small sizes are not entirely desirable for informative evaluation, we view them as practical reflections of data scarcity, supporting our call for more non-English resources.

4 Analysis

4.1 Variations between Languages

We first study the differences in size and snippet length between languages subsets in MCoNaLa. As listed in Table 3, snippet lengths vary across languages, and the average snippet length in Spanish is around 2.5/1.3 times of that in Japanese/Russian. A longer snippet is presumably more complex, suggesting that snippets in Spanish samples are harder to generate, and hence models perform poorer.

4.2 Intent Auto-translation

In § 3.1 we use MMT models for intent translation. To optimize translation quality, we compare three best performing MMT models: OPUS-MT (Tiedemann and Thottingal, 2020), M2M-

original intent (English)	Prepend string 'hello' to all items in list 'a'
translated intent (Spanish)	Preparación (<u>prepare</u>) de la cadena 'hello' a todos los elementos en la lista 'a'
snippet	['hello(0)'.format(i) for i in a]
original intent (English)	add a colorbar to plot 'plt' using image 'im' on axes 'ax'
translated intent (Japanese)	画像'im'を使ってax'の軸にカラーバーを追加
snippet	plt.colorbar(im, ax=ax)
original intent (English)	extend dictionary 'a' with key/ value pairs of dictionary 'b'
translated intent (Russian)	расширить словарь 'a' с ключевыми/ значительными (<u>significant</u>) парами словаря 'b'
snippet	a.update(b)

Figure 4: Examples showing that the translation errors or omits critical words in the original intent.

Language	Size	# Snippet Tokens		
		average	max	min
English	2,879	18.2	170	2
Spanish	341	42.6	343	4
Japanese	210	17.7	94	2
Russian	345	32.0	243	3

Table 3: Data size and snippet length (in number of tokens) of MCoNaLa samples between target languages.

100 (Fan et al., 2021), and M2M-124 used in FLORES-101 (Goyal et al., 2022). Since comparing in *translate-train* needs intensive re-training with different model translations, we ablate in the *translate-test* setting, using each model to translate test intents and evaluate NL-to-Code respectively.

Baseline	MMT	Language		
		Spanish	Japanese	Russian
MBART	M2M-124	2.38	3.08	2.04
	OPUS-MT	2.28	3.21	2.46
	M2M-100	1.83	2.79	2.00
TRANX	M2M-124	2.46	8.41	8.09
	OPUS-MT	2.46	5.09	5.00
	M2M-100	2.04	7.38	8.48
TAE	M2M-124	2.39	9.88	9.57
	OPUS-MT	3.15	3.89	5.30
	M2M-100	2.21	8.20	9.32

Table 4: Comparing MMT models under *translate-test*.

As in Table 4, their results are close, but M2M-124 tends to be more stable across languages and baselines. Despite its relative superiority, its translation quality may still lag behind human performance, with more examples in § 4.3.

4.3 Quality of Auto-translation

To better measure the quality of translated intents, we manually check the semantic alignment be-

tween the original and translated intents, with the assistance of the Google Translate API and dictionaries. Concretely, we take 20 English CoNaLa intents and check if their semantics preserve after being translated into three target languages (*translate-train*). We similarly examine 20 MCoNaLa intents in each target language and check their English translations (*translate-test*). We use the M2M-124 translations given its best results. As shown in Figure 4, MMT translations are still sub-optimal: often mis-translate, even omit, the key words. This is especially severe on verbs that indicate certain Python operations. Hence, the translation step may impair intent-snippet alignment, being one of the major factors to the poor results in translated settings.

5 Conclusion

In this work, we extend the task of NL-to-Code generation from English-centric to multilingual scenarios. We establish the MCoNaLa benchmark that contains NL intent and code snippet pairs available in Spanish, Japanese, and Russian. Our benchmark serves for the multilingual code generation task, requiring models of both multilingual understanding and code synthesis. We conduct systematic experiments on three baseline models and show varying difficulty across languages and settings. We hope to reveal the necessity to develop, and serve as a solid test bed for language-comprehensive approaches regarding code intelligence.

Acknowledgements

We thank all the annotators for their hard work. This work was supported by the National Science Foundation under grant number 1815287.

Limitations

Although the MCoNaLa dataset makes a first step to include more natural languages aside from English, it is currently limited to the languages supported by the StackOverflow forum, since SO provides the source data for the MCoNaLa creation. This can be mitigated by extending to more languages using programming forums in other languages that have a similar purpose to SO. Besides, MCoNaLa dataset only supports literal evaluation methods such as BLEU. Given the executable nature of Python programs, it is beneficial to support more evaluation metrics such as functional correctness, robustness, and conciseness.

Ethics Statement

The MCoNaLa dataset is built to serve as a testbed for evaluating code generation systems from natural languages extending beyond English, given that an English-centric setting can harm universal accessibility to language technologies.

We hire annotators who are proficient in target languages and assist them with clearly documented instructions, flexible annotation interfaces (e.g., Google Sheets), and automated methods (e.g., using a neural classifier to filter out possibly invalid cases) to optimize the annotation efficiency. We carefully check in line with our instructions and standards, to ensure the quality of both the question posts given and the annotation results back from our annotators. We emphasize the differences between samples in different languages, because they are natural reflections of the questions that programmers asked in each specific language, similar to many works in fields such as multilingual question answering (Clark et al., 2020) and named entity recognition (Nothman et al., 2013). We reckon that it is of paramount importance to evaluate on data that was originally produced in the target language, and results may be less reliable otherwise.

Nevertheless, with the advances in models capable of generating code from natural language inputs, we should be aware of the potentially harmful usage such as concealing malicious code (Wallace et al., 2020), or generating code with security vulnerabilities (Verdi et al., 2020; Pearce et al., 2021).

References

- Wasi Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2020. [A transformer-based approach for source code summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4998–5007, Online. Association for Computational Linguistics.
- Miltiadis Allamanis, Hao Peng, and Charles Sutton. 2016. [A convolutional attention network for extreme summarization of source code](#). In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2091–2100. JMLR.org.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. [Program synthesis with large language models](#). *ArXiv preprint*, abs/2108.07732.
- Shaunak Chatterjee, Sudeep Juvekar, and Koushik Sen. 2009. Sniff: A search engine for java using free-form queries. In *International Conference on Fundamental Approaches to Software Engineering*, pages 385–400. Springer.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. [Evaluating large language models trained on code](#). *ArXiv preprint*, abs/2107.03374.
- Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. [TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages](#). *Transactions of the Association for Computational Linguistics*, 8:454–470.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Jacob Devlin. 2018. Multilingual bert readme. <https://github.com/google-research/bert/blob/master/multilingual.md>.
- Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker’s guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at](#)

- scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’ Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.
- Sven Hartrumpf, Ingo Glöckner, and Johannes Leveling. 2008. Efficient question answering with question decomposition and multiple answer streams. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 421–428. Springer.
- Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. 2021. [Measuring coding challenge competence with apps](#). *ArXiv preprint*, abs/2105.09938.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. [XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.
- Xing Hu, Ge Li, Xin Xia, David Lo, Shuai Lu, and Zhi Jin. 2018. [Summarizing source code with transferred API knowledge](#). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 2269–2275. ijcai.org.
- Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2016. [Summarizing source code using a neural attention model](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2073–2083, Berlin, Germany. Association for Computational Linguistics.
- Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2018. [Mapping language to code in programmatic context](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1652, Brussels, Belgium. Association for Computational Linguistics.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Haoran Li, Abhinav Arora, Shuohui Chen, Ankit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. [MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.
- Qingyuan Liang, Zeyu Sun, Qihao Zhu, Wenjie Zhang, Lian Yu, Yingfei Xiong, and Lu Zhang. 2021. [Lyra: A benchmark for turducken-style code generation](#). *ArXiv preprint*, abs/2108.12144.
- Wang Ling, Phil Blunsom, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Fumin Wang, and Andrew Senior. 2016. [Latent predictor networks for code generation](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 599–609, Berlin, Germany. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, et al. 2021. [Codexglue: A machine learning benchmark dataset for code understanding and generation](#). *ArXiv preprint*, abs/2102.04664.
- Mehrad Moradshahi, Giovanni Campagna, Sina Semnani, Silei Xu, and Monica Lam. 2020. [Localizing open-ontology QA semantic parsers in a day using machine translation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5970–5983, Online. Association for Computational Linguistics.
- Dana Movshovitz-Attias and William W. Cohen. 2013. [Natural language models for predicting programming comments](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 35–40, Sofia, Bulgaria. Association for Computational Linguistics.
- Sajad Norouzi, Keyi Tang, and Yanshuai Cao. 2021. [Code generation from natural language with less prior knowledge and more monolingual data](#). In *Proceedings of the 59th Annual Meeting of the Association for*

- Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 776–785, Online. Association for Computational Linguistics.
- Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.
- Yusuke Oda, Hiroyuki Fudaba, Graham Neubig, Hideaki Hata, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Learning to generate pseudo-code from source code using statistical machine translation. In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 574–584. IEEE.
- Sebastiano Panichella, Jairo Aponte, Massimiliano Di Penta, Andrian Marcus, and Gerardo Canfora. 2012. Mining source code descriptions from developer communications. In *2012 20th IEEE International Conference on Program Comprehension (ICPC)*, pages 63–72. IEEE.
- Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. 2021. An empirical cybersecurity evaluation of github copilot’s code contributions. *arXiv preprint arXiv:2108.09293*.
- Chris Quirk, Raymond Mooney, and Michel Galley. 2015. [Language to code: Learning semantic parsers for if-this-then-that recipes](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 878–888, Beijing, China. Association for Computational Linguistics.
- Maxim Rabinovich, Mitchell Stern, and Dan Klein. 2017. [Abstract syntax networks for code generation and semantic parsing](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1139–1149, Vancouver, Canada. Association for Computational Linguistics.
- Sebastian Ruder and Avi Sil. 2021. [Multi-domain multilingual question answering](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 17–21, Punta Cana, Dominican Republic & Online. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Tom Sherborne and Mirella Lapata. 2021. [Zero-shot cross-lingual semantic parsing](#). *ArXiv preprint*, abs/2104.07554.
- Tom Sherborne, Yumo Xu, and Mirella Lapata. 2020. [Bootstrapping a crosslingual semantic parser](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 499–517, Online. Association for Computational Linguistics.
- Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2021. [Cross-lingual training with dense retrieval for document retrieval](#). *ArXiv preprint*, abs/2109.01628.
- Hideki Shima and Teruko Mitamura. 2010. Bootstrap pattern learning for open-domain CLQA. In *Proceedings of NTCIR-8 Workshop Meeting*.
- Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. [Leveraging monolingual data with self-supervision for multilingual neural machine translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2827–2835, Online. Association for Computational Linguistics.
- Jörg Tiedemann and Santhosh Thottingal. 2020. [OPUS-MT – building open translation services for the world](#). In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.
- Morteza Verdi, Ashkan Sami, Jafar Akhondali, Foutse Khomh, Gias Uddin, and Alireza Karami Motlagh. 2020. An empirical study of c++ vulnerabilities in crowd-sourced code examples. *IEEE Transactions on Software Engineering*.
- Eric Wallace, Tony Z Zhao, Shi Feng, and Sameer Singh. 2020. Concealed data poisoning attacks on nlp models. *arXiv preprint arXiv:2010.12563*.
- Yue Wang, Weishi Wang, Shafiq Joty, and Steven C.H. Hoi. 2021. [CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8696–8708, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Edmund Wong, Taiyue Liu, and Lin Tan. 2015. Clocom: Mining existing source code for automatic comment generation. In *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, pages 380–389. IEEE.
- Edmund Wong, Jinqiu Yang, and Lin Tan. 2013. Auto-comment: Mining question and answer sites for automatic comment generation. In *2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 562–567. IEEE.
- Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. [Generalized data](#)

- augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796, Florence, Italy. Association for Computational Linguistics.
- Frank F. Xu, Zhengbao Jiang, Pengcheng Yin, Bogdan Vasilescu, and Graham Neubig. 2020. *Incorporating external knowledge through pre-training for natural language to code generation*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6045–6052, Online. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. *mT5: A massively multilingual pre-trained text-to-text transformer*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Ziyu Yao, Daniel S. Weld, Wei-Peng Chen, and Huan Sun. 2018. *Staqc: A systematically mined question-code dataset from stack overflow*. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1693–1703. ACM.
- Pengcheng Yin, Bowen Deng, Edgar Chen, Bogdan Vasilescu, and Graham Neubig. 2018a. *Learning to mine aligned code and natural language pairs from stack overflow*. In *International Conference on Mining Software Repositories, MSR*, pages 476–486. ACM.
- Pengcheng Yin, Bowen Deng, Edgar Chen, Bogdan Vasilescu, and Graham Neubig. 2018b. *Learning to mine aligned code and natural language pairs from stack overflow*. In *2018 IEEE/ACM 15th international conference on mining software repositories (MSR)*, pages 476–486. IEEE.
- Pengcheng Yin and Graham Neubig. 2018. *TRANX: A transition-based neural abstract syntax parser for semantic parsing and code generation*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 7–12, Brussels, Belgium. Association for Computational Linguistics.
- Alexey Zagalsky, Ohad Barzilay, and Amiram Yehudai. 2012. *Example overflow: Using social media for code recommendation*. In *2012 Third International Workshop on Recommendation Systems for Software Engineering (RSSE)*, pages 38–42. IEEE.
- Victor Zhong, Caiming Xiong, and Richard Socher. 2017. *Seq2sql: Generating structured queries from natural language using reinforcement learning*. *ArXiv preprint*, abs/1709.00103.

A Related Work

Natural Language to Code Generation Datasets

There have been several benchmark datasets for NL-to-Code generation, such as Hearthstone (Ling et al., 2016), Django (Oda et al., 2015), CONCODE (Iyer et al., 2018), and CoNaLa (Yin et al., 2018a). Other examples include datasets for problem solving, such as HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and APPS (Hendrycks et al., 2021). A number of methods have been proposed to mine intent-snippet pairs for the purpose of code search, summarization, or generation. While our work falls in the line of mining from SO (Wong et al., 2013; Iyer et al., 2016; Yao et al., 2018; Yin et al., 2018b), other work also attempts to exploit other data sources such as API documentation (Chatterjee et al., 2009; Movshovitz-Attias and Cohen, 2013; Xu et al., 2020), code comments (Wong et al., 2015), specialized sites (Quirk et al., 2015), and developer communications (Panichella et al., 2012). One prior methodology to automatically collect large-scale parallel data is using heuristics to extract intent-snippet pairs (Chatterjee et al., 2009; Wong et al., 2013; Zagalsky et al., 2012), but this often results in compromised data quality (Xu et al., 2020). Our work resorts to a manual annotation strategy that often yields accurately aligned intent-snippet pairs.

Multilingual Learning While the bulk of code-related tasks have their NL components in English, program developers native in other languages cannot enjoy the advances in code intelligence techniques, leading to the current lacunae in multilingual learning. Our work intends to mitigate this gap by facilitating NL-to-Code generation in multiple languages beyond English. To enable language understanding across multiple languages, a number of works propose to train language models with corpus in multiple languages (Devlin, 2018; Liu et al., 2020; Conneau et al., 2020; Xue et al., 2021). In addition to multilingual training, other data augmentation techniques commonly used in machine translation (MT), such as back-translation (Edunov et al., 2018), monolingual (Sennrich et al., 2016; Siddhant et al., 2020) or generalized data augmentation (Xia et al., 2019), also inspired our experiments. However, these techniques have rarely been utilized for NL-conditioned code generation. We present preliminary attempts in the experiments.

Augmenting pre-trained language models with audio feature embedding for argumentation mining in political debates

Rafael Mestre[†], Stuart E. Middleton, Matt Ryan, Masood Gheasi,
Timothy J. Norman and Jiatong Zhu

University of Southampton

[†]r.mestre@soton.ac.uk

Abstract

The integration of multimodality in natural language processing (NLP) tasks seeks to exploit the complementary information contained in two or more modalities, such as text, audio and video. This paper investigates the integration of often under-researched audio features with text, using the task of argumentation mining (AM) as a case study. We take a previously reported dataset and present an audio-enhanced version (the Multimodal USElecDeb60To16 dataset). We report the performance of two text models based on BERT and GloVe embeddings, one audio model (based on CNN and Bi-LSTM) and multimodal combinations, on a dataset of 28,850 utterances. The results show that multimodal models do not outperform text-based models when using the full dataset. However, we show that audio features add value in fully supervised scenarios with limited data. We find that when data is scarce (e.g. with 10% of the original dataset) multimodal models yield improved performance, whereas text models based on BERT considerably decrease performance. Finally, we conduct a study with artificially generated voices and an ablation study to investigate the importance of different audio features in the audio models.

1 Introduction

In recent years, there has been an increasing interest in multimodal classification, which refers to the task of automatically classifying an input based on multiple modalities or sources of information, such as text, images and audio (Baltrušaitis et al., 2018). Multimodal approaches are beneficial as they can reduce the subjectivity of classification with a single modality and improve the accuracy of the overall classification. However, finding the best representations (especially those that work well with other modalities), aligning and fusing them, and getting the models to co-learn are difficult challenges to overcome (Morency and Baltrušaitis, 2017). A large body of literature has

focused on the combination of image and text for applications like emotion recognition (Illendula and Sheth, 2019), fake news detection (Nakamura et al., 2020), image classification (Guillaumin et al., 2010), or document image classification (Jain and Wigington, 2019). Much less attention has been paid to combining audio with text.

Audio can convey a variety of information about the pitch or intonation of the speaker that can indicate variance in emotional state as well as better identify modes of communication like sarcasm that have been difficult for models to detect. The integration of audio has successfully improved classification tasks like multimodal sentiment analysis and emotion recognition when compared with classic NLP models (Yao et al., 2020; Ho et al., 2020). In this paper, we focus on a less explored NLP area in terms of multimodality: argumentation mining (AM). AM is the computational study of arguments to develop models that can automatically identify, extract, and represent arguments in text or other forms of digital communication such as audio or video. AM has traditionally focused on textual data such as news articles, blog posts, and online comments, but the advantages of using audio to detect arguments have not been extensively explored.

In this work, we expand on an existing AM dataset of US political debates (USElecDeb60To16 by Haddadan et al. 2019) by including audio. We test the performance of several multimodal AM models in different variations of the same dataset, e.g., after balancing the labels and with fractional datasets. Our contribution is three-fold: i) a new fully aligned audio dataset, expanding on an existing AM dataset (Section 3), adding balanced and fractional subsets for researchers to experiment with; ii) original multimodal benchmarking results for this dataset highlighting where audio feature embeddings add most value compared to text-only models (Section 4); iii) analysis of audio features importance, including performance comparison of

human and computerized voices (Section 5) and an ablation study (Section 6).

2 Related work

Multimodal approaches including audio have been mostly used for sentiment analysis or emotion recognition (Yang et al., 2022; Cai et al., 2019), often using the IEMOCAP dataset, one of the oldest datasets that contains 12 hours of dialogue recordings with emotion labels in text, audio and video format (Busso et al., 2008). In recent years newer datasets have been released, such as the SAVEE (Jackson and Haq, 2014) and RAVDESS databases (Livingstone and Russo, 2018), the MELD dataset (Poria et al., 2018), and the CNU-MOSEI dataset (Zadeh et al., 2016). Generally, audio-textual multimodal approaches contain separate pipelines for audio and text features, sometimes connected through attention layers. For instance, Cai et al. (2019) combined GloVe embeddings in a bidirectional long-short term memory (Bi-LSTM) array for text with a combination of a convolutional neural network (CNN) and a Bi-LSTM array for the audio. Likewise, Yoon et al. (2018) used GloVe embeddings and recurrent neural networks (RNN) for both audio and text, reaching accuracies of 71.8 % with the IEMOCAP dataset. Atmaja and Akagi (2020) used either LSTM or CNN (not at the same time) for the acoustic pipeline and LSTM with Fast-Text and GloVe embeddings. Ho et al. (2020) used a multi-level multi-head fusion attention using bidirectional encoder representations (BERT) for the text representations, achieving improved accuracies in three different datasets.

Audio features in this domain have been generally embedded using low level descriptors (LLDs), such as mel-frequency cepstral coefficients (MFCCs) (Atmaja and Akagi, 2020; Ho et al., 2020). MFCCs are computed from the mel spectrogram of the audio signal by performing a discrete cosine transform (DCT) of its log to reduce its dimensionality in a way that is highly related to the raw signal, but approximating the human auditory system and often yielding higher classification performance (Singh et al., 2021). As LLDs do not contain global information about the utterance, high-level statistical functions (HSFs), such as mean, kurtosis and quadratic error, among many others, can also be used. Yao et al. (2020) compared the performance in speech emotion recognition of a HSF classifier based on a deep neural

network (DNN), a LLS classifier based on a recurrent neural network (RNN) and a raw-signal mel-spectrogram classifier based on a CNN, finding similar performance between the HSF and LLD models, and a slightly lower performance for the model using the raw signal, showing the benefits of the low level representations. The use of RNN with LLDs has been shown to offer benefits by considering the temporal dimension of an utterance (Xie et al., 2019), but several researchers have started to use both CNNs and RNNs in combination to learn both temporal and local features in the frequency domain (Zhao et al., 2019; Singh et al., 2021; Yao et al., 2020). Whereas MFCCs are the feature of choice for the great majority of applications, the list of remaining LDDs are virtually endless, including the zero crossing rate, chroma vector, entropy of energy, Hammarberg index, spectral slope, harmonic difference, among many others (Atmaja and Akagi, 2020). While some efforts have been made towards standardization of audio features (Eyben et al., 2016), the choice is generally pragmatic and depends on the package used by the researcher, with openSMILE toolkit (Eyben et al., 2013), Librosa (McFee et al., 2015) and PyAudioAnalysis (Giannakopoulos, 2015) being those most commonly chosen ones.

On the other hand, AM research has focused on a diverse set of applications using the text modality alone, from online interactions (Ghosh et al., 2014) and tweets (Alsinet et al., 2019) to argumentative essays (Stab and Gurevych, 2014) and political debates (Lawrence and Reed, 2017; Visser et al., 2021). Regarding multimodal AM, Lippi and Torroni (2016) presented a first step towards the use of audio features from speech to improve argument detection. In this paper, they used raw input signals, which were passed through a speech recognition API to obtain the text. Then they used bag of words and bi-grams together with discrete HSF features from MFCCs, namely minimum, maximum, average and standard deviation, to train a support vector machine in an argument classification task. The results were positive towards the addition of audio, although the performance was modest due to the small size of the dataset and the limitations of the text and audio representations. The only other work that considered multimodal aspects used the M-arg dataset (Mestre et al., 2021). There, the authors analyzed argumentative relations in the 2016 US presidential debates using text and audio, building an

argumentation mining pipeline based on BERT embeddings for text and a combination of a Bi-LSTM and a CNN for the audio. Although the dataset, annotated for "support" and "attack" between sentences, was rather small and heavily unbalanced towards the "neither" class, the authors reported a slight improvement when considering audio and text together in a multimodal model. Surprisingly, audio features alone showed a better performance than the text-only model based on BERT encodings, suggesting that in small datasets, when the performance of BERT-based models suffers, audio features might provide a handy supplement to classify arguments. The effect of specific audio features on performance was not assessed.

Here, we build upon a previous dataset (USElecDeb60To16) presented by [Haddadan et al. \(2019\)](#), which contained English transcripts of the US presidential debates from 1960 to 2016 labelled with more than 29k annotations of argument components and their boundaries. We used the original videos from the debates to obtain aligned timestamps at the sentence level following the work of [Mestre et al. \(2021\)](#), thus enabling the task of multimodal AM with a total of 28,850 aligned and annotated sentences. Concurrently to the submission of our work, [Mancini et al. \(2022\)](#) also presented and released a multimodal dataset, using the same videos and alignment process, with 26,791 sentences. Both datasets are complementary, although our dataset is slightly larger as we did not drop any of the debate videos (see next section). [Mancini et al. \(2022\)](#) compare datasets and architectures from the two previously mentioned works by [Lippi and Torroni \(2016\)](#) and [Mestre et al. \(2021\)](#), as well as their new dataset, finding generally positive results to the addition of audio. In our work, we report an audio feature analysis, as well as the impact of using computerized voices, and we investigate the benefit of multimodal models on both balanced and fractional small data subsets.

3 Methodology

3.1 Dataset construction

In their USElecDeb60To16 dataset, [Haddadan et al. \(2019\)](#) reported the performance of several models for argument classification, with the highest weighted F-score of 0.673 for the argumentative component classification (ACC) of premise, claim and other. They also collapsed all the premise/claim annotations into one single label,

"argument", and attempted argumentative sentence detection (ASD), with a weighted F-score of 0.843 using an LSTM array. We used this dataset in its collapsed version (argument/other) for ASD to assess whether the addition of audio could improve the reported performance (F-score of 0.843) and to simplify the task using only 2 classes as a first step to studying the potential of multimodal AM. For this, we needed to add the audio of the debates with sentence-level timestamps.

Videos from each debate were downloaded from the YouTube channel of the Commission for Presidential Debates.¹ Before starting the audio alignment process, we fixed a small number of inconsistencies in the dataset resulting from errors in the original transcripts. Some were simple, like sentences lacking a space between periods, which made sentence tokenization algorithms fail. In a couple of debates, full paragraphs were missing from the transcript, possibly due to an error in the original web scraping algorithm by [Haddadan et al. \(2019\)](#). Older debates also had serious transcription issues in the original source, such as full sentences or paragraphs missing or speeches being repeated twice in the transcript. Regarding videos, older ones also had issues, such as debate 5 (the first Carter-Ford Debate in 1976), in which the audio was lost during live transmission, and commentators, not presidential candidates, spoke for almost half an hour. This was not reflected in the transcript, and we had to manually edit the video to match the transcript. For two debates (the first and second Clinton-Bush-Perot debates of 1992), the Commission decided to split the transcript into two parts, even though the debates occurred uninterrupted. Therefore, we split the videos in two to match the transcript. Others had cuts or repeated segments that lasted from a few seconds to several minutes, and we were forced to adapt the original dataset to reflect these changes. Whereas preceding researchers [Mancini et al. \(2022\)](#) were forced to remove full or part of the debates from their multimodal dataset to account for these issues, we rigorously edited the USElecDeb60To16 dataset and videos to reduce unsystematic data loss error, such that we could provide an enhanced comprehensive dataset to researchers for further investigation. We want to highlight that this error reduction does not cast any doubt on the quality and substantive find-

¹<https://www.debates.org/voter-education/debate-transcripts/>

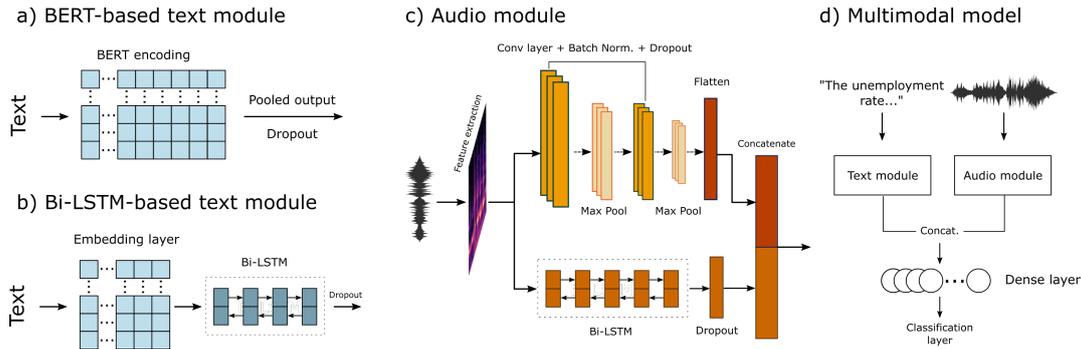


Figure 1: Model architectures used in this paper. a-c) Generic architectures of the text and audio modules. d) Multimodal model that combines text and audio modules.

ings of that preceding work, as it mostly relates to sentence omission.

We aligned the transcripts with the audio using the Aeneas Forced Alignment (v. 1.7.3) tool as proposed by Mestre et al. (2021). Two researchers manually checked every debate for major misalignment (and fixed them) until we obtained an almost perfectly aligned text. After alignment, our dataset contained 28,850 labelled sentences (76.15% of them arguments), with timestamps indicating start and ending times in the audio file. We present the extended dataset we call Multimodal USElecDeb60To16, with the collapsed original annotations, alongside timestamps to match the audio, instructions for obtaining the videos and scripts to extract the audio features, in our GitHub repository page (Mestre et al., 2023).²

3.2 Model architectures

We used the architectures proposed by Mestre et al. (2021), which showed the potential of multimodal argumentation mining in our dataset (Figure 1). We considered two text modules based on GloVe and BERT (Devlin et al., 2018). The former used Wikipedia-trained 200-dimensional GloVe embeddings, whose maximum length was given by its 99th percentile to eliminate very long sentences. They were passed through a Bi-LSTM, followed by a dropout layer, a dense layer and an output layer with softmax activation. The latter module consisted of a BERT pre-processor³ and a BERT encoder with L=12 hidden layers, a size of H=768 and A=12 attention heads.⁴ Its pooled output was also followed by a dropout layer and a dense layer.

²<https://github.com/rafamestre/Multimodal-USElecDeb60To16>.

³bert_en_uncased_preprocess v.3

⁴bert_en_uncased_L-12_H-768_A-12 v.4

The audio module was inspired by Cai et al. (2019). For each utterance in audio form, the Python library Librosa was used for audio feature extraction (McFee et al., 2015). We extracted the following LLD features: MFCCs (Klapuri and Davy, 2006), spectral centroids (Klapuri and Davy, 2006), spectral bandwidth (Klapuri and Davy, 2006), spectral roll-off (McFee et al., 2015), spectral contrast (Jiang et al., 2002a), and a 12-bit chroma vector (McFee et al., 2015). Motivation for selection of features and evaluation is further described in Section 6. For each sentence, the features were concatenated to form a tensor of $(45, T)$, where T is the duration of the utterance. All utterances were padded with zeros to have the same length T_{max} , which was defined by the 99th percentile duration of all utterances. Each utterance was passed in parallel through a CNN and a Bi-LSTM to find both local and temporal features. The CNN consisted of two convolutional layers, two maxpool layers and batch normalization layers. Outputs from both modules were flattened, concatenated and passed through dropout and dense layers.

The multimodal model was a combination of the text and audio modules in which the inputs were the text string and its corresponding audio, each passed in parallel. We considered two multimodal models: one with a BERT text module and another one with a Bi-LSTM text module.

3.3 Hyperparameter tuning

We developed a robust methodological framework to tune the hyperparameters for each model. Model training was performed in a High Performance Computing (HPC) cluster in dedicated GPUs (with either nodes of 4 GTX1080 Ti GPUs or nodes of 2 Nvidia Volta V100 GPUs). The hyperparam-

Model	Class	Original dataset ($N = 28, 850$)			Balanced dataset ($N = 13, 758$)		
		Precision	Recall	F_1	Precision	Recall	F_1
Text Bi-LSTM	Argument	0.844 ± 0.005	0.950 ± 0.011	0.893 ± 0.004	0.707 ± 0.011	0.789 ± 0.037	0.745 ± 0.012
	Other	0.727 ± 0.028	0.429 ± 0.034	0.539 ± 0.023	0.761 ± 0.020	0.669 ± 0.039	0.711 ± 0.017
	Wt. average	0.816 ± 0.006	0.827 ± 0.005	0.810 ± 0.006	0.734 ± 0.007	0.730 ± 0.007	0.729 ± 0.007
	Macro av.	0.785 ± 0.013	0.690 ± 0.013	0.716 ± 0.012	0.734 ± 0.007	0.729 ± 0.007	0.728 ± 0.007
Text BERT	Argument	0.854 ± 0.004	0.951 ± 0.006	0.900 ± 0.002	0.714 ± 0.013	0.839 ± 0.025	0.771 ± 0.011
	Other	0.758 ± 0.018	0.487 ± 0.018	0.593 ± 0.012	0.813 ± 0.017	0.674 ± 0.024	0.737 ± 0.010
	Wt. average	0.831 ± 0.004	0.839 ± 0.003	0.826 ± 0.004	0.764 ± 0.007	0.755 ± 0.006	0.754 ± 0.006
	Macro av.	0.806 ± 0.008	0.719 ± 0.007	0.746 ± 0.006	0.763 ± 0.007	0.757 ± 0.005	0.754 ± 0.006
Audio	Argument	0.785 ± 0.011	0.973 ± 0.028	0.869 ± 0.005	0.628 ± 0.032	0.517 ± 0.279	0.521 ± 0.204
	Other	0.654 ± 0.081	0.135 ± 0.080	0.211 ± 0.089	0.603 ± 0.063	0.670 ± 0.211	0.612 ± 0.064
	Wt. average	0.754 ± 0.011	0.775 ± 0.003	0.714 ± 0.017	0.615 ± 0.017	0.595 ± 0.036	0.567 ± 0.078
	Macro av.	0.720 ± 0.035	0.554 ± 0.026	0.540 ± 0.042	0.615 ± 0.016	0.593 ± 0.034	0.566 ± 0.080
Multimodal (Bi-LSTM +Audio)	Argument	0.879 ± 0.031	0.672 ± 0.255	0.733 ± 0.184	0.765 ± 0.054	0.548 ± 0.259	0.593 ± 0.230
	Other	0.451 ± 0.127	0.681 ± 0.173	0.515 ± 0.053	0.661 ± 0.087	0.812 ± 0.104	0.719 ± 0.021
	Wt. average	0.776 ± 0.016	0.674 ± 0.153	0.680 ± 0.152	0.713 ± 0.019	0.679 ± 0.079	0.656 ± 0.126
	Macro av.	0.665 ± 0.051	0.677 ± 0.046	0.624 ± 0.117	0.713 ± 0.019	0.680 ± 0.078	0.656 ± 0.126
Multimodal (BERT +Audio)	Argument	0.851 ± 0.007	0.940 ± 0.006	0.893 ± 0.006	0.730 ± 0.031	0.773 ± 0.057	0.749 ± 0.014
	Other	0.723 ± 0.016	0.487 ± 0.020	0.581 ± 0.016	0.762 ± 0.033	0.712 ± 0.059	0.734 ± 0.017
	Wt. average	0.820 ± 0.009	0.830 ± 0.008	0.818 ± 0.009	0.746 ± 0.004	0.742 ± 0.005	0.741 ± 0.005
	Macro av.	0.787 ± 0.011	0.713 ± 0.010	0.737 ± 0.010	0.746 ± 0.005	0.743 ± 0.004	0.741 ± 0.005

Table 1: Models’ performance for original and balanced datasets. Errors indicate standard deviation after 5 replicates.

ter training was assisted by the Python package Ray[tune] (Liaw et al., 2018), which allows distributed parallel hyperparameter tuning with different search strategies and schedulers. We defined our hyperparameter search as shown in the appendix (Table A1), including training parameters like the learning rate and batch size, and also architecture-dependent parameters like the kernel size of the CNN or whether the text embeddings should be retrained or not. To search over the defined hyperparameter space, we used the Tree-structured Parzen Estimator algorithm (Bergstra et al., 2011), which considers the performance of previous iterations of the search to choose the next hyperparameters to test, implemented in the HyperOpt library for parallel optimization (Bergstra et al., 2013). The training was implemented in TensorFlow and we included Keras callbacks after each epoch of training to update the hyperparameter search algorithm. We implemented an early stopping scheduler algorithm that monitored validation loss at each epoch, stopping the training before overfitting, with a minimum change of $1e-4$ and a patience of 3 epochs. We considered implementing other schedulers like Asynchronous Successive Halving Algorithm (ASHA), which stops unpromising trials if their performance is worse than that of previous trials (Li et al., 2018), but we discovered in our experiments that this algorithm tends to penalize slow learning models, which sometimes ended up giving the best results. We sampled 50 times the search space, with validation

split of 20% and test split of 20%, and the best hyperparameters, used in the remaining sections, are reported in the appendix (Table A1), as well as average runtimes and number of parameters. With our dataset, we provide full details of the training results, with confusion matrices, training history, validation metrics plots, etc.

4 Models’ performance

4.1 Full original dataset

Table 1 shows the performance of each one of the models after training with the original dataset ($N = 28, 850$) and optimized parameters. The text-only models, particularly the BERT model, perform best in terms of both macro and weighted F-scores, reaching a weighted average of 0.826 (macro average of 0.746), comparable to the weighted average reported by Haddadan et al. (2019) of 0.843 (or macro average of 0.730) using a LSTM network. As in that paper, the precision and recall of the "other" class is low, but classification of the "argument" class performs much better, with a high recall in both the BERT and Bi-LSTM models. The audio-only model yields a rather low macro averaged F-score of 0.540, as it relies on over-classifying the argument class. The BERT-based multimodal model performs significantly better than the audio-only model and similarly to the text models, with a macro average of 0.737. The Bi-LSTM-based multimodal model performs better than the audio-only model in terms of macro

Model	Class	10% original dataset ($N = 2, 885$)			10% balanced dataset ($N = 1, 376$)		
		Precision	Recall	F_1	Precision	Recall	F_1
Text Bi-LSTM	Argument	0.816 ± 0.027	0.946 ± 0.022	0.876 ± 0.015	0.686 ± 0.045	0.749 ± 0.025	0.715 ± 0.031
	Other	0.669 ± 0.067	0.332 ± 0.092	0.436 ± 0.083	0.723 ± 0.027	0.656 ± 0.061	0.687 ± 0.044
	Wt. average	0.781 ± 0.029	0.797 ± 0.024	0.769 ± 0.034	0.705 ± 0.033	0.702 ± 0.035	0.701 ± 0.036
	Macro av.	0.743 ± 0.037	0.639 ± 0.039	0.656 ± 0.046	0.704 ± 0.033	0.702 ± 0.035	0.701 ± 0.036
Text BERT	Argument	0.784 ± 0.042	0.985 ± 0.016	0.873 ± 0.020	0.571 ± 0.117	0.712 ± 0.231	0.610 ± 0.101
	Other	0.391 ± 0.395	0.148 ± 0.192	0.206 ± 0.262	0.535 ± 0.204	0.441 ± 0.352	0.438 ± 0.317
	Wt. average	0.687 ± 0.133	0.782 ± 0.040	0.710 ± 0.082	0.552 ± 0.155	0.570 ± 0.117	0.520 ± 0.174
	Macro av.	0.588 ± 0.218	0.567 ± 0.089	0.539 ± 0.140	0.553 ± 0.152	0.577 ± 0.109	0.524 ± 0.168
Audio	Argument	0.806 ± 0.045	0.782 ± 0.419	0.708 ± 0.360	0.614 ± 0.043	0.658 ± 0.256	0.607 ± 0.131
	Other	0.477 ± 0.380	0.310 ± 0.400	0.237 ± 0.182	0.670 ± 0.112	0.549 ± 0.238	0.563 ± 0.130
	Wt. average	0.731 ± 0.079	0.671 ± 0.232	0.597 ± 0.258	0.639 ± 0.048	0.613 ± 0.023	0.590 ± 0.036
	Macro av.	0.642 ± 0.183	0.546 ± 0.050	0.472 ± 0.160	0.642 ± 0.051	0.604 ± 0.027	0.585 ± 0.041
Multimodal (Bi-LSTM +Audio)	Argument	0.684 ± 0.382	0.635 ± 0.360	0.657 ± 0.369	0.755 ± 0.140	0.386 ± 0.260	0.445 ± 0.258
	Other	0.412 ± 0.103	0.641 ± 0.213	0.472 ± 0.054	0.612 ± 0.059	0.834 ± 0.131	0.697 ± 0.027
	Wt. average	0.620 ± 0.316	0.637 ± 0.227	0.614 ± 0.294	0.679 ± 0.044	0.625 ± 0.052	0.581 ± 0.113
	Macro av.	0.548 ± 0.241	0.638 ± 0.078	0.565 ± 0.210	0.684 ± 0.052	0.610 ± 0.066	0.571 ± 0.124
Multimodal (BERT +Audio)	Argument	0.833 ± 0.030	0.936 ± 0.023	0.881 ± 0.009	0.722 ± 0.017	0.789 ± 0.033	0.753 ± 0.012
	Other	0.710 ± 0.060	0.445 ± 0.095	0.538 ± 0.056	0.771 ± 0.021	0.699 ± 0.037	0.732 ± 0.017
	Wt. average	0.803 ± 0.012	0.810 ± 0.014	0.793 ± 0.024	0.747 ± 0.009	0.743 ± 0.009	0.743 ± 0.010
	Macro av.	0.771 ± 0.017	0.690 ± 0.034	0.709 ± 0.030	0.746 ± 0.009	0.744 ± 0.009	0.743 ± 0.010

Table 2: Models’ performance for 10% of the datasets. Errors indicate standard deviation after 5 replicates.

averaged F-score, with 0.624. Our models perform slightly better than those of Mancini et al. (2022), who used the same architecture (although with their own hyperparameter optimization) and dataset (although slightly smaller). But, like us, they find that the multimodal models do not significantly (if at all) outperform the text-only models, with macro F-scores of 0.674 in both. Their audio-only model was not better than the random baseline at 0.505.

In both the original work of Mestre et al. (2021) and the replication by Mancini et al. (2022), the authors show a beneficial impact of audio embeddings in argument classification. However, that dataset was significantly smaller ($N = 4, 104$) and heavily imbalanced towards one of the classes. Our dataset is also slightly imbalanced towards the "argument" class (76.15% arguments) and the text-only models seem to be reaching saturation, as per the previous paragraph. Moreover, the low precision and recall of the "other" class leads us to believe that the models overly rely on classifying many instances as "argument". Therefore, we asked ourselves what would happen: 1) when the dataset is small and the performance of the text-only model might suffer; 2) when the dataset is balanced and the models cannot rely on over-classifying the "argument" class. Does the addition of audio improve the performance metrics in those cases?

4.2 Fractional and balanced datasets

The right-hand side of Table 1 shows the results with the same models for a balanced dataset. To obtain a balanced dataset, a random number of "ar-

gument" classes were dropped from the original dataset, until we obtained an equal number of both classes, therefore reducing the total size to 13,758 sentences. This table shows how the overall performance of the BERT and Bi-LSTM models has been reduced, reaching macro averaged F-score values of 0.754 for the BERT model and 0.728 for the Bi-LSTM model. Moreover, the precision and recall of both classes are more balanced: whereas in the original dataset the recall of the "argument" and "other" classes of the BERT model were 0.951 and 0.487, respectively, they are now 0.839 and 0.674. The multimodal model continues to perform significantly better than the audio-only model, but still not better than the BERT model, to which it still achieves similar F-score values. It seems, therefore, that a multimodal model does not provide better results than text-only models when the datasets are balanced, at least as long as the number of annotations continues to be large ($N = 13, 758$). The text-only models still seem to reach saturation of what can be accomplished with the dataset.

Table 2 shows the results for a fractional dataset composed of only 10% of the original and balanced data. We hypothesize that the performance of the text-only models will start to suffer with small amounts of training data and the audio features from the multimodal models will be able to partially recover previous performance. Indeed, it has been shown in some work that BERT models tend to decline in performance with small datasets and can be outperformed by simpler models like Bi-LSTM (Ezen-Can, 2020). Likewise, not only does

the overall performance suffer, but also the stability of the model as discussed by Dodge et al. (2020). In our case, we see that the BERT model shows a large drop in macro average F-score, down to 0.539, and high instability, as can be observed from the large standard error of the "other" class. In this case, the Bi-LSTM model outperforms the BERT model at 0.656, suggesting that the BERT model is more sensitive to smaller datasets. The BERT-based multimodal model also outperforms the BERT model, with a macro average F-score of 0.709, very close to the best scenario with the original dataset at 0.746. Surprisingly, the Bi-LSTM-based multimodal model does not outperform the Bi-LSTM model, but worsens its performance. On 10% of the balanced dataset, results are similar, with the BERT model suffering and the BERT-based multimodal model outperforming all with F-scores of 0.743, close to the original balanced case.

For intermediate sizes, such as 50% and 20% (reported in Section C) the change seems to be gradual. For both 50% and 20% of the original dataset, the performance of the only-text BERT model and the BERT-based multimodal model is practically identical. However, for balanced datasets of 20% (with only $N = 2,751$), the performance of the BERT model starts to decrease and is overcome by the multimodal model. All these results suggest that there is a point at around $N = 3,000$ or below where audio features provide an important added value in the performance of the models. Full details of all the replicates, including the model history with validation losses and accuracy, confusion matrices and a full breakdown of the performance metrics can be found in the repository of the project⁵ and its official release (Mestre et al., 2023).

5 Artificial voices

It is not clear what the audio models are specifically looking at when they undergo training, as the features extracted are not always easy to interpret. One hypothesis is that they learn from the words being uttered, their pronunciation and associations thereof in a similar way to text-based models. However, it is also likely that these audio models are picking up intonation or pitch features that are possibly different when one person is making an argument.

As a first step in investigating what the audio-

⁵<https://github.com/rafamestre/Multimodal-USElecDeb60To16>.

based models are paying attention to, we ran our models using artificial voices, instead of the original voices from the presidential candidates. For this, we used the computer generated voices from the Microsoft Windows Text to Speech (TTS) system. Then, we used the text-to-speech conversion library `pyttsx3` (v 2.9) for Python to run each sentence through the Microsoft TTS system and generate utterances spoken by the so-called Microsoft Mark and Microsoft Zira, the male and female version of US voices. We set the speaking rate at 200 words per minute, but it could be interesting for future studies to observe the accuracy of the models when the speech rate is changed. Likewise, each country package in Microsoft Windows has its own set of unique voices, even if they speak the same language, e.g., UK, India, South Africa, and so on, so it could be interesting to check potential differences in accuracy with different accents.

We then ran our audio-only models as described before and the performance metrics are displayed in Table 3. We only report the F-scores, and not precision and recall, for simplicity.⁶ We can see in this table that, generally, there are no differences in F-score by gender of the artificial voices, although there seems to be a bias towards the female Zira voice. When compared to the audio model run with original voices, it is interesting to see that whereas using artificial voices results in a similar score with the full dataset, with the balanced dataset the results are improved, going from 0.566 as reported in 1 to 0.626 using the Zira voice. This improvement is also found with the 10% dataset, which reported 0.536 and 0.594 for the original and balanced datasets, whereas Table 2 reported 0.572 and 0.585, respectively. In the balanced case, these values are even better than the BERT model at 0.524. A potential explanation is that the artificial audio models lack noise coming from the recording, or remove variation coming from different people having different baseline pitches that might confuse the model. There might be a trade-off between taking advantage of pitch or intonation during arguments (which we were not able to prove produces any effect) and benefiting from the noise-reduced nature of artificial audio. In any case, it seems that audio-only models based on artificial voices can learn features and classify arguments with an accuracy comparable to text-only models, and sometimes

⁶Full information, all the computerized utterances and scripts to reproduce our results can be found in our repository (Mestre et al., 2023).

Voice	Class	F_1			
		Original dataset	Balanced dataset	10% original	10% balanced
Female	Argument	0.874 ± 0.002	0.596 ± 0.141	0.865 ± 0.012	0.555 ± 0.055
	Other	0.235 ± 0.029	0.656 ± 0.017	0.207 ± 0.115	0.633 ± 0.051
	Wt. average	0.722 ± 0.006	0.626 ± 0.064	0.706 ± 0.024	0.593 ± 0.025
	Macro av.	0.555 ± 0.014	0.626 ± 0.063	0.536 ± 0.054	0.594 ± 0.022
Male	Argument	0.871 ± 0.003	0.582 ± 0.126	0.875 ± 0.007	0.496 ± 0.173
	Other	0.199 ± 0.043	0.654 ± 0.167	0.191 ± 0.121	0.595 ± 0.127
	Wt. average	0.711 ± 0.010	0.618 ± 0.060	0.720 ± 0.032	0.548 ± 0.057
	Macro av.	0.535 ± 0.021	0.618 ± 0.059	0.533 ± 0.062	0.545 ± 0.059

Table 3: Audio-only models’ performance with artificial voices. Errors indicate standard deviation after 5 replicates.

even better than those when data is scarce. Models based on the original voices often struggle, and this might be due to the inherent noise of the recordings or differences at the speaker level.

6 Ablation study

Finally, to further understand how the different LLD audio features might play a role in argument detection, we perform an ablation study with the audio model in which we eliminate one of the six features at a time and assess how the performance of the model changes. The results are in Table 4 for the full and balanced datasets (we only report F-score for simplicity). The first column indicates the feature that was eliminated from the feature tensor, which originally had dimensions of $(45, T_{max})$, whereas the second column displays the dimensions of the tensor after elimination. In general, none of the cases deviate much from the full-feature model with macro F-scores of 0.540 and 0.566 for the original and balanced datasets (Table 1). The first four features are spectral features, meaning that they are features extracted from the spectrogram of the sound wave. In particular, the spectral centroid and bandwidth characterize the center of mass of the spectrum (where most of the energy is located) and its weighted standard deviation, respectively (Sandhya et al., 2020). The spectral rolloff also characterizes the energy spectrum by identifying the frequency below which a certain percentage (in our case, 85%) of the energy is located, and can be used to differentiate voices from noise (Syed et al., 2021). These three features are 1-dimensional and only reduce the feature space to $(44, T)$. The spectral contrast feature, however, is 7-dimensional and works by dividing the spectrogram into 6 sub-bands, for which the difference between their peaks and valleys are computed, and is commonly used in music identification (Jiang et al., 2002b). The chromagram, or chroma feature,

is a feature that aggregates all information of a waveform into the 12 different pitch classes, which are separated by an octave. This feature is mostly used for music synchronization and singing voice separation (Yuan et al., 2022). Finally, MFCCs are a variable set of features (in our case, we use 12) which describe the shape of the spectral signal. They are based on human auditive perceptions and are widely used in the literature to capture phonetically relevant features (Mansour and Lachiri, 2017).

From the ablation study, we see that skipping features does not have a strong influence on the performance with the original full dataset. With the balanced dataset, the elimination of the spectral roll-off feature seems to have a strong effect, as it decreases its macro F-score to 0.402. A special mention to the MFCCs is deserved. These are the most common LDDs in the literature. Eliminating this feature but keeping the rest does not affect the performance (F-score of 0.544) in the original dataset, and improves it in the balanced case to 0.612. As MFCCs (and many of the other features) are commonly used to distinguish between voices based on their frequency and pitch, they could bias the model by considering information about the speaker, which is not necessarily relevant to the argument. This would also explain why the artificial voices performed better than the original dataset and why some of the simplest features, like spectral roll-off, have a big influence on performance.

7 Conclusion

In this paper, we explored the possibilities of using audio to detect arguments with multimodal machine learning models in a dataset of US presidential debates that was annotated for arguments. We found that, generally, BERT-based text-only models outperformed all models in the original dataset, but multimodal models can improve performance

Feature skipped	Feature space	Class	F_1	
			Original dataset	Balanced dataset
Spectral centroids	(44,T)	Argument	0.858 ± 0.010	0.511 ± 0.185
		Other	0.225 ± 0.156	0.613 ± 0.005
		Wt. average	0.706 ± 0.035	0.561 ± 0.070
		Macro av.	0.542 ± 0.075	0.562 ± 0.069
Spectral bandwidth	(44,T)	Argument	0.869 ± 0.001	0.360 ± 0.270
		Other	0.198 ± 0.018	0.653 ± 0.039
		Wt. average	0.709 ± 0.001	0.508 ± 0.122
		Macro av.	0.534 ± 0.009	0.507 ± 0.122
Spectral roll-off	(44,T)	Argument	0.777 ± 0.109	0.127 ± 0.118
		Other	0.320 ± 0.192	0.677 ± 0.007
		Wt. average	0.668 ± 0.065	0.408 ± 0.058
		Macro av.	0.548 ± 0.075	0.402 ± 0.059
Spectral contrast	(38,T)	Argument	0.866 ± 0.002	0.429 ± 0.278
		Other	0.201 ± 0.110	0.645 ± 0.025
		Wt. average	0.706 ± 0.028	0.537 ± 0.127
		Macro av.	0.533 ± 0.055	0.537 ± 0.128
Chroma	(33,T)	Argument	0.870 ± 0.003	0.408 ± 0.316
		Other	0.196 ± 0.038	0.611 ± 0.061
		Wt. average	0.710 ± 0.012	0.509 ± 0.133
		Macro av.	0.533 ± 0.020	0.509 ± 0.132
MFCCs	(22,T)	Argument	0.869 ± 0.003	0.619 ± 0.032
		Other	0.220 ± 0.018	0.605 ± 0.058
		Wt. average	0.714 ± 0.011	0.611 ± 0.022
		Macro av.	0.544 ± 0.011	0.612 ± 0.021

Table 4: Results from ablation study. The complete feature space has a dimensions of $(45, T_{max})$, where T_{max} is the 99% percentile length of the utterances.

when the datasets are small and BERT encodings start performing poorly, in both balanced and unbalanced versions of the data. Multimodal models are therefore an alternative that could be used to improve classifications of arguments when data is scarce. To further investigate the reasons for these improvements, we ran audio-only models using artificially generated voices of male and female genders. Although we did not find a significant difference in the performance with the artificial voices (only a slight preference toward female), we find that in the most difficult scenarios (balanced and small datasets), the models with artificial voices outperform those with the original audio from the debates. Moreover, we perform an ablation study and we find that removing certain features like MFCCs can improve the performance of the models. We recognize that these features are commonly used to distinguish between speakers, so irrelevant characteristics of speakers might be influencing the capacity of the models to accurately classify arguments. However, all these features are highly correlated with one another, so further work should investigate which features are more independent of the speakers themselves or if they can be normalized before being fed into the network. These results should be compared with artificially generated voices, which could have different accents or speech rates to understand how those features can influence the classification of arguments.

Limitations

This paper assesses the benefits of audio features in the task of argumentation mining. Although the dataset presented has a large number of annotations ($N = 28, 850$) further research should be aimed at studying its cross-domain adaptability in different scenarios and datasets. The model architectures used in this paper are fairly standard in the literature and thus represent benchmarking results for further research in the field, as the application of multimodal sources of data in argumentation mining is still largely unexplored. Newer architectures, for instance those based on cross-attention mechanisms between the different modalities, should be explored next to check whether these results could be improved. It is still not fully clear what information from the audio embeddings are being picked up by the models. Our study on computerized voices offers an interesting avenue of research, but it should be further expanded to include a larger array of voices, different speech rates and embedding in fully multimodal models to assess their performance. The potential biases of models that use audio, especially how different voice’s characteristics (such as pitch frequency, which is correlated to gender) affect the classification, are not fully studied here, but briefly touched upon in the computerized voice study. This is rather important but unexplored territory and normalization strategies should be investigated to solve these issues. Finally, there are a large number of audio features that could be explored in this domain. Those used in this work are just some of the most common ones, but, as mentioned before, the choice is generally based on the extraction package used by the researcher. There is a need for standardization of these features in the community, so different work can be better compared.

Ethics Statement

We acknowledge that the use of audio features for automatic classification can cause potential privacy issues, as the real voice, and not only the speech, is used for classification. At this stage, we do not foresee any outstanding ethical issues from this research, as we use public domain data that was televised in public national television and is widely available on the web. Ethics approval for this research was received from the University of Southampton’s Faculty of Social Science Ethics and Research Governance committee, Ref: 66226,

Date 16/03/2022. The audio-enhanced dataset of this work along with the codes to reproduce our results are made available under license CC0: Public Domain in the project's [GitHub](#) and is also assigned a DOI to ensure reliable access to this work's supplementary data ([Mestre et al., 2023](#)).

Acknowledgements

This work has been funded by UK Research and Innovation (UKRI) funding (grant ref MR/S032711/1) and by the Web Science Institute of the University of Southampton (project PP-2020-Mestre). This work was also supported by the Natural Environment Research Council (NE/S015604/1) and Economic and Social Research Council (ES/V011278/1). The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

References

- Teresa Alsinet, Josep Argelich, Ramón Béjar, and Joel Cemeli. 2019. [A distributed argumentation algorithm for mining consistent opinions in weighted twitter discussions](#). *Soft Computing*, 23:2147–2166.
- Bagus Tris Atmaja and Masato Akagi. 2020. [Dimensional speech emotion recognition from speech features and word embeddings by using multitask learning](#). *APSIPA Transactions on Signal and Information Processing*, 9.
- Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443.
- James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.
- James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR.
- C. Busso, M. Bulut, C. C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan. 2008. [IEMOCAP: Interactive emotional dyadic motion capture database](#). *Language Resources and Evaluation*, 42(4):335–359.
- Linqin Cai, Yaxin Hu, Jiangong Dong, and Sitong Zhou. 2019. Audio-textual emotion recognition based on improved neural networks. *Mathematical Problems in Engineering*, 2019.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.
- Florian Eyben, Klaus R. Scherer, Bjorn W. Schuller, Johan Sundberg, Elisabeth Andre, Carlos Busso, Laurence Y. Devillers, Julien Epps, Petri Laukka, Shrikanth S. Narayanan, and Khiet P. Truong. 2016. [The geneva minimalistic acoustic parameter set \(gemaps\) for voice research and affective computing](#). *IEEE Transactions on Affective Computing*, 7:190–202.
- Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. 2013. [Recent developments in openmille, the munich open-source multimedia feature extractor](#). pages 835–838.
- Aysu Ezen-Can. 2020. A comparison of lstm and bert for small corpus. *arXiv preprint arXiv:2009.05451*.
- Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. 2014. Analyzing argumentative discourse units in online interactions. In *Proceedings of the first workshop on argumentation mining*, pages 39–48.
- Theodoros Giannakopoulos. 2015. [Pyaudioanalysis: An open-source python library for audio signal analysis](#). *PLoS ONE*, 10.
- Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. 2010. [Multimodal semi-supervised learning for image classification](#). pages 902–909.
- Shohreh Haddadan, Elena Cabrio, and Serena Villata. 2019. Yes, we can! mining arguments in 50 years of us presidential campaign debates. In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*, pages 4684–4690.
- Ngoc Huynh Ho, Hyung Jeong Yang, Soo Hyung Kim, and Gueesang Lee. 2020. [Multimodal approach of speech emotion recognition using multi-level multi-head fusion attention-based recurrent neural network](#). *IEEE Access*, 8:61672–61686.
- Anurag Illendula and Amit Sheth. 2019. [Multimodal emotion classification](#).
- Philip Jackson and SJUoSG Haq. 2014. Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*.
- Rajiv Jain and Curtis Wigington. 2019. [Multimodal document image classification](#). pages 71–77. IEEE Computer Society.

- D. N. Jiang, L. Lu, H. J. Zhang, J. H. Tao, and L. H. Cai. 2002a. [Music type classification by spectral contrast feature](#). *Proc. IEEE Int. Conf. on Multimedia and Expo*, pages 113–116.
- Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. 2002b. Music type classification by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116. IEEE.
- A. Klapuri and M. Davy. 2006. [Signal processing methods for music transcription](#). In *Signal Processing Methods for Music Transcription*, chapter 5. Springer Science & Business Media.
- John Lawrence and Chris Reed. 2017. Using complex argumentative interactions to reconstruct the argumentative structure of large-scale debates. In *Proceedings of the 4th Workshop on Argument Mining*, pages 108–117.
- Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Ben Recht, and Ameet Talwalkar. 2018. Massively parallel hyperparameter tuning.
- Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. 2018. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.
- Marco Lippi and Paolo Torrioni. 2016. Argument mining from speech: Detecting claims in political debates. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pages 2979–2985.
- Steven R Livingstone and Frank A Russo. 2018. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391.
- Eleonora Mancini, Federico Ruggeri, Andrea Galassi, and Paolo Torrioni. 2022. [Multimodal argument mining: A case study in political debates](#). pages 158–170.
- Asma Mansour and Zied Lachiri. 2017. Svm based emotional speaker recognition using mfcc-sdc features. *International Journal of Advanced Computer Science and Applications*, 8(4).
- Brian McFee, Colin Raffel, Dawen Liang, Daniel P Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25.
- Rafael Mestre, Stuart E. Middleton, Matt Ryan, Masood Gheasi, Timothy J. Norman, and Jiatong Zhu. 2023. [rafamestre/multimodal-uselecdeb60to16: v1.0.0](#).
- Rafael Mestre, Razvan Milicin, Stuart Middleton, Matt Ryan, Jiatong Zhu, and Timothy J Norman. 2021. M-arg: Multimodal argument mining dataset for political debates with audio and transcripts. In *Proceedings of the 8th Workshop on Argument Mining*, pages 78–88.
- Louis-Philippe Morency and Tadas Baltrušaitis. 2017. Multimodal machine learning: integrating language, vision and speech. In *Proceedings of the 55th annual meeting of the association for computational linguistics: Tutorial abstracts*, pages 3–5.
- Kai Nakamura, Sharon Levy, and William Yang Wang. 2020. [rfakeddit: A new multimodal benchmark dataset for fine-grained fake news detection](#). pages 11–16.
- Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.
- P Sandhya, V Spoorthy, Shashidhar G Koolagudi, and NV Sobhana. 2020. Spectral features for emotional speaker recognition. In *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAIECC)*, pages 1–6. IEEE.
- Prabhav Singh, Ridam Srivastava, K. P.S. Rana, and Vineet Kumar. 2021. [A multimodal hierarchical approach to speech emotion recognition from audio and text\[formula presented\]](#). *Knowledge-Based Systems*, 229.
- C. Stab and I. Gurevych. 2014. [Annotating argument components and relations in persuasive essays](#). In *Proc. 25th Int. Conf. on Computational Linguistics*, pages 1501–1510.
- S Syed, Munaf Rashid, Samreen Hussain, Anoshia Imtiaz, Hammah Abid, and Hira Zahid. 2021. Inter classifier comparison to detect voice pathologies. *Mathematical Biosciences and Engineering*, 18(3):2258–2273.
- Jacky Visser, John Lawrence, Chris Reed, Jean Wage-mans, and Douglas Walton. 2021. *Annotating Argument Schemes*, volume 35.
- Yue Xie, Ruiyu Liang, Zhenlin Liang, Chengwei Huang, Cairong Zou, and Bjorn Schuller. 2019. [Speech emotion classification using attention-based lstm](#). *IEEE/ACM Transactions on Audio Speech and Language Processing*, 27:1675–1685.
- Bo Yang, Bo Shao, Lijun Wu, and Xiaola Lin. 2022. Multimodal sentiment analysis with unidirectional modality translation. *Neurocomputing*, 467:130–137.
- Zengwei Yao, Zihao Wang, Weihuang Liu, Yaqian Liu, and Jiahui Pan. 2020. [Speech emotion recognition using fusion of three multi-task learning-based classifiers: Hsf-dnn, ms-cnn and lld-rnn](#). *Speech Communication*, 120:11–19.

- Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. 2018. Multimodal speech emotion recognition using audio and text. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 112–118. IEEE.
- Siyuan Yuan, Zhepei Wang, Umut Isik, Ritwik Giri, Jean-Marc Valin, Michael M Goodwin, and Arvinth Krishnaswamy. 2022. Improved singing voice separation with chromagram-based pitch-aware remixing. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 111–115. IEEE.
- Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems*, 31(6):82–88.
- Jianfeng Zhao, Xia Mao, and Lijiang Chen. 2019. [Speech emotion recognition using deep 1d 2d cnn lstm networks](#). *Biomedical Signal Processing and Control*, 47:312–323.

A Hyperparameter tuning

Table A1 shows the hyperparameter bounds, as well as the final values selected for each model. After hyperparameter tuning, BERT-based text-models took approximately 45 minutes to train, whereas Bi-LSTM-based text-models took only 2 minutes (on the full dataset). Audio-only models took 25 min to train (as they would converge very slowly taking more than 50 training epochs), BERT-based multimodal models took an average of 3 hours and Bi-LSTM-based multimodal models took 8 minutes.

B Notes on alignment

As mentioned in the main text, we used the Aeneas package, which is a Python package that uses “forced alignment” to match the text and audio from utterances. Briefly, each sentence is provided as text and the tool uses the *espeak* Windows speech synthesizer to generate a computerized voice uttering that sentence. Then, amplitude waves are contrasted from the real and computer-generated sentences, and they are aligned to extract the timestamps. Two researchers manually checked every debate for major misalignment (and fixed them) until we obtained an almost perfectly aligned text.

We indicated omissions to be ignored during alignment: text between brackets that indicates transcription tags like "applause", or the interjection "uh", that appeared (too much) in some of the transcripts and never in others. This tool comes with a handy HTML output that allows the user to click on different parts of the transcript and check the alignment. Two people manually checked every debate for major misalignment (and fixed those cases as described above) until we obtained an almost perfectly aligned text.

Together with the dataset and codes to reproduce the results, we present our code to reproduce the alignment process, as well as an exhaustive list of the problems we encountered during alignment and how we solved them (e.g., modifications to the original transcripts, splitting videos, etc.). We share a folder with all the results from training our models with original and balanced datasets, as well as fractional subsets of 50%, 20% and 10%. We also include the training with artificial voices and from the ablation study. Each subfolder contains the parameters used by the model, confusion matrices of each run (5 runs per model), loss value vs epoch plots, training history with validation met-

rics, and precision/recall/F-score metrics for each run, as well as the average values.

C Results for 50% and 20% datasets

Tables A2 and A3 show the results for all models with 50% and 20% fractional datasets.

Hyper-parameter	Range	Bi-LSTM	BERT	Audio	Multimodal (Bi-LSTM +Audio)	Multimodal (BERT +Audio)
Learning rate	[0.01, 0.000001]	6.68e-4	2.37e-6	1.89e-5	7.73e-5	2.81e-6
Batch size	{16, 32, 64}	16	32	64	16	16
Hidden activation	{ReLU, Sigm., Tanh}	Tanh	ReLU	Sigm.	ReLU	ReLU
Trainable	{True, False}	False	True		True	True
Dropout text	[0, 0.9]	0.6	0.8		0.5	0.7
Dropout audio	[0, 0.9]			0.6	0.6	0.1
Dropout final	[0, 0.9]				0	0.5
# neurons dense layer	{16, 32, 64, 128, 256}	32	256	32	16	16
# neurons Bi-LSTM text	{16, 32, 64, 128, 256}	64			64	
# neurons Bi-LSTM audio	{16, 32, 64, 128, 256}			16	32	256
# filters conv. layer 1	{4, 8, 16, 32, 64}			4	4	8
# filters conv. layer 2	{4, 8, 16, 32, 64}			4	8	4
Kernel size conv. layer 1	{1, 3, 5, 7}			3	3	3
Kernel size conv. layer 2	{1, 3, 5, 7}			5	7	3
Size pooling layer 1	{2, 4}			2	2	2
Size pooling layer 2	{2, 4}			2	2	4
Number of parameters		2,802,074	109,679,619	5,362,110	7,928,618	119,978,775

Table A1: List of hyperparameters, their search range and their optimal value for each model. The range in "learning rate" (in squared brackets) was given as a log uniform distribution, in the dropout layers a uniform distribution in multiples of 0.1, whereas in the remaining cases (represented with curly brackets) the choices were from the discrete set of values shown.

Model	Class	50% original dataset ($N = 14, 425$)			50% balanced dataset ($N = 6, 879$)		
		Precision	Recall	F_1	Precision	Recall	F_1
Text Bi-LSTM	Argument	0.836 ± 0.014	0.944 ± 0.012	0.886 ± 0.005	0.704 ± 0.025	0.784 ± 0.027	0.741 ± 0.004
	Other	0.705 ± 0.044	0.415 ± 0.033	0.521 ± 0.019	0.760 ± 0.020	0.673 ± 0.049	0.713 ± 0.026
	Wt. average	0.805 ± 0.006	0.816 ± 0.007	0.798 ± 0.011	0.732 ± 0.008	0.728 ± 0.013	0.727 ± 0.014
	Macro av.	0.770 ± 0.016	0.680 ± 0.011	0.704 ± 0.011	0.732 ± 0.008	0.728 ± 0.012	0.727 ± 0.014
Text BERT	Argument	0.841 ± 0.041	0.952 ± 0.028	0.892 ± 0.013	0.719 ± 0.010	0.803 ± 0.023	0.759 ± 0.009
	Other	0.776 ± 0.127	0.402 ± 0.224	0.474 ± 0.262	0.773 ± 0.018	0.680 ± 0.027	0.723 ± 0.016
	Wt. average	0.826 ± 0.004	0.822 ± 0.031	0.794 ± 0.071	0.746 ± 0.010	0.742 ± 0.010	0.741 ± 0.010
	Macro av.	0.809 ± 0.043	0.677 ± 0.098	0.683 ± 0.137	0.746 ± 0.010	0.742 ± 0.010	0.741 ± 0.010
Audio	Argument	0.795 ± 0.032	0.897 ± 0.176	0.833 ± 0.079	0.630 ± 0.030	0.521 ± 0.211	0.547 ± 0.122
	Other	0.398 ± 0.246	0.211 ± 0.253	0.216 ± 0.153	0.601 ± 0.067	0.681 ± 0.164	0.623 ± 0.050
	Wt. average	0.703 ± 0.066	0.738 ± 0.078	0.690 ± 0.042	0.616 ± 0.022	0.599 ± 0.034	0.584 ± 0.050
	Macro av.	0.597 ± 0.126	0.554 ± 0.042	0.524 ± 0.052	0.615 ± 0.023	0.601 ± 0.029	0.585 ± 0.048
Multimodal (Bi-LSTM +Audio)	Argument	0.870 ± 0.036	0.756 ± 0.190	0.794 ± 0.113	0.800 ± 0.096	0.330 ± 0.282	0.398 ± 0.271
	Other	0.493 ± 0.119	0.622 ± 0.181	0.521 ± 0.049	0.582 ± 0.091	0.887 ± 0.117	0.692 ± 0.029
	Wt. average	0.780 ± 0.013	0.722 ± 0.106	0.728 ± 0.092	0.694 ± 0.027	0.603 ± 0.091	0.542 ± 0.154
	Macro av.	0.682 ± 0.044	0.689 ± 0.034	0.657 ± 0.073	0.691 ± 0.023	0.609 ± 0.084	0.545 ± 0.149
Multimodal (BERT +Audio)	Argument	0.842 ± 0.010	0.946 ± 0.011	0.891 ± 0.223	0.727 ± 0.025	0.778 ± 0.028	0.751 ± 0.020
	Other	0.736 ± 0.037	0.457 ± 0.029	0.563 ± 0.014	0.756 ± 0.021	0.701 ± 0.022	0.727 ± 0.010
	Wt. average	0.816 ± 0.002	0.825 ± 0.003	0.810 ± 0.005	0.742 ± 0.013	0.740 ± 0.014	0.739 ± 0.013
	Macro av.	0.789 ± 0.014	0.701 ± 0.009	0.727 ± 0.006	0.741 ± 0.014	0.739 ± 0.013	0.739 ± 0.013

Table A2: Models' performance for 50% of the datasets. Errors indicate standard deviation after 5 replicates.

Model	Class	20% original dataset ($N = 5, 770$)			20% balanced dataset ($N = 2, 751$)		
		Precision	Recall	F_1	Precision	Recall	F_1
Text Bi-LSTM	Argument	0.833 ± 0.012	0.941 ± 0.026	0.883 ± 0.012	0.699 ± 0.024	0.723 ± 0.064	0.709 ± 0.032
	Other	0.679 ± 0.066	0.386 ± 0.043	0.488 ± 0.019	0.728 ± 0.038	0.701 ± 0.053	0.713 ± 0.028
	Wt. average	0.797 ± 0.015	0.810 ± 0.015	0.790 ± 0.013	0.714 ± 0.023	0.712 ± 0.024	0.711 ± 0.024
	Macro av.	0.756 ± 0.030	0.663 ± 0.010	0.686 ± 0.009	0.714 ± 0.023	0.712 ± 0.040	0.711 ± 0.024
Text BERT	Argument	0.844 ± 0.015	0.952 ± 0.015	0.895 ± 0.023	0.623 ± 0.112	0.691 ± 0.166	0.648 ± 0.128
	Other	0.757 ± 0.047	0.453 ± 0.026	0.566 ± 0.021	0.648 ± 0.117	0.570 ± 0.184	0.598 ± 0.147
	Wt. average	0.823 ± 0.019	0.831 ± 0.017	0.815 ± 0.018	0.635 ± 0.113	0.632 ± 0.116	0.624 ± 0.122
	Macro av.	0.801 ± 0.026	0.703 ± 0.013	0.731 ± 0.016	0.635 ± 0.113	0.631 ± 0.115	0.623 ± 0.122
Audio	Argument	0.802 ± 0.024	0.770 ± 0.246	0.766 ± 0.131	0.410 ± 0.422	0.207 ± 0.422	0.155 ± 0.279
	Other	0.453 ± 0.157	0.365 ± 0.289	0.304 ± 0.132	0.575 ± 0.127	0.817 ± 0.384	0.589 ± 0.205
	Wt. average	0.718 ± 0.026	0.677 ± 0.121	0.657 ± 0.079	0.488 ± 0.237	0.519 ± 0.031	0.378 ± 0.051
	Macro av.	0.628 ± 0.068	0.568 ± 0.028	0.535 ± 0.037	0.492 ± 0.240	0.512 ± 0.020	0.372 ± 0.048
Multimodal (Bi-LSTM +Audio)	Argument	0.873 ± 0.089	0.432 ± 0.398	0.423 ± 0.429	0.751 ± 0.045	0.457 ± 0.118	0.558 ± 0.099
	Other	0.357 ± 0.107	0.792 ± 0.200	0.467 ± 0.070	0.610 ± 0.039	0.841 ± 0.068	0.705 ± 0.026
	Wt. average	0.748 ± 0.073	0.522 ± 0.254	0.473 ± 0.342	0.681 ± 0.022	0.649 ± 0.037	0.631 ± 0.055
	Macro av.	0.615 ± 0.067	0.612 ± 0.102	0.470 ± 0.248	0.680 ± 0.022	0.649 ± 0.035	0.631 ± 0.054
Multimodal (BERT +Audio)	Argument	0.843 ± 0.005	0.955 ± 0.011	0.896 ± 0.006	0.714 ± 0.014	0.793 ± 0.047	0.751 ± 0.019
	Other	0.745 ± 0.055	0.422 ± 0.027	0.538 ± 0.031	0.761 ± 0.045	0.673 ± 0.288	0.713 ± 0.019
	Wt. average	0.820 ± 0.014	0.830 ± 0.010	0.812 ± 0.010	0.738 ± 0.021	0.734 ± 0.017	0.732 ± 0.017
	Macro av.	0.794 ± 0.028	0.689 ± 0.015	0.717 ± 0.017	0.738 ± 0.021	0.733 ± 0.018	0.732 ± 0.017

Table A3: Models' performance for 20% of the datasets. Errors indicate standard deviation after 5 replicates.

Improving Retrieval Augmented Neural Machine Translation by Controlling Source and Fuzzy-Match Interactions

Cuong Hoang*, Devendra Sachan*, Prashant Mathur, Brian Thompson, Marcello Federico

AWS AI Labs

pramathu@amazon.com

Abstract

We explore zero-shot adaptation, where a general-domain model has access to customer or domain specific parallel data at inference time, but not during training. We build on the idea of Retrieval Augmented Translation (RAT) where top- k in-domain fuzzy matches are found for the source sentence, and target-language translations of those fuzzy-matched sentences are provided to the translation model at inference time. We propose a novel architecture to control interactions between a source sentence and the top- k fuzzy target-language matches, and compare it to architectures from prior work. We conduct experiments in two language pairs (En-De and En-Fr) by training models on WMT data and testing them with five and seven multi-domain datasets, respectively. Our approach consistently outperforms the alternative architectures, improving BLEU across language pair, domain, and number k of fuzzy matches with almost no trade-off on inference latency.

1 Introduction

Domain adaptation techniques such as fine-tuning (Freitag and Al-Onaizan, 2016; Luong and Manning, 2015) are highly effective at increasing in-domain performance of neural machine translation (NMT) systems, but are impractical in many realistic settings. For example, consider a single machine serving translations to thousands of customers, each with a private Translation Memory (TM). In this case, adapting, storing and loading large adapted models for each customer is computationally infeasible. In this paper we thus consider zero-shot adaptation instead, with a single general-domain model trained from heterogeneous sources that has access to the customer or domain specific TM only at inference time.

Our work builds on Retrieval Augmented Translation (RAT) (Li et al., 2022; Bulte and Tezcan, 2019; Xu et al., 2020; He et al., 2021; Cai et al., 2021), a paradigm which combines a translation model (Vaswani et al., 2017) with an external retriever module that retrieves the top- k most similar source sentences from a TM (i.e. "fuzzy matches") (Farajian et al., 2017; Gu et al., 2017; Bulte and Tezcan, 2019). The encoder encodes the input sentence along with the translations of the top- k fuzzy-matches and passes the resulting encodings to the decoder.

Prior RAT methods for NMT have fallen into two camps: Early work (Bulte and Tezcan, 2019; Zhang et al., 2018) concatenated the source sentence and the top- k fuzzy matches before encoding, relying on the encoder’s self-attention to compare the source sentence to each target sentences and determine which target phrases are relevant for the translation. More recent work (He et al., 2021; Cai et al., 2021) has opted to encode the source sentences and the top- k fuzzy matches *independently*, effectively shifting the entire burden of determining which target phrases are relevant to the decoder. We hypothesize that neither approach is ideal: In the first, the encoder has access to the information that we expect to be important (namely, the source and the fuzzy matches), but the self-attention also has potentially confusing/spurious connections. In the second, the encoder lacks the self-attention connections between the source and the fuzzy matches.

To address these issues, we propose a novel architecture which has self-attention connections between the source sentence and each fuzzy-match, but not between fuzzy-matches. We denote this method **RAT with Selective Interactions (RAT-SI)**. Our method is illustrated in Figure 1, along with two previously discussed approaches.

Experiments in five English-German (En-De) domain-specific test sets (Aharoni and Goldberg, 2020) and seven English-French (En-Fr) domain

*Work done while the authors were at AWS AI Labs.

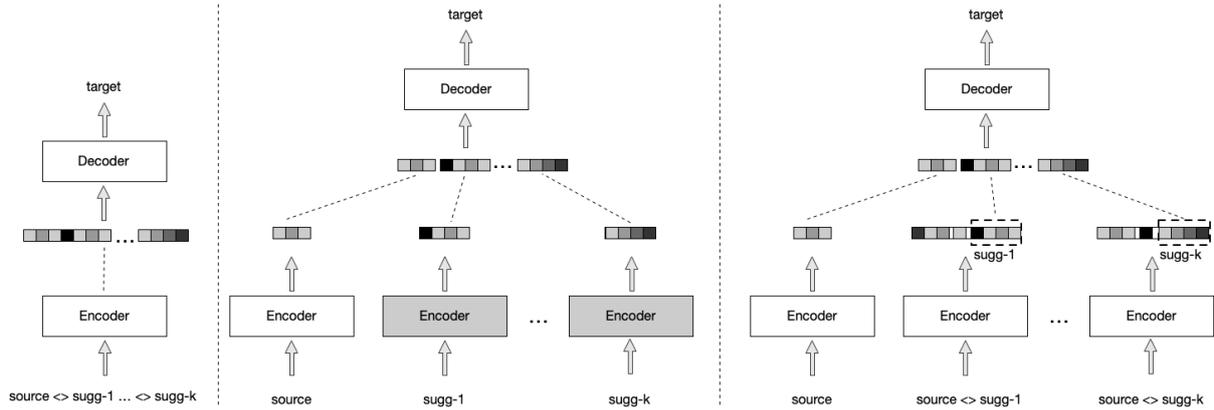


Figure 1: Architectures for retrieval augmented NMT. Left: Plain transformer ingesting source and retrieved fuzzy matches concatenated with a separator symbol (Bulte and Tezcan, 2019), denoted herein as RAT-CAT. Center: Transformer with dual encoder, one for encoding the source and one for encoding each retrieved fuzzy-matches, inspired by He et al. (2021), denoted herein as RAT-SEP. Right: Transformer separately encoding the source and each source + fuzzy-match pair (this work), denoted herein as RAT-SI.

specific test sets (Pham et al., 2021), for $k = \{3, 4, 5\}$, demonstrate that our proposed method outperforms both prior approaches in 32 out of 36 cases considered. The proposed method outperforms the closest competitor by +0.82 to +1.75 BLEU for En-De and +1.57 to +1.93 for En-Fr.

2 Method

To isolate the effects of the underlying modeling strategy from the various tricks and implementation details employed in prior papers, we build baseline models which distill the two primary modeling strategies used in prior works:

The first concatenates a source sentence with target-language fuzzy matches and then encodes the entire sequence, as in Bulte and Tezcan (2019) and Xu et al. (2020). In this approach, the cross-attention of the encoder must learn to find the relevant parts of target-language fuzzy-matches by comparing each fuzzy-match to the source sentence, while ignoring potential spurious fuzzy-match to fuzzy-match interactions (see the left diagram in Figure 1). We denote this method **RAT-CAT**.

The second encodes the source and each target-language fuzzy-match separately (with two distinct encoders), and instead concatenates the encoded representations, inspired by He et al. (2021) and Cai et al. (2021). In this approach, the spurious connections between the target-language fuzzy-matches are eliminated, but the connections between the source and each fuzzy-match are also eliminated, forcing the attention in the decoder to

find the relevant portions in the fuzzy-match that are relevant to the source (see the center diagram in Figure 1). We denote this method **RAT-SEP**.

Finally, we propose a third method which attempts to build on the strengths of each of the prior methods. As in RAT-SEP, our method separately encodes (with the same encoder) the source and each target-language fuzzy-match; however, each fuzzy-match is jointly encoded with a copy of the source, as in RAT-CAT, allowing the encoder to find portions of the fuzzy-match which are relevant to the input. Finally, all the encoded inputs are concatenated and exposed to the decoder; However, the encoding of the source is only provided to the encoder once, to avoid potentially spurious interactions between copies of the input (see the right diagram in Figure 1). We denote our proposed method **RAT-SI**.

3 Experimental Setup

Our experiments are in two language directions: English-German (En-De) and English-French (En-Fr). We train models using the public WMT 2014 (Bojar et al., 2014) data set, with 4.5M En-De sentences and 36M En-Fr sentences.

During training, the model sees target-language fuzzy-match sentences from the same dataset it is being trained on (i.e. WMT14), but at inference, models must perform zero-shot adaptation to five En-De domain-specialized TMs¹ and seven En-Fr domain-specialized TMs.² En-De data is taken

¹Medical, Law, IT, Religion and Subtitles.

²News, Medical, Bank, Law, IT, TED and Religion.

from Aharoni and Goldberg (2020), which is a re-split version of the multi-domain data set from Koehn and Knowles (2017) while En-Fr data set is taken from the multi-domain data set of Pham et al. (2021).

To find target-language fuzzy matches for our model from domain specific TMs, we use Okapi BM25 (Robertson and Zaragoza, 2009), a classical retrieval algorithm that performs search by computing lexical matches of the query with all sentences in the evidence, to obtain top-ranked sentences for each input. To enable fast retrieval, we leverage the implementation provided by the ElasticSearch library.³ Specifically, we built an index using source sentences of each TM, and for every input source sentence, we collect top- k similar source side sentences and then use their corresponding target side sentences as inputs to the model.

To explore how each method performs (and how robust they are) under different conditions, we run a full set of experiments for $k = \{3, 4, 5\}$. We train separate models for each language pair and k value, and then apply that model to each of the 5 (En-De) or 7 (En-Fr) domains.

We report translation quality with BLEU scores computed via Sacrebleu (Post, 2018).⁴ We use compare-mt (Neubig et al., 2019) to perform pairwise significance testing with `bootstrap = 1000` and `prob_thresh = 0.05` for all pairs.

All models employed transformers (Vaswani et al., 2017) with 6 encoder and 6 decoder layers. Hidden size was set to 1024 and maximum input length truncated to 1024 tokens. All models employed a joint source-target language subword vocabulary of size $32K$ using Sentencepiece algorithm (Kudo and Richardson, 2018).

We use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$; and (ii) increase the learning rate linearly for the first $4K$ training steps and decrease it thereafter; (iii) use batch size of $32K$ source tokens and $32K$ target tokens. Checkpoints are saved after every $10K$ iterations during training. We train models with maximum of $300K$ iterations. We use dropout of 0.1 and label-smoothing of 0.1.

³<https://github.com/elastic/elasticsearch-py>

⁴SacreBleu signature: nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0.

4 Results

Results for En-De are shown in Table 1 and results for En-Fr are shown in Table 2.

We observe several trends in the results. First, our proposed RAT-SI method outperforms both the RAT-CAT and RAT-SEP methods across both language pairs, having the best performance in 32/36 cases considered. In En-De, the proposed RAT-SI method has an average improvement of 1.43 BLEU over RAT-CAT and 2.35 BLEU over RAT-SEP, while in En-Fr we observe an average improvement of 1.73 BLEU over RAT-CAT and 2.98 over RAT-SEP. These results support our hypothesis that attention connections between the source sentences and each fuzzy match are critical to translation quality and the connections between the fuzzy matches are actually harmful.

Second, on average, $k = 5$ produces the best results for the RAT-SI method, but only by a small amount. However, considering individual language pair / domain combinations, there are many cases where $k = 5$ does not produce the best results, sometimes by several BLEU points. We hypothesize that this is due to the different domains containing, on average, different amounts of relevant data. This observation underscores the importance of tuning k , as well as testing new RAT methods under a variety of conditions, including different k values.

Finally, consistent with prior work, we see large improvements for all online domain-adapted methods (RAT-CAT, RAT-SEP, and RAT-SI) over the non-domain-adapted baseline, with improvements of up to +13.85 BLEU. This is not surprising, since the baseline model does not take advantage of any domain-specific data.

4.1 Latency

While not the focus of this work, we did a preliminary study of latency, comparing a baseline transformer to RAT-CAT and RAT-SI models during inference. We follow Domhan et al. (2020) and measure latency values as the 90th percentile of inference time when translating each sentence individually (no batching). We run experiments on an EC2 p3.2xlarge instance with a Tesla V100 GPU and report encoding latency results in Table 3. We use a batch size of 1 and $k=3$ for all experiments.

We observe a small increase of encoding latency by using RAT-SI (i.e. 17.48 ms) compared to of RAT-CAT. We provide a breakdown of the total

Model	k	IT	LAW	REL	MED	SUBT	Average
Baseline	n/a	27.92	35.59	11.26	30.74	19.46	24.99
RAT-CAT		33.97	50.34	25.14	45.05	19.89	34.88
RAT-SEP	k=3	32.78	49.04	22.92	44.28	20.48	33.90
RAT-SI (this work)		33.08	52.02*	26.40*	46.16	20.83*	35.70
RAT-CAT		33.67	49.59	23.40	44.87	20.27	34.36
RAT-SEP	k=4	31.84	48.38	24.37	43.55	19.99	33.63
RAT-SI (this work)		33.68	52.00*	28.42*	46.13	20.23	36.09
RAT-CAT		33.44	49.67	24.95	44.16	20.01	34.45
RAT-SEP	k=5	30.84	47.92	23.91	44.10	20.27	33.41
RAT-SI (this work)		33.84	52.17*	27.53*	46.95*	20.49	36.20

Table 1: BLEU scores for En-De experiments. The best BLEU for RAT models with a specific top-k value is **bolded**, and "*" indicates the best result is statistically significant compared to both the other methods. The proposed method (RAT-SI) produces the best results in 13/15 cases considered, with an average improvement of 1.43 BLEU over RAT-CAT and 2.35 BLEU over RAT-SEP.

Model	k	LAW	MED	IT	NEWS	BANK	REL	TED	Average
Baseline	n/a	52.68	31.12	32.22	35.09	41.04	14.51	35.55	34.60
RAT-CAT		66.32	37.09	39.91	35.09	49.01	61.83	36.39	46.52
RAT-SEP	k=3	64.93	37.06	38.02	35.57	49.14	53.34	37.29	45.05
RAT-SI (this work)		66.56	41.30*	40.61	35.53	50.19*	67.55*	37.42	48.45
RAT-CAT		65.71	37.31	38.71	34.60	49.43	63.55	36.10	46.49
RAT-SEP	k=4	64.35	37.89	38.89	35.45	49.28	53.41	37.13	45.20
RAT-SI (this work)		66.63*	39.50*	41.90*	35.71	50.04	65.20*	37.47	48.06
RAT-CAT		65.60	37.35	38.74	34.46	49.33	63.21	36.08	46.40
RAT-SEP	k=5	64.62	38.50	39.53	35.59	49.97	52.56	37.09	45.41
RAT-SI (this work)		67.03*	39.05	41.33*	35.93*	49.82	65.49*	37.90*	48.08

Table 2: BLEU scores for En-Fr experiments. The best BLEU for RAT models with a specific top-k value is **bolded**, and "*" indicates the best result is statistically significant compared to both the other methods. The proposed method (RAT-SI) produces the best results in 19/21 cases considered, with average improvements of 1.73 BLEU over RAT-CAT and 2.98 over RAT-SEP.

Model	Encoding Latency
Transformer	14.80 ms
RAT-CAT	15.23 ms
RAT-SI	17.48 ms

Table 3: Encoding latency in milliseconds of models (lower is better).

encoding latency in Table 4 which shows encoding the inputs in RAT-SI is faster than RAT-CAT but it requires an extra overhead for extracting the encoding of fuzzy matches from the joint encoding of source with fuzzy match. However, the encoding time is a very small fraction of overall latency (see Table 5) and thus this difference appears to be negligible.

We find that RAT-CAT and RAT-SI have nearly identical latencies, and each is only slightly slower than the baseline transformer (see Table 5). This is somewhat surprising since both methods make the input to the decoder significantly longer. We hypothesize that we are under-utilizing the GPU in all cases, and thus the increased computations does not increase latency. Further investigation of this is

RAT-SI Model	Encoding Latency
Encoding input	14.91 ms
Extra overheads	2.57 ms
Total time	17.48 ms

Table 4: Encoding latency of RAT-SI in milliseconds (lower is better). Extra overheads include (1): Concatenate input and $k = 3$ input-suggestion pairs (2): Extract $k = 3$ suggestion encodings and append them to the input encoding.

left for future work.

5 Related Work

Bulte and Tezcan (2019) proposed augmenting the input to NMT with target-language fuzzy-match sentences from a TM, concatenating the input and fuzzy-matches together. Their method was simpler than prior works such as (Zhang et al., 2018), which manipulated n-gram probabilities based on their occurrence in the fuzzy-matches. Xu et al. (2020) proposed several enhancements using the same architecture, including fine-tuning models

Model	Translation Latency
Transformer	574.02 ms
RAT-CAT	597.28 ms
RAT-SI	597.41 ms

Table 5: Translation latency in milliseconds of RAT-CAT and our model RAT-SI (lower is better). Batch size was set to one to simulate an on-demand system.

and masking out or marking words not related to the input sentence, and matching arbitrarily large n-grams instead of sentences.

More recent work has explored using separate encoders for input and fuzzy-match (He et al., 2021; Cai et al., 2021). He et al. (2021) also considers the realistic scenario where a TM may include noise, while Cai et al. (2021) explores finding target sentences in monolingual data instead of relying on a TM at inference time.

Xia et al. (2019) and Xu et al. (2020) explore aspects of filtering fuzzy-matches by applying similarity thresholds, leveraging word alignment information (Zhang et al., 2018; Xu et al., 2020; He et al., 2021) or re-ranking with additional score (e.g. word overlapping) (Gu et al., 2018; Zhang et al., 2018).

Our work is related to the use of k -nearest-neighbor for NMT (Khandelwal et al., 2021; Zheng et al., 2021) but it is less expensive and does not require storage and search over a large data store of context representations and corresponding target tokens (Meng et al., 2021).

Other works have considered online adaptation outside the context of RAT, including Vilar (2018), who proposes Learning Hidden Unit Contributions (Swietojanski et al., 2016) as a compact way to store many adaptations of the same general-domain model. For an overview of fuzzy-match augmentation outside of NMT, see Li et al. (2022).

Domain adaptation can also be performed offline, typically via fine tuning (Luong and Manning, 2015). Regularization is often applied during fine tuning to avoid catastrophic forgetting (Khayrallah et al., 2018; Thompson et al., 2019a,b).

TMs are commonly used in the localization industry to provide suggestions to translators in order to boost their productivity (Federico et al., 2012). Enhancing translation quality of MT system by leveraging fuzzy-matches extracted from TMs has been explored widely for statistical MT (Koehn and Senellart, 2010; Mathur et al., 2013) and neural MT

systems (Farajian et al., 2017; Gu et al., 2017; Cao and Xiong, 2018; Bulte and Tezcan, 2019).

6 Conclusion

Previous work in retrieval augmented translation has used architectures which either have full connections between source and all fuzzy matches, or independently encode the source and each fuzzy match. Based on our hypothesis that the attention connections between source and each fuzzy match are helpful, but that the connections between different fuzzy matches are harmful, we propose a new architecture (RAT-SI) with the former connections but not the latter. Experiments on several language pairs, domains, and different numbers of fuzzy matches (k) demonstrate that RAT-SI substantially outperforms the prior architectures.

7 Limitations

Due to the availability of domain specific datasets, we perform experiments on two high-resource languages, both out of English. It is unclear if our conclusions would hold on low-resource language pairs. Furthermore, our domains may or may not match real world use cases where an MT customer has their own TM. Real TMs may be significantly larger/smaller, contain multiple domains, etc.

References

- Roe Aharoni and Yoav Goldberg. 2020. [Unsupervised domain clusters in pretrained language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. [Findings of the 2014 workshop on statistical machine translation](#). In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Bram Bulte and Arda Tezcan. 2019. [Neural fuzzy repair: Integrating fuzzy matches into neural machine translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.
- Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. 2021. [Neural machine translation with monolingual translation memory](#). In *Proceedings*

- of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7307–7318, Online. Association for Computational Linguistics.
- Qian Cao and Deyi Xiong. 2018. [Encoding gated translation memory into neural machine translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3042–3047, Brussels, Belgium. Association for Computational Linguistics.
- Tobias Domhan, Michael Denkowski, David Vilar, Xing Niu, Felix Hieber, and Kenneth Heafield. 2020. [The sockeye 2 neural machine translation toolkit at AMTA 2020](#). In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 110–115, Virtual. Association for Machine Translation in the Americas.
- M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. [Multi-domain neural machine translation through unsupervised adaptation](#). In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.
- Marcello Federico, Alessandro Cattelan, and Marco Trombetti. 2012. [Measuring user productivity in machine translation enhanced computer assisted translation](#). In *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas: Research Papers*, San Diego, California, USA. Association for Machine Translation in the Americas.
- Markus Freitag and Yaser Al-Onaizan. 2016. [Fast domain adaptation for neural machine translation](#). *arXiv preprint arXiv:1612.06897*.
- Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2017. [Search engine guided non-parametric neural machine translation](#). *CoRR*, abs/1705.07267.
- Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. [Search engine guided neural machine translation](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5133–5140. AAAI Press.
- Qiuxiang He, Guoping Huang, Qu Cui, Li Li, and Lemao Liu. 2021. [Fast and accurate neural machine translation with translation memory](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3170–3180, Online. Association for Computational Linguistics.
- Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. [Nearest neighbor machine translation](#). In *International Conference on Learning Representations*.
- Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. [Regularized training objective for continued training for domain adaptation in neural machine translation](#). In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 36–44, Melbourne, Australia. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Philipp Koehn and Rebecca Knowles. 2017. [Six challenges for neural machine translation](#). In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.
- Philipp Koehn and Jean Senellart. 2010. [Convergence of translation memory and statistical machine translation](#). In *Proceedings of the Second Joint EM+/CNGL Workshop: Bringing MT to the User: Research on Integrating MT in the Translation Industry*, pages 21–32, Denver, Colorado, USA. Association for Machine Translation in the Americas.
- Taku Kudo and John Richardson. 2018. [Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.
- Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. [A survey on retrieval-augmented text generation](#).
- Minh-Thang Luong and Christopher Manning. 2015. [Stanford neural machine translation systems for spoken language domains](#). In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam.
- Prashant Mathur, Mauro Cettolo, and Marcello Federico. 2013. [Online learning approaches in computer assisted translation](#). In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 301–308, Sofia, Bulgaria. Association for Computational Linguistics.
- Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang, and Jiwei Li. 2021. [Fast nearest neighbor machine translation](#). *CoRR*, abs/2105.14528.

- Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. [compare-mt: A tool for holistic comparison of language generation systems](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. Association for Computational Linguistics.
- MinhQuang Pham, Josep Maria Crego, and François Yvon. 2021. [Revisiting Multi-Domain Machine Translation](#). *Transactions of the Association for Computational Linguistics*, 9:17–35.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Stephen Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: Bm25 and beyond](#). *Found. Trends Inf. Retr.*, 3(4):333–389.
- Pawel Swietojanski, Jinyu Li, and Steve Renals. 2016. Learning hidden unit contributions for unsupervised acoustic model adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(8):1450–1463.
- Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019a. [Overcoming catastrophic forgetting during domain adaptation of neural machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2062–2068, Minneapolis, Minnesota. Association for Computational Linguistics.
- Brian Thompson, Rebecca Knowles, Xuan Zhang, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019b. [HABLex: Human annotated bilingual lexicons for experiments in machine translation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1382–1387, Hong Kong, China. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- David Vilar. 2018. [Learning hidden unit contribution for adapting neural machine translation models](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 500–505, New Orleans, Louisiana. Association for Computational Linguistics.
- Mengzhou Xia, Guoping Huang, Lemao Liu, and Shuming Shi. 2019. [Graph based translation memory for neural machine translation](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7297–7304.
- Jitao Xu, Josep Crego, and Jean Senellart. 2020. [Boosting neural machine translation with similar translations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.
- Jingyi Zhang, Masao Utiyama, Eiichiro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. [Guiding neural machine translation with retrieved translation pieces](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1325–1335, New Orleans, Louisiana. Association for Computational Linguistics.
- Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. [Adaptive nearest neighbor machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 368–374, Online. Association for Computational Linguistics.

CALM-Bench: A Multi-task Benchmark for Evaluating Causality Aware Language Models

Dhairya Dalal[†] and Mihael Arcan[‡] and Paul Buitelaar^{†‡}

[†]SFI Centre for Research and Training in Artificial Intelligence

^{†‡}Insight SFI Research Centre for Data Analytics

Data Science Institute, University of Galway

d.dalal1@nuigalway.ie,

{mihael.arcana,paul.buitelaar}@universityofgalway.ie

Abstract

Causal reasoning is a critical component of human cognition and is required across a range of question-answering (QA) tasks (such as abductive reasoning, commonsense QA, and procedural reasoning). Research on causal QA has been underdefined, task-specific, and limited in complexity. Recent advances in foundation language models (such as BERT, ERNIE, and T5) have shown the efficacy of pre-trained models across diverse QA tasks. However, there is limited research exploring the causal reasoning capabilities of those language models and no standard evaluation benchmark. To unify causal QA research, we propose *CALM-Bench*, a multi-task benchmark for evaluating causality-aware language models (CALM). We present a standardized definition of causal QA tasks and show empirically that causal reasoning can be generalized and transferred across different QA tasks. Additionally, we share a strong multi-task baseline model which outperforms single-task fine-tuned models on the *CALM-Bench* tasks.

1 Introduction

Causal reasoning is a crucial aspect of human cognition and is critical to the development of our mental models of reality (Neeleman et al., 2012; Johnson-Laird and Khemlani, 2017; Griffiths, 2017). Theories of causation have been studied extensively across philosophy (Beebe et al., 2009), physics (Dowe, 2009), cognitive science (Waldmann, 2017), and probability and statistics (Pearl, 2009), amongst many other fields. Explorations of causality in the language domain tend to be semantic, linguistic, or logical in nature as access to direct observational data or event probabilities is not assumed nor is required. Descriptions of causality can be linguistically valid but factually incorrect (e.g. butter is the leading cause of factory deaths). Therefore, causal reasoning in language should ideally be logically consistent and grounded

Premise: Air pollution in the city worsened. Question: What is the CAUSE of this? Alternative 1: Factories increased their production Alternative 2: Factories shut down.
1. Identify Causal Concepts Air pollution, factories, and production
2. Causal Knowledge Linking Air pollution is the introduction of toxic substances and poisonous gasses into the air which make it harmful for humans and other living beings to breathe. Industrial factories release chemical byproducts and harmful gasses into the atmosphere during the production process.
3. Reasoning over causal knowledge Factory, cause-effect, pollution Increased production, cause-effect, air pollution

Figure 1: An CQA example from the COPA (Gordon et al., 2012). CQA requires identifying causal concepts, linking those concepts to causal relations, and reasoning over those relations.

in commonsense knowledge. The *counterfactual theory of causation* (Lewis, 1973) provides a useful definition of causation for language applications. It posits that causation is relational (there is a cause and effect), temporal (the cause must precede the effect), and counterfactual (if the causing event had not occurred, the effect would not have occurred). Various natural language processing (NLP) applications require identifying causal relations and reasoning over those relations.

These NLP applications can be split into two general categories: causal relation identification (CRI) and causal question-answering (CQA). CRI tasks aim to identify and extract cause/effect spans from descriptions of causal events. CRI requires linguistic knowledge - relying on lexical triggers (i.e. causative verbs and causal connectives) and grammatical structures (Neeleman et al., 2012; Girju, 2003). Historically, the majority of NLP research on causality has focused on CRI.

In contrast to CRI, CQA tasks require both background causal knowledge and reasoning. Consider the question *Air pollution in the city worsened. What is the cause of this?* (Figure 1). To answer this question, commonsense knowledge about factories, pollution, and the ability to infer both causal and counterfactual relations is required. General

Task	Example	Size	Question Type	Format	Knowledge
aNLI (Bhagavatula et al., 2020)	Context: Jessie wants to save the planet. This summer has been the hottest in all history. Question: Which hypothesis is the most plausible for the provided observations? A: Jessie decides to buy a new truck. B: Jessie decides to sell her truck and use public transportation instead.	174,226 Train: 169,654 Val: 1,532 Test: 3,040	cause prediction	multiple-choice	social, world
COPA (Gordon et al., 2012)	Question: Air pollution in the city worsened. What is the cause of this? A: Factories increased their production. B: Factories shut down.	1,000 Train: 800* Val: 200 Test: 500	cause prediction effect prediction	multiple-choice	world
CosmosQA (Huang et al., 2019)	Context: Two things happened today in Beijing. First off, incoming journalists were amazed to find China had successfully lifted the brown haze in city. Skies were crystal blue and the air felt noticeably lighter. Question: Why did the sky appear clearer? A: None of the above choices. B: The citizens learned to ignore the gloomy skies. C: The citizens made an effort to cut down on pollution. D: A large storm had recently passed.	35,210 Train: 25,262 Val: 2,985 Test: 6,963	cause prediction effect prediction	multiple-choice	social, world
E-Care (Du et al., 2022)	Question: The city is determined to control air pollution. What is the effect? A: They have to reduce the number of automobiles. B: Environmental pollution has been increased.	17,051 Train: 14,929 Val: 2,122 Test: blind	cause prediction effect prediction	multiple-choice	social, world, science
ROPES (Lin et al., 2019)	Context: There are two planets, Glarnak and Bornak, that share the same atmospheric composition. The planets have nearly identical ecosystems and topography. The main difference between the two planets is the level of global warming on each planet. Glarnak is experiencing a strong impact from global warming. Bornak, though, is experiencing practically no effects of global warming. Question: Which planet has more pollutants in the atmosphere? Glarnak	14,322 Train: 10,924 Val: 1,688 Test: 1,710	cause prediction cause comparison effect prediction effect comparison	reading comprehension	science, world
WIQA (Tandon et al., 2019)	Context: 1. A seed is in soil. 2. The seed germinates. 3. The plant grows roots. 4. The plant grows out of the ground. 5. The plant gets bigger. 6. The plant flowers. 7. The flower produces fruit. 8. The fruit releases seeds. 9. The plant dies. Question: Suppose less pollution in the environment happens, how will it affect the population of plants? A: More B: Less C: No Effect	39,705 Train: 29,808 Val: 6,894 Test: 3,003	effect prediction	multiple-choice	science, world

Table 1: CALM-Bench is a multi-task causal question answering benchmark consisting of six diverse QA tasks requiring both causal reasoning and knowledge.

work on CQA is often under-defined and limited based on the task definition. For example, previous work defined CQA as answering variations of *What is the cause/effect of X?* style questions where the model had to select the most plausible cause or effect from a set of candidate options. While this task requires causal knowledge, it could be recast as an information retrieval problem with no further requirement of causal reasoning. A stronger definition of CQA would allow for more principled explorations of causal reasoning (e.g. reasoning over causal chains, abductive inference, counterfactual reasoning, etc) and aid in the development of stronger NLP models.

Recent advances in foundation language models have demonstrated the effectiveness of pre-trained models across a wide range of NLP and general language understanding tasks. The term *foundation model* (Bommasani et al., 2021) describes any monolithic neural model (e.g. BERT (Devlin et al., 2019)) that captures general knowledge through pre-training and is able to transfer that knowledge to a wide range of downstream tasks. Foundation language models exhibit general reasoning capabilities (Clark et al., 2021), factual knowledge recall (Petroni et al., 2019), and superior performance on a wide range of QA tasks (Khashabi et al., 2020; He et al., 2021; Lourie et al., 2021a). Knowledge in foundation language models is usually injected through denoising objectives (e.g. masked token prediction) (Sun et al., 2020). However, interpreting and extracting that knowledge is difficult (requiring specialized probing tasks) and these

models can be susceptible to exploiting superficial (Kavumba et al., 2019). CQA tasks could provide a unique opportunity to develop both explainable models (through producing causal explanation chains) and expand the reasoning capabilities of those models in QA settings. To date, no comprehensive study has explored the causal reasoning capabilities of foundation language models.

We aim to unify research around CQA research by providing a definition for CQA rooted in the cognitive understanding of causal learning and propose *CALM-Bench*, a multi-task causal question-answering benchmark for evaluating causality-aware language models (CALM). *CALM-Bench* (Table 1) consists of six different QA tasks (aNLI (Bhagavatula et al., 2020), COPA (Gordon et al., 2012), CosmosQA (Huang et al., 2019), E-Care (Du et al., 2022), WIQA (Tandon et al., 2019), and ROPES (Lin et al., 2019)) that require both causal knowledge and causal reasoning. We show empirically that causal reasoning can be generalized across the different tasks in *CALM-Bench*. We present a multi-task learning (MTL) setup that outperforms all single-task fine-tuned baselines and demonstrates strong results on the COPA task in a zero-shot setting. Relevant details about the code and model weights can be found on GitHub ¹.

2 Causal question-answering

We define CQA broadly as any QA task which requires both *causal reasoning* and *causal knowl-*

¹<https://github.com/dhairyalal/CALM-Bench>

edge provided a real or hypothetical description of events. Cognitive theories of causal learning provide a framework for understanding and evaluating the process of causal question-answering in NLP applications. The inferential theory of causal learning posits that causal learning is a slow and effortful cognitive process that involves drawing causal conclusions over propositional premises (Boddez et al., 2017).

Propositions represent our causal knowledge and contain both qualified relational information (e.g. increase of greenhouse gasses in the atmosphere causes global warming) and propositional beliefs (I believe that greenhouse gasses cause global warming). Propositions are compositional (given the propositions: factories cause air pollution and pollution leads to global warming, we can infer that factories cause global warming) and directional (i.e. we would not infer that global warming causes factories). A key aspect of causal learning is the ability to generalize specific causal knowledge to new situations which is known as causal mechanism knowledge. (Johnson and Ahn, 2017; Ahn et al., 1995).

Causal mechanism knowledge is the mental representation of a system of physical or abstract parts/processes and the expectation of causal interactions between those components that can be generalized to new situations. For example, an arson investigator relies on their mechanism knowledge of fire catalysts and forensic experience to ascertain human involvement. Causal mechanism knowledge can be succinctly represented as propositional statements. Causal bridging inferences describe the relationship between causal knowledge and reasoning. Singer et al. (1992) found that individuals invoke causal statements to bridge two events and then validate those statements against prior commonsense and causal knowledge. For example, given the events *Anna added butter to the hot pan.* and *The butter melted.*, we implicitly invoke the bridging statement *heat caused the butter to melt* based on our prior knowledge.

Solving CQA tasks can be decomposed into three general steps: *causal concept identification*, *causal knowledge linking*, and *causal reasoning*. Consider Figure 1, the causal concepts of air pollution and factories are identified and then linked to background knowledge in order to produce causal knowledge. Causal knowledge can be expressed as relational triples (e.g. factory, cause-effect, pol-

lution) which are effectively propositional statements. The final step requires reasoning over that knowledge through both inferential and counterfactual reasoning. We infer that the increase in factory production results in worsening air pollution based on causal knowledge that factory production causes pollution. The counterfactual, if factories shut down then air pollution would not increase, allows us to eliminate the second option. Arriving at the correct answer in this example is difficult without any background causal knowledge and reasoning over that knowledge.

An important aspect of causal learning is the ability to generalize causal mechanism knowledge to novel situations and task settings. We can see in Table 1 that while thematically all the examples are about the causal relationship between global warming and air pollution, each question requires different types of reasoning over the same knowledge. With the aNLI example, global warming is not mentioned explicitly but must be inferred from social commonsense knowledge (i.e. through the bridging inferences that *saving the planet* and *the hottest summer* are related to global warming) and then use abductive reasoning to select the most plausible hypothesis. The COPA example requires counterfactual reasoning to eliminate the option that factories shutting down would not contribute to air pollution and inferential reasoning to infer that increased factory production results in more air pollution. The WIQA example requires both understanding the life cycle of a plant as a procedural chain and predicting the magnitude impact of environmental pollution as a downstream effect on the plant population. Finally, the ROPES example involves generalizing mechanism knowledge to a fictional setting in order to identify which planet is more likely to have pollutants in the air.

CALM-bench consists of diverse QA tasks requiring social, world, and science knowledge. Our empirical experiments aim to validate the assumption that causal reasoning is transferable across these QA tasks in *CALM-Bench* and produce strong baselines for future research in this space.

3 Related Work

3.1 Causal question-answering

COPA was one of the first QA benchmark tasks which required both background commonsense knowledge and causal reasoning. It is also included as part of the SuperGlue (Wang et al., 2019) bench-

mark. COPA can be considered solved by modern massive foundation models which achieve near human performance (99% accuracy). However, these models are very large (the top three models having more than 10 billion+ parameters), are trained on multi-terabyte scale corpora, and require significant computing resources. Sharp et al. (2016) constructed the first CQA dataset from the Yahoo! Answers corpus using the templates *What causes ...* and *What is the result of ...* to identify causal questions. Sharp et al. (2016) and Xie and Mu (2019) investigated different strategies for training distributed causal embeddings for re-ranking answer options for those causal questions. Hassanzadeh et al. (2019) and Kayesh et al. (2020) explored binary causal questions (i.e. could X cause y) answering using a mixture of co-occurrence statistics and cosine similarity threshold derived from fixed BERT embeddings. The proposed solutions were specific to the task format (i.e. learning threshold values for predicting the yes option). causalqa introduced CausalQA, a corpus of 1.1 million causality-related questions and answers extracted from various datasets primarily related to open-domain web queries (e.g. GooAQ (Khashabi et al., 2021), MS-Marco (Nguyen et al., 2016)). Causal questions were identified using templates spanning *What*, *How*, and *Why* style questions whose intent is to enquire about causes and effects.

Both the CausalQA and the Yahoo Answers! causal questions focus on causal knowledge retrieval or basic reading comprehension without further requirement of causal reasoning. Causal knowledge retrieval can be generalized to information retrieval where the goal is to ensure the retrieved passage contains causal explanations related to the query. Here linguistic cues (Khoo et al., 1998; Girju et al., 2007; Neeleman et al., 2012) or semantic similarity (Dalal et al., 2021b) can be used to identify relevant passages. Likewise, answering *What*, *How*, and *Why* style questions in the context of reading comprehension (e.g. SQuAD (Rajpurkar et al., 2016)) focus more on the lexical overlap between the question and supporting text and linguistic cues associated with the question typologies. CALM-Bench aims to address this gap by focusing QA tasks that require both causal knowledge and causal reasoning.

Most recently, CQA research has investigated augmenting foundation language models with external knowledge for CQA. Dalal et al. (2021a)

proposes augmentation with external causal knowledge graph embeddings derived from CauseNet (Heindorf et al., 2020) for QA on the COPA and WIQA tasks and Hosseini et al. (2022) explores injecting the commonsense knowledge from the ATOMIC (Sap et al., 2019) commonsense knowledge base using the BERT masked language modeling pretraining objective for the COPA task. Recent interest in question-answering has led to the development of many large-scale and complex QA tasks. *CALM-bench* consists of curated tasks that require causal reasoning and are described in Section 4.

3.2 Commonsense Reasoning

Commonsense reasoning is closely related to CQA and can be considered a broader superset of CQA depending on the task. Several of the *CALM-Bench* tasks (aNLI, COPA, CosmosQA, and E-CARE) require causal reasoning over commonsense knowledge, and the aNLI, COPA, and CosmosQA tasks were first introduced as commonsense QA tasks. Recent work on commonsense reasoning has focused on probing commonsense knowledge found in foundation language models (Zhou et al., 2020), strategies for effective knowledge augmentation (Fan et al., 2020), and the generation of commonsense knowledge (Bosselut et al., 2019). Lourie et al. (2021b) introduced the first multi-task commonsense QA benchmark (RAINBOW) and a universal model (UNICORN) for general commonsense QA. UNICORN is a T5-11b model (Raffel et al., 2020) trained on the RAINBOW multi-set tasks and fine-tuned in a multi-task setting. Our approach and motivation for multi-task CQA benchmark were greatly inspired by (Lourie et al., 2021b). *CALM-bench* shares two of its tasks (aNLI and CosmosQA) with the RAINBOW benchmark and we consider multi-task learning in our experiments.

3.3 Causal Relation Identification

CRI is often the first step for aggregating causal knowledge when building automated CQA systems (Hassanzadeh et al., 2020). Extracted causal relations are often useful for generating causal knowledge graphs (Heindorf et al., 2020) and developing causal knowledge representations (Sharp et al., 2016; Dalal et al., 2021a) which can be used to improve model performance in CQA tasks. CRI tasks have been studied extensively in the computational linguistics and NLP domain (Yang et al., 2022; Drury et al., 2022). Early methods relied on lexical triggers and linguistic cues (Khoo et al.,

1998; Girju et al., 2007; Neeleman et al., 2012). More recent approaches have explored using neural methods with word embedding features (Dasgupta et al., 2018), self-supervision (Zuo et al., 2021), and external knowledge (Liu et al., 2020). Several efforts have been undertaken to unify CRI research. Tan et al. (2022) introduced the UniCausal benchmark which consolidates six annotated CRI corpora across the tasks of causal sequence classification, cause-effect span detection, and causal pair classification. (Hosseini et al., 2021) introduced the CREST schema and toolkit which converts thirteen commonly used CRI datasets into a unified format.

4 CALM-Bench Tasks

CALM-Bench (Table 1) consists of five multiple-choice tasks (aNLI, COPA, Cosmos QA, E-Care, and WIQA) and a reading comprehension task (ROPES). These tasks require diverse causal knowledge which can be broadly summarized as social (sociological norms of human behavior), world (general commonsense knowledge), and science (specific scientific knowledge of natural processes such as the precipitation cycle or plant life cycle). Questions either require predicting the cause or effect (i.e. cause and effect prediction) provided a description of events or comparing entities (i.e. cause and effect comparison) in a causal system.

Abductive Natural Language Inference (aNLI) (Bhagavatula et al., 2020) is an abductive reasoning task over narratives of social situations. Provided a sequential pair of social observations, the model must predict which of the two provided hypotheses best explains the observations.

Choice of Plausible Alternatives (COPA) (Gordon et al., 2012) is a commonsense causal reasoning task. Provided a premise, the goal is to select the most likely cause or effect from a pair of options. (Kavumba et al., 2019) introduced 500 additional training examples in Balanced-COPA to mitigate the corpus-level artifacts that were likely to be exploited by language models during fine-tuning.

COSMOS QA (Huang et al., 2019) is a multiple-choice QA task requiring social commonsense knowledge. Provided a narrative about people in everyday situations, the goal is to identify the most plausible cause or effect about agents in the story.

E-Care (Du et al., 2022) consists of two causal reasoning tasks. The first task, similar to COPA, requires identifying the most likely cause or effect

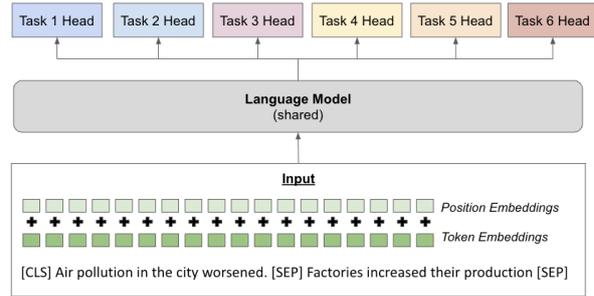


Figure 2: Our MTL model adapts the hard-parameter sharing architecture (Baxter, 2004) where the language model is shared across all the task heads. During training, the task losses are averaged and backpropagated to produce causality-aware contextual embeddings which are effective across all the *CALM-Bench* tasks (Table 4).

of the provided premise. The second task requires generating a causal explanation of the correct answer option. We only consider the first task as part of *CALM-Bench*.

Reasoning over Paragraph Effects (ROPES) (Lin et al., 2019) is a reading comprehension task. Provided a knowledge passage, the model is required to reason over the causal and qualitative relations in the passage and apply them to answering questions about a hypothetical situation. 70% of background passages contain causal relations and 26% contain both causal and qualitative relations.

What If question-answering (WIQA) (Tandon et al., 2019) is a multiple-choice QA task requiring reasoning over procedural descriptions of natural processes. WIQA requires predicting the downstream magnitude (more, less, no effect) effect of a perturbation to an individual step in the procedural chain.

5 Methodology

5.1 Language Models

Our experiments consider two different foundation language models, BERT (Devlin et al., 2019) and ERNIE 2.0 (Sun et al., 2020). BERT and derivative models (e.g. RoBERTa (Liu et al., 2019b), DeBERTa (He et al., 2021), etc) contain unspecified distributional knowledge which is learned through the random masked language modeling pretraining objective. In a contrast, ERNIE 2.0 injects external knowledge through a variety of pretraining objectives including masked knowledge prediction, discourse relation prediction, and the IR relevance task. ERNIE 2.0’s underlying transformer encoder has the same architecture and parameters as the

BERT Transfer Results						
Trained On ↓ Evaluated On ⇒	<i>aNLI</i>	<i>COPA</i>	<i>CosmosQA</i>	<i>E-Care</i>	<i>WIQA</i>	<i>ROPES</i>
<i>Single Task FT Baseline</i>	0.61	0.64	0.57	0.76	0.65	0.58
<i>aNLI</i>	-	+0.11	+0.04	0	-0.01	+0.01
<i>COPA</i>	+0.02	-	+0.04	-0.04	0	-0.06
<i>CosmosQA</i>	+0.01	+0.05	-	0	-0.01	+0.02
<i>E-Care</i>	+0.02	+0.13	-0.02	-	-0.02	-0.05
<i>WIQA</i>	0	0	+0.03	-0.02	-	-0.05
<i>ROPES</i>	+0.02	+0.07	+0.03	-0.04	-0.02	-

Table 2: This table contains the transfer learning results for the BERT model. Results are read across the rows where the first column in each row contains the base task selected for transfer learning and the remainder of the columns are the evaluation results across the target tasks. We provide the single-task finetuned baseline in the second row and the pp difference between for each experiment. All results presented are accuracy scores with exception of ROPES which is exact match.

ERNIE 2.0 Transfer Results						
Trained on ↓ Evaluated On ⇒	<i>aNLI</i>	<i>COPA</i>	<i>CosmosQA</i>	<i>E-Care</i>	<i>WIQA</i>	<i>ROPES</i>
<i>Single Task FT Baseline</i>	0.64	0.71	0.63	0.76	0.64	0.53
<i>aNLI</i>	-	+0.07	0	+0.02	+0.01	+0.08
<i>COPA</i>	0	-	-0.01	+0.01	+0.02	-0.03
<i>CosmosQA</i>	+0.02	+0.01	-	0	+0.02	-0.12
<i>E-Care</i>	0	+0.08	+0.02	-	+0.02	+0.11
<i>WIQA</i>	0	+0.01	0	-0.01	-	+0.03
<i>ROPES</i>	+0.02	-0.06	-0.01	+0.01	+0.02	-

Table 3: This table contains the transfer learning results for the ERNIE 2.0 model. In contrast the BERT model, we observe general consistent positive improvement across nearly all tasks. This suggests that language models with grounded knowledge tend to both do better on CQA tasks and are able to transfer causal reasoning across tasks more effectively.

BERT model and is trained on similar data. ERNIE 2.0 is trained on additional Reddit and Discovery data but the primary difference is in its knowledge-focused pretraining objectives.

We hypothesize that ERNIE 2.0 will outperform BERT across the CQA task as grounded knowledge is a requisite for causal reasoning in our definition. The BERT and ERNIE 2.0 implementations come from the Huggingface Transformers library (Wolf et al., 2020). We use the pretrained base models for both (bert-base-uncased² and

nghuyong/ernie-2.0-base-en respectively³).

5.2 Language Model Training

Single-task fine-tuning and multi-task fine-tuning are used to train our models on the CQA tasks. Sequential fine-tuning (Pratt, 1992) was also investigated but found to be inconsistent and not as effective as the other methods (Appendix A.6.2). Following the task head paradigm introduced in Devlin et al. (2019), we develop separate classification heads for each task (see Appendix A.1 for

²<https://huggingface.co/bert-base-uncased>

³<https://huggingface.co/nghuyong/ernie-2.0-base-en>

	<i>aNLI</i>	<i>COPA</i>	<i>CosmosQA</i>	<i>E-Care</i>	<i>ROPES</i>	<i>WIQA</i>	Score
Fine-tuned Baseline							
Bert-base	0.61	0.64	0.57	0.76	0.58	0.65	0.64
ERNIE-base	0.64	0.71	0.63	0.76	0.53	0.64	0.65
MTL Baseline							
Bert-base MTL	0.62	0.75	0.58	0.72	0.61	0.72	0.67
ERNIE-base MTL	0.65	0.80	0.65	0.78	0.58	0.77	0.71

Table 4: We present the baselines results for CALM-bench. All the task are evaluated using the accuracy metric with the exception of ROPES which displays exact match. Results are presented for the test sets for COPA and WIQA and on validations sets for aNLI, CosmosQA, E-Care, and ROPES. We find that MTL models outperform the single-task finetuned models consistently with ERNIE-base MTL model having the best results.

more details). The pooled *CLS* embedding from the last layer in the language model is fed into the classification head to map the language model’s contextualized output into the task’s classification space. In the single-task setting, each task is trained independently. The cross-entropy loss is calculated per training batch and back-propagated through all the layers in the language model.

For the multi-task learning (MTL) model, we adapt a hard-parameter sharing model (Baxter, 2004) and train it using the multi-task fine-tuning strategy (Liu et al., 2019a). Our MTL model (Figure 2) consists of a shared base language model and separate task heads for tasks in CALM-Bench. For each train step, a train batch is sampled for each task and the task-specific losses are calculated. The task losses are averaged before backpropagation. The MTL model is trained for 8,000 steps on the aNLI, CosmosQA, E-Care, and WIQA tasks. The ROPES task is not included in training as its format is significantly different from the multiple-choice tasks and resulted in lower performance in our early experiments. COPA was also omitted from the MTL training given its small size (800 training examples) and instead saved for zero-shot evaluation. At evaluation time, we fine-tune the MTL model on each target task for one additional epoch and then evaluate the model on the target evaluation set.

A hyperparameter search is run to identify the optimal random seed and the learning rate for each task (see Appendix A.5.1). Four of the tasks (aNLI, CosmosQA, E-CARE, and ROPES) have private test sets and a public leaderboard. For these tasks, we treat the validation set as the test set during evaluation and generate a new validation split from the training data to be used for training validation. The general intuition is that fine-tuned language models

should have the best task-specific performance. If causal reasoning is transferrable, we should see improvements over the single-task fine-tuned models in both the transfer learning and multi-task learning experiments.

6 Empirical Findings

6.1 Single-task Fine-tuned Baselines

The baseline results for the single-task fine-tuned language models for all tasks can be found in Table 4. We find the ERNIE model on average outperforms the BERT model across most of the CQA tasks with an average improvement of 5.3pp on the aNLI, COPA, and CosmosQA tasks. However, ERNIE does underperform the BERT model on both the ROPES and WIQA tasks and shows no improvement on the E-Care task. These results are used as the baseline for the transfer learning and MTL experiments in Table 4.

6.2 Transferability of Causal Reasoning

We conduct sixty experiments to see if causal reasoning can be generalized and transferred across the QA tasks in CALM-Bench. For each experiment, we select a base task (e.g. aNLI) and a different target evaluation task (e.g. COPA). The language model is first fine-tuned on the base task and then fine-tuned on the target task. That model is then evaluated on the target task. Each transfer learning experiment is independent and the final results are summarized in Table 2 and 3.

Across both BERT and ERNIE 2.0 models, we observe that task-specific causal knowledge and reasoning are transferable. However, the pattern of transference differs across both models.

For the BERT model, the E-Care, WIQA, and ROPES tasks generally see degradation in accuracy

and exact match. However, there is improvement across aNLI, COPA, and CosmosQA tasks with COPA receiving an average of 7pp gain. We hypothesize this may due to two factors. As noted earlier, there is no grounded knowledge in BERT. BERT has to learn both task-specific knowledge and reasoning processes associated with each task. Tasks with similar knowledge requirements (aNLI, COPA, and CosmosQA) benefit from each other and the shared task format (multiple-choice). In contrast, the ROPES and WIQA tasks have different task heads and knowledge requirements. BERT is likely suffering from catastrophic forgetting when fine-tuning on the target task.

In contrast, we find consistent general improvement across all the tasks with the ERNIE 2.0 model. ERNIE 2.0 contains grounded knowledge which allows for better transfer learning across the tasks. This was observed with WIQA and ROPES seeing average improvements of 1.8pp and 1.4pp in contrast to the average losses of -0.08pp and -1.5pp with the BERT model.

To summarize, we provide empirical evidence that causal reasoning and knowledge can be transferred across different CQA tasks. We further find validate our assumptions that CQA requires both reasoning capabilities and grounded knowledge as the knowledge-rich ERNIE demonstrates more consistent improvement across the *CALM-Bench* tasks.

6.3 Multi-Task Learning Results

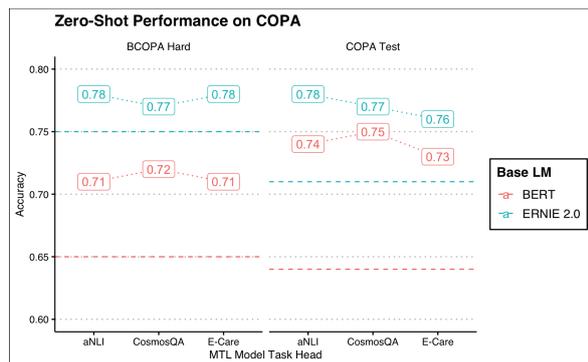


Figure 3: Zero-shot results on the COPA task. We present results for both the primary COPA Test set and the BCOPA Hard (Kavumba et al., 2019) subset. The dashed lines are the single-task fine-tuned baselines. Despite each task head having not seen the COPA examples during the MTL training, they outperform the single-task fine-tuned baselines.

In Table 4, we provide the baseline results for both the single-task fine-tuned and multi-task mod-

els on the CALM-Benchmark. The score column equally averages all metrics to provide a single value for comparing the different approaches. The MTL baselines outperform all the single-task fine-tuned baselines with the ERNIE MTL model providing the best results. These results further corroborate our claim that causal knowledge and reasoning are generalizable across diverse QA tasks. However, we do observe that task format matters. The inclusion of ROPES (a reading comprehension task) during multi-task training resulted in generally lower performance. As a result, our final MTL model was only trained on the subset of multiple-choice tasks (aNLI, CosmosQA, E-Care, and WIQA). Future work may consider alternative ways to weight task-specific losses or different model architectures (e.g. T5 (Raffel et al., 2020)) which can map all tasks to the same text-to-text format.

In the context of multiple-choice CQA, we find consistent and positive improvement across all tasks in both the single-task transfer learning and multi-task learning scenarios. We run an additional zero-shot experiment where is task head in the MTL model is used to evaluate the COPA test and BCOPA hard test examples. Figure 3 shows that both the BERT and ERNIE single-task fine-tuned baselines are outperformed by an average of +10pp and +6.6pp on the test set and see an average of +6.3pp and +1.3pp improvement on the BCOPA hard subset. For comparison, Hosseini et al. (2022) fine-tune a BERT large (345 million parameters) model on 780,000 knowledge triples from the ATOMIC commonsense knowledge base. Their BERT-Large-ATOMIC model achieves 88% accuracy on the COPA test set and 73% accuracy on the BCOPA hard subset. Our smaller ERNIE 2.0 MTL model achieves 80% fine-tuned accuracy on the COPA test set with fewer parameters (110 million) and less training data. Further, our MTL model outperforms the BERT-Large-ATOMIC model on the BCOPA hard subset with the zero-shot MTL heads averaging around 77% accuracy and fine-tuned model achieving 79% accuracy.

7 Conclusion

In this paper, we provide a unified definition of causal question-answering in the context of natural language applications. Drawing from the cognitive science literature, we posit that CQA tasks

require both causal reasoning and causal knowledge. Based on this definition, we introduce the *CALM-bench*, the first multi-task CQA benchmark to evaluate the general causal reasoning capabilities of foundation language models. We provide empirical evidence which validates the intuition that causal reasoning and knowledge are transferable across the CQA tasks. Knowledge-enriched language models like ERNIE are likely to outperform distributional models (i.e. BERT) across all tasks in both the single-task fine-tuning and multi-task fine-tuning settings. Finally, we provide a set of strong baselines for future work exploring causal question-answering and the causal reasoning capabilities of language models.

While our experiments show causal knowledge is transferable, these models are still opaque. CQA provides a unique opportunity for model explainability through causal explanation structures and reasoning chains. The E-Care and WIQA task have annotated explanations that provide a useful starting point. Causal knowledge sources like CauseNet (Heindorf et al., 2020), ConceptNet (Speer et al., 2017), and Wikidata⁴ can also be used to generate causal explanations. We believe the next evolution of foundation language models will have stronger causal reasoning capabilities and implicit structured causal knowledge. CALM-bench provides a starting point for further research on causal question-answering.

Limitations

Our research assumes the English language due to the lack of multi-lingual QA datasets. Future work may consider developing CQA tasks in other languages.

Additionally, we used the base models for BERT and ERNIE 2.0 in our experiments for all experiments. The public leaderboards for most of the tasks in *CALM-Bench* feature larger models with the billion parameter plus models occupying the top spots. Future work can explore scaling our experimental setup to the large and extra-large versions of our language models used as well as considering more modern architectures such DeBERTa (He et al., 2021) and ERNIE 3.0 (Sun et al., 2021). A challenge for multi-task training with large models is that the batch size for each task must be significantly reduced to ensure the model fits in GPU memory. Smaller batch sizes lead to unstable

training and convergence. Tricks like gradient accumulation and modern optimization libraries (e.g. DeepSpeed⁵ and Fairscale⁶) can be explored.

Finally, our multi-task model is not truly universal in the sense that a new task head is required for each additional CQA task. While there is transferability across the multiple-choice formats, the model does struggle to generalize causal reasoning across different formats like reading comprehension. Our encoder-only approach is unable to handle generation tasks. As a result, the E-CARE and aNLI explanation tasks are excluded. Lourie et al. (2021b) found success using encoder-decoder models where all tasks are converted to a text-to-text format. While (Lourie et al., 2021b) only considered multiple-choice tasks, future work could explore including reading comprehension and explanation generation tasks using models like UnifiedQA (Khashabi et al., 2020) and T5.

Acknowledgements

This work was supported by the Science Foundation Ireland under grants SFI/18/CRT/6223 (Centre for Research Training in Artificial Intelligence), SFI/12/RC/2289_P2 (Insight), and co-funded by the European Regional Development Fund.

References

- Woosung Ahn, Charles W. Kalish, Douglas L. Medin, and Susan A. Gelman. 1995. The role of covariation versus mechanism information in causal attribution. *Cognition*, 54(3):299–352.
- Jonathan Baxter. 2004. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28:7–39.
- Helen Beebe, Christopher Hitchcock, and Peter Menzies. 2009. Introduction. In *The Oxford Handbook of Causation*. Oxford University Press.
- Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.
- Yannick Boddez, Jan De Houwer, and Tom Beckers. 2017. The inferential reasoning theory of causal learning: Toward a multi-process prepositional account. *Oxford library of psychology.*, pages 53–64. New York, NY, US.

⁵<https://github.com/microsoft/DeepSpeed>

⁶<https://github.com/facebookresearch/fairscale>

⁴<https://www.wikidata.org/>

- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.
- Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*.
- Dhairya Dalal, Mihael Arcan, and Paul Buitelaar. 2021a. Enhancing multiple-choice question answering with causal knowledge. In *Proceedings of Deep Learning Inside Out (DeeLIO): The Second Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. Association for Computational Linguistics.
- Dhairya Dalal, Sharmi Dev Gupta, and Bentolhoda Binaei. 2021b. A semantic search pipeline for causality-driven adhoc information retrieval.
- Tirthankar Dasgupta, Rupsa Saha, Lipika Dey, and Abir Naskar. 2018. Automatic extraction of causal relations from text using linguistically informed deep neural networks. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 306–316. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.
- Phil Dowe. 2009. Causal Process Theories. In *The Oxford Handbook of Causation*. Oxford University Press.
- Brett Drury, Hugo Gonalo Oliveira, and Alneu de Andrade Lopes. 2022. A survey of the extraction and applications of causal relations. *Natural Language Engineering*, 28(3):361–400.
- Li Du, Xiao Ding, Kai Xiong, Ting Liu, and Bing Qin. 2022. e-CARE: a new dataset for exploring explainable causal reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- William Falcon and The PyTorch Lightning team. 2019. *PyTorch Lightning*.
- Zhihao Fan, Yeyun Gong, Zhongyu Wei, Siyuan Wang, Yameng Huang, Jian Jiao, Xuanjing Huang, Nan Duan, and Ruofei Zhang. 2020. An enhanced knowledge injection model for commonsense generation. *CoRR*, abs/2012.00366.
- Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 76–83. Association for Computational Linguistics.
- Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 13–18, Prague, Czech Republic. Association for Computational Linguistics.
- Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics.
- Thomas L Griffiths. 2017. Formalizing prior knowledge in causal induction. *The oxford handbook of causal reasoning*, pages 115–126.
- Oktie Hassanzadeh, Debarun Bhattacharjya, Mark Feblowitz, Kavitha Srinivas, Michael Perrone, Shirin Sohrabi, and Michael Katz. 2019. Answering binary causal questions through large-scale text mining: An evaluation using cause-effect pairs from human experts. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5003–5009. International Joint Conferences on Artificial Intelligence Organization.
- Oktie Hassanzadeh, Debarun Bhattacharjya, Mark Feblowitz, Kavitha Srinivas, Michael Perrone, Shirin Sohrabi, and Michael Katz. 2020. Causal knowledge extraction through large-scale text mining. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13610–13611.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Wei Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *2021 International Conference on Learning Representations*.
- Stefan Heindorf, Yan Scholten, Henning Wachsmuth, Axel-Cyrille Ngonga Ngomo, and Martin Potthast. 2020. Causenet: Towards a causality graph extracted from the web. In *CIKM*. ACM.

- Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.
- Pedram Hosseini, David A Broniatowski, and Mona Diab. 2021. Predicting directionality in causal relations in text. *arXiv preprint arXiv:2103.13606*.
- Pedram Hosseini, David A. Broniatowski, and Mona Diab. 2022. Knowledge-augmented language models for cause-effect relation classification. In *Proceedings of the First Workshop on Commonsense Representation and Reasoning (CSRR 2022)*, pages 43–48. Association for Computational Linguistics.
- Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: machine reading comprehension with contextual commonsense reasoning. *CoRR*, abs/1909.00277.
- Samuel G. B. Johnson and Woo-kyoung Ahn. 2017. 127Causal Mechanisms. In *The Oxford Handbook of Causal Reasoning*. Oxford University Press.
- Philip Nicholas Johnson-Laird and Sangeet Khemlani. 2017. Mental models and causation. *Oxford handbook of causal reasoning*, pages 1–42.
- Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reisert, and Kentaro Inui. 2019. When choosing plausible alternatives, clever hans can be clever. *arXiv preprint arXiv:1911.00225*.
- Humayun Kayesh, Md. Saiful Islam, Junhu Wang, Shikha Anirban, A.S.M. Kayes, and Paul Watters. 2020. Answering binary causal questions: A transfer learning based approach. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9.
- Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–907. Association for Computational Linguistics.
- Daniel Khashabi, Amos Ng, Tushar Khot, Ashish Sabharwal, Hannaneh Hajishirzi, and Chris Callison-Burch. 2021. *GooAQ: Open question answering with diverse answer types*. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 421–433, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Christopher S. G. Khoo, Jaklin Kornfilt, Robert N. Oddy, and Sung-Hyon Myaeng. 1998. Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing. *Literary and Linguistic Computing*, 13:177–186.
- David Kellogg Lewis. 1973. *Counterfactuals*. Blackwell.
- Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. *CoRR*, abs/1908.05852.
- Jian Liu, Yubo Chen, and Jun Zhao. 2020. Knowledge enhanced event causality identification with mention masking generalizations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3608–3614. International Joint Conferences on Artificial Intelligence Organization.
- Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *ACL*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.
- Ilya Loshchilov and Frank Hutter. 2017. *Fixing weight decay regularization in adam*. *CoRR*, abs/1711.05101.
- Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021a. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. In *AAAI*.
- Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021b. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. *arXiv preprint arXiv:2103.13009*.
- Ad Neeleman, Hans Van de Koot, et al. 2012. The linguistic expression of causation. *The theta system: Argument structure at the interface*, 20.
- Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset.
- Judea Pearl. 2009. *Causality: Models, Reasoning and Inference*, 2nd edition. Cambridge University Press.
- Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases?
- L. Y. Pratt. 1992. Discriminability-based transfer between neural networks. In *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.
- Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning.
- Rebecca Sharp, Mihai Surdeanu, Peter Jansen, Peter Clark, and Michael Hammond. 2016. Creating causal embeddings for question answering with minimal supervision. In *ACL 2016 Proceedings of Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Murray Singer, Michael Halldorson, Jeffrey C Lear, and Peter Andrusiak. 1992. Validation of causal bridging inferences in discourse understanding. *Journal of Memory and Language*, 31(4):507–524.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI 17*, page 4444–4451. AAAI Press.
- Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiayang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. [Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation](#).
- Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975.
- Fiona Anting Tan, Xinyu Zuo, and See-Kiong Ng. 2022. Unicausal: Unified benchmark and model for causal text mining. *arXiv preprint arXiv:2208.09163*.
- Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for “what if...” reasoning over procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6076–6085. Association for Computational Linguistics.
- Michael R. Waldmann. 2017. *The Oxford Handbook of Causal Reasoning*. Oxford University Press.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Zhipeng Xie and Feiteng Mu. 2019. Distributed representation of words in cause and effect spaces. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7330–7337.
- Jie Yang, Soyeon Caren Han, and Josiah Poon. 2022. A survey on extraction of causal relations from natural language text. *Knowl. Inf. Syst.*, 64(5):1161–1186.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [XLNet: Generalized Autoregressive Pretraining for Language Understanding](#). Curran Associates Inc., Red Hook, NY, USA.
- Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. [SWAG: A large-scale adversarial dataset for grounded commonsense inference](#). *CoRR*, abs/1808.05326.
- Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *AAAI*.
- Xinyu Zuo, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Weihua Peng, and Yuguang Chen. 2021. Improving event causality identification via self-supervised representation learning on external causal statement. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2162–2172, Online. Association for Computational Linguistics.

A Appendix

A.1 Training Details

A.1.1 Training Environment

All models were trained on a single Nvidia A100 GPU and the a2-highgpu-1g Google Cloud Compute (GCP) instance. The GCP instance has 12 virtual CPUs and 85 GB of memory.

Model training was implemented using the Pytorch Lightning library ([Falcon and The PyTorch](#)

Lightning team, 2019). To ensure reproducibility we use the Pytorch Lightning `seed_everything` function which sets the random seed for the pytorch, numpy and the python.random libraries and the seeds used for data sampling.

The AdamW optimizer (Loshchilov and Hutter, 2017) and FP16 precision were used during training. Task specific learning rates were selected through a hyperparameter search (see Appendix A.5.1). For single-task fine-tuning experiments, the model was trained for 5 epochs and the model with the best validation accuracy was selected for evaluation. For the MTL experiment we train the model for 10,000 steps and checkpoint the model every 1,000 steps. The checkpoint (8,000 steps) with best average validation accuracy/exact match was selected for evaluation on the test set.

A.2 Multiple-Choice Tasks

In this section we detail the input format and the classification heads for the multiple-choice tasks in *CALM-Bench*. The aNLI, COPA, CosmosQA, and E-Care tasks all converted to the SWAG data format (Zellers et al., 2018) and we adapt the Huggingface BERTforMultipleChoice task head as the classification head.

The WIQA task is treated as simple multi-class classification problem. Provided a procedural description, question, and the answer options (more, less, and no effect) the input format is as follows: [CLS] procedural description [SEP] question [SEP] more [SEP] less [SEP] no effect[SEP]. The classification head is a single layer feed forward network which maps the pooled CLS token embedding of the language model’s last layer into the label space.

A.3 Reading Comprehension Task

We treat the ROPES task as a SQuAD (Rajpurkar et al., 2016) style reading comprehension task and adapt the XLNET reading comprehension task head (Yang et al., 2019). Provided a question, hypothetical situation, and background passage we format the input as follows: [CLS] question [SEP] hypothetical situation [SEP] background [SEP]

The objective of the task head is to identify the answer span in the provided input text. The pooled CLS embedding of the last layer in the language model is fed to a feed forward network which independently predicts the start and end positions of the answer span in the input text. Beam search is run

to identify the most probable start and end position, after which the answer text is extracted. Unlike SQuAD, the answer span is not always present in the situation description or background passage, but it is guaranteed to specified in the question text. As a result, we do not mask the question token positions during for the task head.

A.4 Sequence Classification Tasks

The causal sequence identification and counterfactual sequence identification tasks (Appendix A.6.1) are treated as binary classification tasks. The pooled CLS embedding of the last layer in the language model is fed to a feed forward network which maps it to a binary classification space.

A.5 Relation Extraction Tasks

We treat causal and counterfactual relation extraction tasks (Appendix A.6.1) as token classification tasks and adopt a custom BIO tagging format (Ramshaw and Marcus, 1995). Causal and counterfactual entities are tagged with the <cause>, <effect>, <antecedent>, and <consequent> begin and inside tags (e.g. <B-cause> and <I-cause>). All other tokens are labelled with the outside tag (<O>). The token embeddings of the last layer in the model are fed into a single layer feed forward network which predicts for each token the most probable tag.

A.5.1 Hyperparameter Details

We run a hyperparameter search for the random seed and learning rate for each task in *CALM-Bench*. We search over the following learning rates: [0, 1, 42, 1988, 2022, 3023] and randomly selected seeds: [1e-5, 3e-5, 2e-5, 5e-5]. The search is conducted in a two-stage process where we first identify the best learning rate and then identify the best random seed. During the search trial, the model is trained for 100 steps with the provided hyperparameter and then evaluated on validation set. The best hyperparameters are summarized in Table 5 and Table 6.

A.6 Additional Experiments

A.6.1 Transfer Learning Across CRI and CALM-Bench

For analyzing the relationship between CRI and CQA in the transfer learning context, we consider the following CRI tasks:

- **Causal sequence identification:** a binary classification task to evaluate if the sentence

Model	Huggingface Alias	Parameters	Task	Seed	Learning Rate	Batch Size
BERT	bert-base-uncased	110 million	aNLI	3023	2e-5	24
BERT	bert-base-uncased	110 million	COPA	1	1e-5	24
BERT	bert-base-uncased	110 million	CosmosQA	3023	2e-5	24
BERT	bert-base-uncased	110 million	E-CARE	42	2e-5	24
BERT	bert-base-uncased	110 million	ROPES	0	5e-5	24
BERT	bert-base-uncased	110 million	WIQA	1988	2e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	aNLI	0	2e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	COPA	42	2e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	CosmosQA	0	3e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	E-CARE	2022	3e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	ROPES	42	3e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	WIQA	1988	2e-5	24

Table 5: This table summarizes the best single-task fine-tuning hyperparameters task in *CALM-Bench*.

contains causal relata (i.e. cause and effect entities and a causal relation)

- **Causal relation tagging:** a sequence tagging task that requires identifying cause and effect spans provided a sequence of a token representing a sentence.
- **Counterfactual sequence identification:** a binary classification task to evaluate if the sentence contains counterfactual relata (i.e. antecedent and consequent entities and a counterfactual relation)
- **Counterfactual relation tagging:** a sequence tagging task requires the identification of consequent and antecedent spans from a sequence of tokens representing a sentence

SemEval 2007 Task 4 (Girju et al., 2007) and 2010 Task 8 (Hendrickx et al., 2010) tasks require classifying the relation given a pairs of entities in a sentence. We combine the SemEval 2007 Task 4 and 2010 Task 8 datasets to generate examples for causal relation identification and tagging. CREST (Hosseini et al., 2022) is used to convert all examples from the 2007 and 2010 tasks into a standardized sequence tagging format. For counterfactual tasks, we use the SemEval 2020 Task 5a and 5b datasets.

Table 8 and Table 9 summarize the results for these additional experiments. We find similar patterns to our CQA transfer learning experiments. With BERT, transfer between CRI and CQA tasks is not consistent. However, the ERNIE 2.0 model shows consistent improvement from CQA tasks to the Causal Id and Causal Relation identification

tasks. Across both models there seems to be no transfer learning improvements on the counterfactual relation identification tasks.

A.6.2 Sequential fine-tuning Results

Table 7 summarizes the results of the sequential fine-tuning experiment with the BERT model. We start with a pretrained BERT model and then sequentially train it on the following multiple-choice tasks: WIQA, aNLI, CosmosQA, and E-Care. The model initially sees improvements over the single-task fine-tuned baseline results. However, as additional tasks are added, performance starts to degrade across several tasks. Due to the unstable results of sequential fine-tuning, we choose instead to pursue multi-task learning.

Model	Huggingface Alias	Parameters	Task	Seed	Learning Rate	Batch Size
BERT	bert-base-uncased	110 million	Causal Sequence Identification	42	5e-5	24
BERT	bert-base-uncased	110 million	Causal Relation Identification	1	5e-5	24
BERT	bert-base-uncased	110 million	Counterfactual Sequence Identification	1	1e-5	24
BERT	bert-base-uncased	110 million	Counterfactual Relation Identification	3023	5e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	Causal Sequence Identification	0	2e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	Causal Relation Identification	0	5e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	Counterfactual Sequence Identification	42	3e-5	24
ERNIE 2.0	nghuyong/ernie-2.0-base-en	110 million	Counterfactual Relation Identification	2022	5e-5	24

Table 6: This table summarizes the best hyperparameter used for all CRI transfer learning experiments.

	aNLI	COPA	CosmosQA	E-Care	ROPES	WIQA
<i>BERT single task-fine baseline</i>	0.61	0.64	0.57	0.76	0.51	0.65
+ WIQA and aNLI	0.61	0.74	0.60	0.75	0.34	0.77
+ CosmosQa	0.61	0.72	0.57	0.75	0.30	0.75
+ E-Care	0.60	0.72	0.59	0.76	0.45	0.70

Table 7: Results from the sequential fine-tuning experiment. As additional tasks are added the model’s performance starts to degrade across all tasks.

	Causal QA Tasks						Relation Identification Tasks			
	aNLI	COPA	CosmosQA	E-Care	ROPES	WIQA	Causal Id.	Causal Rel.	CF Id.	CF Rel.
Baseline	.61	.64	.57	.76	.58	.65	.96	.68	.96	.62
aNLI	N/A	+0.11	+0.04	0	+0.01	-0.01	0	+0.01	0	-0.02
COPA	+0.02	N/A	+0.04	-0.04	-0.06	0	+0.01	0	+0.01	-0.02
CosmosQA	+0.01	+0.05	N/A	0	+0.02	-0.01	0	-0.01	+0.01	-0.02
E-Care	+0.02	+0.13	-0.02	N/A	-0.05	-0.02	-0.02	+0.02	+0.01	0
ROPES	+0.02	+0.07	+0.03	-0.04	N/A	-0.02	-0.04	0	0	-0.03
WIQA	0	0	+0.03	-0.02	-0.05	N/A	+0.01	+0.02	+0.01	0
Causal Id.	0	0	+0.01	-0.02	-0.11	-0.01	N/A	+0.02	0	-0.03
Causal Rel.	+0.01	-0.17	+0.02	-0.05	0	0	+0.01	N/A	0	-0.02
CF Id.	+0.01	+0.04	+0.02	0	-0.05	+0.01	+0.01	+0.01	N/A	-0.01
CF Rel.	0	+0.01	+0.03	-0.04	+0.02	+0.01	+0.01	0	0	N/A

Table 8: This heatmap table summarizes the transfer learning results of BERT model on the CALM-bench and CRI tasks.

	Causal QA Tasks						Relation Identification Tasks			
	aNLI	COPA	CosmosQA	E-Care	ROPES	WIQA	Causal Id.	Causal Rel.	CF Id.	CF Rel.
Baseline	.64	.71	.63	.76	.53	.64	.94	.66	.96	.64
aNLI	N/A	+0.07	0	+0.02	+0.08	+0.01	+0.01	+0.03	0	-0.02
COPA	0	N/A	-0.01	+0.01	-0.03	+0.02	+0.03	+0.02	0	0
CosmosQA	+0.02	+0.01	N/A	-0.01	-0.12	+0.01	+0.02	+0.03	+0.01	-0.03
E-Care	0	+0.08	+0.02	N/A	+0.11	+0.02	+0.02	+0.05	0	-0.02
ROPES	+0.02	-0.06	-0.01	+0.01	N/A	+0.02	+0.02	+0.01	0	+0.04
WIQA	0	+0.01	0	-0.01	+0.03	N/A	+0.03	+0.02	+0.01	-0.01
Causal Id.	+0.01	0	+0.02	-0.01	-0.11	-0.03	N/A	+0.03	0	0
Causal Rel.	+0.01	-0.08	-0.02	-0.01	+0.04	+0.01	+0.03	N/A	0	-0.06
CF Id.	0	+0.03	+0.02	-0.01	+0.11	+0.02	+0.03	+0.01	N/A	-0.02
CF Rel.	+0.02	+0.01	0	-0.01	-0.03	+0.02	+0.03	+0.03	0	N/A

Table 9: This heatmap table summarizes the transfer learning results of ERNIE 2.0 model on the CALM-bench and CRI tasks.

ezCoref: Towards Unifying Annotation Guidelines for Coreference Resolution

Ankita Gupta[♣] Marzena Karpinska[♣] Wenlong Zhao[♣] Kalpesh Krishna[♣]
Jack Merullo[◇] Luke Yeh[♡] Mohit Iyyer[♣] Brendan O'Connor[♣]

[♣]University of Massachusetts Amherst, [◇]Brown University, [♡]Google
{ankitagupta, mkarpinska, wenlongzhao, kalpesh, miyyer, brenocon}@cs.umass.edu
john_merullo@brown.edu, lukeyeh@google.com

Abstract

Large-scale, high-quality corpora are critical for advancing research in coreference resolution. However, existing datasets vary in their definition of coreferences and have been collected via complex and lengthy guidelines that are curated for linguistic experts. These concerns have sparked a growing interest among researchers to curate a unified set of guidelines suitable for annotators with various backgrounds. In this work, we develop a crowdsourcing-friendly coreference annotation methodology, ezCoref, consisting of an annotation tool and an interactive tutorial. We use ezCoref to re-annotate 240 passages from seven existing English coreference datasets (spanning fiction, news, and multiple other domains) while teaching annotators only cases that are treated similarly across these datasets.¹ Surprisingly, we find that reasonable quality annotations were already achievable (>90% agreement between crowd and experts) even without extensive training. On carefully analyzing the remaining disagreements, we identify the presence of linguistic cases that our annotators unanimously agree upon but lack unified treatments (e.g., generic pronouns, appositives) in existing datasets. We propose the research community should revisit these phenomena when curating future unified annotation guidelines.

1 Introduction

Coreference resolution is the task of identifying and clustering together all textual expressions (*mentions*) that refer to the same discourse entity in a given document. Impressive progress has been made in developing coreference systems (Lee et al., 2017; Moosavi and Strube, 2018; Joshi et al., 2020), enabled by datasets annotated by experts (Hovy et al., 2006; Bamman et al., 2020; Uryupina et al., 2019) and crowdsourcing (Chamberlain et al., 2016). However, these datasets vary widely in

¹Our platform’s code and collected data is available at <https://github.com/gnkitaa/ezCoref>

OntoNotes: Maybe we need a [CIA] version of the Miranda warning: You have the right to conceal your coup intentions, because we may rat on you.

ARRAU: Maybe [we]e1 need [a [CIA] version of [the Miranda warning]]: [You]e4 have [the right to conceal [[your]e5 [coup] intentions]], because [we]e6 may rat on [you]e7.

Crowd (this work): Maybe [we]e1 need [a [CIA] version of [the [Miranda] warning]]: [You]e3 have [the right] to conceal [[your]e3 coup intentions], because [we]e1 may rat on [you]e3.

Figure 1: We visualize a common sentence from news domain annotated by two expert-curated datasets, OntoNotes (Hovy et al., 2006) and ARRAU (Uryupina et al., 2019), along with the crowd annotations collected via our ezCoref platform. OntoNotes does not mark generic pronouns. ARRAU does not consider them as coreferent and annotates them using a special relation “undef-reference” (markables with vague interpretations). On the contrary, our crowdworkers assign all mentions of the generic pronoun “you” to the same coreference chain. The situation is also similar for the generic “we.”

their definitions of coreference (expressed via annotation guidelines), resulting in inconsistent annotations both within and across domains and languages. For instance, as shown in Figure 1, while ARRAU (Uryupina et al., 2019) treats generic pronouns as non-referring, OntoNotes (Hovy et al., 2006) chooses not to mark them at all.

It is thus unclear which guidelines one should employ when collecting coreference annotations in a new domain or language. Traditionally, existing guidelines have leaned towards lengthy explanations of complex linguistic concepts, such as those in the OntoNotes guidelines (Weischedel et al., 2012), which detail what should and should not be coreferent (e.g., how to deal with head-sharing noun phrases, premodifiers, and generic mentions). As a result, coreference datasets have traditionally been annotated by linguists (experts) already familiar with such concepts, which makes the process expensive and time-consuming. Crowd-

sourced coreference data collection has the potential to be significantly cheaper and faster; however, teaching an exhaustive set of linguistic guidelines to non-expert crowd workers remains a formidable challenge. As a result, there has been a growing interest among researchers in curating a unified set of guidelines (Poesio et al., 2021) suitable for annotators with various backgrounds.

More recently, games-with-a-purpose (GWAPs) (von Ahn, 2006; Poesio et al., 2013) were proposed to aid crowdsourcing of large coreference datasets (e.g., Chamberlain et al., 2016; Yu et al., 2022). While GWAPs make it enjoyable for crowdworkers to learn complex guidelines and perform annotations using them (Madge et al., 2019b), they also require significant effort to attract and maintain workers. For instance, Phrase Detectives Corpus 1.0 was collected over a span of six years (Chamberlain et al., 2016; Poesio et al., 2013; Yu et al., 2022), which motivates us to instead study coreference collection on more efficient payment-based platforms.

Specifically, our work investigates the quality of crowdsourced coreference annotations when annotators are taught only simple coreference cases that are treated uniformly across existing datasets (e.g., pronouns). By providing only these simple cases, we are able to teach the annotators the concept of coreference, while allowing them to freely interpret cases treated differently across the existing datasets. This setup allows us to identify cases where our annotators unanimously agree with each other but disagree with the expert, thus suggesting cases that should be revisited by the research community when curating future guidelines.

Our main contributions are:

- We develop a crowdsourcing-friendly coreference annotation methodology—ezCoref—which includes an intuitive, open-sourced annotation tool supported by a short crowd-oriented interactive tutorial.²
- We use ezCoref to re-annotate 240 passages from seven existing English coreference datasets on Amazon Mechanical Turk (AMT), and conduct a comparative analysis of crowd and expert annotations. We find that high-quality annotations are already achievable from non-experts without extensive train-

²Our tutorial received overwhelmingly positive feedback. One annotator commented that it was “*absolutely beautiful, intuitive, and helpful. Legitimately the best one I’ve ever seen in my 2 years on AMT! Awesome job.*” (Table A4 in Appendix)

ing (>90% B3 (Bagga and Baldwin, 1998a) agreement between crowd and experts).

- We further qualitatively analyze remaining disagreements among crowd and expert annotations and identify linguistic cases that crowd unanimously marks as coreferent but lack unified treatment in existing datasets (e.g., generic pronouns as shown in Figure 1). Additionally, analyzing inter-annotator agreement among the crowd reveals that crowd exhibits higher agreement when annotating familiar texts (e.g., childhood stories or fiction) compared to texts rich in cataphoras or those requiring world knowledge. Finally, our qualitative analysis also provides an empirical evidence to support previous findings in literary studies (Szakolczai’s (2016) analysis of Bleak House) and psychology (Orvell et al.’s (2020) claims about generic “you”).

2 Related Work

Existing coreference datasets: Table 1 provides an overview of seven prominent coreference datasets, which differ widely in their annotator population, mention detection, and coreference guidelines.³ Many datasets are annotated by experts heavily trained in linguistic standards, including ARRAU (Uryupina et al., 2019), LitBank (Bamman et al., 2020), GUM (Zeldes, 2017), and OntoNotes (Hovy et al., 2006). Due to its scale and quality, OntoNotes is likely the most widely used for NLP coreference research, including in two CoNLL shared tasks (Pradhan et al., 2011, 2012). QuizBowl (Guha et al., 2015) has been annotated by domain (but not linguistic) experts. Few coreference datasets exist which are annotated by non-experts, including those created by part-time non-native English speakers (PreCo; Chen et al., 2018), and gamified crowdsourcing without financial compensation (Phrase Detectives; Chamberlain et al., 2016; Yu et al., 2022).

Coreference annotation tools: Several coreference annotation tools have been developed (See Table A3 in Appendix for more details). However, these are difficult to port to a crowdsourced workflow, as they require users to install software on their local machine (Widlöcher and Mathet, 2012; Landragin et al., 2012; Kopeć, 2014; Mueller and Strube, 2001; Reiter, 2018), or have complicated

³Many others exist too; for example, see Jonathan Kummerfeld’s spreadsheet list (accessed Jan. 2022).

Dataset	Domains #(doc, ment, tok)	Annotators	Mention Detection	Mention Types		Coreference Links			
				Singletons	Entity Restrictions	Copulae	Appositives	Generics	Ambiguity
ARRAU (Uryupina et al., 2019)	Multiple (552, 99K, 350K)	Single Expert	Manual	Yes	None	Special Link	No Link	Yes	Explicit
OntoNotes (Hovy et al., 2006)	Multiple (1.6K, 94K, 950K)	Experts	Mixed	No	None	Special Link	Special Link	Only with Pronominals	None
LitBank (Bamman et al., 2020)	Single (100, 29K, 210K)	Experts	Manual	Yes	ACE (selected)	Special Link	Special Link	Only with Pronominals	None
GUM (Zeldes, 2017)	Multiple (25, 6K, 20K)	Experts (Linguistics Students)	Manual	Yes	None	Coref (Sub-Types)	Coref (Sub-Type)	Yes	None
QuizBowl (Guha et al., 2015)	Single (400, 9.4K, 50K)	Domain Experts	Manual & CRF*	Yes	Characters, Books, Authors*	Coref	Coref	If Applicable	None
PreCo*** (Chen et al., 2018)	Multiple (38K, 3.58M, 12.5M)	Non-Expert, Non-Native	Manual**	Yes	None	Coref	Coref	Yes	None
Phrase Detectives (PD) (Chamberlain et al., 2016)	Multiple (542, 100K, 400K)	Crowd (gamified) + 2 Experts	Semi Automatic	Yes	None	Special Link	Special Link	Yes	Implicit
ezCoref Pilot Dataset (this work)	Multiple	Crowd (paid)	Fully Automatic	Yes	None	Annotator's Intuition	Annotator's Intuition	Annotator's Intuition	Implicit

Table 1: Summary of seven datasets analyzed in this work, which differ in domain, size, annotator qualifications, mention detection procedures, types of mentions, and types of links considered as coreferences between these mentions. *Allows other types of mention only when this mention is an answer to a question. **We interpret manual identification based on illustrations presented in the original publication (Chen et al., 2018). ***Inaccessible, see Footnote 8.

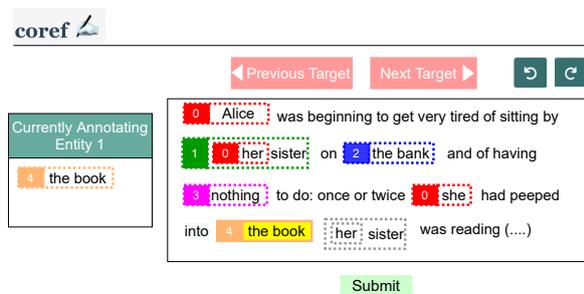


Figure 2: Part of the ezCoref interface (§3)

UI design with multiple drag and drop actions and/or multiple windows (Stenetorp et al., 2012; Widlöcher and Mathet, 2012; Landragin et al., 2012; Yimam et al., 2013; Girardi et al., 2014; Kopeć, 2014; Mueller and Strube, 2001; Oberle, 2018). Closest to ezCoref is CoRefi (Bornstein et al., 2020), a web-based coreference annotation tool that can be embedded into crowdsourcing websites. Subjectively, we found its user interface difficult to use (e.g., users have to memorize multiple key combinations). It also does not allow for nested spans, reducing its usability.

Crowdsourcing linguistic annotations: Several efforts have been made to crowdsource linguistic annotations (Snow et al., 2008; Callison-Burch, 2009; Howe, 2008; Lawson et al., 2010), including on payment-based microtasks via platforms like AMT and GWAPs (von Ahn, 2006). Many GWAPs (Poesio et al., 2013; Kicikoglu et al., 2019; Madge et al., 2019a; Fort et al., 2014) have been used in NLP to collect linguistic annotations including coreferences; with some broader platforms (Venhuizen et al., 2013; Madge et al.,

2019b) aiming to gamify the entire text annotation pipeline. One solution to teaching crowd workers complex guidelines is to incorporate *learning by progression* (Kicikoglu et al., 2020; Madge et al., 2019b; Miller et al., 2019), where annotators start with simpler tasks and gradually move towards more complex problems, but this requires subjective judgments of task difficulty. In contrast to the payment-based microtask setting studied in this work, GWAPs are not open-sourced, need significant development, take longer to collect data, and require continuous efforts to maintain visibility (Poesio et al., 2013).

3 ezCoref: A Crowdsourced Coreference Annotation Platform

The ezCoref user experience consists of (1) a step-by-step interactive tutorial and (2) an annotation interface, which are part of a pipeline including automatic mention detection and AMT Integration.

Annotation structure: Two annotation approaches are prominent in the literature: (1) a local pairwise approach, annotators are shown a pair of mentions and asked whether they refer to the same entity (Hladká et al., 2009; Chamberlain et al., 2016; Li et al., 2020; Ravenscroft et al., 2021), which is time-consuming; or (2) a cluster-based approach (Reiter, 2018; Oberle, 2018; Bornstein et al., 2020), in which annotators group all mentions of the same entity into a single cluster. In ezCoref we use the latter approach, which can be faster but requires the UI to support more complex actions for creating and editing cluster structures.

Example	Phenomena Taught
[John] doesn't like [Fred], but [he] still invited [him] to [the party].	(1) personal pronouns (2) singletons
[This dog] likes to play [catch]. [It]'s better than other [dogs] at [this game]. [Its] owner is really proud.	(1) possessive pronouns (2) semantically similar expression which are not corefering (3) non-person entities (animals)
[Director [Mackenzie]] spent [last two years] working on a ["Young Adam"]. During [this time] [he] often had to make [compromises] but [the movie] turned out to exceed expectations.	(1) nested spans (2) non-person entities (time, item)
[The office] wasn't exactly small either. [I]'m sure that 50, or maybe even 60, [people] could easily fit [there].	(1) non-person entities (place)

Table 2: Simple coreference cases explained in tutorial.

User interface: We spent two years iteratively designing, implementing, and user testing the interface to make it as simple and crowdsourcing-friendly as possible (Figure 2).⁴ Marked mentions are surrounded by color-coded frames with entity IDs. The currently selected mention (“the book”), is highlighted with a flashing yellow cursor-like box. The core annotation action is to select other mentions that corefer with the current mention, and then advance to a later unassigned mention; annotators can also re-assign a previously annotated mention to another cluster. Advanced users can exclusively use keyboard shortcuts, undo and redo actions were added to allow error correction. Finally, ezCoref provides a side panel showing mentions of the entity currently being annotated to spot mentions assigned to the wrong cluster.

Coreference tutorial: To teach crowdworkers the basic definition of coreference and familiarize them with the interface, we develop a tutorial (aimed to take ~ 20 minutes) that introduces them to the mechanics of the annotation tool, and then trains them on simple cases of coreferences. These cases (e.g., personal/possessive pronouns or determinative phrases which corefer with their antecedents as shown in Table 2) are annotated similarly across all existing datasets and are unlikely to be disputed. The tutorial concludes with a quality control example to exclude poor quality annotators.⁵ These training examples, feedback, and annotation guidelines can be easily customized using a simple JSON schema.

Annotation workflow: The annotators are presented with one passage (or “document”) at a time (Figure 2), and all mentions have to be annotated before proceeding to the next passage. There is no limitation to the length or language of the passage.

⁴The interface is implemented in [ReactJS](#).

⁵Examples of the tutorial interface and the quality control example are provided in Appendix.

In this work, we divide an initial document into a sequence of shorter passages of complete sentences, on average 175 tokens, as shorter passages minimize the need to scroll, reducing annotator effort. While this obviously cannot capture longer distance coreference,⁶ a large portion of important coreference phenomena is local: within the OntoNotes written genres, for pronominal mentions, the closest antecedent is contained within the current or previous two sentences more than 95% of the time.

Automatic mention detection: As a first step to collect coreference annotations, we must identify mentions in the documents from each of the seven existing datasets; this process is done in a diverse array of ways (from manually to automatic) in prior work as shown in Table 1. We decided to automatically identify mentions to give all crowdworkers an identical set of mentions, which simplifies the annotation task and also allows us to easily compare and study their coreference annotations via inter-annotator agreement. Specifically, we implement a simple algorithm that yields a high average recall over all seven datasets.⁷

Our algorithm considers all noun phrases (including proper nouns, common nouns, and pronouns) as markables, extracting them using the Stanza dependency parser (version 1.3.0; Qi et al., 2020). We allow for nested mentions and proper noun premodifiers (e.g., [U.S.] in “U.S. policy”). We include all conjuncts with the entire coordinated noun phrase ([Mark], [Mary], as well as [Mark and Mary], are all considered mentions); details in Appendix A.3.

4 Using ezCoref to Re-annotate Existing Coreference Datasets

We deploy ezCoref on the AMT crowdsourcing platform to re-annotate 240 passages from seven existing datasets, covering seven unique domains. In total, we collect annotations for 12,200 mentions and 42,108 tokens. We compare our workers’ an-

⁶We leave this for future work—for example, more sophisticated user interfaces to support longer documents, or merging coreference chains between short passages. As documents get progressively longer, such as book chapters or books, the task takes on aspects of cross-document coreference and entity linking (e.g. Bagga and Baldwin, 1998b; FitzGerald et al., 2021; Logan IV et al., 2021).

⁷We acknowledge that any algorithm can be used as long as its recall across all datasets is high, and ours is only one such algorithm. However, we do not conduct an ablation study to compare crowd annotations for mentions obtained from these potential algorithms as it would be prohibitively expensive. Furthermore, while advanced mention detection methods can improve annotation quality, our goal is not to collect the highest-quality coreference dataset, but to study annotator behavior when a common set of mentions is provided.

notations both quantitatively and qualitatively to each other and to existing expert annotations.

Datasets: We collect coreference annotations for the seven existing datasets described in Table 1: OntoNotes (Hovy et al., 2006), LitBank (Bamman et al., 2020), PreCo⁸ (Chen et al., 2018), ARRAU (Uryupina et al., 2019), GUM (Zeldes, 2017), Phrase Detectives (Chamberlain et al., 2016), and QuizBowl (Guha et al., 2015). The sample covers seven domains: news, opinionated magazines, weblogs, fiction, biographies, Wikipedia articles, and trivia questions from Quiz Bowl. For each dataset with multiple domains, we manually select a broad range of domain(s) for re-annotation. From each domain in each dataset, we then select documents and divide them into shorter passages (on average 175 tokens each), creating 20 such passages per dataset. For datasets with multiple domains, we choose 20 such passages per domain (see Appendix A.1 for detail). Overall, we collect annotations for 240 passages with 5 annotations per passage to measure inter-annotator agreement.

Procedure: We first launch an annotation tutorial and recruit the annotators on the AMT platform.⁹ At the end of the tutorial, each annotator is asked to annotate a short passage (around 150 words). Only annotators with a B3 score (Bagga and Baldwin, 1998a) of 0.90 or higher are then invited to participate in the annotation task.

Training Annotators with Simplified Guidelines using ezCoref: As the goal of our study is to understand what crowdworkers perceive as coreference, we train our annotators with simple guidelines. We carefully draft our training examples to include only cases which are considered as coreference by all the existing datasets. The objective is to

⁸The PreCo dataset is interestingly large but seems difficult to access. In November 2018 and October 2021 we filled out the data request form at the URL provided by the paper, and attempted to contact the PreCo official email directly, but did not receive a response. To enable a precise research comparison, we scraped all documents from PreCo’s public demo in November 2018 (no longer available as of 2021); its statistics match their paper and our experiments use this version of the data. PreCo further suffers from data curation issues (Gebru et al., 2018; Jo and Gebru, 2020); it uses text from English reading comprehension tests collected from several websites, but the original document sources and copyright statuses are undocumented. When reading through PreCo documents, we found many domains including opinion, fiction, biographies, and news (Table A1 in Appendix); we use our manual categories for domain analysis.

⁹We allow only workers with a $\geq 99\%$ approval rate and at least 10,000 approved tasks who are from the US, Canada, Australia, New Zealand, or the UK.

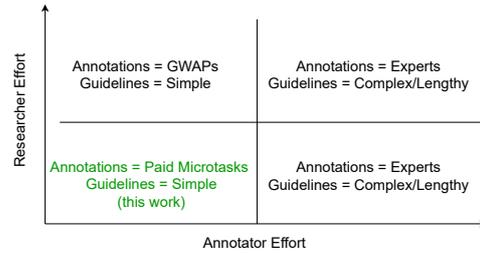


Figure 3: Existing expert annotated datasets entail high annotator effort (e.g., OntoNotes, ARRAU). Existing crowdsourced coreference datasets (e.g., Phrase Detectives) entail significant researcher effort. In this work, we explore the minimum effort scenario for both annotators (by providing them simplified guidelines) and researchers (by open-sourcing ezCoref).

teach crowdworkers the broad definition of coreference while leaving space for different interpretations of ambiguous cases or those resolved differently across the existing datasets. Note that a comparable experiment with more complex guidelines is infeasible since it is unclear which guidelines to choose, and also providing complex linguistic guidelines to crowdworkers remains an open challenge. Overall, ezCoref is aimed to minimize both researcher and annotator effort for new coreference data collection, compared to prior work (Figure 3).

Worker details: Overall, 73 annotators (including 44 males, 20 females, and one non-binary person)¹⁰ completed the tutorial task, which took 19.4 minutes on average (sd=11.2 minutes). They were aged between 21 and 69 years (mean=38.9, sd=11.3) and identified themselves as native English speakers. Most of the annotators had at least a college degree (47 vs 18). 89.0% of annotators, who did the tutorial, received a B3 score of 0.90 or higher for the final screening example, and were invited to the annotation task. 50.7% of the invited annotators returned to participate in the main annotation task, and 29.2% of them annotated five or more passages. Annotation of one passage took, on average, 4.15 minutes, a rate of 2530 tokens per hour. The total cost of the tutorial was \$460.70 (\$4.50 per tutorial). We paid \$1 per passage for the main annotation task, resulting in a total cost of \$1440.¹¹

5 Analysis

In this section, we perform quantitative and qualitative analyses of our crowdsourced coreference annotations. First, we evaluate the performance

¹⁰We did not collect demographic data for the remaining eight individuals, from an earlier pilot experiment.

¹¹All reported costs include 20% AMT fee.

of our mention detection algorithm, comparing it to gold mentions across seven datasets. Next, we measure the quality of our annotations and their agreement with other datasets. Finally, we discuss interesting qualitative results.

5.1 Mention Detector Evaluation

Datasets differ in the way they define their mention boundaries and thus the boundaries for the same mention may differ. To fairly compare our mentions with the gold standards, we employ a headword-based comparison. We find the head of the given phrase by identifying, in the dependency tree, the most-shared ancestor of all tokens within the given mention. Two mentions are considered same if their respective headwords match.

Table 3 compares our mention detector to the gold mentions in existing datasets. Our method obtains high recall across most datasets (>0.90), which shows that most of the mentions annotated in existing datasets are correctly identified and allows a direct comparison of crowd annotations with expert annotations. It has the lowest recall with ARRAU (0.84) and PreCo (0.88), which is to be expected as ARRAU marks all referring premodifiers (identified manually) and PreCo allows common noun modifiers, while we identify only the premodifiers which are proper nouns.¹²

For most datasets, the precision is >0.80 , suggesting that the algorithm identifies most of the relevant mentions. We observe a substantially lower score for OntoNotes, LitBank, and QuizBowl as these datasets restrict their mention types to limited entities (refer to Table 1). However, this does not limit our analysis. In fact, an algorithm with high precision on LitBank or OntoNotes would miss a huge percentage of relevant mentions and entities on other datasets (constraining our analysis) and when annotating new texts and domains. Furthermore, our algorithm identifies more mentions than in the original datasets, which in the best case allows us to discover new entities and, in the worst case, may result in more singletons. Finally, the mention density (number of mentions per token) from our detector remains roughly consistent across all datasets when using our method, allowing us to fairly compare statistics (e.g., agreement rates) across datasets.

¹²We made this decision as identifying automatically all premodifiers would result in many singletons and lead to more arduous annotation effort.

Dataset	Recall	Precision	Mentions / Tokens	
			Gold	This Work
OntoNotes	0.957	0.376	0.112	0.286
LitBank	0.962	0.415	0.121	0.280
QuizBowl	0.956	0.543	0.188	0.318
PD (Gold)	0.953	0.803	0.259	0.273
PD (Silver)	0.938	0.791	0.265	0.274
GUM	0.906	0.848	0.269	0.287
PreCo	0.881	0.883	0.287	0.287
ARRAU	0.840	0.870	0.289	0.279

Table 3: Comparison of mentions identified by our mention detection algorithm with the gold mentions annotated in the respective datasets. We use head-word based comparison to compare mentions of different lengths. Our method obtains high recall across most datasets and the mention-density using our mention-detector remains roughly consistent across datasets, allowing us to do fair analysis (e.g., agreement) across datasets.

5.2 Agreement with Existing Datasets

How well do annotations from ezCoref agree with annotations from existing datasets?

Aggregating annotations: To compare crowd-sourced annotations with gold annotations, we first require an aggregation method that can combine annotations from multiple crowdworkers to infer coreference clusters. We use a simple aggregation method that determines whether a pair of mentions is coreferent by counting the number of annotators who marked the two mentions in the same cluster.¹³ Two mentions are considered as coreferent when the number of annotators linking them together is greater than a threshold (τ). After inferring these pairs of mentions, we construct an undirected graph where nodes are mentions and edges represent coreference links. Finally, we find connected components in the graph to obtain coreference clusters.¹⁴ We compare aggregated annotations from ezCoref with gold annotations across the seven datasets using B3 scores (precision, recall, and F1),¹⁵ as illustrated in Figure 4.

High agreement with OntoNotes, GUM, LitBank, ARRAU: Our annotators achieve the high-

¹³Future data collection efforts interested in creating large resources can utilize more advanced aggregation methods (Poesio et al., 2019).

¹⁴This method resolves to majority voting-based aggregation when the τ is set so that more than half of annotators should agree. For $\tau = N$, this method is very conservative, adding a link between two mentions only when all annotators agree unanimously. Conversely, for $\tau = 1$, only a single vote is required to add a link between two mentions.

¹⁵For a mention in a given document, B3 recall is the fraction of mentions that are correctly predicted by the system as coreferent with it out of all mentions that are actually coreferent with it. B3 precision is the fraction of mentions that are correctly predicted by the system as coreferent with it out of all system-predicted mentions.

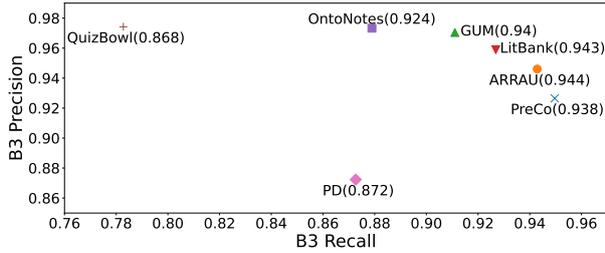


Figure 4: Agreement with gold annotations across datasets. B3 (F1) scores shown in parentheses are computed with singletons included.

est precision with OntoNotes (Figure 4), suggesting that most of the entities identified by crowdworkers are correct for this dataset. In terms of F1 scores, the datasets which are closest to crowd annotations are GUM, LitBank, and ARRAU, all of which are annotated by experts. This result shows that high-quality annotations can be obtained from non-experts using ezCoref without extensive training. We further conducted a qualitative analysis of high agreement cases for each dataset. Overall, we observe that non-experts agree with experts on chains containing pronouns and named entities. However, non-experts also mark noun phrases in appositive constructions as coreferent, consistent with GUM guidelines. Finally, non-experts also assign generic mentions to the same coreference chain, consistent with their treatment by GUM and ARRAU, and leads to higher agreement with these datasets.

Low precision with Phrase Detectives and PreCo, low recall with Quiz Bowl:

We observe that Phrase Detectives has a very low precision compared to all other datasets, implying that crowdworkers add more links compared to gold annotations. Our qualitative analysis reveals that PD annotators miss some valid links, splitting entities which are correctly linked together by our annotators (see Table 4). Another dataset with lower precision is PreCo, which also contains many missing links. In general, we observe more actual mistakes in PreCo and PD than in the other datasets, which is not surprising as they were not annotated by experts.¹⁶ This result is further validated by our agreement analysis of the fiction domain (Table 5), in which ezCoref annotations agree far more closely with expert annotations (GUM, LitBank) than PreCo and PD. Finally, Quiz Bowl has by far the lowest recall with ezCoref annotations, which is ex-

¹⁶That said, both PreCo and PD were additionally validated by multiple non-expert annotators.

PD	Not long after [a suitor] appeared, and as [he] appeared to be very rich and the miller could see nothing in [him] with which to find fault, he betrothed his daughter to [him]. But the girl did not care for [the man] (...). She did not feel that she could trust [him], and she could not look at [him] nor think of [him] without an inward shudder.
PreCo	When I listened to the weather report, I was afraid to see [the advertisements]. [Those colorful advertisements] always made me crazy.

Table 4: Cases of split entities (missing links) in annotations provided with Phrase Detectives and PreCo. Instead, our crowd annotators mark all mentions as referring to the same entity in each of these examples.

pected given the difficulty with cataphora and factual knowledge (examples (c) and (e) in Table 6).

Domain	Dataset	B3		
		Precision	Recall	F1
Fiction	GUM	0.982	0.921	0.950
	LitBank	0.959	0.927	0.943
	PreCo	0.805	0.963	0.877
	Phrase Detectives	0.784	0.775	0.780

Table 5: Agreement with existing datasets for fiction.

Varying the aggregation threshold τ : What is the effect of varying the aggregation threshold (τ) on precision and recall with gold annotations? Figure 5 shows that the Quiz Bowl dataset has the highest drop in recall (36% absolute drop) when increasing τ from 1 to 5.¹⁷ This indicates that the number of unanimous clusters ($\tau = 5$) is considerably lower than the total number of clusters found individually by all annotators ($\tau = 1$); as such, our annotators heavily disagree about gold clusters in the QuizBowl dataset. We observe a similar trend in OntoNotes (26% drop in recall), whereas Phrase Detectives has the lowest drop in recall (0.07) with the increase in the number of annotators, which is expected since Phrase Detectives is crowdsourced.

5.3 What domains are most suitable for crowdsourcing coreference?

We use the B3 metric¹⁸ (Bagga and Baldwin, 1998a) to compute IAA for each domain, excluding singletons¹⁹ (see Table 7). We obtain the highest agreement on fiction (72.6%) and biographies (72.4%). This is because both domains contain a high frequency of pronouns (see examples *a* and

¹⁷We analyze variations in recall which is more interpretable than precision, since the denominator is fixed in recall when varying number of annotators.

¹⁸Krippendorff’s alpha/kappa are other possible measures for IAA. However, prior work (Paun et al., 2022) has raised concerns over using Krippendorff’s alpha/kappa for anaphora resolution. Instead, we found B3 intuitive to understand as a measure of agreement among annotators at the mention level, i.e. fraction of mentions two annotators agree should be coreferent with a given mention.

¹⁹IAA including singletons is much higher (Appendix A.4).

Phenomena	Dataset (Domain)	Example
Pronouns	LitBank (Fiction)	A Wolf had been gorging on an animal [he] had killed, when suddenly a small bone in the meat stuck in [his] throat and [he] could not (a) swallow [it]. [He] soon felt a terrible pain in [his] throat (...) [He] tried to induce everyone [he] met to remove the bone. "[I] would give anything," said [he] , " if [you] would take [it] out. "
	GUM (Biographies)	Despite Daniel's attempts at reconciliation, [his] father carried the grudge until [his] death. Around schooling age, [his] father, Johann, (b) encouraged [him] to study business (...). However, Daniel refused because [he] wanted to study mathematics. [He] later gave in to [his] father's wish and studied business. [His] father then asked [him] to study in medicine.
Cataphora	QuizBowl (Quizzes)	[One character in this work] is forgiven by [magenta] wife for an affair with a governess before beginning one with a ballerina. [Another (c) character in this work] is a sickly, thin man who eventually starts dating a reformed prostitute, Marya Nikolaevna. In addition to [Stiva] and [Nikolai] , [another character in this work] (...) had earlier failed in [his] courtship of Ekaterina Shcherbatskaya.
Factual Knowledge	OntoNotes (News)	(d) The Soviet Union's jobless rate is soaring (...), [Pravda] said. Unemployment has reached 27.6 % in Azerbaijan, (...) and 16.3% in Kirgizia, [the Communist Party newspaper] said.
	QuizBowl (Quizzes)	(...) [another character in this work] (...) had earlier failed in [his] courtship of [Ekaterina Shcherbatskaya]. Another character in this work (e) rejects [Ekaterina] before (...) moving to St. Petersburg. For 10 points name this work in which [Levin] marries [Kitty] , (...) a novel by Leo Tolstoy.

Table 6: Representative examples showing unique phenomena in each dataset (coreferences are color coded).

Fiction	Biographies	Opinion	Web	News	Wikipedia	Quiz
72.6	72.4	69.5	65.9	62.3	61.8	59.7

Table 7: Domain-wise IAA: B3% scores using CoNLL script (Pradhan et al., 2014), excluding singletons.

b in Table 6), which our annotators found easier to annotate. We also observe that the fiction domain contains many well-known children stories (e.g., Little Red Riding Hood) that are likely familiar to our annotators, which may have made them easier to annotate. Annotators have the least agreement on Quiz Bowl coreference (59.73%), as this dataset is rich in challenging cataphoras (example *c* in Table 6) and often require world knowledge about books, characters, and authors to identify coreferences (example *e* in Table 6).

5.4 Qualitative analysis

To better understand the differences in annotation quality, we conduct a manual analysis²⁰ of all 240 passages, comparing our ezCoref annotations to gold annotations from each dataset. Specifically, we look at each link that was annotated by our workers but not in the gold data, or vice versa. For each link, we determine whether crowd or the gold annotations contained a mistake, or whether the discrepancy is reasonable under specific guidelines. We find that ezCoref annotations contain fewer mistakes than non-expert annotated datasets (PreCo and PD), almost twice as many mistakes as those of expert datasets (OntoNotes and GUM), and seven times as many mistakes as those in the esoteric Quiz Bowl dataset (Appendix Table A2).

Disagreements and deviations from expert guidelines: As in Poesio and Artstein (2005), we identify cases of genuine ambiguity, where a mention can refer to two different antecedents. The

²⁰By a linguist who studied guidelines of all datasets.

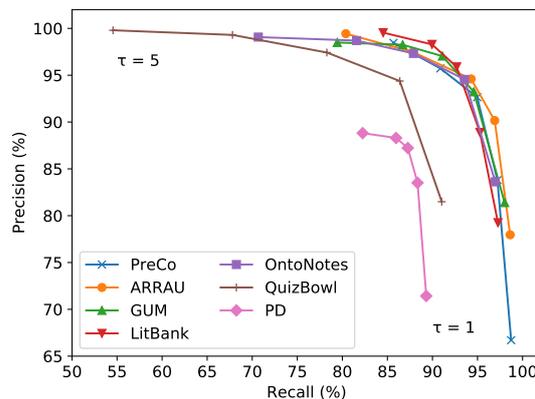


Figure 5: Agreement with gold annotations with varying voting threshold τ . $\tau = 3$ is majority voting (Figure 4). B3 scores computed with singletons included.

first row of Table 8 shows an example from Dickens' *Bleak House*, where the pronoun "it" could reasonably refer to either the "fog" or the "river." Our annotators have high disagreement on this link, which is understandable given the literary analysis of Szokolczai (2016) who interprets the ambiguity of this pronoun as Dickens' way to show indeterminacy attributed to elements in the scene.²¹

We observe that generic mentions, especially generic pronouns, are almost always annotated as coreferring by crowd, while existing datasets lack consensus (Table 1). Table 8 (second row) shows an example where annotators unanimously connected all instances of generic "you." This observation is in line with Orvell et al.'s (2020) study which explains that by using the same linguistic form ("you"), one invites readers (annotators) to consider how the situation refers to them. Finally, while datasets tend to treat copulae and appositive constructions identically and annotate them

²¹In LitBank, the source of this passage, the pronoun "it" is annotated as referring to the "river" as only "river" is a potential markable per entity restriction (ACE entities only).

Ambiguity	[Fog] everywhere. [Fog] up [the river] , where [it] flows among green aits and meadows; [fog] down [the river] , where [it] rolls defiled among the tiers of shipping and the waterside pollutions of a great (and dirty) city. - Charles Dickens, <i>Bleak House</i>
Generic	Please , Ma'am , is this New Zealand or Australia? (and she tried to curtsey as she spoke – fancy CURTSEYING as [you] 're falling through the air! Do [you] think [you] could manage it?) - Lewis Carroll, <i>Alice in Wonderland</i>

Table 8: Examples of genuine ambiguity and generic “you” observed in our data.

in a similar way, our annotators intuitively annotate them differently. While crowdworkers almost always mark noun phrases in appositive constructions as coreferent, the noun phrases in copulae are linked by majority vote only in $\sim 35\%$ of cases.

6 Conclusion

Existing coreference datasets vary in their definition of coreferences and have been collected via complex guidelines. In this work, we investigate the quality of annotations when crowdworkers are taught only few coreference cases that are treated similarly across existing datasets. We develop a crowdsourcing-friendly coreference annotation methodology, ezCoref and use it to re-annotate 240 passages from seven existing English coreference datasets. We observe reasonable quality annotations were already achievable even without extensive training. On analyzing the remaining disagreements, we identify linguistic cases that crowd unanimously agree upon but lack unified treatments in existing datasets, suggesting cases the researchers should revisit when curating future unified annotation guidelines.

7 Limitations

We list some of the limitations of our study which researchers and practitioners would hopefully benefit from when interpreting our analysis. Firstly, our analysis is only applicable to the English language and how native English speakers understand coreferences. In this work, we have taken a step towards building a framework to facilitate the comparison of the crowd and expert annotations, and the variations observed in non-native speakers should be explored in future studies. Secondly, as a result of resource constraints, we limited ourselves to one set of guidelines and compared crowd annotations under these guidelines with expert annotations. Understanding the effects of various guidelines on annotator behavior is left for future research. Thirdly, even the best automatic mention detection algorithm could have errors, especially when tested out-of-domain. Despite this limitation,

we decided to use an automatic method as it allows us to study annotators’ behavior when a “common set of mentions” is provided. Some of the proposed solutions to address this issue are to directly crowdsource mentions or verify the automatically identified mentions via crowdsourcing (Madge et al., 2019b), which can be utilized for future collection of high-quality corpora. Finally, we also acknowledge that the tool cannot handle split-antecedents or separate tags for different relations, which we leave for future work. As a result, our approach focuses on cases of identity coreferences. However, we believe that identity coreference supported by our tool has value as an NLP tool (e.g., studying characters in narratives (Bamman et al., 2013)), allowing the collection of more in-domain annotations, necessary to advance such practical applications.

8 Ethics Statement

The data collection protocol was approved by the coauthors’ institutional review board. All annotators were presented with a consent form (mentioned below) prior to the annotation. They were also informed that only satisfactory performance on the screening example will allow them to take part in the annotation task. All data collected during the tutorial and annotations (including annotators’ feedback and demographics) will be released anonymized. We also ensure that the annotators receive at least \$13.50 per hour. Since base compensation is per unit of work, not by time (the standard practice on Amazon Mechanical Turk), we add bonuses for workers whose speed caused them to fall below that hourly rate.

Consent Before participating in our study, we requested every annotator to provide their consent. The annotators were informed about the purpose of this research study, any risks associated with it, and the qualifications necessary to participate. The consent form also elaborated on task details describing what they will be asked to do and how long it will take. The participants were informed that they could choose as many documents as they would like to annotate (by accepting new Human Intelligence Tasks at AMT) subject to availability, and they may drop out at any time. Annotators were informed that they would be compensated in the standard manner through the Amazon Mechanical Turk crowdsourcing platform, with the amount specified in the Amazon Mechanical Turk interface. As part of this study, we also collected demographic information, including their age, gender,

native language, education level, and proficiency in the English language. We ensured our annotators that the collected personal information would remain confidential in the consent form.

Acknowledgements

We are very grateful to the crowd annotators on AMT for participating in our annotation tasks and providing positive reviews. We are grateful to Abe Handler, Aditya Jain, Anna Rogers, Julian Richardson, Kavya Jeganathan, Neha Kennard, Nishant Yadav, Timothy O’Gorman, and the UMass NLP group for several useful discussions during the course of the project. We also thank Massimo Poesio for sharing the GNOME portion of AR-RAU dataset. This material is based upon work supported by National Science Foundation awards 1925548, 1814955, and 1845576, and a Google PhD Fellowship awarded to KK.

References

- Amit Bagga and Breck Baldwin. 1998a. Algorithms for scoring coreference chains. In *Proceedings of the First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, Volume 1*, pages 563–566.
- Amit Bagga and Breck Baldwin. 1998b. [Entity-based cross-document coreferencing using the vector space model](#). In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.
- David Bamman, Olivia Lewke, and Anya Mansoor. 2020. [An annotated dataset of coreference in English literature](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.
- David Bamman, Brendan T. O’Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Annual Meeting of the Association for Computational Linguistics*.
- Ari Bornstein, Arie Cattan, and Ido Dagan. 2020. [CoRefi: A crowd sourcing suite for coreference annotation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 205–215, Online. Association for Computational Linguistics.
- Chris Callison-Burch. 2009. [Fast, cheap, and creative: Evaluating translation quality using Amazon’s Mechanical Turk](#). In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 286–295, Singapore. Association for Computational Linguistics.
- Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. 2016. [Phrase Detectives Corpus 1.0 Crowdsourced Anaphoric Coreference](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 2039–2046, Portorož, Slovenia. European Language Resources Association.
- Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. 2018. [PreCo: A large-scale dataset in preschool vocabulary for coreference resolution](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181, Brussels, Belgium. Association for Computational Linguistics.
- Nicholas FitzGerald, Dan Bikel, Jan Botha, Daniel Gillick, Tom Kwiatkowski, and Andrew McCallum. 2021. [MOLEMAN: Mention-only linking of entities with a mention annotation network](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 278–285, Online. Association for Computational Linguistics.
- Karén Fort, Bruno Guillaume, and Hadrien Chastant. 2014. [Creating Zombilingo, a game with a purpose for dependency syntax annotation](#). In *Proceedings of the First International Workshop on Gamification for Information Retrieval*, pages 2–6, New York, NY, USA. Association for Computing Machinery.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. 2018. [Datasheets for Datasets](#). *CoRR*, abs/1803.09010.
- Christian Girardi, Manuela Speranza, Rachele Sprugnoli, and Sara Tonelli. 2014. [CROMER: A tool for cross-document event and entity coreference](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 3204–3208, Reykjavik, Iceland. European Language Resources Association.
- Anupam Guha, Mohit Iyyer, Danny Bouman, and Jordan Boyd-Graber. 2015. [Removing the Training Wheels: A coreference dataset that entertains humans and challenges computers](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1108–1118, Denver, Colorado. Association for Computational Linguistics.
- Barbora Hladká, Jiří Mírovský, and Pavel Schlesinger. 2009. [Play the Language: Play Coreference](#). In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 209–212, Suntec, Singapore. Association for Computational Linguistics.

- Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. [OntoNotes: The 90% solution](#). In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.
- Jeff Howe. 2008. *Crowdsourcing: How the power of the crowd is driving the future of business*. London, England: Random House Books.
- Eun Seo Jo and Timnit Gebru. 2020. [Lessons from Archives: Strategies for collecting sociocultural data in machine learning](#). In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. ACM.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. [SpanBERT: Improving pre-training by representing and predicting spans](#). *Transactions of the Association for Computational Linguistics*, 8:64–77.
- Doruk Kicikoglu, Richard Bartle, Jon Chamberlain, and Massimo Poesio. 2019. [Wormingo: A ‘true gamification’ approach to anaphoric annotation](#). In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–7.
- Osman Doruk Kicikoglu, Richard Bartle, Jon Chamberlain, Silviu Paun, and Massimo Poesio. 2020. [Aggregation driven progression system for GWAPs](#). In *Workshop on Games and Natural Language Processing*, pages 79–84, Marseille, France. European Language Resources Association.
- Mateusz Kopeć. 2014. [MMAX2 for coreference annotation](#). In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 93–96, Gothenburg, Sweden. Association for Computational Linguistics.
- Frédéric Landragin, Thierry Poibeau, and Bernard Victorri. 2012. [ANALEC: A new tool for the dynamic annotation of textual data](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 357–362, Istanbul, Turkey. European Language Resources Association.
- Nolan Lawson, Kevin Eustice, Mike Perkowitz, and Meliha Yetisgen-Yildiz. 2010. [Annotating large email datasets for Named Entity Recognition with Mechanical Turk](#). In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 71–79, Los Angeles. Association for Computational Linguistics.
- Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. [End-to-end neural coreference resolution](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.
- Belinda Z. Li, Gabriel Stanovsky, and Luke Zettlemoyer. 2020. [Active learning for coreference resolution using discrete annotation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8320–8331, Online. Association for Computational Linguistics.
- Robert L Logan IV, Andrew McCallum, Sameer Singh, and Dan Bikel. 2021. [Benchmarking scalable methods for streaming cross document entity coreference](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4717–4731, Online. Association for Computational Linguistics.
- Chris Madge, Richard Bartle, Jon Chamberlain, Udo Kruschwitz, and Massimo Poesio. 2019a. [Making text annotation fun with a clicker game](#). In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, New York, NY, USA. Association for Computing Machinery.
- Chris Madge, Juntao Yu, Jon Chamberlain, Udo Kruschwitz, Silviu Paun, and Massimo Poesio. 2019b. [Progression in a Language Annotation Game with a Purpose](#). In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 77–85.
- Josh Aaron Miller, Uttkarsh Narayan, Matthew Hantsbarger, Seth Cooper, and Magy Seif El-Nasr. 2019. [Expertise and engagement: Re-designing citizen science games with players’ minds in mind](#). In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–11.
- Nafise Sadat Moosavi and Michael Strube. 2018. [Using linguistic features to improve the generalization capability of neural coreference resolvers](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Brussels, Belgium. Association for Computational Linguistics.
- Christoph Mueller and Michael Strube. 2001. [Annotating anaphoric and bridging relations with MMAX](#). In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.
- Bruno Oberle. 2018. [SACR: A drag-and-drop based tool for coreference annotation](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, Miyazaki, Japan. European Language Resources Association.
- Ariana Orvell, Ethan Kross, and Susan A. Gelman. 2020. [“You” speaks to me: Effects of generic-you in creating resonance between people and ideas](#). *Proceedings of the National Academy of Sciences*, 117(49):31038–31045.
- Silviu Paun, Ron Artstein, and Massimo Poesio. 2022. [Statistical Methods for Annotation Analysis](#), volume 1 of *Synthesis Lectures on Human Language Technologies*. Springer, Cham.

- Massimo Poesio and Ron Artstein. 2005. Annotating (Anaphoric) Ambiguity. In *Proceedings of the corpus linguistics conference*.
- Massimo Poesio, Jon Chamberlain, Udo Kruschwitz, Livio Robaldo, and Luca Ducceschi. 2013. **Phrase Detectives: Utilizing collective intelligence for internet-scale language resource creation**. *ACM Trans. Interact. Intell. Syst.*, 3(1).
- Massimo Poesio, Jon Chamberlain, Silviu Paun, Juntao Yu, Alexandra Uma, and Udo Kruschwitz. 2019. **A crowdsourced corpus of multiple judgments and disagreement on anaphoric interpretation**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1778–1789. Association for Computational Linguistics.
- Massimo Poesio, Amir Zeldes, Anna Nedoluzhko, Sapan Khosla, Ramesh Manuvinaurike, Nafise Moosavi, Vincent Ng, Maciej Ogrodniczuk, Sameer Pradhan, Carolyn Rose, Michael Strube, Juntao Yu, Yulia Grishina, Yufang Hou, and Fred Landragin. 2021. Universal anaphora 1.0. <https://sites.google.com/view/universalanaphora/>. Accessed: 2021-10-30.
- Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. **Scoring coreference partitions of predicted mentions: A reference implementation**. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. **CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes**. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.
- Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. **CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes**. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA. Association for Computational Linguistics.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. **Stanza: A Python Natural Language Processing Toolkit for many human languages**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- James Ravenscroft, Amanda Clare, Arie Cattan, Ido Dagan, and Maria Liakata. 2021. **CD²CR: Co-reference resolution across documents and domains**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 270–280, Online. Association for Computational Linguistics.
- Nils Reiter. 2018. **CorefAnnotator - A new annotation tool for entity references**. In *Abstracts of EADH: Data in the Digital Humanities*.
- Rion Snow, Brendan O’Connor, Dan Jurafsky, and Andrew Y Ng. 2008. **Cheap and fast – but is it good? Evaluating non-expert annotations for natural language tasks**. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263.
- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii. 2012. **BRAT: A Web-based Tool for NLP-assisted text annotation**. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.
- Árpád Szokolczai. 2016. *Novels and the Sociology of the Contemporary*. Routledge, Milton Park, Abingdon, Oxon New York, NY.
- Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa J. Rodriguez, and Massimo Poesio. 2019. **Annotating a broad range of anaphoric phenomena, in a variety of genres: The ARRAU corpus**. *Natural Language Engineering*, 26(1):95–128.
- Noortje J. Venhuizen, Valerio Basile, Kilian Evang, and Johan Bos. 2013. **Gamification for Word Sense Labeling**. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 397–403, Potsdam, Germany. Association for Computational Linguistics.
- Luis von Ahn. 2006. **Games with a purpose**. *Computer*, 39(6):92–94.
- Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Nianwen Xue, Martha Palmer, Jena D. Hwang, Claire Bonial, Jinho Choi, Aous Mansouri, Maha Foster, Abdel-aati Hawwary, Marcus Mitchell, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, and Ann Houston. 2012. **Ontonotes Release 5.0**. <https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf>. Accessed: 2022-01-15.
- Antoine Widlöcher and Yann Mathet. 2012. **The Glozz platform: A corpus annotation and mining tool**. In *2012 ACM symposium on Document Engineering*.
- Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. **WebAnno: A flexible, web-based and visually supported system for distributed annotations**. In *Proceedings of the*

51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 1–6, Sofia, Bulgaria. Association for Computational Linguistics.

Juntao Yu, Silviu Paun, Maris Camilleri, Paloma Carretero Garcia, Jon Chamberlain, Udo Kruschwitz, and Massimo Poesio. 2022. [Aggregating crowd-sourced and automatic judgments to scale up a corpus of anaphoric reference for fiction and Wikipedia texts](#). *Computing Research Repository*, arXiv:2210.05581.

Amir Zeldes. 2017. [The GUM corpus: Creating multilayer resources in the classroom](#). *Language Resources and Evaluation*, 51(3):581–612.

A Appendix

A.1 Details of our crowdsourced data

Table A1 mentions all datasets that we re-annotate in this work with their breakdown based on domains, number of documents, passages, tokens and mentions annotated.

Dataset	Domain	#Docs	#Passages	#Tokens	#Mentions
OntoNotes	News	6	30	4923	1365
	Weblogs	5	20	3452	1001
	Opinion	12	20	3861	1157
LitBank	Fiction	4	30	5455	1494
QuizBowl	Quizzes	20	20	3304	1083
ARRAU	News	3	20	3336	885
GUM	Biographies	4	20	3422	1119
	Fiction	4	20	3299	1008
Phrase	Wikipedia	7	20	3509	1003
Detectives	Fiction	4	20	4007	1063
	Opinion	7	9	1692	495
PreCo	News	4	8	1318	369
	Fiction	2	2	378	105
	Biographies	1	1	152	53
Total	All	83	240	42108	12200

Table A1: All datasets analyzed in this work with their breakdown based on domains, number of documents, passages, tokens and mentions annotated.

A.2 Manual Qualitative Analysis

Dataset	Mistakes (our)	Mistakes (gold)
PD (silver)	22	76
OntoNotes	81	49
PreCo	12	33
GUM	48	25
ARRAU	33	16
LitBank	21	13
QuizBowl	67	10

Table A2: Number of mistakes in our crowd annotations vs. gold datasets, obtained through a manual analysis.

A.3 Detailed Mention Detection Algorithm

- We identify all noun phrases using the Stanza dependency parser (Qi et al., 2020). For each word with a noun-related part-of-speech tag,²² we recursively traverse all of its children in the dependency graph until a dependency relation is found in a whitelist.²³ The maximal span considered as a candidate mention thus covers all words related by relations in the whitelist.
- Possessive nominal modifiers are also considered as candidate mentions. For instance, in the sentence “Mary’s book is on the table,” we consider both “Mary” and “Mary’s book” as mentions.

²²Pronouns, nouns, proper nouns, and numbers.

²³The whitelist includes all multi-word expression relations (i.e., compound, flat, and fixed) and modifier relations (i.e., determiners, adjectival modifiers, numeric modifiers, nominal modifiers, and possessive nominal modifiers).

- Modifiers that are proper nouns in a multi-word expression are considered as mentions. For instance, in “U.S. foreign policy,” the modifier “U.S.” is also considered as a mention.
- All conjuncts, including the headword and other words depending on it via the conjunct relation, are considered mentions in a coordinated noun phrase. For instance, in the sentence, “John, Bob, and Mary went to the party.”, the detected mentions are “John,” “Bob,” “Mary,” and the coordinated noun phrase “John, Bob, and Mary.”
- Finally, we remove mentions if a larger mention with the same headword exists. We allow nested spans (e.g., [[my] hands]) but merge any intersecting spans into one large span (e.g., [western [Canadian] province] is merged into [western Canadian province]).

A.4 Inter-Annotator Agreement Among Our Annotators Across Domains

Figure 6 illustrates agreement among our annotators computed with B3 scores including singletons.

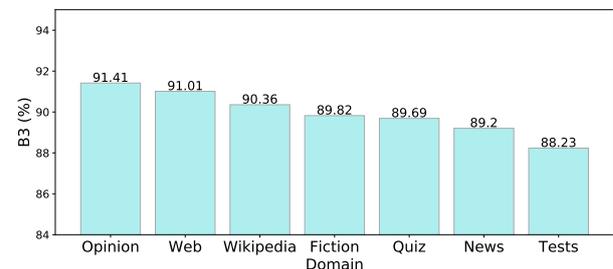


Figure 6: Inter Annotator Agreement across different domains. B3 scores with Singletons included.

A.5 Another illustrative example

An example of a single sentence annotated by two datasets, OntoNotes and ARRAU. These annotations differ widely from each other in kinds of mentions and links between mentions.

OntoNotes: [Lloyd’s, once a pillar of [the world insurance market]e1,]e2 is being shaken to [its]e2 very foundation.

ARRAU: [Lloyd’s, once [a pillar of [the world [insurance]e3 market]e2]eS1]e1, is being shaken to [[its]e1 very foundation]eS2.

System	Annotate all clusters	Pre-identified Mentions	Open Source	Webapp	Coref only	Keyboard and Mouse	MTurk Tested	Non-expert Terminology	Nested Span Support	Interactive Tutorial
Stenetorp et al. (2012)	✓	✗	✓	✓	✗	✗	✗	✓	✗*	✗
Widlöcher and Mathet (2012)	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Landragin et al. (2012)	✓	✗	✓	✗	✗	✗	✗	✗	✓	✗
Yimam et al. (2013)	✓	✗	✓	✓	✗	✗	✗*	✗	✓	✗
Poesio et al. (2013)	✗	✓	✗	✓	✓	✗	✗	✓	✓	✓
Girardi et al. (2014)	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗
Mueller and Strube (2001)	✓	✗	✓	✗	✓	✗	✗	✗	✓	✗
Kopeć (2014)	✓	✗	✓	✗	✓	✗	✗	✗	✓	✗
Guha et al. (2015)	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗
Oberle (2018)	✓	✗	✓	✓	✓	✗	✗	✗	✓	✗
Reiter (2018)	✓	✗	✓	✗	✓	✗	✗	✗	✓	✗
Bornstein et al. (2020)	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓
Prodigy*	✓	✓	✗	✓	✓	✗	✓	✗	✗	✓
ezCoref (this work)	✓	✓	✓*	✓	✓	✓	✓	✓	✓	✓

Table A3: A comparison of different coreference annotation tools. (* — ezCoref code will be open-sourced upon paper publication; Stenetorp et al. (2012) did not implement nested spans originally, but later added them with limited functionality. Yimam et al. (2013) have APIs for CrowdFlower integration, but suggest expert annotators.).

*Accessible at: <https://prodi.gy/>

Tutorial feedback from our crowd annotators

1. This was a really interesting task. The tutorial was very clear and easy to understand. I think it was very helpful when I completed the final passage.
2. Very great tutorial, I loved how it walked me through each and every step making sure I understood.
3. excellent interface and very precise instructions! out of curiosity, what is the time-frame and scale for this project? several weeks? months? hundreds or thousands of hits? I have a ton of projects during the autumn normally but will definitely make time for this if it's going to be around for more than a day or two. Looking forward to working with you folks if possible!
4. I actually enjoyed this. Thank you for the opportunity.
5. it was interesting a bit difficult but overall gave a lot of feedback necessary to do a good job.
6. I loved the tutorial and the layout. I am still a little bit unsure about a couple of the entities and hope I got it right. For example: would 'legs' be in 'his' because it refers to that person? I wasn't sure and made them separate.
7. I loved how this tutorial was set up. It was easy to use and made me very interested in doing the actual HITs. It would have been nice to be able to print out a quick reference guide or something, so we could refer to the instructions from before while we completed the final task. I don't think it would be needed for very long after starting the real HITs, but it would still be nice to have.
8. On the last test section, there was no place for feedback. There was a section that said ""it was getting dark"" ""It was getting late"" Both of those refer to a time of day, but one is light, one is the hour, so I marked them as different. Not sure of how broad or narrow we need to be when justifying ""same"" entities, as there is an argument either way.
9. I just wanted to say that I really appreciated how efficiently put together and clear this tutorial was.
10. This was a unique task. Thank you.
11. I feel much better with the help and feedback. It was interesting and definitely way different in a good way than the usual survey. I did my best and I hope I did well enough. Keep safe and Happy Holidays no matter what happens.

Table A4: Some of the comments received from our annotators after completing the tutorial. We received overwhelmingly positive feedback; annotators sometimes also mentioned cases they found confusing.

Coreference Tutorial

Welcome!

This is a **paid tutorial** for the "**Large-Scale Coreference Annotation Task.**"

In this tutorial you will learn how to annotate **coreferences**, that is, words and phrases that refer to the same people or things.

Upon completing the tutorial, you will get a **completion code**. You **MUST enter this code** in the textbox below and **submit the HIT** in order to receive the payment.

Depending on your performance, you might be invited to participate in our "Large-Scale Coreference Annotation Task."

Before proceeding to the tutorial, please **fill in the following survey**:

What is your **gender**?

What is your **age**?

What is your **native language**?

How is your **English level**?

- Beginner
- Intermediate
- Advanced (near native)
- Native speaker

What is your **education level**?

- Primary
- Secondary
- College
- Graduate School

[Click this link to begin.](#)

[OPTIONAL] We would love to hear **your feedback** about **this tutorial**.

Submit your code below:

Figure 7: Screenshot of tutorial task invitation on AMT with detailed instructions.

Coreference Tutorial Mode

Welcome to the coreference tutorial mode. Here you will learn how to use the interface efficiently to label text for coreferences.

What are **coreferences**?

A coreference is when **two words** or **spans** (sequence of words) refer to **the same thing**.

In the examples below, the following words are coreferences (they refer to the same "thing"):

- (1) **"John"** and **"He"**
- (2) **"Robert"** and **"He"**
- (3) **"Alice"** and **"Her"**

John is cool. He is nice.

Robert loves Alice. He talks to her everyday.

Let's get started.

Figure 8: Tutorial Interface (Introductory prompt)

Select Spans (Task 1 of 10)

Step 1 of 2

Observe how the border around "Mary" is flashing. This means the span "Mary" is the current target.
Click on all the spans that refer to the target "Mary."

Happy Annotating!

Currently Annotating Entity 0

Mary

Mary is fun.
She jokes a lot.
That's why Mark likes her.

← Previous Target Next Target →

⌂ ⌂

Shortcuts

Function	Key
Previous Target	a
Next Target	d
Undo	Ctrl + Z
Redo	Ctrl + Y

Continue

Figure 9: Tutorial interface: A sample prompt teaching tool functionality.

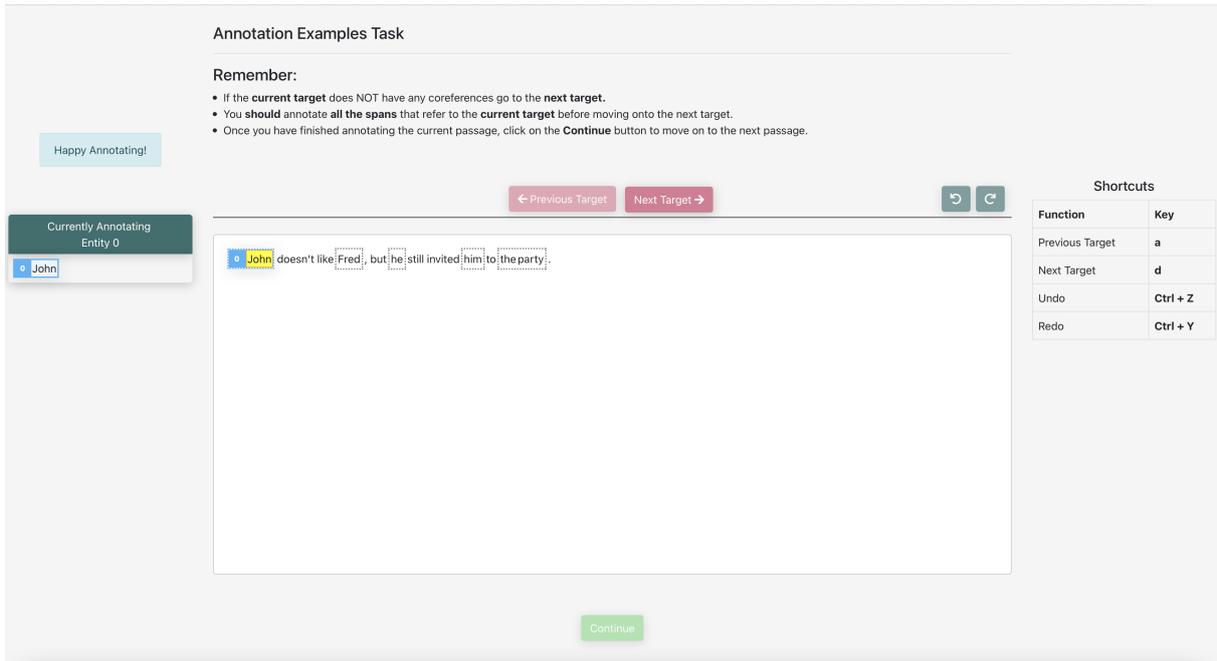


Figure 10: Tutorial interface: A sample prompt teaching basic coreferences.

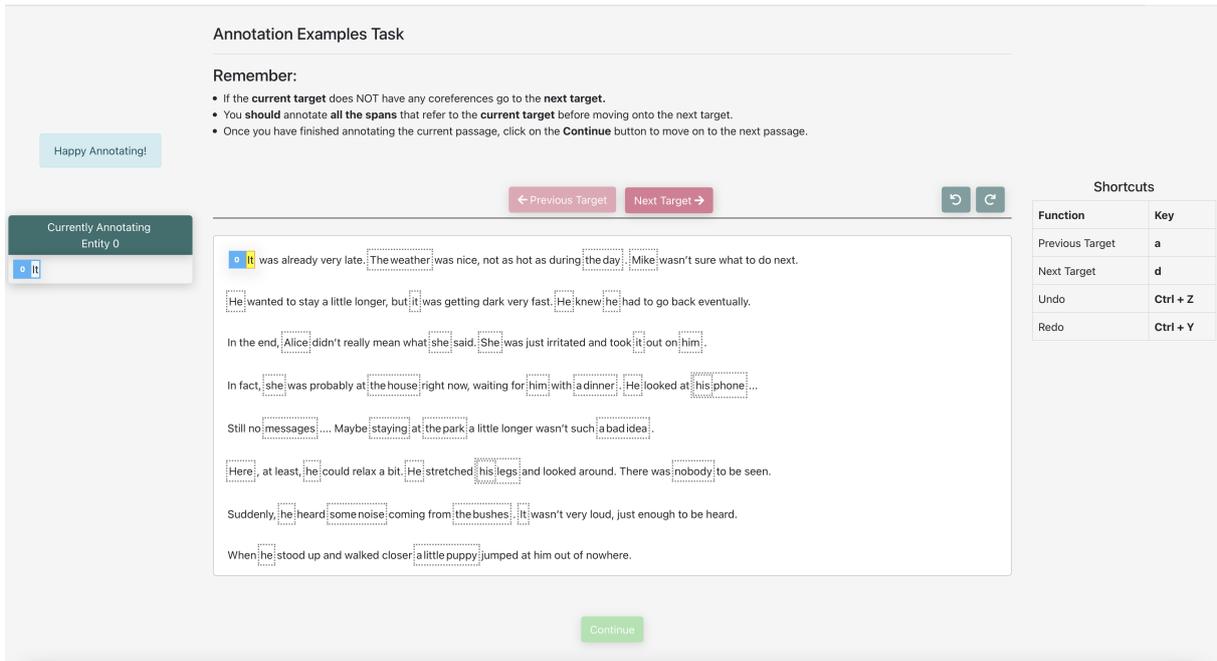


Figure 11: Tutorial interface: quality control example.

Coreference Annotation Task

Welcome to the coreference annotation task. In this task you will be asked to annotate a short paragraph for coreferences. If you need to review the tutorial, please follow this [link](#).

What are **coreferences**?

A coreference is when **two words** or **spans** (sequence of words) refer to **the same thing**.

In the examples below, the following words are coreferences (they refer to the same “thing”):

- (1) "John" and "He"
- (2) "Robert" and "He"
- (3) "Alice" and "Her"

John is cool. He is nice.

Robert loves Alice. He talks to her everyday.

[Click this link to begin annotation.](#)

[OPTIONAL] We would love to hear **your feedback**. Let us know if anything was unclear or particularly challenging.

Submit your code below:

Submit

Figure 12: Annotation task invite on AMT with detailed instructions

PREME: Preference-based Meeting Exploration through an Interactive Questionnaire

Negar Arabzadeh
University of Waterloo
narabzad@uwaterloo.ca

Julia Kiseleva
Microsoft Research

Ali Ahmadvand
Emory University

Yang Liu
Microsoft Research

Ahmed Hassan Awadallah
Microsoft Research

Ming Zhong
University of Illinois

Milad Shokouhi
Microsoft Research

Abstract

The recent increase in the volume of online meetings necessitates automated tools for organizing the material, especially when an attendee has missed the discussion and needs assistance in quickly exploring it. In this work, we propose a novel end-to-end framework for generating interactive questionnaires for preference-based meeting exploration. As a result, users are supplied with a list of suggested questions reflecting their preferences. Since the task is new, we introduce an automatic evaluation strategy by measuring how much the generated questions via questionnaire are answerable to ensure factual correctness and covers the source meeting for the depth of possible exploration.

1 Introduction

In recent years, video conferencing technology has gained substantial improvements, and thus, online meetings have become easily accessible and more prominent. Primarily due to the pandemic and working from home, the need for video calling has grown significantly. Therefore, the high volume of online meetings necessitates automated tools for managing and organizing essential information for attendees. Especially when an attendee has missed an online meeting, it is critical to access the required information since quickly reading through the transcript is quite time-consuming.

Providing meeting summaries is a promising direction (Wang and Cardie, 2013; Jacquenet et al., 2019; Zhao et al., 2019; Singhal et al., 2020). However, recent studies show that 1) users’ needs do not fully align with current approaches to automatic text summarization (ter Hoeve et al., 2020, 2022) and 2) approaches designed for document summarization could not effectively apply to meetings transcripts (Murray et al., 2010; Mehdad et al., 2013; Li et al., 2019) due to the following potential reasons: **(R1) Structure:** standard documents are well structured compared to meeting transcripts;

The following subjects were discussed in the meeting.
Which subject are you more interested in?

Remote Control Cases Remote Control Design
 Remote Control Functions Remote Control Buttons
 New Remote Control Remote control Price

What do you want to know more about the New Remote Control?

Fronts Features advantages
 Disadvantages Think

Additional questions you might be interested in:

- What is the new feature of the front of the remote control?
- What are different colors of the front for the remote control?
- What are the latest trends for a front under remote control?
- What is the difference between front and back of the remote control?

Figure 1: An example of exploring one of the meetings from the collection (Carletta et al., 2005) based on user preferences through an interactive questionnaire.

(R2) Language: spoken language used in meetings is less regular than documents; and **(R3) Multiple speakers:** the speaker role is essential. Moreover, there is little meeting data publicly available that can be used for experimentation compared to regular documents such as news or articles. In contrast with document summarization, when summarizing a meeting, different users tend different preferences on what content should be included in the summary. Therefore, there is an increasing calling for alternative ways of summarizing, especially for meetings transcripts. Recently, Zhong et al. (2021) attempted to tackle this problem by proposing a query-based multi-domain meeting summary, where a user provides a query in question form, e.g., ‘*What was the discussion about the jog dial’s function when talking about changes in the current design?*’ to locate the part of the transcript that related to the query and then summarize. However, when attendees have missed the meeting, they cannot formulate such questions due to no prior knowledge about the meeting. To overcome this, we aim to address the following **research challenge:** *How can attendees effectively explore a meeting content without having prior knowledge about it?*

This work is motivated by the fact that asking questions is a more efficient way for humans to ac-

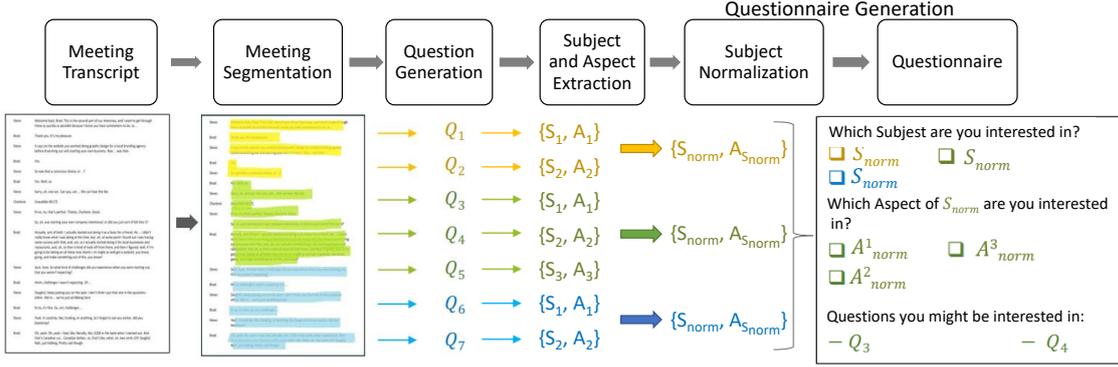


Figure 2: Overview of our framework, Preference-based Meeting Exploration through an Interactive Questionnaire (PREME), where Q is a comprehensive set of questions, and S_i and A_j are extracted pairs of subjects and aspects.

quire information than notes in plain text (Lawson et al., 2007, 2006; Aliannejadi et al., 2021). Thus, we address preference-based meeting exploration by automatically generating a structured interactive questionnaire for a transcript that covers most of the discussed topics and quickly walks users through the discussed content. An example of the desired questionnaire is shown in Fig. 1. First, the user has the ability to express their preferences regarding *subjects* that have been discussed (Solbiati et al., 2021; Huang et al., 2018; Zhang and Zhou, 2019; Sehikh et al., 2017). Next, the questionnaire interactively suggests narrowing down their exploration if possible by displaying a list of possible related *aspects*. As a result, a ranked list of questions reflecting user preferences is generated. Next, the user can pick a question that demonstrates their seeking needs the most and is redirected to the meeting part containing an answer. Interactively asking for preferences in the questionnaire is beneficial because the user oversees what has been covered during the meeting they have missed. In section 4.2 we elaborate on a user study on a number of professionals who find such application useful for their daily job. Hence, the goal of proposed questionnaires is two-fold: **(G1)** to compactly represent the discussed content; **(G2)** to guide users to form questions that express their preference regarding the transcript. We require the generated questionnaire to satisfy the following properties:

- P1 Coverage:** coverage is the amount of the information from the source text that a questionnaire points to. The generated questionnaire must cover the meeting as much as possible;
- P2 Answerable:** a given meeting transcript should contain the answers to the questions generated as a result of the questionnaire.

To address the defined challenge, we propose

a framework, PREME, which consists of several concrete sequential steps highlighted in Fig. 2. We start by enchaining the method to extract meeting segments (Solbiati et al., 2021). Due to the conversational nature of the meeting, topic detection from the segments is challenging (Huang et al., 2018; Zhang and Zhou, 2019; Sehikh et al., 2017). Thus, we indirectly extract the topics as follows. First, we generate questions from each segments (Brown et al., 2020) since extracting topics from the questions is much more well studied. Further, we employ a trained Conditional Random Field (CRF) model to tag subjects and aspects (Fig. 1) from generated questions originated from each segments (Wallach, 2004). Once we got each segment’s topic list, we proposed a strategy to normalize them to reduce the number of options in the questionnaire. Recently, Deutsch et al. (2020) demonstrated that QA-Based evaluation is strongly correlated with human opinion. Thus, to evaluate PREME, we employ a similar QA-based strategy.

To summarize, the main contributions are:

- C1** We propose PREME, a novel framework to enable meetings exploration based on user’s preferences through an interactive questionnaire;
- C2** We propose a new method for subject normalization which returns the most informative subject from a set of phrases and keywords;
- C3** We introduce a new automatic evaluation strategy for measuring the effectiveness of the proposed questionnaire to assess the required properties **P1** and **P2**, which according to (Deutsch et al., 2020) has a strong correlation with human judgments; and
- C4** We open-source a dataset that includes 1000 questions comprehensively annotated with subject to their subjects and aspects at <https://github.com/microsoft/preme>

2 Related Work

2.1 Automatic Textual Summarization

Automatic text summarization task has attracted lots of attention across Natural Language Processing (NLP) community recently. Many systems are proposed to summarize documents in different domains, including news (Rush et al., 2015; Nallapati et al., 2017; See et al., 2017; Celikyilmaz et al., 2018; Liu and Lapata, 2019; Zhang et al., 2020), academic papers (Manakul and Gales, 2021; Huang et al., 2021) and books (Kryściński et al., 2021). Meeting summarization has also emerged as a widespread need recently. Due to the unique discourse structure of dialogues, conventional document summarization systems are facing challenges when summarizing meetings (Li et al., 2019; Zhu et al., 2020). Thus, new models are proposed for tackling this task. Wang and Cardie (2013) employ decisions, and action items in dialogues to generate the summary progressively. Oya et al. (2014) propose a template-based meeting summarization system by learning the relationship between summaries and their source meeting transcripts. Shang et al. (2018) design an unsupervised meeting summarization model with multi-sentence compression techniques. Li et al. (2019) introduce multi-modal information into meeting summarization with a hierarchical attention mechanism. Zhu et al. (2020) propose a hierarchical meeting summarizer that can process both word-level and turn-level information of dialogues. Furthermore, the community noted that due to the lengthy content and distributed information, a general summary of the meetings does not necessarily satisfy what users seek. Thus, Query-based summarization methods have become more prevailing for generating concise and specific summaries. (Litvak and Vanetik, 2017; Nema et al., 2017; Baumel et al., 2018; Ishigaki et al., 2020; Kulkarni et al., 2020, 2021; Pasunuru et al., 2021). Recently, Zhong et al. (2021) proposed a new framework of query-based summarization for meetings, in which they annotate QMSUM, a query-based multi-domain meeting dataset. Each QMSUM meetings come along with a set of queries with different levels of abstractness, i.e., general queries and specific queries. Human annotators write these queries, and the summaries align with these queries after reading the meeting transcripts.

While query-based summarization can be a proper path to provide users with meeting information at different specificity levels, we argue that

issuing such specific queries still requires a certain degree of background knowledge. In real-life scenarios, users might not be equipped with that knowledge and issue informative queries, especially when they did not attend the meeting. Hence, they can not benefit from query-based summarization techniques to explore the meetings. We address the drawbacks of query-based summarizers by providing users with an interactive questionnaire which provides them with potential queries and allows them to explore the meetings more flexibly.

2.2 Evaluation of Summaries Factuality

The summaries often has called out for hallucination issues (Maynez et al., 2020). Thus, Wang et al. (2020) propose a framework to evaluate factual consistency of summaries with the source text. Similarly, Deutsch et al. (2020) propose a Question Answering (QA)-based evaluation approach on summaries' content quality. They measure how much information is contained in a candidate summary by calculating the proportion of questions it can answer. These approaches inspired us for automated end-to-end evaluations of the questionnaires.

2.3 Question Generation and Filtering

Initial works in Question Generation task leveraged crowd-sourcing or rule-based methods to generate pre-defined question templates (Mostow and Chen, 2009; Rus et al., 2010; Lindberg et al., 2013; Fabri et al., 2020; Mazidi and Nielsen, 2014; Labutov et al., 2015). Heilman and Smith (2010) tackled this problem by over-generating candidate questions and then using a learning to rank framework to rank them to filter the low-quality questions. SQUASH (Krishna and Iyyer, 2019) is one of the recent works in which authors used question generation methods to convert a document into a hierarchy of question-answer pairs with the focus on questions' granularity level. They employed a neural encoder-decoder model trained on three reading comprehension data sets, i.e., SQuAD (Rajpurkar et al., 2016), QuAC (Choi et al., 2018), and CoQ (Reddy et al., 2019) to generate the questions, and further, they filtered out the unanswerable questions using some heuristics and question answering models. While question generation using question answering data sets seems a general approach, this method does not work well on meeting-related questions generated due to many reasons, including: (1) Different structure of meetings compared to documents; (2) There are not many ques-

tion-answering datasets available from meetings; (3) Sometimes, the answer to questions generated from meetings could be very long, making it hard to fit the context in neural models. In our work, we introduce an automatic method that can generate questions regarding the meeting to overcome the high price of collecting with annotators.

2.4 Questionnaire Organization

Obtaining users preferences has always shown to be a challenging task (Jiang et al., 2008; Rokach and Kisilevich, 2012; Anava et al., 2015; Christakopoulou et al., 2016; Sepliarskaia et al., 2018). The task becomes more challenging when we aim to minimize the number of interactions with users to get to know their preferences. Sepliarskaia et al. (2018) reformulate this task as an optimization problem. They propose a static questionnaire by choosing a minimal and diverse set of questions. Similarly, in Liu et al. (2019) proposed a dynamic questionnaire generation method for search of clinical trials. Quiz-style question generation has also been explored recently by Lelkes et al. (2021). The authors have formulated the problem as two sequence to sequence tasks, including the question-answer generation step and incorrect answer generation step. We argue that while the former step seems relevant to our work, it could not be adapted to meeting transcripts since their proposed dataset has been trained on factual question answering data sets and cannot be used for meeting purposes. All in all, we can conclude that creating questionnaires are still under exploration in different domain. Hence, our effort in organizing a questionnaire, especially for meetings, is timely and useful for future research.

3 Proposed Framework: PREME

This section explains PREME, our proposed novel methodology to explore meetings based on users' preferences through an interactive questionnaire. An overview of our methodology is shown in Fig. 2 in which we first apply a topic segmentation method (Solbiati et al., 2021) on meeting transcript to retrieve segments with different topics (Section 3.1). Then, we generate a set of all possible questions from each segment (Section 3.2). Further, we extract the most informative part of the questions, i.e., the subject and aspect of each question (Section 3.3). In the last step, we map the normalized subjects and aspects with generated questions and form the questionnaire (Section 3.4).

3.1 Meeting Segmentation

A meeting transcript can be extremely long and contain discussions of various topics. Therefore, our goal is to divide the meeting text into a sequence of topically coherent chunks. Thus, we adopted an unsupervised topic segmentation method based on the contextualized presentation of meeting (Solbiati et al., 2021). In this topic segmentation method, the authors compute the BERT embeddings for every utterance of the meeting transcript. Further, they curated blocks of utterances and performed a block-wise max-pooling operation to generate contextualized embedding for each block. Then, the semantic similarity between two adjacent blocks is captured, and a change in the topic is detected if two adjacent blocks show similarity below a certain threshold. This approach has several advantages, including: (1) It is unsupervised; (2) Since we are just converting the meeting into smaller pieces, and we are not losing any part of the meeting.

3.2 Question Generation

For question generation from a segment, we leveraged the powerful GPT-3 model (Brown et al., 2020). An impressive capability of the GPT-3 is to generate very realistic results from few training samples or even no training sample (few-shot and zero-shot learning). The variety of the generated content can be controlled using a temperature hyper-parameter. To expand the size of generated questions' pool as much as possible, in each segment, the API is called in a zero shot learning model with different temperature values between [0-1] with a 0.05 margin, where the value closer to 1 means more diversified questions. We set the maximum output length to 128 tokens and then we repeat the process for 10 trials for each specific temperature. Given that the maximum context window for the API was 2048 tokens, we truncate and slide by half-a-window size of 2048 tokens whenever a segment includes more than 2048 tokens. As a results, A list of questions is extracted based on random initialization in each API call, meaning different results are achieved even with the same hyper-parameters. We extracted five questions on average per segment in each call. Finally, a union across all runs is used to form our question pool.

3.3 Subject and Aspect Extraction

Every of the generated questions has one or more *subject(s)* that is defined as the principal matter that attendees have discussed, i.e., the main con-

Table 1: Examples of annotated questions with their subjects and aspects. Subjects are highlighted in red and Aspects are highlighted in green.

Q1	What is the arrow symbol on the remote control for?
Q2	What are the main frustrations people have with the remote control ?
Q3	How will the logo and color scheme be incorporated into the product ?
Q4	What are pros and cons of having a remote with a large number of buttons ?
Q5	What is the most difficult part of the project from the industrial engineer's point of view?

cern of the questions. Some questions might point to a specific *aspect(s)* of the subject which is defined as the mentioned details about a given subject. We aim to extract the primary subjects from any question and the detailed aspect if it is mentioned. Table 1 shows examples of annotated *subjects* and *aspects* for a few questions. For instance, in the question “What is the arrow symbol on the remote control for?”, “remote control” is annotated as the subject and the “arrow symbol” is the specific aspect of the subject. To extract the subjects and aspects from the questions, we use CRF (Wallach, 2004). We examined SOTA keyword extraction and contextualized neural embedding-based topic extraction models; however, the CRF model which uses word’s identity, suffix, shape and POS tags as features, seems to work the best among them. To train the CRF model, we were required to have annotated questions with subjects and aspects labels. We designed an annotation study using the UHRS¹ crowd-sourcing platform, where we carefully trained annotators with detailed instructions to label randomly selected 1000 questions generated by GPT3 with their subject and aspects². Each question has been assigned to two annotators, and we report the annotators’ agreement in Section 4. Further, we employ the trained CRF model to extract subjects and aspects from the questions.

3.4 Questionnaire Generation

Given a meeting transcript, for each of its segment T which was initially supposed to coherently point out one subject, we generate Q_T , a set of generated questions from T . In other words, given an ideal meeting segmentation method, each segment

¹<https://prod.uhrs.playmsn.com/uhrs/>

²We invested in having a few well-trained annotators rather than having a high number of annotators who have not been trained well. Thus, annotators were paid hourly and by the quality of their work and they had no intentions for cheating.

is supposed to be pointed to one subject. Thus, we assume that each segment has only one valid topic and as shown in Figure 2, each segment is being represented with one S_{norm} . We create a set S_{Q_T} by extracting the subjects from each question in Q_T . Therefore, for the segment T , we have at least $|Q_T|$ number of subjects. Extracted subjects from a question set with the same origin segment must be normalized so that one comprehensive, general, and informative subject presents a segment. The more the selected subject representative covers other concepts in S_{Q_T} , the better normalization we employed. This subject normalization reduces the number of subjects shown to the user at the first step of the questionnaire and will decrease the user’s effort, causing figuring out users’ preferences by asking them the minimum number of questions. In other words, our goal is to select a single subject S_{norm} from S_{Q_T} which represents S_{Q_T} in the most informative way. To do so, we define the notion of the subject network as follows.

Definition 3.1. Given a segment T , a set of generated questions Q_T , and extracted subjects S_{Q_T} , a subject-network for $G(S_{Q_T})$ is denoted as $G(S_{Q_T}) = (\mathbb{V}, \mathbb{E}, w)$. It is a weighted undirected graph, where $\mathbb{V} = \{s_i \in S_{Q_T}\}$, and $\mathbb{E} = \{e_{s_i, s_j} : \forall s_i, s_j \in \mathbb{V}\}$. The function $w : \mathbb{E} \rightarrow [0, 1]$ is the cosine similarity between the semantic relatedness of the contextualized embedding vectors of two incident subjects of an edge e_{s_i, s_j} , i.e., v_{s_i} and v_{s_j} .

In Def. 3.1, we propose a subject-network where subjects are connected, and edge weights represent the semantic similarity between the two subjects. We hypothesize that the node with highest similarity and connection to others is the most central one. In other words, since it has great similarity to other subjects, there is a high probability that it points to a more generic concept and that covers the other subjects. Hence, the node S_{norm} should have high centrality attribute to represent the main subject of segment S . We employed PageRank (Haveliwala, 2003) value to find the most important and informative node in this network. Similarly, PageRank has shown to have a high correlation with the most important nodes and has been used in tackling different tasks such as quantifying term’s specificity or ranking problems in different information retrieval tasks (Arabzadeh et al., 2020, 2019; Kurland and Lee, 2010). We measure the PageRank score of each node and select the node

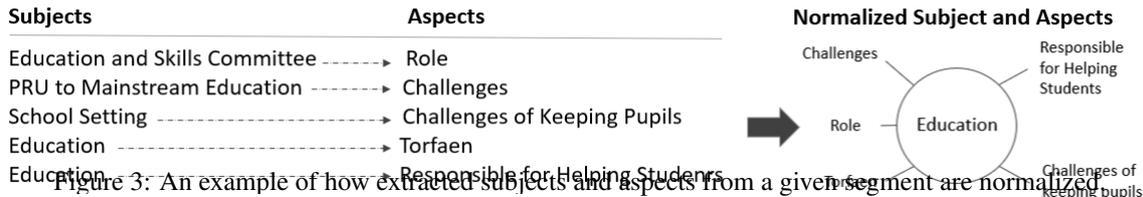


Figure 3: An example of how extracted subjects and aspects from a given segment are normalized

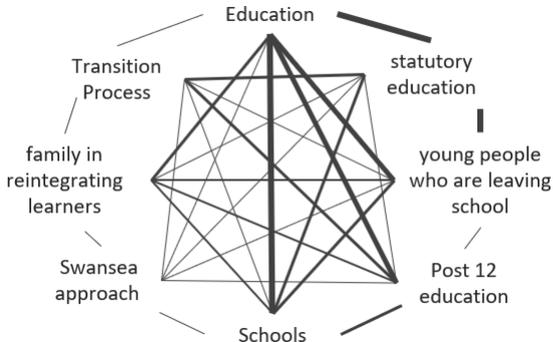


Figure 4: An example of subject-network built for one extracted segments from (Janin et al., 2003). The edge weights represent the semantic similarity between each nodes. Higher weights are shown with higher width.

with the highest PageRank value as the representative subject S_{norm} of the subject set S_{Q_T} for segment T . In other words, we represent each segment T by subject S_{norm} where $PageRank(S_{norm}) > PageRank(s_i)$ for every $s_i \in \mathbb{V}$.

Fig. 4 displays a subject-network generated from extracted subjects from one of the meetings’ segments in the QMSUM dataset. subjects such as “Education”, “Schools,” “Young people who are leaving school” are included in this subject set and represented by nodes in this subject-network. Further, we connect every pair of nodes in this graph, and the edge weight is directly related to their semantic similarity. As presented in Fig. 4, some nodes have higher edge weights which their connected lines are shown with greater width. We measure page rank in this weighted network. Here “Education” got the highest PageRank value in this subject-network. Hence, we present these subjects by one subject, i.e., “Education”. “Education” can be a promising representative for these subjects as it covers more specific concepts such as “schools”, “statutory education,” and “post 12 education.”

Next, the extracted aspects from each question set should be mapped to their representative subject. We remove the redundant and repetitive aspects and subjects by removing those who have highly similar n-grams. Plus, There might be several subjects existing in S_{Q_T} which all point out to S_{norm} , and they might be semantically very similar. In this step, we must be concerned not to lose any aspect because of subject normalization. We aim to

Table 2: Annotators agreement on annotated questions with respect to subjects and aspects using Krippendorff’s score (Krippendorff, 2011)

	Subject	Aspect
Hard [Exact Match]	0.459	0.415
Soft [At least one term matched]	0.490	0.485

map every aspect from S_{norm} and every s_i in S_{Q_T} which is highly similar to S_{norm} to maximize the potential of questions we might want to show at the end of the questionnaire. For instance, in Fig 3 we display a few extracted subjects and aspects from one segment. If we only consider “education” and its related aspect, we will lose many aspects that users might be interested in, and as a result, the questionnaire coverage will drop. On the other hand, if we merge the highly similar representative subjects with, e.g., “school setting” and “Education and Skills Committee,” we will have a broader host of questions to suggest to users. Therefore, we will filter out dissimilar subjects from S_{Q_T} to S_{norm} and map extracted aspects from filtered S_{Q_T} to S_{norm} as it is shown in Fig. 3. As a result, if “education” is the subject of interest for a user, they have the opportunity to select which aspects of education they are more interested in, such as “Role” of education or “challenges” of education. Finally, we will show users the questions in which the selected aspects and normalized subjects have appeared.

4 Evaluation Methodology

For experiments, we use the QMSUM dataset (Zhong et al., 2021), which includes 232 product, academic, and committee meetings (Janin et al., 2003; Carletta et al., 2005). Each meeting comes with a set of general and specific questions; the general ones are out of the scope of this work since they refer to very broad concepts, e.g., “summarize the whole meeting.”. Further evaluations are conducted on the QMSUM test set.

4.1 Evaluating Framework Components

The proposed framework consists of several steps (Fig. 2). The used *meeting segmentation* (Solbiati et al., 2021) method has shown to outperform baselines (Hearst, 1997; Beeferman et al., 1999; Badjatiya et al., 2018). Hence, we refer to original paper for evaluation results.

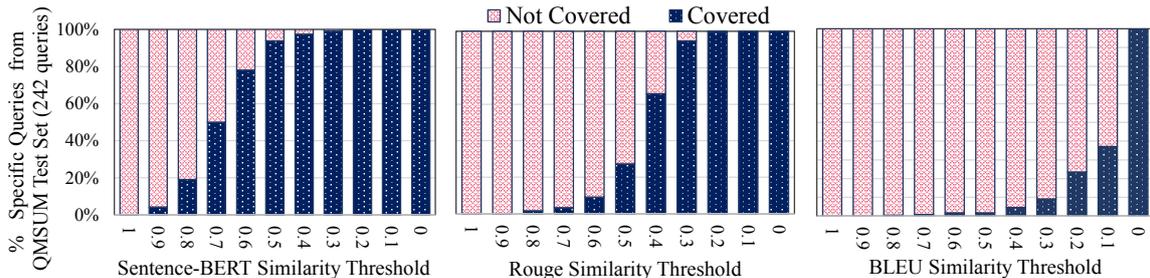


Figure 5: Coverage of PREMEon QMSUM test set considering different similarity metrics and threshold

Table 3: CRF performance on extracting subjects and aspects of questions using 10-fold cross validation

	Precision	Recall	F1-Score
Subject	0.64	0.69	0.67
Aspect	0.89	0.80	0.84
N/A	0.63	0.73	0.68

Evaluating Question Generation: We evaluate the quality of our generated questions by measuring the fraction of generated questions by human annotators in QMSUM that we covered in PREME. We assume the specific queries in the QMSUM dataset enjoy relatively high quality because annotators issued them after comprehensively reading the transcript (gold standard questions). Hence, Fig. 5 reports the similarity between most similar questions generated by PREME and the gold questions by three different similarity metrics i.e., Sentence-BERT similarity (Reimers and Gurevych, 2019), Rouge F-1 score (Lin, 2004), and BLEU-4 score (Papineni et al., 2002). We assume a questions from QMSUM is covered if there is at least a question generated by PREME that has similarity is higher than a certain threshold $t \in [1, 0.9, \dots, 0.1, 0]$. We report the percentage of ‘Covered/Not Covered’ questions based on different similarity matching thresholds. Based on Fig. 5 we conclude while we cover a relatively fair number of specific questions, there is still room for improvement. However, we should note that the questions in QMSUM are very limited, and initially, they were not supposed to cover all possible questions that one could raise from the meeting. Additionally, we observe that questions in QMSUM, which are issued by humans, include more abstractive questions while our generated questions inclined toward more factual ones.

Evaluating Subject and Aspect Extraction: To assess the quality of the collected dataset, we measure Krippendorff’s alpha agreement between annotators (Krippendorff, 2011) for extracted *subject* and *aspect* of the 1000 questions generated from the training set. Tab. 2 shows annotators have

agreement ~ 0.4 , which is interpreted as “Moderate” agreement for such a challenging task. Since different annotators might selected different section of the text, Tab. 2 reports both *hard* and *soft* agreements. we trained the CRF model using *crfsuite* library and evaluated it by 10-fold cross-validation. Given each term in the questions, the model predicts whether the term is considered the subject, aspect, or not applicable for labeling (N/A). Tab. 3 shows the result of the CRF model evaluation in terms of precision, recall, and F1 scores. We notice that the model shows better performance on detecting aspects compared to the subject.

4.2 Evaluating Questionnaires

To the best of our knowledge, we are first to propose a preference-based questionnaire as a way for meeting exploration; thus, no particular gold standard benchmark or evaluation metrics. Since we require users to express their preference, it makes it challenging to simulate ‘enough imaginative context’ among annotators. Thus, we conducted a user study to highlight the usefulness of exploring meetings through an interactive questionnaire. We provided 20 participants who were professional workers and graduate students aged between 24-41 with detailed explanations and examples of results generated by PREME such as in Figure 1. Participants on average had over 5 hours of online meetings per week. Among which, over 80% of them reported that they need to explore the content of a past meeting, at least a couple of times a week. Finally, over 80% of participants agreed on finding PREME useful for meetings exploration. Also, we introduce a new evaluation strategy that satisfies the desired properties on coverage (P1) and the existence of answers in the transcript (P2). The proposed automatic metrics capture if our framework is ready to be tested through a more comprehensive user study in the future, when we can run a pair-wise preference-based comparison between PREME and other meeting exploration methods.

Table 4: Test set statistics and PREME Performance: Average number of generated questions and Coverage.

	#Meetings	Average # Turns	Average # Questions	Coverage (%)
Academic	9	893	1257	83.07%
Committee	6	214	1105	64.04%
Product	20	569	724	86.25%
All	35	591	927	81.62%

Automatic evaluation: We utilize the model SOTA called Locator in (Zhong et al., 2021) in which, given the query, it can extract the relevant spans from the meeting. The Locator employs a hierarchical ranking-based model structure based on CNN (Kim, 2014) and Transformers (Vaswani et al., 2017) architecture. The Locator embeds each utterance of the meeting and feeds it to a CNN network by capturing the local features, and utilize Transformer layers to obtain contextualized turn-level representations. In addition, the speaker’s embedding is also concatenated to the features list. Finally, the model uses MLP to score each turn, and the turns with the highest scores are considered the relevant spans for each question.

To measure the coverage (to satisfy **P1**), we adopt the newly proposed QA-style of evaluation (Deutsch et al., 2020; Wang et al., 2020) which has shown to have substantial correlation with human judgments in terms of questions quality assessments. *Coverage* is defined as the fraction of a meeting that a questionnaire encompasses. To measure the coverage, first, the relevant answer spans for the existing questions in a questionnaire are located. Further, the proportion of utterances that were already located as relevance answer spans w.r.t. the whole meeting transcripts, is measured as the coverage. We believe that that is a promising indicator of questionnaire informativeness. The coverage is basically how much of the original meeting was covered by the questionnaire. We hypothesize that a good questionnaire should ideally include questions from all parts of a meeting. i.e., the questionnaire includes questions related to every part of the meeting so that users are able to explore their section of interest from the meeting. Therefore, the more the questionnaire covers the meeting, the better it is. To do so, we find the answer spans to the generated question in each questionnaire and we report the percentage of utterances that the locator detected as the answer span for all the questions in the questionnaire from the whole meeting. We run our experiments on the QMSUM test set. Tab. 4 shows the details of this test set. We over generate the questions and after

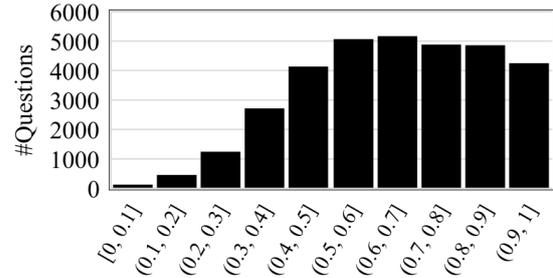


Figure 6: Histogram of Confidence Scores of Question-Answering model on generated questions from PREME.

removing the duplicates, on average, the questionnaire has 1257 unique questions from Academic meetings, 1105 questions from Committee meetings, and 724 questions from Product meetings. Further, Tab. 4 reports the percentage of utterances covered in each meeting. On average, our proposed questionnaire can cover 81% of the meeting. We also compared the coverage on different types of meetings. While our generated questionnaire covered Committee meetings the least (64%), the Product and Academic meetings show higher coverage (over 80%). Further, we evaluate how much the generated questions in PREME are answerable (to satisfy **P2**). Inspired by (Krishna and Iyyer, 2019), we run a pretrained QA model (Sanh et al., 2019) over generated questions and report the confidence score for each QA pair in Fig. 6. We use DistilBERT fine-tuned on SQUAD (Rajpurkar et al., 2016) dataset. We observe that more than 73% of generated questions from PREME on meetings in test set of QMSUM shows confidence score higher than 0.5 and more than 42% of questions shows confidence score greater than 0.7. The results confirm that a promising portion of generated questions are answerable.

5 Conclusions and Future Work

We proposed an end-to-end framework, called PREME, that allows automatically building a questionnaire that will enable users to explore the most of discussed subjects and their aspects if desired. As a result, users are supplied with questions about the meetings that express their information needs, and answers can be found in the transcript. Since simulating actual users’ preferences is challenging and requires hired annotators, we have ran a small user study as well running an automatic end-to-end evaluation strategy to demonstrate the desired properties (**P1** and **P2**) of the generated questionnaires. We publicly release the collected dataset of annotated questions concerning its subjects and aspects, the code for questionnaires generation, and our

evaluation procedure to carry forward the proposed state-of-the-art for the newly formulated problem. In future, and by proposing a new method for questionnaire generation will allow us to run a user study for pair-wise comparison of the methods and reveal the correlation between human and automatic evaluation metrics for the suggested task.

6 Limitations

Generally, there is not much data available for meeting exploration. Thus, all studies on this domain are limited by small training and exploratory data. Therefore, it would be beneficial for the community to collect more labelled meeting data for meeting exploration and organization purposes. Since PREME is made of different SOTA components, its performance is also limited by individual components. In future, novel attempts can be made to address this problem as an end-to-end framework. In addition, the future works should include an extensive human evaluation that will reveal additional requirements for the PREME to satisfy, which will suggest additional evaluation metrics. Plus, since this the first work on to tackle meeting exploration via questionnaire, the preference-based evaluation is not possible.

References

- Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2021. [Building and evaluating open-domain dialogue corpora with clarifying questions](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4473–4484, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Oren Anava, Shahar Golan, Nadav Golbandi, Zohar Karnin, Ronny Lempel, Oleg Rokhlenko, and Oren Somekh. 2015. Budget-constrained item cold-start handling in collaborative filtering recommenders via optimal design. In *Proceedings of the 24th international conference on world wide web*, pages 45–54.
- Negar Arabzadeh, Fattane Zarrinkalam, Jelena Jovanovic, Feras Al-Obeidat, and Ebrahim Bagheri. 2020. Neural embedding-based specificity metrics for pre-retrieval query performance prediction. *Information Processing & Management*, 57(4):102248.
- Negar Arabzadeh, Fattaneh Zarrinkalam, Jelena Jovanovic, and Ebrahim Bagheri. 2019. Geometric estimation of specificity within embedding spaces. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2109–2112.
- Pinkesh Badjatiya, Litton J Kurisinkel, Manish Gupta, and Vasudeva Varma. 2018. Attention-based neural text segmentation. In *European Conference on Information Retrieval*, pages 180–193. Springer.
- Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704*.
- Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34(1):177–210.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.
- Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*.
- Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.
- Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 815–824.
- Daniel Deutsch, Tania Bedrax-Weiss, and Dan Roth. 2020. [Towards question-answering as an automatic metric for evaluating the content quality of a summary](#). *CoRR*, abs/2010.00490.
- Alexander R. Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. [Template-based question generation from retrieved sentences for improved unsupervised question answering](#). *CoRR*, abs/2004.11892.
- Taher H Haveliwala. 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE transactions on knowledge and data engineering*, 15(4):784–796.
- Marti A Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64.

- Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617.
- Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*.
- Tai-Chia Huang, Chia-Hsuan Hsieh, and Hei-Chia Wang. 2018. Automatic meeting summarization and topic detection system. *Data Technologies and Applications*.
- Tatsuya Ishigaki, Hen-Hsen Huang, Hiroya Takamura, Hsin-Hsi Chen, and Manabu Okumura. 2020. Neural query-biased abstractive summarization using copying mechanism. In *European Conference on Information Retrieval*, pages 174–181. Springer.
- François Jacquenet, Marc Bernard, and Christine Largeron. 2019. Meeting summarization, a challenge for deep learning. In *International Work-Conference on Artificial Neural Networks*, pages 644–655. Springer.
- Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. 2003. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*, volume 1, pages I–I. IEEE.
- Bin Jiang, Jian Pei, Xuemin Lin, David W Cheung, and Jiawei Han. 2008. Mining preferences from superior and inferior examples. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 390–398.
- Yoon Kim. 2014. [Convolutional neural networks for sentence classification](#). *CoRR*, abs/1408.5882.
- Klaus Krippendorff. 2011. Computing krippendorff’s alpha-reliability.
- Kalpesh Krishna and Mohit Iyyer. 2019. Generating question-answer hierarchies. *arXiv preprint arXiv:1906.02622*.
- Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2021. Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209*.
- Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. Aquamuse: Automatically generating datasets for query-based multi-document summarization. *arXiv preprint arXiv:2010.12694*.
- Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2021. Comsum and sibert: A dataset and neural model for query-based multi-document summarization. In *International Conference on Document Analysis and Recognition*, pages 84–98. Springer.
- Oren Kurland and Lillian Lee. 2010. Pagerank without hyperlinks: Structural reranking using links induced by language models. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38.
- Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 889–898.
- Timothy J. Lawson, James H. Bodle, Melissa A. Houlette, and Richard R. Haubner. 2006. [Guiding questions enhance student learning from educational videos](#). *Teaching of Psychology*, 33(1):31–33.
- Timothy J Lawson, James H Bodle, and Tracy A McDonough. 2007. Techniques for increasing student learning from educational videos: Notes versus guiding questions. *Teaching of Psychology*, 34(2):90–93.
- Adam D Lelkes, Vinh Q Tran, and Cong Yu. 2021. Quiz-style question generation for news stories. In *Proceedings of the Web Conference 2021*, pages 2501–2511.
- Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. [Keep meeting summaries on topic: Abstractive multi-modal meeting summarization](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196, Florence, Italy. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114.
- Marina Litvak and Natalia Vanetik. 2017. [Query-based summarization using MDL principle](#). In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 22–31, Valencia, Spain. Association for Computational Linguistics.
- Cong Liu, Chi Yuan, Alex M Butler, Richard D Carvajal, Ziran Ryan Li, Casey N Ta, and Chunhua Weng. 2019. Dquest: dynamic questionnaire for search of clinical trials. *Journal of the American Medical Informatics Association*, 26(11):1333–1343.

- Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.
- Potsawee Manakul and Mark Gales. 2021. Long-span summarization via local attention and content selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6026–6041.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Karen Mazidi and Rodney D. Nielsen. 2014. [Linguistic considerations in automatic question generation](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 321–326, Baltimore, Maryland. Association for Computational Linguistics.
- Yashar Mehdad, Giuseppe Carenini, Frank Tompa, and Raymond Ng. 2013. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 136–146.
- Jack Mostow and Wei Chen. 2009. Generating instruction automatically for the reading strategy of self-questioning. In *AIED*, pages 465–472.
- Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2010. Generating and validating abstracts of meeting conversations: a user study. In *Proceedings of the 6th International Natural Language Generation Conference*.
- Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. [Diversity driven attention model for query-based abstractive summarization](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072, Vancouver, Canada. Association for Computational Linguistics.
- Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. [A template-based abstractive meeting summarization: Leveraging summary and source text relationships](#). In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 45–53, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021. Data augmentation for abstractive query-focused multi-document summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13666–13674.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). *CoRR*, abs/1908.10084.
- Lior Rokach and Slava Kisilevich. 2012. Initial profile generation in recommender systems using pairwise comparison. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1854–1859.
- Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge.
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). *CoRR*, abs/1704.04368.
- Imran Sehikh, Dominique Fohr, and Irina Illina. 2017. Topic segmentation in asr transcripts using bidirectional rnns for change detection. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 512–518. IEEE.
- Anna Sepiarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference elicitation as an optimization problem. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 172–180.

- Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. 2018. [Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 664–674, Melbourne, Australia. Association for Computational Linguistics.
- Daksha Singhal, Kavya Khatte, A Tejaswini, and R Jayashree. 2020. Abstractive summarization of meeting conversations. In *2020 IEEE International Conference for Innovation in Technology (INOCON)*, pages 1–4. IEEE.
- Alessandro Solbiati, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. 2021. Unsupervised topic segmentation of meetings with bert embeddings. *arXiv preprint arXiv:2106.12978*.
- Maartje ter Hoeve, Julia Kiseleva, and Maarten de Rijke. 2020. What makes a good summary? reconsidering the focus of automatic summarization.
- Maartje ter Hoeve, Julia Kiseleva, and Maarten de Rijke. 2022. Summarization with graphical elements. *arXiv preprint arXiv:2204.07551*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.
- Hanna M Wallach. 2004. Conditional random fields: An introduction. *Technical Reports (CIS)*, page 22.
- Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. [Asking and answering questions to evaluate the factual consistency of summaries](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.
- Lu Wang and Claire Cardie. 2013. [Domain-independent abstract generation for focused meeting summarization](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405, Sofia, Bulgaria. Association for Computational Linguistics.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Leilan Zhang and Qiang Zhou. 2019. Topic segmentation for dialogue stream. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1036–1043. IEEE.
- Zhou Zhao, Haojie Pan, Changjie Fan, Yan Liu, Linyin Li, Min Yang, and Deng Cai. 2019. Abstractive meeting summarization via hierarchical adaptive segmental network learning. In *The World Wide Web Conference*, pages 3455–3461.
- Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. [QMSum: A new benchmark for query-based multi-domain meeting summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.
- Chenguang Zhu, Ruo Chen Xu, Michael Zeng, and Xuedong Huang. 2020. [A hierarchical network for abstractive meeting summarization with cross-domain pretraining](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.

Sentence Identification with BOS and EOS Label Combinations

Takuma Udagawa, Hiroshi Kanayama, Issei Yoshida

IBM Research - Tokyo, Japan

Takuma.Udagawa@ibm.com, {hkana, issei}@jp.ibm.com

Abstract

The sentence is a fundamental unit in many NLP applications. Sentence segmentation is widely used as the first preprocessing task, where an input text is split into consecutive sentences considering the end of the sentence (EOS) as their boundaries. This task formulation relies on a strong assumption that the input text consists only of sentences, or what we call the sentential units (SUs). However, real-world texts often contain non-sentential units (NSUs) such as metadata, sentence fragments, non-linguistic markers, etc. which are unreasonable or undesirable to be treated as a part of an SU. To tackle this issue, we formulate a novel task of sentence identification, where the goal is to identify SUs while excluding NSUs in a given text. To conduct sentence identification, we propose a simple yet effective method which combines the beginning of the sentence (BOS) and EOS labels to determine the most probable SUs and NSUs based on dynamic programming. To evaluate this task, we design an automatic, language-independent procedure to convert the Universal Dependencies corpora into sentence identification benchmarks. Finally, our experiments on the sentence identification task demonstrate that our proposed method generally outperforms sentence segmentation baselines which only utilize EOS labels.

1 Introduction

The sentence, which we refer to as the sentential unit (SU), is a fundamental unit of processing in many NLP applications including syntactic parsing (Dozat and Manning, 2017), semantic parsing (Dozat and Manning, 2018), and machine translation (Liu et al., 2020). Existing works mostly rely on *sentence segmentation* (a.k.a. *sentence boundary detection*) as the first preprocessing task, where we predict the end of the sentence (EOS) to split a text into consecutive SUs (Kiss and Strunk, 2006; Gillick, 2009). This approach relies on a strong

assumption that the text only consists of SUs; however, real-world texts like web contents often contain non-sentential units (NSUs) such as the metadata of attachments embedded in the email body, repetition of symbols for separating texts, irregular series of nouns, etc. (just to name a few). Such NSUs may cause detrimental or unexpected results in the downstream tasks if considered as parts of the SUs and are more desirable to be distinguished from SUs in the first preprocessing step.

To tackle this problem, we formulate a novel task of *sentence identification*, where the goal is to identify SUs while excluding NSUs in a given text (§3). This can be regarded as an SU span extraction task, where each SU span is represented by the beginning of the sentence (BOS) and the EOS labels.¹ We illustrate the difference between sentence segmentation and sentence identification in Table 1. In sentence segmentation, the text fragment of an embedded file (“- TEXT.htm << File: TEXT.htm >>”) needs to be considered as a part of an SU. In contrast, sentence identification can regard it as an NSU and exclude it for downstream applications such as dependency parsing.

To conduct sentence identification, we propose a simple method which effectively combines the BOS and EOS probabilities to determine both SUs and NSUs (§4). To be specific, we first train the BOS and EOS labeling models based on either the sentence identification dataset (with SUs and NSUs) or sentence segmentation dataset (only SUs). Then, we search for the most probable spans of SUs and NSUs using a simple dynamic programming framework. Theoretically, our method can be considered as a natural generalization of existing sentence segmentation algorithms.

To evaluate this task, we design an automatic pro-

¹For simplicity, we assume that the input text can be segmented into consecutive, non-overlapping units of SUs and NSUs. This way, we can also represent and evaluate SU extraction as an equivalent BIO labeling task (§5-§7).

Input Text (from EWT)	Thank you. - TEXT.htm << File: TEXT.htm >> I was thinking of converting it to a hover vehicle. I might just sell the car and get you to drive me around all winter.	
Sentence Segmentation	E	
		Thank you. - TEXT.htm << File: TEXT.htm >> I was thinking of converting it to a hover vehicle. I might just sell the car and get you to drive me around all winter.
Sentence Identification	B E	
		Thank you. - TEXT.htm << File: TEXT.htm >> I was thinking of converting it to a hover vehicle. I might just sell the car and get you to drive me around all winter.

Table 1: Illustration of sentence segmentation and sentence identification. In sentence segmentation, EOS labels (E) are used to segment the input text into consecutive SUs (in blue). In sentence identification, only the spans bracketed by the BOS (B) and EOS labels are extracted as SUs, while the rest can be excluded as NSUs.

cedure to convert the Universal Dependencies (UD) corpora (de Marneffe et al., 2021) into sentence identification benchmarks (§5). To be specific, (i) we use the original sentence boundaries in UD as the unit (SU and NSU) boundaries and (ii) classify each unit as an SU iff it contains at least one clausal predicate with a core/non-core argument. Importantly, our classification rule follows the definition of *lexical sentence* in linguistics (Nunberg, 1990), is easily customizable with language-independent rules, and makes reasonable classification within the scope of our experiments.

To conduct our experiments, we focus on the English Web Treebank (Silveira et al., 2014) as the primary benchmark for sentence identification and train the BOS/EOS labeling models by finetuning RoBERTa (Liu et al., 2019) (§6). We also propose techniques to develop these models using a standard sentence segmentation dataset, i.e. the Wall Street Journal corpus (Marcus et al., 1993), which only contains clean, edited SUs without any NSUs.

Based on our experimental results, we demonstrate that our proposed method generally outperforms sentence segmentation baselines which only utilize EOS labels (§7). These results highlight the importance of combining the BOS labels in addition to the EOS labels for accurate sentence identification under various conditions.

2 Background

Sentence segmentation, a.k.a. sentence boundary detection, is the task of segmenting an input text into the unit of sentences. Despite the long history of study (Riley, 1989) and its importance in the entire NLP pipeline (Walker et al., 2001), this area has received relatively little attention. For one reason, the task has been recognized as “long

solved” (Read et al., 2012) with the most recent approach reporting 99.8% F1 score on the standard English Wall Street Journal (WSJ) dataset (Wicks and Post, 2021). Their state-of-the-art method ER-SATZ combines (i) a regular-expression based detector of candidate sentence boundaries, followed by (ii) a Transformer-based (Vaswani et al., 2017) binary classifier which predicts whether the candidate boundary is EOS based on the local context, i.e. surrounding few words. This modern context-based approach has been shown to outperform competitive, widely used baselines such as SPLITTA (Gillick, 2009), PUNKT (Kiss and Strunk, 2006), and MOSES (Koehn et al., 2007).

However, two important aspects are not fully addressed in the current literature. First is the coverage of *diverse domains, genres, and writing styles*. Existing works (including Wicks and Post, 2021) focus on formal/edited text and assume the existence of sentence ending punctuations (e.g. full stops) at the sentence boundaries. However, social media texts often lack such punctuations and contain various types of non-linguistic noise, which can lead to a substantial degradation in the segmentation performance (Read et al., 2012; Rudrapal et al., 2015). Speech transcription texts also usually contain disfluent, ungrammatical, or fragmented structures and lack both punctuations and casing (Wang et al., 2019; Rehbein et al., 2020). Considering the amount of such informal or non-standard texts in the real world, it is compelling to expand the capability of sentence segmentation beyond formal, standardized text.

The second aspect is the coverage of *multiple languages*. Different languages involve different complexities in sentence segmentation, e.g. Chinese requires the disambiguation of commas as the

sentence ending punctuation (Xue and Yang, 2011) and Thai does not mark EOS with any type of punctuations (Aroonmanakun et al., 2007; Zhou et al., 2016). To advance NLP from a multilingual perspective, it is crucial to develop and evaluate models in multiple languages: Wicks and Post (2021) make an important step in this direction, proposing a language-agnostic, unified sentence segmentation model covering a total of 87 languages.

Based on these observations, we first propose to extend the task of sentence segmentation to *sentence identification*, which expands the capability of sentence segmentation beyond formal, standardized text (§3, §4). Secondly, we propose a cross-lingual method of benchmarking sentence identification based on the UD corpora, considering every word or character as the candidate boundary to cover diverse domains, genres, and languages that lack sentence ending punctuations (§5). Finally, we follow Wicks and Post (2021) to develop modern neural-based models that require no language-specific engineering and can be developed for different languages in a unified manner (§6).

3 Task Formulation

3.1 Sentence Segmentation Task

First, we introduce a precise (re-)formulation of the sentence segmentation task. Let $\mathbf{W} = (w_0, w_1, \dots, w_{N-1})$ represent the input text, where each w_i denotes a word (but can also be a subword or character). We also define the text span $\mathbf{W}[i:j] = (w_i, \dots, w_{j-1})$, their concatenation $\mathbf{W}[i:j] \oplus \mathbf{W}[j:k] = \mathbf{W}[i:k]$, and SU boundary indices $\mathbf{B} = (b_0, b_1, \dots, b_M)$ where $b_0 = 0$, $b_M = N$, and $\bigoplus_{i=1}^M \mathbf{W}[b_{i-1}:b_i] = \mathbf{W}$ (i.e. the concatenation of all SUs recovers the input text).

Next, we introduce the SU probability $p_{\text{SU}}(\mathbf{W}[i:j])$ which corresponds to the probability of the text span $\mathbf{W}[i:j]$ being an SU. Based on this probability, the task of sentence segmentation can be formalized as searching for the boundaries \mathbf{B} which maximize the following probability:²

$$\arg \max_{\mathbf{B}} \prod_{i=1}^M p_{\text{SU}}(\mathbf{W}[b_{i-1}:b_i]) \quad (1)$$

The most standard approach is to define $p_{\text{SU}}(\mathbf{W}[i:j])$ based on a pretrained EOS labeling model, as we describe in §4.1. However, our (re-)formulation

² M is a variable and need not be fixed during the search.

as Eq. (1) is more general and permits other definitions of SU probability as well.

3.2 Sentence Identification Task

In sentence identification, we consider the input text \mathbf{W} can be segmented into consecutive, non-overlapping units of SUs and NSUs. Hence, we regard $\mathbf{B} = (b_0, b_1, \dots, b_M)$ as the unit (SU and NSU) boundaries and define the unit indicators $\mathbf{A} = (a_1, a_2, \dots, a_M)$ for each unit as follows:

$$a_i = \begin{cases} 1 & \text{if } \mathbf{W}[b_{i-1}:b_i] \text{ is an SU} \\ 0 & \text{if } \mathbf{W}[b_{i-1}:b_i] \text{ is an NSU} \end{cases}$$

Next, we introduce the NSU probability $p_{\text{NSU}}(\mathbf{W}[i:j])$ which corresponds to the probability of the text span $\mathbf{W}[i:j]$ being an NSU. Based on p_{SU} and p_{NSU} , we can formalize the task of sentence identification as searching for the unit boundaries \mathbf{B} and unit indicators \mathbf{A} which maximize the following probability:

$$\arg \max_{\mathbf{B}, \mathbf{A}} \prod_{i=1}^M p_{\text{SU}}(\mathbf{W}[b_{i-1}:b_i])^{a_i} p_{\text{NSU}}(\mathbf{W}[b_{i-1}:b_i])^{1-a_i} \quad (2)$$

Note that this strictly generalizes the sentence segmentation task in Eq. (1), which is a special case where $a_i = 1, \forall a_i \in \mathbf{A}$. Based on this task formulation, we discuss how we can define $p_{\text{SU}}(\mathbf{W}[i:j])$ and $p_{\text{NSU}}(\mathbf{W}[i:j])$ to derive our sentence identification algorithm in §4.2.

4 Methods

4.1 Sentence Segmentation Method

In the most standard approach, sentence segmentation employs an EOS labeling model p_{EOS} to define the SU probability p_{SU} in Eq. (1). To be specific, let $p_{\text{EOS}}(w_i|\mathbf{W};\theta)$ denote the EOS labeling model, which computes the probability of w_i being EOS in \mathbf{W} (θ denotes the model parameters). Typically, it is straightforward to train this model in a *supervised learning* setup using a dataset annotated with gold EOS boundaries (Wicks and Post, 2021). For brevity, we use the notation $p_{\text{EOS}}(w_i)$ as a shorthand for $p_{\text{EOS}}(w_i|\mathbf{W};\theta)$, i.e. we omit \mathbf{W} and θ (unless required) in the rest of this paper.

Based on the pretrained model p_{EOS} , we can define the SU probability as $p_{\text{SU}}(\mathbf{W}[i:j]) = p_{\text{EOS}}(w_{j-1}) \prod_{i \leq k < j-1} (1 - p_{\text{EOS}}(w_k))$, which requires the last word w_{j-1} to be EOS and all other

words to be non-EOS. By substituting this definition, we can decompose Eq. (1) as follows:

$$\begin{aligned}
(1) &= \arg \max_{\mathbf{B}} \sum_{i=1}^M \log p_{\text{SU}}(\mathbf{W}[b_{i-1}:b_i]) \\
&= \arg \max_{\mathbf{B}} \sum_{i=1}^M \left\{ \log p_{\text{EOS}}(w_{b_{i-1}}) + \sum_{b_{i-1} \leq j < b_i} \log(1 - p_{\text{EOS}}(w_j)) \right\} \\
&= \arg \max_{\mathbf{B}} \sum_{i \in \mathbf{B}_{\text{EOS}}} \log p_{\text{EOS}}(w_i) + \sum_{i \notin \mathbf{B}_{\text{EOS}}} \log(1 - p_{\text{EOS}}(w_i))
\end{aligned} \tag{3}$$

where $\mathbf{B}_{\text{EOS}} = \{b_i - 1 \mid i \in (1, 2, \dots, M)\}$ represents all the EOS indices defined by \mathbf{B} .

This is a trivial optimization problem where we can simply choose $\mathbf{B}_{\text{EOS}} = \{i \in (0, 1, \dots, N - 1) \mid p_{\text{EOS}}(w_i) \geq 0.5\}$ to maximize Eq. (3). This also shows that sentence segmentation can be conducted by predicting the EOS independently for each w_i based on $p_{\text{EOS}}(w_i)$. In contrast, sentence identification involves a more complex optimization problem which we solve using dynamic programming (§4.2).

4.2 Sentence Identification Method

We extend the method of sentence segmentation (§4.1) to conduct sentence identification. To be specific, we employ pretrained BOS and EOS labeling models $p_{\text{BOS}}, p_{\text{EOS}}$ to define the SU and NSU probabilities $p_{\text{SU}}, p_{\text{NSU}}$ in Eq. (2). As a first step, we need to train the BOS and EOS labeling models: this can be conducted in a supervised manner using a dataset containing gold BOS and EOS labels, as we explain in §6.1.

Based on the pretrained BOS and EOS labeling models, we can define the SU and NSU probabilities as follows:

$$\begin{aligned}
p_{\text{SU}}(\mathbf{W}[i:j]) &= p_{\text{BOS}}(w_i) \prod_{i < k \leq j-1} (1 - p_{\text{BOS}}(w_k)) \\
&\quad \times p_{\text{EOS}}(w_{j-1}) \prod_{i \leq k < j-1} (1 - p_{\text{EOS}}(w_k)) \\
p_{\text{NSU}}(\mathbf{W}[i:j]) &= \prod_{i \leq k \leq j-1} (1 - p_{\text{BOS}}(w_k)) \times \prod_{i \leq k \leq j-1} (1 - p_{\text{EOS}}(w_k))
\end{aligned}$$

In the SU probability p_{SU} , the first word w_i is required to be BOS, the last word w_{j-1} to be EOS, and all other words to be neither BOS nor EOS. Note that this definition of p_{SU} is a natural generalization from §4.1 which only relies on the EOS probability p_{EOS} .

In contrast, the NSU probability p_{NSU} requires all words to be neither BOS nor EOS. Notably, this definition does not distinguish contiguous NSUs in the sense that $p_{\text{NSU}}(\mathbf{W}[i:k]) = p_{\text{NSU}}(\mathbf{W}[i:j]) \times p_{\text{NSU}}(\mathbf{W}[j:k])$ if $\mathbf{W}[i:j] \oplus \mathbf{W}[j:k] = \mathbf{W}[i:k]$.

This is convenient as we are only interested in the extraction of SUs and do not need to seek the exact boundaries between consecutive NSUs.

By substituting these definitions of p_{SU} and p_{NSU} , we can decompose Eq. (2) as follows:

$$\begin{aligned}
(2) &= \arg \max_{\mathbf{B}, \mathbf{A}} \sum_{i=1}^M \left\{ a_i \log p_{\text{SU}}(\mathbf{W}[b_{i-1}:b_i]) \right. \\
&\quad \left. + (1 - a_i) \log p_{\text{NSU}}(\mathbf{W}[b_{i-1}:b_i]) \right\} \\
&= \arg \max_{\mathbf{B}, \mathbf{A}} \sum_{i \in \mathbf{B}_{\text{BOS}}^{\mathbf{A}}} \log p_{\text{BOS}}(w_i) + \sum_{i \notin \mathbf{B}_{\text{BOS}}^{\mathbf{A}}} \log(1 - p_{\text{BOS}}(w_i)) \\
&\quad + \sum_{i \in \mathbf{B}_{\text{EOS}}^{\mathbf{A}}} \log p_{\text{EOS}}(w_i) + \sum_{i \notin \mathbf{B}_{\text{EOS}}^{\mathbf{A}}} \log(1 - p_{\text{EOS}}(w_i))
\end{aligned} \tag{4}$$

where $\mathbf{B}_{\text{BOS}}^{\mathbf{A}} = \{b_{i-1} \mid i \in (1, 2, \dots, M), a_i = 1\}$ denotes the BOS indices and $\mathbf{B}_{\text{EOS}}^{\mathbf{A}} = \{b_i - 1 \mid i \in (1, 2, \dots, M), a_i = 1\}$ denotes the EOS indices, both defined by \mathbf{B} and \mathbf{A} .

Therefore, our goal is to choose $\mathbf{B}_{\text{BOS}}^{\mathbf{A}}$ and $\mathbf{B}_{\text{EOS}}^{\mathbf{A}}$ which maximize Eq. (4). To this end, we need to consider the restrictions that (i) the first label should be BOS, (ii) the last label should be EOS, and (iii) BOS and EOS labels need to appear alternately. These restrictions can be incorporated in our dynamic programming framework to find the argmax of Eq. (4). For the precise algorithm, we refer the readers to Appendix A.

5 Evaluation

Due to the novelty of the task, currently there exists no benchmark for evaluating sentence identification. To address this issue, we propose a fully automatic procedure to convert the Universal Dependencies (UD) corpora (de Marneffe et al., 2021) into sentence identification benchmarks.

Concretely speaking, we conduct the following two steps based on the gold UD annotation: (i) the detection of unit (SU and NSU) boundaries and (ii) the classification of each unit into SU or NSU. As for (i), we simply use the original *sentence boundaries* in the UD annotation, where UD uses the term *sentence* in a broader sense including both SUs and NSUs (e.g. sentence fragments). Note that the exact boundaries between consecutive NSUs (which we call NSU–NSU boundaries) do not need to be accurate or consistent, since we are only interested in extracting the spans of SUs. However, we do expect that the original boundaries are generally reliable in all other cases (SU–SU and SU–NSU boundaries), which seems to be the case.

The main problem is (ii), i.e. how to classify

by adding a binary BOS/EOS classifier on top of the encoder.

To enable our models to handle various lengths of the input texts, we concatenate the consecutive L units of gold SUs and NSUs as the input during training, where L is sampled from a geometric distribution with parameter p_{CC} .⁵ However, the RoBERTa encoder has the restriction that the input text size cannot exceed 512 subwords. Therefore, if the input text size is too large, we replace L with the maximum $L' < L$ which satisfies this restriction. Note that this is a common procedure to sample variable (instead of fixed) lengths of concatenated units (Joshi et al., 2020).

Assuming the existence of the in-domain sentence identification dataset (EWT Train/Dev), it is straightforward to train the BOS/EOS labeling models based on our unit concatenation procedure. However, we may not always have the gold annotation of SUs and NSUs for the target domain. To take such cases into account, we also consider a setup where we only have the standard sentence segmentation dataset (WSJ Train/Dev) to train the BOS/EOS labeling models.

When using the sentence segmentation dataset (WSJ), we need to apply the unit concatenation procedure using only clean, edited SUs. Unfortunately, this can yield the following data priors which do not actually hold in a sentence identification dataset (EWT): (i) an SU (almost) always starts with a capitalization and ends with punctuation, (ii) the first word of the input is always BOS and the last word is always EOS, and (iii) BOS always directly follows EOS.

To address (i) and (ii), we propose a simple data augmentation technique to alleviate the discrepancy in the data priors. To address (iii), we propose an ensembling technique with the unidirectional (instead of bidirectional) models which are agnostic to this data prior.

6.1.1 Data Augmentation (+AUG)

To address (i), we conduct a unit-level data augmentation, i.e. we modify each unit based on the following rules with a small probability p_{DA} :

- Convert all words in the unit to lower-case, upper-case, or title-case (e.g. “hello world”,

⁵With parameter $p_{CC} \in (0, 1]$, the probability mass function of the geometric distribution is $p(L = l) = (1 - p_{CC})^{l-1} p_{CC}$ where $l \in \{1, 2, 3, \dots\}$. As p_{CC} decreases, the distribution gets more skewed towards larger L . With $p_{CC} = 0$, we consider $p(L = \infty) = 1$.

Orig.	B	E B
	Joe went to school.	After that he ...
(i) Unit	B	E B
Aug.	Joe went to school	AFTER THAT HE ...
(ii) Unit	B	E B
Trunc.	Joe went to school	AFTER THAT HE ...

Table 4: Illustration of our data augmentation technique. In (i) *unit-level augmentation*, we randomly change the casing or remove the last punctuations of each unit. In (ii) *unit truncation*, we randomly truncate the first and last units of the input (and regard them as NSUs).

“HELLO WORLD”, or “Hello World”).

- Remove sentence ending punctuations based on a regular-expression matcher (following ERSATZ, Wicks and Post, 2021).

After the unit-level augmentation, we can apply the unit concatenation in the exact same manner.

Finally, to address (ii), we randomly apply a unit truncation to the first and last units of the concatenated input. To be specific, we choose a random word in the first (last) unit and remove all words prior (posterior) to it with a small probability p_{TR} . If the truncation is conducted, we regard the unit as an NSU and fix the gold BOS/EOS labels accordingly. See Table 4 for an illustration.

Based on this procedure, we can expect to alleviate the data priors (i) and (ii). For more details, we refer the readers to Appendix D.

6.1.2 Unidirectional Model (+UNI)

Simply concatenating SUs (without NSUs) yields the data prior (iii), i.e. BOS always directly follows EOS. This prior can be easily captured by the bidirectional models $p_{\text{BOS}}(w_i | \mathbf{W})$, $p_{\text{EOS}}(w_i | \mathbf{W})$ conditioned on the whole input \mathbf{W} , including our RoBERTa-based models. For instance, as shown in Figure 1, the model may predict EOS at the end of the first unit ($w_2 = \#$) just because the next word ($w_3 = \text{This}$) is likely predicted as BOS.

To alleviate this issue, we propose to combine the predictions of the unidirectional models for BOS and EOS labeling. To be precise, let $\mathbf{W}^{\leq i} = (w_0, \dots, w_i)$ and $\mathbf{W}^{\geq i} = (w_i, \dots, w_{N-1})$. Then, we can represent the unidirectional BOS model as $p_{\text{BOS}}^{\text{uni}}(w_i | \mathbf{W}^{\geq i})$ (looking the context right-to-left) and EOS model as $p_{\text{EOS}}^{\text{uni}}(w_i | \mathbf{W}^{\leq i})$ (looking left-to-right). As illustrated in Figure 1, these models are agnostic to the data prior (iii). In practice, we can simply use different attention masks and share the

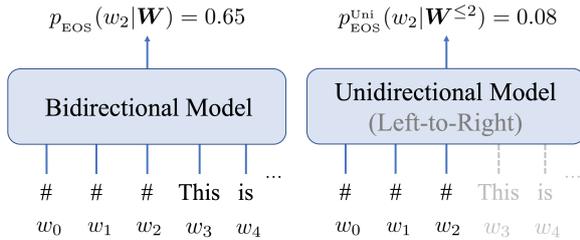


Figure 1: Illustration of the bidirectional EOS model (left) and the unidirectional EOS model (right).

encoder parameters (except the last classifier) for the unidirectional and bidirectional models.

We can utilize these unidirectional models by taking a linear interpolation with the bidirectional models as follows:

$$p_{\text{BOS}}^{+\text{Uni}}(w_i|\mathbf{W}) = \lambda \cdot p_{\text{BOS}}^{\text{Uni}}(w_i|\mathbf{W}^{\geq i}) + (1-\lambda) \cdot p_{\text{BOS}}(w_i|\mathbf{W})$$

$$p_{\text{EOS}}^{+\text{Uni}}(w_i|\mathbf{W}) = \lambda \cdot p_{\text{EOS}}^{\text{Uni}}(w_i|\mathbf{W}^{\leq i}) + (1-\lambda) \cdot p_{\text{EOS}}(w_i|\mathbf{W})$$

Then, we can use $p_{\text{BOS}}^{+\text{Uni}}$ and $p_{\text{EOS}}^{+\text{Uni}}$ in place of p_{BOS} and p_{EOS} (respectively) to conduct sentence identification, as described in §4.2.

Finally, we compare our proposed methods against sentence segmentation baselines which only utilize EOS labels.⁶ As for the baselines, we use the EOS labeling model developed in the same manner to segment the input text based on EOS. Note that we can optionally force the last word in the input to be EOS: in this case, the result will only contain SUs since all segments will end with EOS. By default, we do not force the last EOS: in this case, the segment after the last EOS (if exists) is considered as an NSU.

As a default configuration, we use $p_{CC} = 0.5$, $p_{DA} = 0.3$, $p_{TR} = 0.1$, and $\lambda = 0.5$ in our experiments. To ensure reproducibility, we report more details on the hyperparameters and model setup in Appendix D. For the precise procedure on how we convert between the word-, character-, and subword-level labels (for RoBERTa), we refer the readers to Appendix C.

6.2 Evaluation Setup

In the evaluation phase, we consider three ways of assembling the input texts on which we conduct sentence identification. Firstly, we can apply the same unit concatenation procedure as described in §6.1. To be specific, we use $p_{CC} = 0.5$ (same as the

⁶This EOS-only method is the most reasonable baseline to quantify the precise advantage from combining BOS labels in addition to EOS, which is proposed in our methods.

training phase) and $p_{CC} = 0$ (which concatenates the units up to the maximal length) to simulate both shorter and longer lengths of the input texts.

However, this approach is relatively *synthetic* in the sense that we take the gold unit boundaries for granted. They are usually unavailable at the inference time, so we should consider a more realistic setting for evaluating the methods without relying on the gold unit boundaries.

To this end, we propose to evaluate sentence identification as a *postprocessing* of sentence segmentation. To be specific, we first apply the state-of-the-art method ERSATZ (Wicks and Post, 2021) on the raw text of EWT and then apply sentence identification to each segmented text. Note that ERSATZ has high precision but still predicts false EOS which can fragment a gold SU: in such cases, we consider the fragmented SUs as NSUs and fix the labels accordingly (just as we did in unit truncation, cf. §6.1 and Table 4).

As for the evaluation metrics, we convert the predictions of our methods into word/character-level BIO labels (cf. Appendix C) and compute the F1 score for each label prediction. Then, we summarize the results as the macro average F1 and weighted average F1. We also compute the F1 score of the exact SU span extraction at the word/character-level. Finally, we run each experiment (from model training to testing) five times with different random seeds and report the average and standard deviation as the final results.

7 Results

Table 5 summarizes the word-level evaluation results. The results for the character-level evaluation show similar tendencies, so we put them in Appendix E. The F1 score for each BIO label prediction is also available in Appendix E.

Firstly, we take a look at the results when we have the in-domain sentence identification dataset (EWT Train/Dev) for model development. In this setup, we can verify that our proposed method (BOS&EOS) significantly outperforms the baselines (EOS-Only) in all metrics. For instance, our method achieves consistently high performance of 84~89% F1 for the exact SU span extraction, both at the word- and character-level. This is a very promising result that demonstrates the effectiveness of our method when we can leverage the gold SUs and NSUs from the target domain.

Secondly, we focus on the results where we

Train/Dev Datasets	Model	EWT Test ($p_{CC} = 0.5$)			EWT Test ($p_{CC} = 0$)			EWT Test (Postprocess)		
		BIO Macro	BIO Weighted	Span	BIO Macro	BIO Weighted	Span	BIO Macro	BIO Weighted	Span
EWT Train/Dev	EOS-Only	83.2±1.5	93.9±0.6	72.8±1.8	59.7±0.2	86.4±0.1	58.2±1.1	86.3±2.7	94.6±1.1	81.6±2.4
	EOS-Only (force last)	58.6±0.1	86.6±0.0	60.4±0.8	57.6±0.2	85.9±0.1	57.7±1.0	59.1±0.1	85.7±0.0	62.3±0.3
	BOS&EOS	93.0±1.4	97.3±0.6	87.3±1.6	91.0±1.8	96.4±0.7	84.1±2.6	92.3±1.0	96.7±0.4	88.8±0.9
WSJ Train/Dev	EOS-Only	71.7±0.7	88.9±0.4	59.2±2.4	56.9±0.6	85.2±0.3	48.2±2.5	71.5±0.3	87.8±0.3	67.8±0.3
	EOS-Only (force last)	57.5±0.3	86.2±0.2	53.6±2.1	55.4±0.7	85.0±0.3	48.2±2.5	58.9±0.1	85.7±0.0	61.1±0.2
	EOS-Only (+AUG)	66.4±1.5	88.3±0.4	59.5±1.4	58.3±0.5	86.1±0.3	54.4±2.5	71.1±1.3	88.5±0.6	66.2±1.9
	BOS&EOS	71.5±0.2	89.1±0.2	59.1±1.5	57.7±0.9	85.4±0.2	48.8±1.6	71.0±0.3	87.9±0.2	68.4±0.3
	BOS&EOS (+UNI)	70.4±0.7	88.2±0.3	60.0±1.1	63.3±0.8	86.0±0.4	53.0±1.3	70.8±0.4	87.6±0.2	68.4±0.1
	BOS&EOS (+UNI +AUG)	72.5±0.4	89.5±0.1	66.6±0.2	72.4±1.3	89.1±0.5	63.7±1.0	74.3±1.1	89.6±0.4	71.9±1.4

Table 5: **Overall Results** (Word-Level). We report the macro/weighted average F1 of the BIO labeling task and the F1 score of the exact SU span extraction task. Details of our experimental setup are discussed in §6.

only utilize the standard sentence segmentation dataset (WSJ Train/Dev) for model development. In this setup, we also report the results of applying our data augmentation (+AUG) and unidirectional model (+UNI) techniques from §6.1.⁷

Due to the data discrepancy between WSJ and EWT, we find a natural drop in performance compared to the previous setup using in-domain EWT Train/Dev. However, we can verify that our techniques (+AUG, +UNI) generally help to alleviate this issue, and our proposed method performs on par or slightly better than the EOS-only baselines when applying these techniques. It is especially worth noting the improvement in the exact SU span extraction task (reaching 64~72% F1), where the advantage of our method is the most conspicuous and consistent in both word- and character-level evaluation. This improvement can also be explained by the higher performance in the B-label prediction with our method (Appendix E), which is a prerequisite for accurate SU span extraction.

Finally, we note that the EOS-only baseline without forcing the last EOS can be quite competitive with shorter inputs ($p_{CC} = 0.5$ and postprocessing) but performs considerably worse when the input texts are longer ($p_{CC} = 0$). This is because the baseline can only predict the last segment of the input as an NSU, which is less problematic when the input texts are shorter but becomes increasingly problematic with longer inputs (since most NSUs will not be able to be removed). In contrast, our proposed method performs much more robustly under various input lengths.

Through further experiments and analyses, we

⁷We did not observe any improvement from applying these techniques to the in-domain dataset (EWT Train/Dev), which is consistent with our motivation and expectation.

found that (i) the results are stable across different hyperparameter choices, (ii) predictions are reasonable especially when using the in-domain dataset (EWT Train/Dev) for model development, and (iii) our methods do not sacrifice performance on the formal/edited texts of the sentence segmentation dataset (WSJ Test). These detailed evidences can be found in Appendix F.

8 Conclusion

In this paper, we introduced a novel task of sentence identification, where we aim to identify SUs while excluding NSUs in a given text (§3). Through sentence identification, we can clearly distinguish the portions of the text that are appropriate (or not) for the prediction and evaluation of sophisticated linguistic analyses, such as dependency parsing, semantic role labeling, etc.

To conduct sentence identification, we proposed a simple yet effective method of combining the BOS and EOS labeling models to determine the SUs and NSUs (§4). To evaluate sentence identification, we designed an automatic, language-independent procedure to convert the UD corpora into sentence identification benchmarks (§5).

In our experiments, we developed the BOS/EOS labeling models by finetuning pretrained RoBERTa (§6). Based on the experimental results, we showed that our proposed method combining the BOS and EOS labels outperforms sentence segmentation baselines which only utilize EOS labels in all of the considered settings (§7). Overall, we expect sentence identification to be a fundamental framework for the preprocessing of noisy, informal, or non-standard texts in the real world.

Limitations

Firstly, our current experiments are limited to English and cover only five domains of web media texts in EWT. However, our task formulation (§3), method (§4), and evaluation framework (§5) are fully agnostic to the language and domain. Hence it is straightforward to conduct experiments in different languages or domains (as long as they are supported in the UD). While we expect similar results with different languages/domains, we leave further investigation as a future work.

Secondly, while our method performs reliably when the in-domain dataset is available, there is still a huge room left for improvement without relying on such resources (e.g. only using the standard sentence segmentation dataset). To make our method fully practical, we still need to improve on the accuracy and robustness in such cross-domain scenarios. One potential approach is to refine the definitions of SU and NSU probabilities from §4.2 to make sentence identification more robust. For instance, we can incorporate span-level scores instead of only using word-level BOS/EOS probabilities to define the SU/NSU probabilities. We leave further improvement and extension of our approach as an important future work.

Finally, our methods are currently evaluated on the (exact) SU span extraction task. Ideally, we should also evaluate the methods on downstream applications such as POS tagging, syntactic parsing, semantic role labeling, etc. However, we still expect that the (exact) SU span extraction will play a primary role in the evaluation, since accurate (say human-level) identification of SUs/NSUs will likely provide unprecedented benefits on a wide variety of NLP applications dealing with real-world texts. While we leave the precise analyses on downstream applications as future work, our contributions make the first foundational step towards expanding the capability of the long-established sentence segmentation task.

References

- Wirote Aroonmanakun et al. 2007. Thoughts on word and sentence segmentation in thai. In *Proceedings of the Seventh Symposium on Natural language Processing*, pages 85–90.
- Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. **Universal Dependencies**. *Computational Linguistics*, 47(2):255–308.
- Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proc. of ICLR*.
- Timothy Dozat and Christopher D. Manning. 2018. **Simpler but more accurate semantic dependency parsing**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.
- Dan Gillick. 2009. **Sentence boundary detection and the problem with the U.S.** In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244, Boulder, Colorado. Association for Computational Linguistics.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. **SpanBERT: Improving pre-training by representing and predicting spans**. *Transactions of the Association for Computational Linguistics*, 8:64–77.
- Tibor Kiss and Jan Strunk. 2006. **Unsupervised multilingual sentence boundary detection**. *Computational Linguistics*, 32(4):485–525.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. **Moses: Open source toolkit for statistical machine translation**. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. **Multilingual denoising pre-training for neural machine translation**. *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. **RoBERTa: A robustly optimized BERT pretraining approach**. *arXiv preprint arXiv:1907.11692*.

- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. [Building a large annotated corpus of English: The Penn Treebank](#). *Computational Linguistics*, 19(2):313–330.
- Geoffrey Nunberg. 1990. *The linguistics of punctuation*. 18. Center for the Study of Language (CSLI).
- Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. 2012. [Sentence boundary detection: A long solved problem?](#) In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India. The COLING 2012 Organizing Committee.
- Ines Rehbein, Josef Ruppenhofer, and Thomas Schmidt. 2020. [Improving sentence boundary detection for spoken language transcripts](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7102–7111, Marseille, France. European Language Resources Association.
- Michael D. Riley. 1989. [Some applications of tree-based modelling to speech and language](#). In *Speech and Natural Language: Proceedings of a Workshop Held at Cape Cod, Massachusetts, October 15-18, 1989*.
- Dwijen Rudrapal, Anupam Jamatia, Kunal Chakma, Amitava Das, and Björn Gambäck. 2015. [Sentence boundary detection for social media text](#). In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 254–260, Trivandrum, India. NLP Association of India.
- Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. [A gold standard dependency corpus for English](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NeurIPS*.
- Daniel J. Walker, David E. Clements, Maki Darwin, and Jan W. Amtrup. 2001. [Sentence boundary detection: a comparison of paradigms for improving MT quality](#). In *Proceedings of Machine Translation Summit VIII*, Santiago de Compostela, Spain.
- Xiaolin Wang, Masao Utiyama, and Eiichiro Sumita. 2019. [Online sentence segmentation for simultaneous interpretation using multi-shifted recurrent neural network](#). In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 1–11, Dublin, Ireland. European Association for Machine Translation.
- Rachel Wicks and Matt Post. 2021. [A unified approach to sentence segmentation of punctuated text in many languages](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3995–4007, Online. Association for Computational Linguistics.
- Nianwen Xue and Yaqin Yang. 2011. [Chinese sentence segmentation as comma classification](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 631–635, Portland, Oregon, USA. Association for Computational Linguistics.
- Nina Zhou, AiTi Aw, Nattadaporn Lertcheva, and Xuancong Wang. 2016. [A word labeling approach to Thai sentence boundary detection and POS tagging](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 319–327, Osaka, Japan. The COLING 2016 Organizing Committee.

A Dynamic Programming Algorithm

To find the maximum value (and the argmax) of Eq. 4 from §4.2, we rely on a simple dynamic programming framework. To be specific, we consider the partial labeling of BOS and EOS up to $\mathbf{W}^{\leq k} = (w_0, \dots, w_k)$, where $k \leq N - 1$. Then, we aim to compute the maximum log probability of Eq. 4 based on the partial labeling, i.e. using $\mathbf{W}^{\leq k}$ in place of \mathbf{W} .

Since the labeling is partial, $\mathbf{W}^{\leq k}$ may end inside the SU (i.e. the last label is BOS) or outside the SU (i.e. the last label is EOS). Let $\log p_{\text{IS}}(k+1)$ denote the maximum log probability when $\mathbf{W}^{\leq k}$ ends inside the SU and $\log p_{\text{OS}}(k+1)$ the maximum log probability when $\mathbf{W}^{\leq k}$ ends outside the SU. Then, we can initialize $\log p_{\text{IS}}(0) = \log 0 = -\infty$, $\log p_{\text{OS}}(0) = \log 1 = 0$ (since we always start outside the SU) and iteratively update the two values as follows:

$$\begin{aligned} \log p'_{\text{IS}}(i) &= \max \{ \log p_{\text{IS}}(i) + \log(1 - p_{\text{BOS}}(w_i)), \\ &\quad \log p_{\text{OS}}(i) + \log p_{\text{BOS}}(w_i) \} \\ \log p'_{\text{OS}}(i) &= \log p_{\text{OS}}(i) + \log(1 - p_{\text{BOS}}(w_i)) \\ \log p_{\text{IS}}(i+1) &= \log p'_{\text{IS}}(i) + \log(1 - p_{\text{EOS}}(w_i)) \\ \log p_{\text{OS}}(i+1) &= \max \{ \log p'_{\text{IS}}(i) + \log p_{\text{EOS}}(w_i), \\ &\quad \log p'_{\text{OS}}(i) + \log(1 - p_{\text{EOS}}(w_i)) \} \end{aligned} \quad (5)$$

Note that we first update $p_{\text{IS}}(i) \rightarrow p'_{\text{IS}}(i)$ and $p_{\text{OS}}(i) \rightarrow p'_{\text{OS}}(i)$ based on the BOS probability $p_{\text{BOS}}(w_i)$. Then, we update $p'_{\text{IS}}(i) \rightarrow p_{\text{IS}}(i+1)$ and $p'_{\text{OS}}(i) \rightarrow p_{\text{OS}}(i+1)$ based on the EOS probability $p_{\text{EOS}}(w_i)$.⁸ The iterative procedure is illustrated in Figure 2.

Finally, we can compute the log probability $\log p_{\text{OS}}(N)$ (since we always end outside the SU), which corresponds to the maximum value of Eq. 4. To obtain the argmax, we can simply incorporate backtracking during the iterative updates of Eq. 5. Through this dynamic programming framework, we can ensure that the restrictions from §4.2 are satisfied: namely, (i) the first label should be BOS, (ii) the last label should be EOS, and (iii) BOS and EOS labels need to appear alternately.

In practice, we can limit the candidates of BOS indices to the subset where $p_{\text{BOS}}(w_i)$ is higher than a certain threshold c . This can be efficiently implemented by simply skipping the updates of $p'_{\text{IS}}(i)$ and $p'_{\text{OS}}(i)$, i.e. using $p'_{\text{IS}}(i) = p_{\text{IS}}(i)$ and $p'_{\text{OS}}(i) = p_{\text{OS}}(i)$, if $p_{\text{BOS}}(w_i) < c$.⁹ Likewise, we

⁸Note that if a single word w_i is labeled as both BOS and EOS at the same time, we can extract it as a single SU.

⁹This is equivalent to forcing w_i to be non-BOS, i.e. setting

can limit the candidates of EOS indices by skipping the updates of $p_{\text{IS}}(i+1)$ and $p_{\text{OS}}(i+1)$ if $p_{\text{EOS}}(w_i) < c$. Generally speaking, this leads to a more efficient algorithm: therefore, we use the candidate threshold of $c = 0.1$ for restricting both BOS and EOS indices throughout our experiments.

B SU and NSU Examples

In Table 6, we provide more examples of SUs and NSUs identified based on our procedure described in §5. As for the SUs, we can verify that EWT contains clean, formal SUs with appropriate capitalization and punctuation. We can also verify that EWT contains various types of *informal* SUs, e.g. that lack capitalization/punctuation, use non-standard casing, end with emoticons, include spelling errors, concatenate consecutive SUs without a space, etc.

C Label Assignment and Conversion

In this section, we explain the precise procedure on how we (i) assign the gold character-level labels, (ii) convert the character-level labels to word/subword-level labels, and (iii) convert the subword-level labels to character/word-level labels. We limit our explanation to BIO labels, since it is straightforward to convert them to the combination of BOS and EOS labels (and vice versa).

Firstly, we can assign the gold character-level labels from the UD annotation by taking the character-level alignment, which determines the exact spans of SUs and NSUs. From the character-level labels, we can assign the word- or subword-level labels based on the following rule:

- If the word (or subword) contains a character with the B-label, assign it the B-label.
- Else if it contains a character with the I-label, assign the I-label.
- Otherwise assign the O-label.

For instance, this procedure is used to create the subword-level labels for training our BOS/EOS labeling models.

To evaluate our methods, we need to convert the subword-level labels produced by our methods into the character-level labels, which can then be converted into the word-level labels (based on the previous procedure). To convert a subword-level label into a sequence of character-level labels, we

$p_{\text{BOS}}(w_i) = 0$ in Eq. 5.

GPU. We use a batch size of 8, accumulate the gradients for 32 batches, and apply the gradient clipping at 1.0 before updating the model weights. As for the optimizer, we use Adam with the initial learning rate of 0.0001 and exponentially decay the learning rate by $\gamma = 0.95$ after each epoch. We check the validation loss every 200 batches and stop the training early if there is no improvement for 5 consecutive evaluations.

E The Full Experimental Results

In this section, we report the full results of our experiments which did not fit in §7. Table 7 shows the word-level F1 scores for each B-, I-, and O-label prediction. Table 8 shows the overall results for the character-level evaluation.

Generally speaking, we can confirm the same results as observed in §7. Firstly, our proposed method significantly outperforms the baselines when we use the EWT Train/Dev dataset for model development. Secondly, our method performs slightly better than (or at least on par with) the baselines when developed on the WSJ Train/Dev dataset. Finally, the baseline without forcing the last EOS is competitive with shorter inputs ($p_{CC} = 0.5$ and postprocessing) but performs considerably worse when the input texts are longer ($p_{CC} = 0$).

F Further Experiments and Analyses

In this section, we provide further experiments and analyses to complement our study. To be specific, we provide discussions on the effect of the choice of hyperparameters (F.1), qualitative analyses based on example model outputs (F.2), and evaluation of sentence identification based on the sentence segmentation dataset (F.3).

F.1 Effect of Hyperparameters

As a default configuration, we used $p_{DA} = 0.3$, $p_{TR} = 0.1$ for the data augmentation (+AUG) and $\lambda = 0.5$ for the unidirectional model ensembling (+UNI). To examine the effect of the choice of these hyperparameters, we conducted further experiments by changing these default hyperparameters. Note that all evaluation results in this subsection are based on BOS&EOS (+UNI +AUG) developed on WSJ Train/Dev.

Firstly, we focus on the data augmentation and report the results of our method trained with different sets of p_{DA} and p_{TR} (with λ fixed at 0.5). Since increasing p_{DA} leads to higher recall (and

lower precision) of SU extraction and increasing p_{TR} leads to higher precision (and lower recall), we used a fixed ratio of $p_{DA} : p_{TR} = 3 : 1$ which seemed to make a good trade-off. As shown in Table 9, the results are generally stable with the different choices of the hyperparameters. However, more data augmentation (with larger values of p_{DA} and p_{TR}) tends to slightly improve the performance, especially for the exact SU span extraction.

Secondly, we focus on the unidirectional model ensembling and report the results of changing the linear interpolation rate $\lambda \in [0, 1]$, where $\lambda = 0$ is equivalent to using only the bidirectional models and $\lambda = 1$ only the unidirectional models. We fix $p_{DA} = 0.3$ and $p_{TR} = 0.1$ and only change λ at the inference time without retraining the unidirectional or bidirectional models. As shown in Figure 3, we found that unidirectional and bidirectional models generally have complementary benefits, and choosing the intermediate value of λ leads to the best performance. The results also indicate that we may be able to obtain further improvement by tuning λ on the validation set, although we simply fixed $\lambda = 0.5$ throughout our experiments.

F.2 Qualitative Analyses

In Table 10 and 11, we show the actual predictions made by our proposed method developed on EWT Train/Dev and WSJ Train/Dev. For the latter, we applied +UNI and +AUG with the default hyperparameters.

In the first example (Table 10), we can verify that both models identify the correct SU span while removing the non-sentential header as the NSU. This is a relatively easy example, since the start of the SU is capitalized and less ambiguous.

In the second example (Table 11), we can observe that our method using in-domain data (EWT Train/Dev) extracts the correct SU span, while our method developed on out-of-domain data (WSJ Train/Dev) incorrectly excludes a part of an SU. This seems to be a relatively difficult example, since the start of the SU is not capitalized and more ambiguous. It is worth noting that such SUs can be reliably extracted when we can leverage the in-domain annotation of gold SUs and NSUs.

F.3 Evaluation on the Sentence Segmentation Dataset

Finally, we report the results of sentence identification on the standard sentence segmentation dataset (WSJ Test).

Train/Dev Datasets	Model	EWT Test ($p_{CC} = 0.5$)			EWT Test ($p_{CC} = 0.0$)			EWT Test (Postprocess)		
		B-Label	I-Label	O-Label	B-Label	I-Label	O-Label	B-Label	I-Label	O-Label
EWT Train/Dev	EOS-Only	85.6±0.8	97.3±0.3	66.6±3.5	78.0±0.6	95.1±0.1	6.0±0.4	90.2±1.2	97.5±0.5	71.3±6.6
	EOS-Only (force last)	79.8±0.2	95.9±0.0	0.0±0.0	77.8±0.6	95.1±0.1	0.0±0.0	81.7±0.2	95.7±0.0	0.0±0.0
	BOS&EOS	94.3±0.6	98.7±0.3	86.1±3.5	93.0±1.0	98.2±0.4	81.7±4.0	94.7±0.3	98.4±0.2	83.9±2.4
WSJ Train/Dev	EOS-Only	78.7±1.3	94.4±0.4	42.1±1.3	71.8±1.9	94.3±0.2	4.5±0.4	83.3±0.2	93.8±0.3	37.3±0.9
	EOS-Only (force last)	76.7±0.9	95.6±0.1	0.0±0.0	71.7±1.9	94.6±0.2	0.0±0.0	81.0±0.4	95.6±0.0	0.0±0.0
	EOS-Only (+AUG)	79.4±1.0	95.4±0.2	24.5±4.1	78.1±1.8	93.3±0.2	1.4±1.1	82.7±1.1	94.9±0.5	35.6±3.2
	BOS&EOS	79.4±0.9	94.8±0.2	40.5±0.6	72.9±1.2	94.3±0.2	5.8±2.0	83.9±0.2	94.1±0.2	35.2±0.9
	BOS&EOS (+UNI)	79.8±0.6	93.9±0.2	37.5±1.5	76.2±1.3	93.3±0.3	20.2±1.2	83.8±0.1	93.8±0.1	34.7±0.9
	BOS&EOS (+UNI +AUG)	83.7±0.1	95.0±0.2	38.7±1.2	83.0±0.6	94.6±0.3	39.7±3.5	85.9±0.6	95.2±0.2	41.9±2.8

Table 7: **BIO Labeling Results** (Word-Level). We report the F1 scores for each B-, I- and O-label prediction.

Train/Dev Datasets	Model	EWT Test ($p_{CC} = 0.5$)			EWT Test ($p_{CC} = 0.0$)			EWT Test (Postprocess)		
		BIO	BIO	Span	BIO	BIO	Span	BIO	BIO	Span
		Macro	Weighted		Macro	Weighted		Macro	Weighted	
EWT Train/Dev	EOS-Only	83.8±1.1	92.7±0.5	72.8±1.8	58.5±0.2	81.5±0.0	58.2±1.1	87.7±2.3	93.9±1.2	81.6±2.4
	EOS-Only (force last)	57.7±0.1	81.0±0.0	60.4±0.8	56.9±0.2	80.9±0.0	57.7±1.0	58.1±0.1	79.9±0.0	62.3±0.3
	BOS&EOS	94.0±1.0	97.2±0.6	87.3±1.6	92.2±1.5	96.3±0.7	84.1±2.6	93.5±0.6	96.6±0.4	88.9±0.8
WSJ Train/Dev	EOS-Only	72.8±0.6	86.9±0.4	59.1±2.3	56.0±0.6	80.9±0.1	48.2±2.5	73.3±0.4	85.6±0.2	67.7±0.4
	EOS-Only (force last)	56.6±0.3	80.9±0.0	53.5±2.0	54.9±0.6	80.7±0.1	48.2±2.5	57.8±0.2	79.9±0.0	61.0±0.3
	EOS-Only (+AUG)	64.3±1.5	83.5±0.6	59.5±1.4	57.4±0.5	81.0±0.2	54.4±2.5	69.2±1.7	84.0±0.8	66.2±1.9
	BOS&EOS	72.7±0.7	87.1±0.2	59.1±1.5	57.8±1.9	81.6±0.7	48.8±1.6	72.4±1.0	85.2±0.5	68.3±0.3
	BOS&EOS (+UNI)	72.4±0.6	86.3±0.3	59.6±1.0	65.3±1.0	83.6±0.5	52.9±1.3	72.8±0.4	85.3±0.2	68.0±0.2
	BOS&EOS (+UNI +AUG)	72.2±1.3	86.1±0.6	66.5±0.3	72.8±1.8	86.5±0.9	63.6±1.0	73.2±1.9	85.7±0.9	71.8±1.5

Table 8: **Overall Results** (Character-Level). We report the macro/weighted average F1 of the BIO labeling task and the F1 score of the exact SU span extraction task.

In Table 12, we summarize the WSJ dataset statistics. Note that WSJ only contains SUs and do not contain any NSUs (O-labels). However, we can still evaluate the performance using the same metrics, i.e. the macro/weighted average F1 of the BIO labeling task and the F1 of the exact SU span extraction task.¹¹

Table 13 summarizes the word-level evaluation results. Since we are evaluating on WSJ Test, the performance is naturally better when the models are trained on WSJ Train/Dev rather than EWT Train/Dev (which is now out-of-domain).

When the models are trained on EWT, we found that the baseline (EOS-Only) forcing the last EOS performs the best. This is natural, since this baseline better reflects the nature of the sentence segmentation dataset where all units are SUs. However, our method (BOS&EOS) is still comparable to this baseline and do not (or minimally) sacrifice performance on such datasets.

When the models are trained on WSJ, we found that our method without +UNI or +AUG performs

the best. This is most likely because we can leverage the knowledge of BOS to predict EOS. When we apply the data augmentation (+AUG) and uni-directional model ensembling (+UNI), we observe a slight decrease in performance compared to our vanilla method. However, the results are still comparable and even outperforms the baselines in some metrics (e.g. the exact SU span extraction task).

Overall, we can conclude that our methods do not sacrifice the performance on the the clean, edited texts of the sentence segmentation dataset.

¹¹Since the O-label does not exist, we report the macro average F1 as the average F1 scores of the B-label and I-label predictions.

Evaluation	Augmentation Rates	EWT Test ($p_{CC} = 0.5$)			EWT Test ($p_{CC} = 0$)			EWT Test (Postprocess)		
		BIO Macro	BIO Weighted	Span	BIO Macro	BIO Weighted	Span	BIO Macro	BIO Weighted	Span
Word-Level	$p_{DA} = 0.15, p_{TR} = 0.05$	71.3 \pm 1.1	89.0 \pm 0.5	65.7 \pm 1.3	71.5 \pm 0.9	88.6 \pm 0.5	62.3 \pm 1.7	73.5 \pm 1.4	89.2 \pm 0.6	71.2 \pm 1.8
	$p_{DA} = 0.3, p_{TR} = 0.1$	72.5 \pm 0.4	89.5 \pm 0.1	66.6 \pm 0.2	72.4 \pm 1.3	89.1 \pm 0.5	63.7 \pm 1.0	74.3 \pm 1.1	89.6 \pm 0.4	71.9 \pm 1.4
	$p_{DA} = 0.45, p_{TR} = 0.15$	73.2\pm1.0	90.0\pm0.1	67.3\pm0.8	73.0\pm0.9	89.5\pm0.6	64.0\pm1.8	75.1\pm1.3	90.0\pm0.4	72.1\pm0.7
Character-Level	$p_{DA} = 0.15, p_{TR} = 0.05$	72.3\pm1.9	86.3\pm1.0	65.4 \pm 1.4	73.6\pm0.7	86.8\pm0.3	62.2 \pm 1.7	73.8\pm1.5	86.1\pm0.8	71.0 \pm 1.8
	$p_{DA} = 0.3, p_{TR} = 0.1$	72.2 \pm 1.3	86.1 \pm 0.6	66.5 \pm 0.3	72.8 \pm 1.8	86.5 \pm 0.9	63.6 \pm 1.0	73.2 \pm 1.9	85.7 \pm 0.9	71.8 \pm 1.5
	$p_{DA} = 0.45, p_{TR} = 0.15$	71.9 \pm 1.1	86.1 \pm 0.3	67.2\pm0.8	72.3 \pm 0.9	86.3 \pm 0.6	64.0\pm1.8	73.6 \pm 1.5	86.0 \pm 0.6	72.1\pm0.7

Table 9: **Effect of Data Augmentation Rates** (Word/Character-Level). We use different data augmentation rates (p_{DA} and p_{TR}) and evaluate BOS&EOS (+UNI +AUG) developed on WSJ Train/Dev. We report the macro/weighted average F1 of the BIO labeling task and the F1 score of the exact SU span extraction task.

Developed on EWT	... 06/04/2001 05:54 PM	B Can you pass this along to Elizabeth to ensure Sanders E is on board as well?
Developed on WSJ	... 06/04/2001 05:54 PM	B Can you pass this along to Elizabeth to ensure Sanders E is on board as well?

Table 10: **Example Outputs (Both Correct)**. We show the predictions made by our proposed method (BOS&EOS) developed on EWT Train/Dev (top) or WSJ Train/Dev (bottom). We can verify that both methods identify the correct SU span while removing the non-sentential header as the NSU.

Developed on EWT	B with my breakfast I like bacon and sausage when I having a big breakfast like E a grand slam with pancakes and the works.
Developed on WSJ	B with my breakfast I like bacon and sausage when I having a big breakfast like E a grand slam with pancakes and the works.

Table 11: **Example Output with One Incorrect Case**. We show the predictions made by our proposed method (BOS&EOS) developed on EWT Train/Dev (top) or WSJ Train/Dev (bottom). We can verify that the former extracts the correct SU span, while the latter incorrectly excludes the first prepositional phrase as an NSU.

		Train	Dev	Test
	Total SUs	37,447	2,021	7,442
	Total NSUs	0	0	0
Word-Level	B-Label	37,447	2,021	7,442
	I-Label	805,387	44,354	163,132
	O-Label	0	0	0
Character-Level	B-Label	37,447	2,021	7,442
	I-Label	4,308,729	236,798	876,461
	O-Label	0	0	0

Table 12: WSJ dataset statistics.

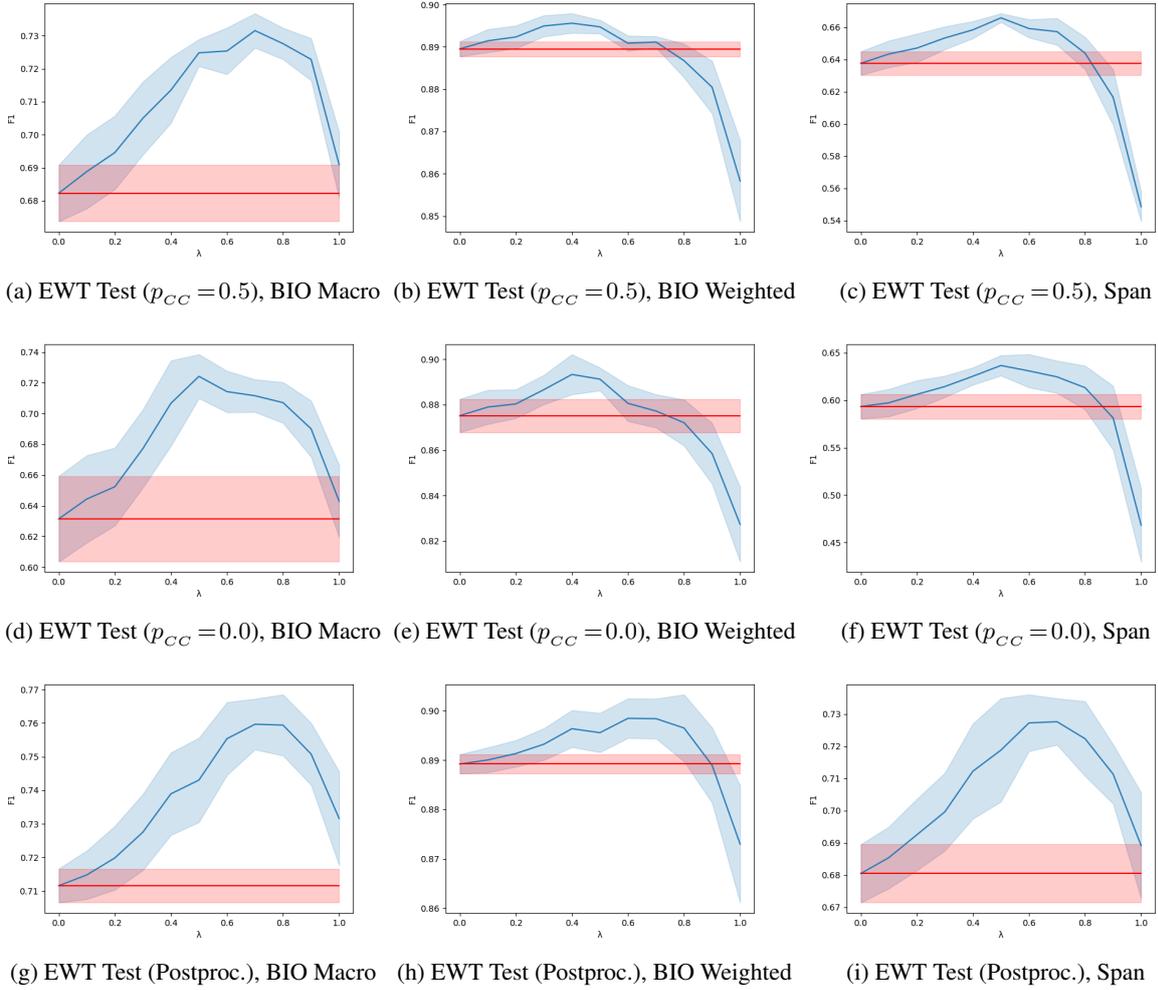


Figure 3: **Effect of the Unidirectional Model Interpolation Rate (Word-Level)**. We change $\lambda \in [0, 1]$ and report the macro/weighted average F1 of the BIO labeling task and the F1 score of the exact SU span extraction task. Interpolated results are shown in blue and non-interpolated results (i.e. $\lambda = 0$) shown in red. The line shows the mean and the shade shows the standard deviation from the five experimental runs.

Train/Dev Datasets	Model	WSJ Test ($p_{CC} = 0.5$)			WSJ Test ($p_{CC} = 0$)		
		BIO Macro	BIO Weighted	Span	BIO Macro	BIO Weighted	Span
EWT Train/Dev	EOS-Only	97.4 \pm 0.1	99.5 \pm 0.0	87.3 \pm 0.3	97.3\pm0.0	99.5 \pm 0.0	87.2 \pm 0.2
	EOS-Only (force last)	97.6\pm0.1	99.9\pm0.0	87.8\pm0.3	97.3\pm0.0	99.6\pm0.0	87.3\pm0.2
	BOS&EOS	97.1 \pm 0.2	99.4 \pm 0.0	86.7 \pm 0.5	97.0 \pm 0.1	99.3 \pm 0.0	86.5 \pm 0.3
WSJ Train/Dev	EOS-Only	98.4 \pm 0.6	99.7\pm0.1	92.1 \pm 2.9	98.2 \pm 0.4	99.7\pm0.1	90.6 \pm 1.8
	EOS-Only (force last)	98.4 \pm 0.6	99.7\pm0.1	92.1 \pm 2.9	98.2 \pm 0.4	99.7\pm0.1	90.6 \pm 1.8
	EOS-Only (+AUG)	98.2 \pm 1.1	99.1 \pm 1.0	92.6 \pm 2.5	97.3 \pm 1.9	99.3 \pm 0.8	87.8 \pm 6.3
	BOS&EOS	99.2\pm0.2	99.7\pm0.3	95.5\pm0.5	98.7\pm0.1	99.7\pm0.2	93.1\pm0.4
	BOS&EOS (+UNI)	98.5 \pm 0.3	98.9 \pm 0.5	92.9 \pm 1.0	98.1 \pm 0.3	98.8 \pm 0.5	91.4 \pm 0.8
	BOS&EOS (+UNI +AUG)	98.7 \pm 0.2	99.3 \pm 0.4	94.0 \pm 0.7	98.2 \pm 0.3	99.1 \pm 0.3	91.8 \pm 1.1

Table 13: **Overall Results on WSJ Test (RoBERTa, Word-Level)**. We report the macro/weighted average F1 of the BIO labeling task and the F1 score of the exact SU span extraction task.

Gauging the Gap Between Human and Machine Text Simplification Through Analytical Evaluation of Simplification Strategies and Errors

Daichi Yamaguchi Rei Miyata Sayuka Shimada Satoshi Sato
Nagoya University, Nagoya, Japan
{yamaguchi.daichi.e4, shimada.sayuka.y9}@s.mail.nagoya-u.ac.jp
{miyata, ssato}@nuee.nagoya-u.ac.jp

Abstract

This study presents an analytical evaluation of neural text simplification (TS) systems. Because recent TS models are trained in an end-to-end fashion, it is difficult to grasp their abilities to perform particular simplification operations. For the advancement of TS research and development, we should understand in detail what current TS systems can and cannot perform in comparison with human performance. To that end, we first developed an analytical evaluation framework consisting of fine-grained taxonomies of simplification strategies (at both the surface and content levels) and errors. Using this framework, we annotated TS instances produced by professional human editors and multiple neural TS systems and compared the results. Our analyses concretely and quantitatively revealed a wide gap between humans and systems, specifically indicating that systems tend to perform deletions and local substitutions while excessively omitting important information, and that the systems can hardly perform information addition operations. Based on our analyses, we also provide detailed directions to address these limitations.

1 Introduction

Text simplification (TS) is the task of reducing the content and structural complexity of text while retaining the core part of the original meaning (Alva-Manchego et al., 2020). TS can not only facilitate the text reading by children or language learners, but also improve the performance of downstream NLP applications, including machine translation and summarization (Siddharthan et al., 2004; Štajner and Popovic, 2016).

Early studies on TS have separately dealt with lexical simplification (Glavaš and Štajner, 2015) and syntactic simplification (Scarton et al., 2017), and developed simplification techniques specialized for particular linguistic phenomena. In contrast, recent studies have tackled TS as a task of

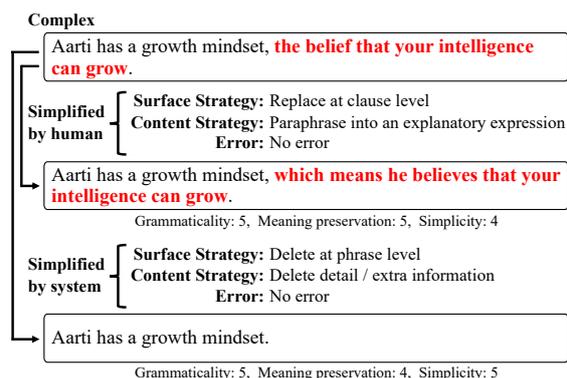


Figure 1: Example of an analytical evaluation in terms of editing strategies and errors.

monolingual translation from a complex to a simplified language using deep neural networks. While neural TS has demonstrated steady improvement, few studies have attempted to assess what kind of editing operations are performed by the systems in concrete terms. To further advance TS research and development, we should understand the potential and limitations of current TS technologies and precisely grasp the gap between human and machine TS. To do so, we need analytical frameworks that can be applied to both human and machine TS. In contrast to (machine) translation research and practice, where several frameworks have been developed for analyzing translation strategies (Chesterman, 2016) and errors (Lommel et al., 2014), no well-established framework tailored for TS tasks is available.

Therefore, in this study, we first propose an analytical evaluation framework consisting of taxonomies of editing strategies (both at the surface and content levels) and errors. We then report an experiment in which we apply our framework to instances of human and machine TS in various settings, and concretely describe the gap between them. Figure 1 shows an example of the evaluation using our framework, illustrating the detailed differences in editing operations between humans and

TS systems. Our results revealed that current neural TS systems can frequently replace local spans, while excessively deleting important parts. Moreover, TS systems cannot perform operations related to content addition, such as the addition of detail information. These findings enable us to understand the fundamental challenges of current technologies and pursue a promising avenue to fill the gap between humans and machines.

2 Related Work

To evaluate TS systems, automatic evaluation metrics such as SARI (Xu et al., 2016), BLEU (Papineni et al., 2002), and Flesch–Kincaid Grade Level (FKGL) (Kincaid et al., 1975) are widely used. SARI and BLEU use n-gram overlap between target sentences and human-created references, whereas FKGL uses the number of syllables and words in the output sentences. These metrics can be easily calculated if references are available and are indispensable for the rapid cycle of system development and evaluation. However, their limitations and pitfalls have been acknowledged. Sulem et al. (2018a), for example, reported that BLEU is negatively correlated with human evaluation scores, such as simplicity and grammaticality. Tanprasert and Kauchak (2021) also showed that the FKGL score can easily be manipulated by minor modifications, such as adding periods randomly.

Subjective human evaluation has also been implemented (Štajner and Nisioi, 2018; Sulem et al., 2018b; Al-Thanyyan and Azmi, 2021). In many cases, certain aspects of TS quality, namely grammaticality/fluency, meaning preservation/adequacy, and simplicity, are rated on a three- to five-point Likert scale based on the evaluation criteria.

Importantly, all the abovementioned evaluation methods only provide summative numerical scores. These scores are useful for comparing the general performances of different systems, but do not necessarily provide a guidepost for achieving higher system performance. To gain a detailed understanding of what TS systems can/cannot do vis-à-vis editing operations by humans, analytical evaluation methods are required.

The analytical evaluation of TS can be broadly divided into strategy and error analyses. The former concerns the type of editing operation (strategy) performed to produce the simplified text. Previous studies have acknowledged general strategies, such as paraphrasing, deletion, and splitting (Shard-

low, 2014), and document-level strategies, such as sentence reordering and sentence-joining operations (Alva-Manchego et al., 2019b). However, these roughly typify superficial textual changes rather than detailed content-level changes that capture editing operations peculiar to TS. The latter concerns the type of error in the resulting simplified text. In contrast to automatic and human evaluations, fewer attempts have been made to conduct an error analysis (Maddela et al., 2021).¹

Proper implementation of analytical evaluation requires well-formulated frameworks to classify textual phenomena observed in the outputs. The general editing strategies mentioned above and some guidelines for human writers (Mitkov and Štajner, 2014) are not sufficiently concrete for fine-grained analysis. Although several typologies of simplification operations (e.g., Amancio and Specia, 2014; Brunato et al., 2014; Koptient et al., 2019) and editing guidelines for human writers (Mitkov and Štajner, 2014) have been proposed, the following limitations can be generally acknowledged: (1) content-level operations are not fully covered; (2) their applicability to outputs of automatic TS systems has not been verified. In the field of translation studies, a wide variety of translation strategies have been proposed to describe the differences between source and target texts (e.g., Vinay and Darbelnet, 1958; Molina and Hurtado Albir, 2002). Chesterman (2016), for example, developed a comprehensive taxonomy of translation strategies that consists of syntactic, semantic, and pragmatic categories, and each includes ten strategies. Taxonomies of translation errors have also been developed and are widely used in practice, such as Multidimensional Quality Metrics (MQM) (Lommel et al., 2014). Although these existing frameworks may be useful as points of departure, detailed ones dedicated to TS tasks are still lacking.

3 Framework of Analytical Evaluation

We developed taxonomies of simplification strategies and errors as the analytical evaluation framework for TS. Simplification strategies consist of two independent components: **surface strategies**, which capture superficial operations for grammatical or textual elements, and **content strategies**, which capture semantic or content changes from the viewpoint of simplification. In this framework,

¹The under-reporting of error analysis is a general problem in NLG literature (van Miltenburg et al., 2021).

each TS instance (i.e., a minimally decomposed editing operation) is first judged as an error category listed in the error taxonomy. If it is not, the instance is then independently labeled a surface and content strategy (see also Figure 1). Our framework includes guidelines for annotating surface and content strategies in the form of a decision tree.²

3.1 Taxonomy Construction

Specifically referring to Alva-Manchego et al. (2019b), Chesterman (2016), and Shardlow (2014), we created taxonomies of the simplification strategies and errors through an analysis of human and machine TS instances. As manual simplification data, we used original and simplified news articles from Newsela,³ which were produced by professional editors and are expected to include various types of editing operations, including creative ones. We selected four articles from Newsela’s Popular category. Each article has four simplified versions with different degrees of simplicity, from Lv0 (the original document) to Lv4 (the simplest document). We manually aligned sentences from all adjacent-level documents (e.g., Lv0–Lv1, Lv1–Lv2) and acquired 551 complex–simplified pairs that exhibited any sort of rewriting.⁴ Next, we decomposed the rewriting from complex to simplified sentences into minimum edits (see Figure 2).⁵ Consequently, we acquired 1,133 minimum editing instances of human simplification.

First, using these instances, we created prototype taxonomies of the simplification strategies in the bottom-up procedures: (i) for each instance, we devised labels for describing surface and content strategies; and (ii) we aggregated and revised the labels to form systematic taxonomies. Edit (3) in Figure 2 is an example of a minimum edit instance: “hard work will help you reach your goals” → “hard work is important”. The same editing operation is annotated differently with the surface strategy (“Replace at sentence level”) and content strategy (“Paraphrase into a direct expression”).

Second, using simplified instances generated by TS systems, we expanded and modified the proto-

²The decision trees are shown in Appendix A.2.

³<https://newsela.com/data>

⁴These pairs included the sentences that were not aligned because we considered such sentences as instances of addition or deletion of a sentence.

⁵Following Miyata and Fujita (2021), we defined a minimum edit as “a small edit that is difficult to be further decomposed into more than one independent edit” and that does not induce “ungrammaticality in the edited sentence” (p. 1541).

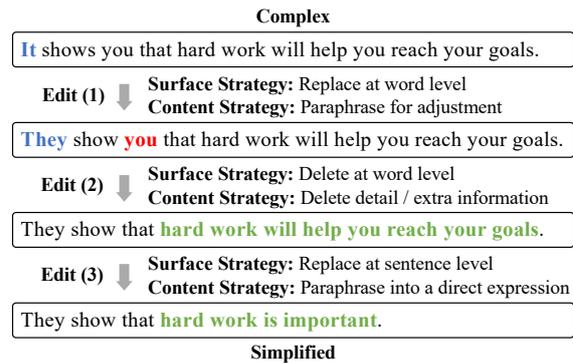


Figure 2: Example of the decomposition of rewriting instances and labeling of strategies.

type taxonomies to improve their applicability. At this stage, we created an error taxonomy by analyzing the system errors.⁶ We used three neural TS models, Transformer (Jiang et al., 2020), DRESS (Zhang and Lapata, 2017), and SUC (Sun et al., 2020) that were trained on Newsela data.⁷ From the same Newsela’s articles, we selected 166, 38, and 38 sentences for Transformer, DRESS, and SUC, respectively, to generate their simplified versions.⁸ We decomposed 125 outputs that exhibited any sort of rewriting to acquire 217 minimum edits. We then separated the non-error and error instances. Using the prototype taxonomies of the surface and content strategies, we classified non-error instances and, if necessary, modified the taxonomies to properly cover all instances. We also created an error taxonomy by analyzing the error instances.

3.2 Taxonomy of Surface Strategies

We defined 22 surface strategies, S1–S22, under the seven general categories: **Replacement**, **Deletion**, **Addition**, **Integration**, **Splitting**, **Move**, and **No change**.⁹ **Replacement**, **Deletion**, and **Addition** have the same set of linguistic focuses, i.e., punctuations, words, phrases, clauses, and sentences.

Note that, if the head of a phrase changes (e.g., “playing the video games” → “the video games), it is classified into the **Replacement** category. If the head of a phrase is retained (e.g., “the video games” → “the games”), it is classified into the **Deletion** rather than **Replacement** category.

⁶In principle, simplification instances by professional human editors seldom include errors.

⁷We explain the detailed implementation in §4.1.

⁸We first used the same set of 38 sentences, but we found out that the outputs of DRESS and SUC consisted of many error instances. To collect a wide range of non-error instances, we added another 128 sentences only for Transformer.

⁹See Table 2 for the detailed surface strategies and Appendix B for the example sentences.

3.3 Taxonomy of Content Strategies

We defined 30 content strategies, C1–C30, under five general categories: **No content change**, **Content deletion**, **Content addition**, **Content change**, and **Document-level adjustment**.¹⁰

Note that while the content strategies in **No content change**, except for (C5) **Remain unchanged**, change the surface structure or textual element, they do not change the propositional meaning of the sentence. **Document-level adjustment** includes the change of the sentence order in a document and the secondary edits that need to be performed due to changes made to a different sentence; for example, some lexical changes might entail changing the pronouns in the later sentences.

3.4 Error Taxonomy

We defined four error categories: **Inappropriate deletion**, **Inappropriate addition**, **Inappropriate paraphrase**, and **Non-sentence**.¹¹ The first three categories roughly correspond to **Deletion**, **Addition**, and **Replacement** in the surface strategies. **Non-sentence** covers other error types that make the sentence ungrammatical or unintelligible.

4 Experimental Setup

To clarify the potential and limitations of current TS systems in comparison with human performance, we designed an experiment to annotate TS instances produced by human editors and recent neural TS systems using our taxonomies of simplification strategies and errors described in §3. We also conducted a human evaluation to better understand the general tendencies of how strategies and errors affect the TS quality.

4.1 Neural Text Simplification Systems

We implemented six systems, that is, three neural models below trained separately with Newsela (in-domain setting) or Wikipedia (out-of-domain setting).¹² It should be noted that the training data size and pre-processing methods differed depending on the models, as we aimed to replicate the models described in the original papers as much as possible.

¹⁰See Table 4 for the detailed content strategies and Appendix B for the example sentences.

¹¹See Appendix B for the example sentences.

¹²We calculated scores of automatic evaluation metrics and verified that we had appropriately reproduced the implementations reported in the original papers. See Appendix C.1 for details on the automatic evaluation.

Transformer (Jiang et al., 2020)¹³ This BERT-initialized Transformer model is a state-of-the-art model. We used the Newsela and Wikipedia models distributed by the authors.

DRESS (Zhang and Lapata, 2017)¹⁴ This model exploits reinforcement learning, which rewards rewriting. Many studies have used this as a baseline (e.g., Vu et al., 2018; Nassar et al., 2019; Omelianchuk et al., 2021). To train the Newsela model, we used newsela_data_share-20150302 from the Newsela corpus, excluding Lv0–Lv1, Lv1–Lv2, and Lv2–Lv3 pairs, following the original paper. We also excluded sentences that were more than 85 words per sentence or included “/” because the original code could not process them. The remaining 94,635 sentences¹⁵ were used for training after the named entities were tagged with Stanford CoreNLP (Manning et al., 2014).¹⁶ We used processed Wikilarge to train the Wikipedia model.

SUC (Sun et al., 2020)¹⁷ This model uses one target sentence and two preceding and following sentences as input. Only this model exploits the context among these three models. Because Sun et al. (2020) did not provide a Newsela model, we trained it using Newsela-Auto, the same dataset used in the Transformer model above. Excluding Lv0–Lv1, Lv1–Lv2, and Lv2–Lv3 pairs following Zhang and Lapata (2017), we used the 640,867 sentences with context and 173,105 sentences without context. To train the Wikipedia model, we used the first 116,020 sentences with context and all of the 40,893 sentences without context from the distributed dataset. We created the vocabularies for Newsela and Wikipedia models, respectively, from the training data using spaCy (Honnibal and Montani, 2017).¹⁸

4.2 Annotation of Strategies and Errors

As evaluation data, from three original Newsela articles (Lv0) in the Popular category, we respectively extracted 13, 11, and 22 sequential sentences while retaining the textual cohesion. For these 46

¹³<https://github.com/chaojiang06/wiki-auto>

¹⁴<https://github.com/XingxingZhang/dress>

¹⁵Zhang and Lapata (2017) reported using 94,208 sentences. Although we processed the corpus in the same manner, we could not obtain the same number of sentences.

¹⁶<https://github.com/stanfordnlp/CoreNLP>

¹⁷<https://github.com/RLSNLP/>

[Document-Context-to-Sentence-Simplification](https://github.com/Document-Context-to-Sentence-Simplification)

¹⁸<https://github.com/explosion/spaCy>

Score	Grammaticality (G)	Meaning preservation (M)	Simplicity (S)
5	Native speaker level fluent	Adequately preserved	Much simpler
4	Non-native speaker level fluent	Mostly preserved	Simpler
3	Understandable	Partially preserved	The same simplicity
2	Partially understandable	Completely different	More difficult
1	Completely unintelligible	Unintelligible	Unintelligible

Table 1: Abridged guidelines for human evaluations. The full version is shown in Appendix A.1.

complex sentences, we extracted 54 corresponding simplified sentences from Newsela’s articles (Lv1) as human references¹⁹ and generated 276 simplified sentences (46 sentences \times 6 systems) as system outputs. We decomposed 39 and 191 sentences that exhibited any sort of rewriting to acquire 105 and 389 minimum edits, respectively, for human references and system outputs.²⁰

Each editing instance was annotated with strategies and error categories based on the classification procedures explained in §3. We counted the sentences that were not rewritten as instances of strategy. The annotation was carried out independently by the first and third authors, who can adequately understand the English text and have a good command of the analytical evaluation framework, i.e., the taxonomies and guidelines. The inter-annotator agreement scores (Cohen’s unweighted kappa) for the surface strategies, content strategies, and errors were 0.806, 0.745, and 0.851, respectively, indicating substantial agreement (Landis and Koch, 1977).²¹ After the independent annotation, the annotators resolved any disagreement in judgments through discussions to obtain the final labels.

4.3 Human Evaluation

Using the sentence data used in §4.2, we also conducted a subjective human evaluation to assess the grammaticality (G), meaning preservation (M), and simplicity (S) of the simplified sentences generated by the six systems.

The annotators were two professional translators who were familiar with Japanese–English translation, English proofreading, and native language checking. They assigned a score to each sentence using a five-point Likert scale by referring to the

¹⁹Because human references include the instances of sentence addition and splitting, the number of simplified sentences is larger than that of complex sentences.

²⁰This means that the average rewriting rate for human editors was 2.69 times per sentence and that of systems was 2.03 times.

²¹When calculating the agreement scores for the strategies, we aggregated the annotations for the errors into one class and vice versa. The detailed distributions of annotations are presented in Appendix D.

evaluation guidelines, an abridged version of which is shown in Table 1.²² Before commencement of the formal evaluation, they evaluated another 29 sentences as a practice to properly understand the task. They evaluated the same set of sentences that exhibited any sort of rewriting. We consistently gave scores of 5, 5, and 3 for G, M, and S, respectively, to the non-rewritten sentences. The inter-annotator agreement scores (Cohen’s quadratic weighted kappa) for G, M, and S were 0.541, 0.257, and 0.628, respectively.²³

5 Results and Discussions

5.1 Surface Strategies

Table 2 lists the annotation results for the surface strategies with human evaluation scores for the system outputs.²⁴ Note that for each strategy, the human evaluation score was calculated using *sentences that exhibit the strategy*. As single sentences may include multiple strategies, the scores may be influenced by other strategies. Nevertheless, the general impact of each strategy can be inferred.

All the systems performed **Replacement** less frequently than humans did. The systems chiefly performed (S2) **Replace at word level** and could not perform (S4) **Replace at clause level** or (S5) **Replace at sentence level**, whereas humans performed **Replacement** strategies at various linguistic levels. This indicates the incapability of current models to learn replacement operations for linguistic units larger than phrases.

Deletion was the dominant strategy for the systems; the Transformer systems and in-domain DRESS system performed **Deletion** more frequently than humans. Human evaluation scores suggest the trade-off between meaning preservation and simplicity according to the size of the linguistic unit that is deleted; the deletion of a larger

²²The detailed guidelines are presented in Appendix A.1.

²³When calculating the inter-annotator agreement scores, we excluded the non-rewritten sentences. If we include them, the scores for G, M, and S rise to 0.618, 0.433, and 0.725, respectively.

²⁴The overall results of the human evaluation are presented in Appendix C.2.

Surface strategy	Human ref.	Number of annotated instances						Human evaluation		
		Transformer		DRESS		SUC		G	M	S
		IND	OOD	IND	OOD	IND	OOD			
Replacement	29	20	12	19	12	12	0			
(S1) Replace at punctuation level	(3)	(3)	(0)	(1)	(0)	(0)	(0)	4.63	3.25	3.88
(S2) Replace at word level	(4)	(10)	(5)	(13)	(11)	(10)	(0)	3.80	3.72	3.14
(S3) Replace at phrase level	(11)	(7)	(6)	(5)	(1)	(2)	(0)	4.26	3.81	3.76
(S4) Replace at clause level	(4)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
(S5) Replace at sentence level	(7)	(0)	(1)	(0)	(0)	(0)	(0)	5.00	4.50	4.50
Deletion	30	39	35	32	17	16	0			
(S6) Delete at punctuation level	(4)	(0)	(3)	(1)	(0)	(0)	(0)	3.38	3.50	3.00
(S7) Delete at word level	(6)	(5)	(10)	(5)	(2)	(7)	(0)	3.67	3.31	3.29
(S8) Delete at phrase level	(12)	(16)	(10)	(10)	(2)	(5)	(0)	3.84	3.29	3.78
(S9) Delete at clause level	(3)	(18)	(12)	(16)	(13)	(4)	(0)	3.91	3.12	3.91
(S10) Delete at sentence level	(5)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
Addition	20	1	0	0	0	1	0			
(S11) Add at punctuation level	(0)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
(S12) Add at word level	(3)	(1)	(0)	(0)	(0)	(1)	(0)	3.75	4.00	3.50
(S13) Add at phrase level	(8)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
(S14) Add at clause level	(1)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
(S15) Add at sentence level	(8)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
Integration	0	0	0	0	0	0	0			
(S16) Integrate two sentences	(0)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
(S17) Integrate more than two sentences	(0)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
Splitting	3	0	0	0	0	0	0			
(S18) Split by phrase	(0)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
(S19) Split by clause	(3)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
Move	8	0	1	0	0	1	0			
(S20) Move constituents	(4)	(0)	(1)	(0)	(0)	(1)	(0)	3.50	2.50	2.25
(S21) Move a sentence	(4)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-
No transformation	15	4	16	8	23	16	18			
(S22) Use an identical sentence	(15)	(4)	(16)	(8)	(23)	(16)	(18)	5.00	5.00	3.00
Total	105	64	64	59	52	46	18			
Precision		0.313	0.297	0.288	0.327	0.283	0.278			
Recall		0.190	0.181	0.162	0.162	0.124	0.048			

Table 2: Number of annotated instances for the surface strategies. Three TS systems are trained with in-domain Newsela data (IND) and out-of-domain Wikipedia data (OOD). Human evaluation scores are the averaged scores for system outputs that involve each strategy (G: grammaticality; M: meaning preservation; S: simplicity).

unit can increase the simplicity score, but decrease the meaning preservation score. It is also notable that the number of **(S9) Delete at clause level** performed by the systems was much larger than that performed by humans. This is attributable to the structure of the training data, which are aligned at the single-sentence level. Consider the case in which the complex sentence “I bought an apple and ate it” is split into two sentences, “I bought an apple” and “I ate it”. In the current research practices for preparing training data, the two simplified sentences are separately aligned with the complex sentence. This would induce the systems to excessively learn large deletions, such as Examples 1 in Table 3. It is also important to note that none of the systems performed **(S10) Delete at sentence level**. This may be because the training data did not include instances of sentence deletion, as the alignment of such cases is difficult.

The systems seldom performed **Addition**, whereas humans performed this 20 times at var-

ious linguistic levels. Although instances of addition were included in the training data, even word- and phrase-level addition strategies were hardly observed. This implies the fundamental difficulties of addition operations for the current models and training data.

The systems used in this study cannot learn **Integration (S16 and S17)**²⁵, **Splitting (S18 and S19)**, and **(S21) Move a sentence** because of the aforementioned data structure problem. Although some end-to-end systems can perform **Splitting** and **Integration** (Scarton and Specia, 2018), these suprasentential operations remain to be fully achieved in the neural TS research. To address the **Splitting** operation, we can refer to the rich accumulation of linguistically motivated studies on syntactic simplification (Scarton et al., 2017).

The final two rows in Table 2 show the overall

²⁵Neither did the humans perform Integration in the evaluation dataset. We observed six instances of Integration in the 1,133 instances used for taxonomy creation.

#	Model	Text
1	Input	But when schools start later, teens get to class on time and find it easier to stay awake, a new study finds.
	Transformer (IND)	but when schools start later , teens get to class on time .
	DRESS (IND)	But when schools start later , teens get to class on time .
	Human reference	A new study finds that when schools start later, teens get to class on time. They also find it easier to stay awake.
2	Input	Not everyone can become a genius or a star athlete, but they can improve the skills they have and develop new ones.
	Transformer (IND)	not everyone can be a genius or a star athlete .
3	Input	Knowing this, schools in several districts have begun to shift their start times.
	Human reference	Knowing this about teens, schools in several districts have begun to shift their start times.
4	Input	Information from millions of cones reaches our brains as electrical signals that communicate all the types of light reflected by what we see, which is then interpreted as different shades of color.
	Human reference	The cones then send information to our brains, which interprets the light we see as different colors.

Table 3: Examples of system outputs and references.

precision and recall of adopted strategies by the systems in comparison with humans’ strategies for the same sentences. The precision scores are about 0.2–0.3 and the recall scores are all below 0.2, which means that humans and systems tend to adopt different strategies even for the same sentence. Although further investigations are needed to draw insights from these results, we should be aware of the substantial differences between humans and machines in terms of simplification operations.²⁶

5.2 Content Strategies

Table 4 lists the annotation results for the content strategies. While humans performed **(C1) Transform syntactic structure** nine times, the systems rarely did. Most **C1** cases by humans involved sentence splitting, and as previously mentioned, the systems could not learn this operation from the current training data.

The systems generally performed various **Content deletion** strategies. It is worth noting that **(C10) Delete important information**, a large deletion corresponding to **(S9) Delete at clause level** in surface strategies, was performed frequently. Example 2 in Table 3 illustrates the deletion of the latter clause. Although the output can be regarded as a simplified version of the input at the sentence level, this deletion might be inappropriate in terms of logical flow in the entire document. In this sense, these categories might be regarded as **(E1) Inappropriate deletion** in the error taxonomy. Indeed, humans did not adopt this strategy.

The systems did not perform any **Content addition**, which corresponds to the lack of **Addition** of the surface strategies. These strategies require contextual information in many cases like “this” →

“this about teens” (See Example 3 in Table 3). However, even the context-aware SUC models cannot perform these addition operations.

As for **Content change**, the Transformer and DRESS performed **(C20) Paraphrase into a similar phrase** more than humans. The neural systems generally have abilities to perform local rewriting like “become” → “be” (see Example 2 in Table 3). Similarly, **(C25) Paraphrase into an essential point**, which substantially concerns deleting or altering local elements like “color production” → “color”, was performed well by the in-domain systems. By contrast, the systems cannot perform **(C21) Paraphrase into an explanatory expression** and **(C24) Paraphrase into a concrete expression**. The former requires external or contextual knowledge to add information, such as “the belief that your intelligence can grow” → “which means he believes that your intelligence can grow”. The latter requires word sense disambiguation or anaphora resolution to explicitly indicate the hidden meaning, such as “ones” → “friendships”. In general, current systems have limitations in performing these sophisticated **Content change** operations, such as Example 4 in Table 3.

The systems hardly performed **Document-level adjustment**. **(C27) Change information flow** corresponds to **(S21) Move a sentence** at surface level, which is architecturally impossible for the systems used in this study. The other strategies, C28–C30, depend on the results of other operations in the document, and are fundamentally difficult for current systems that do not exploit the output-side context.

5.3 Errors

Table 5 lists the annotation results for the simplification errors. As mentioned in §5.2, the number of **(E1) Inappropriate deletion** can increase if we consider **(C10) Delete important information** as an error. The instances of **(E2) Inappropriate**

²⁶These differences might be attributed not only to the inability of systems to replicate human performance, but also to the nature of TS tasks. Examining the differences between human editors would be an important future task.

Content strategy	Number of annotated instances								Human evaluation		
	Human	Transformer		DRESS		SUC		G	M	S	
	ref.	IND	OOD	IND	OOD	IND	OOD				
No content change	31	7	24	11	24	17	18				
(C1) Transform syntactic structure	(9)	(0)	(1)	(0)	(0)	(1)	(0)	3.50	2.50	2.25	
(C2) Paraphrase into an abbreviation	(0)	(0)	(0)	(1)	(0)	(0)	(0)	3.00	3.50	4.00	
(C3) Paraphrase into a non-abbreviation	(1)	(0)	(2)	(0)	(0)	(0)	(0)	3.50	3.25	2.25	
(C4) Paraphrase into standard form	(6)	(3)	(5)	(2)	(1)	(0)	(0)	4.23	3.73	3.55	
(C5) Remain unchanged	(15)	(4)	(16)	(8)	(23)	(16)	(18)	5.00	5.00	3.00	
Content deletion	24	37	32	31	16	14	0				
(C6) Delete introduction / conclusion	(2)	(1)	(0)	(1)	(0)	(1)	(0)	2.83	2.83	4.00	
(C7) Delete a parallel element	(1)	(5)	(1)	(6)	(0)	(1)	(0)	3.50	3.27	3.69	
(C8) Delete information for cohesion	(5)	(6)	(6)	(7)	(3)	(3)	(0)	3.94	3.30	3.64	
(C9) Delete a modifier	(9)	(6)	(10)	(4)	(2)	(5)	(0)	3.94	3.37	3.46	
(C10) Delete important information	(0)	(5)	(6)	(4)	(1)	(1)	(0)	3.91	3.06	3.97	
(C11) Delete detail / extra information	(7)	(14)	(9)	(9)	(10)	(3)	(0)	3.93	3.19	3.92	
Content addition	17	0	0	0	0	0	0				
(C12) Add introduction / conclusion	(0)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
(C13) Add a parallel element	(2)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
(C14) Add contextual information	(0)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
(C15) Add information for cohesion	(1)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
(C16) Add a modifier	(4)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
(C17) Add detail / extra information	(10)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
Content change	22	19	8	17	11	15	0				
(C18) Change aspect	(1)	(2)	(0)	(0)	(0)	(1)	(0)	3.67	3.83	3.67	
(C19) Change modality	(0)	(1)	(1)	(0)	(0)	(2)	(0)	3.00	3.25	2.50	
(C20) Paraphrase into a similar phrase	(2)	(4)	(3)	(7)	(5)	(1)	(0)	3.75	3.60	3.18	
(C21) Paraphrase into an explanatory expression	(4)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
(C22) Paraphrase into a direct expression	(6)	(3)	(0)	(1)	(2)	(2)	(0)	3.88	3.56	3.44	
(C23) Paraphrase into a brief expression	(1)	(1)	(1)	(1)	(0)	(0)	(0)	4.00	3.33	4.33	
(C24) Paraphrase into a concrete expression	(1)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
(C25) Paraphrase into an essential point	(4)	(6)	(1)	(7)	(2)	(5)	(0)	4.02	3.81	3.55	
(C26) Paraphrase into a different view	(3)	(2)	(2)	(1)	(2)	(4)	(0)	4.09	3.95	3.00	
Document-level adjustment	11	1	0	0	1	0	0				
(C27) Change information flow	(4)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
(C28) Delete for adjustment	(2)	(1)	(0)	(0)	(1)	(0)	(0)	4.25	3.25	4.00	
(C29) Add for adjustment	(2)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
(C30) Paraphrase for adjustment	(3)	(0)	(0)	(0)	(0)	(0)	(0)	-	-	-	
Total	105	64	64	59	52	46	18				
Precision		0.219	0.250	0.237	0.308	0.239	0.278				
Recall		0.133	0.152	0.133	0.152	0.105	0.048				

Table 4: Number of annotated instances for the content strategies.

addition were also observed. In particular, SUC produced many such instances. Considering the observation that almost no instance was annotated as **Addition** in Table 2, what the systems added to output sentences were not judged as (successful) strategies but as errors. **(E3) Inappropriate paraphrase** is the most frequent error type for most systems, which includes, for example, “intelligence” (being intellectual) → “spy” and “cells called neurons” → “DNA”. These errors are problematic because incorrect information can be conveyed to readers without being noticed as errors.

For Transformer and DRESS, the in-domain systems trained on Newsela generally produced more errors than the out-of-domain systems trained on Wikipedia. Considering the fewer number of **No transformation** cases of in-domain systems (see Table 2), in-domain systems tended to be more aggressive but erroneous than out-of-domain systems.

For all the human evaluation scores, except for

the meaning preservation score for **(E2) Inappropriate addition**, the averaged scores are below 3. This indicates that inclusion of any error can lead to an unacceptable output sentence.

6 Conclusions and Outlook

To better advance TS research and practice, in this study, we conducted an analytical evaluation of current neural TS systems and showed their potential and limitations in comparison with human performance. Using our proposed evaluation framework consisting of taxonomies of surface strategies, content strategies, and errors, we annotated both the human references and outputs of six systems (three models trained on in-domain and out-of-domain datasets). The results demonstrated that, while current TS systems can perform deletions and local substitutions, their performance is far behind human parity, owing to the following limitations:

Error category	Number of annotated instances						Human evaluation		
	Transformer		DRESS		SUC		G	M	S
	IND	OOD	IND	OOD	IND	OOD			
(E1) Inappropriate deletion	6	2	7	0	9	1	2.62	2.82	2.80
(E2) Inappropriate addition	6	2	4	4	12	51	2.68	3.86	2.11
(E3) Inappropriate paraphrase	17	5	28	14	15	0	2.83	2.53	2.76
(E4) Non-sentence	0	0	0	1	3	0	1.00	1.25	1.00
Total	29	9	39	19	39	52			

Table 5: Number of annotated instances for the error categories.

- The systems have difficulties in substituting a linguistic unit larger than a phrase, including sentence splitting.
- Excessive deletion of clause-level important information has occurred frequently.
- The systems tried to perform addition operations; however, they always failed to produce correct results.

Our analytical evaluation also suggests detailed paths to overcome these issues. For example, in addition to improving the capacity of end-to-end neural models, utilizing technologies tailored to particular operations such as sentence splitting and explanation generation can be helpful. To mitigate the excessive deletion, it would be effective to refine the alignment methods. Exploiting document-level contexts on both input and output sides and/or document-external knowledge is a necessary task for successful content addition.

Limitations

Applicability. The primary limitation in our study is that we chiefly used the Newsela dataset to build the annotation framework, i.e., the taxonomies and decision trees, and conduct the analytical evaluation. While we assume that the Newsela dataset includes diverse simplification operations as mentioned in §3.1, the applicability of our framework to other domains or datasets, such as Simple Wikipedia, needs to be investigated.²⁷

The diversity of adopted TS systems is also limited. As the aim of this pilot study is to demonstrate the usefulness of analytical evaluation, we mainly selected orthodox baseline models. To further improve the applicability, it is important to examine other types of TS models, such as controllable models (e.g., Maddela et al., 2021; Nishihara et al., 2019; Scarton and Specia, 2018) and edit-based models (e.g., Dong et al., 2019; Stahlberg and Kumar, 2020). Further investigation of various

²⁷The characteristics of Simple Wikipedia as a TS data resource have been extensively discussed (Xu et al., 2015).

document-level models other than SUC used in this study will also be needed (Sun et al., 2021).

Although our taxonomies are mostly language independent, the forms of decision trees for strategy annotation may need to be changed depending on the language because the decision order was defined based on the degree of difficulty in identifying the strategies, which might be language dependent.²⁸

Feasibility. The annotation of simplification strategies and errors was conducted by the authors, who were involved in the development of the annotation framework. Although the authors independently conducted the annotation task and substantial inter-annotator agreement was achieved, the feasibility of annotation by those outside this study has not been examined. To improve the feasibility, more detailed instructions and a sufficient training session may be needed. Although sharing the annotated data would be beneficial for the feasibility, it is difficult due to copyright issues.

Acknowledgments

We are grateful to Newsela for sharing the data. This work was supported by JSPS KAKENHI Grant Number 19H05660 and by the Research Grant Program of KDDI Foundation, Japan.

References

- Suha S. Al-Thanyyan and Aqil M. Azmi. 2021. [Automated text simplification: A survey](#). *ACM Computing Surveys*, 54(2):1–36.
- Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019a. [EASSE: Easier automatic sentence simplification evaluation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong, China.

²⁸The details of decision trees are presented in Appendix A.2

- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2019b. [Cross-sentence transformations in text simplification](#). In *Proceedings of the 2019 Workshop on Widening NLP*, pages 181–184, Florence, Italy.
- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. [Data-driven sentence simplification: Survey and benchmark](#). *Computational Linguistics*, 46(1):135–187.
- Marcelo Amancio and Lucia Specia. 2014. [An analysis of crowdsourced text simplifications](#). In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 123–130, Gothenburg, Sweden.
- Dominique Brunato, Felice Dell’Orletta, Giulia Venturi, and Simonetta Montemagni. 2014. [Defining an annotation scheme with a view to automatic text simplification](#). In *Proceedings of the First Italian Conference on Computational Linguistics (CLiC-it)*, pages 87–92, Pisa, Italy.
- Andrew Chesterman. 2016. *Memes of translation: The spread of ideas in translation theory*, 2nd edition. Amsterdam: John Benjamins.
- Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. [EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3393–3402, Florence, Italy.
- Goran Glavaš and Sanja Štajner. 2015. [Simplifying lexical simplification: Do we need simplified corpora?](#) In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 63–68, Beijing, China.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.
- Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. [Neural CRF model for sentence alignment in text simplification](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7943–7960, Online.
- J. Peter Kincaid, Robert P. Fishburne Jr, Richard L. Rogers, and Brad S. Chissom. 1975. [Derivation of new readability formulas \(Automated Readability Index, Fog Count and Flesch Reading Ease Formula\) for Navy enlisted personnel](#). Technical report, Institute for Simulation and Training, University of Central Florida.
- Anaïs Koptient, Rémi Cardon, and Natalia Grabar. 2019. [Simplification-induced transformations: Typology and some characteristics](#). In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 309–318, Florence, Italy.
- J. Richard Landis and Gary G. Koch. 1977. [The measurement of observer agreement for categorical data](#). *Biometrics*, 33(1):159–174.
- Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. [Multidimensional Quality Metrics \(MQM\): A framework for declaring and describing translation quality metrics](#). *Tradumática*, 12:455–463.
- Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. [Controllable text simplification with explicit paraphrasing](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 3536–3553, Online.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. [The Stanford CoreNLP natural language processing toolkit](#). In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, pages 55–60, Baltimore, Maryland, USA.
- Ruslan Mitkov and Sanja Štajner. 2014. [The fewer, the better? A contrastive study about ways to simplify](#). In *Proceedings of the Workshop on Automatic Text Simplification - Methods and Applications in the Multilingual Society (ATS-MA)*, pages 30–40, Dublin, Ireland.
- Rei Miyata and Atsushi Fujita. 2021. [Understanding pre-editing for black-box neural machine translation](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1539–1550, Online.
- Lucía Molina and Amparo Hurtado Albir. 2002. [Translation techniques revisited: A dynamic and functionalist approach](#). *Meta*, 47(4):498–512.
- Islam Nassar, Michelle Ananda-Rajah, and Gholamreza Haffari. 2019. [Neural versus non-neural text simplification: A case study](#). In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association (ALTA)*, pages 172–177, Sydney, Australia.
- Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. [Controllable text simplification with lexical constraint loss](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop (ACL SRW)*, pages 260–266, Florence, Italy.
- Kostiantyn Omelianchuk, Vipul Raheja, and Oleksandr Skurzhanyski. 2021. [Text Simplification by Tagging](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 11–25, Online.

- Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. **BLEU: A method for automatic evaluation of machine translation**. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, Pennsylvania, USA.
- Carolina Scarton, Alessio Palmero Aprosio, Sara Tonelli, Tamara Martín Wanton, and Lucia Specia. 2017. **MUSST: A multilingual syntactic simplification tool**. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP): System Demonstrations*, pages 25–28, Taipei, Taiwan.
- Carolina Scarton and Lucia Specia. 2018. **Learning simplifications for specific target audiences**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 712–718, Melbourne, Australia.
- Matthew Shardlow. 2014. **A survey of automated text simplification**. *International Journal of Advanced Computer Science and Applications (IJACSA), Special Issue on Natural Language Processing 2014*, 4(1):58–70.
- Advait Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. **Syntactic simplification for improving content selection in multi-document summarization**. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 896–902, Geneva, Switzerland.
- Felix Stahlberg and Shankar Kumar. 2020. **Seq2Edits: Sequence transduction using span-level edit operations**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online.
- Sanja Štajner and Sergiu Nisioi. 2018. **A detailed evaluation of neural sequence-to-sequence models for in-domain and cross-domain text simplification**. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*, pages 3026–3033, Miyazaki, Japan.
- Sanja Štajner and Maja Popovic. 2016. **Can text simplification help machine translation?** In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 230–242, Riga, Latvia.
- Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. **BLEU is not suitable for the evaluation of text simplification**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 738–744, Brussels, Belgium.
- Elior Sulem, Omri Abend, and Ari Rappoport. 2018b. **Simple and effective text simplification using semantic and neural methods**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 162–173, Melbourne, Australia.
- Renliang Sun, Hanqi Jin, and Xiaojun Wan. 2021. **Document-level text simplification: Dataset, criteria and baseline**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7997–8013, Online and Punta Cana, Dominican Republic.
- Renliang Sun, Zhe Lin, and Xiaojun Wan. 2020. **On the helpfulness of document context to sentence simplification**. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, pages 1411–1423, Barcelona, Spain (Online).
- Teerapaun Tanprasert and David Kauchak. 2021. **Flesch-kincaid is not a text simplification evaluation metric**. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 1–14, Online.
- Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. **Underreporting of errors in NLG output, and what to do about it**. In *Proceedings of the 14th International Conference on Natural Language Generation (INLG)*, pages 140–153, Aberdeen, Scotland, UK.
- Jean-Paul Vinay and Jean Darbelnet. 1958. *Stylistique comparée du français et de l'anglais*. Didier, Paris, trans. and ed. by J. C. Sager & M.-J. Hamel (1995) as *Comparative Stylistics of French and English: A Methodology for Translation*. John Benjamins, Amsterdam.
- Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu. 2018. **Sentence simplification with memory-augmented neural networks**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 79–85, New Orleans, Louisiana, USA.
- Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. **Problems in current text simplification research: New data can help**. *Transactions of the Association for Computational Linguistics (TACL)*, 3:283–297.
- Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. **Optimizing statistical machine translation for text simplification**. *Transactions of the Association for Computational Linguistics (TACL)*, 4:401–415.
- Xingxing Zhang and Mirella Lapata. 2017. **Sentence simplification with deep reinforcement learning**. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 584–594, Copenhagen, Denmark.

Score	Grammaticality/Fluency (G)	Meaning preservation/Adequacy (M)	Simplicity (S)
5	The target sentence is fluent (native speaker level) and grammatically correct.	The target sentence adequately conveys the core meaning of the original sentence.	The target sentence is much simpler than the original sentence.
4	The target sentence is almost fluent (non-native speaker level) and grammatically correct	The target sentence mostly conveys the core meaning of the original sentence.	The target sentence is simpler than the original sentence.
3	The target sentence is less fluent with some ungrammatical parts, but understandable	The core meaning of the original text is not conveyed, but the information of the the original text is partially preserved.	The target sentence is as simple/difficult as the original sentence.
2	The target sentence is ungrammatical, but partially understandable.	The meaning of the target sentence is completely different from that of the original sentence.	The target sentence is more difficult than the original sentence.
1	The target sentence is completely unintelligible.	It is impossible to assess the meaning of the target sentence because of its unintelligibility.	It is impossible to assess the simplicity of the target sentence because of its unintelligibility.

Table 6: Guidelines for human evaluation.

A Guidelines

A.1 Guidelines for Human Evaluation

Table 6 lists the guidelines for human evaluations. We instructed annotators to consider document-level coherence when evaluating each sentence. Additionally, we instructed them to give an S score of 1 to the sentence that was given an M score of 1 or 2.

A.2 Annotation Guidelines for Simplification Strategies

Figures 3 and 4 show the guidelines, i.e., decision trees, for the annotation of simplification strategies. The procedures to build a decision tree were as follows: (1) through the classification of sample instances by trial and error, the first author created the prototype decision tree in a way that easier decisions can be made in earlier stages; (2) the third author validated the prototype by classifying sample instances using it; (3) based on the feedback from the third author, the first author refined the prototype.

L1 represents the category of strategy, and L2 represents the strategy. Note that S# and C# in the figures do not indicate the strategy numbers. In the annotation task described in §4.2, we used the Japanese versions.

B Examples of Strategies and Errors

Tables 7 and 8 list examples of the surface and content strategies. Table 9 lists examples of errors. These sentences were extracted from the in-

stances of human simplification,²⁹ which are based on Newsela articles (see §3.1 for detail).

C Additional Evaluation Results

C.1 Automatic Evaluation Scores

Table 10 shows the overall results of the automatic evaluation in terms of SARI, BLEU, and FKGL, all of which were measured by using EASSE (Alva-Manchego et al., 2019a)³⁰ at the corpus level.

For preparing the evaluation data, we manually aligned complex–simplest sentences for five Newsela articles. To properly implement SUC, we excluded sentences that do not have two preceding or following sentences and that consist of less than four words. We finally used 1,010 sentences for the automatic evaluation.

C.2 Overall Human Evaluation Scores

Table 11 shows the overall results of the human evaluation. The evaluation guidelines are presented in Appendix A.1.

²⁹An exception is (C10) Delete important information in Table 8, the example of which was extracted from the outputs of Transformer.

³⁰<https://github.com/feralvam/easse>

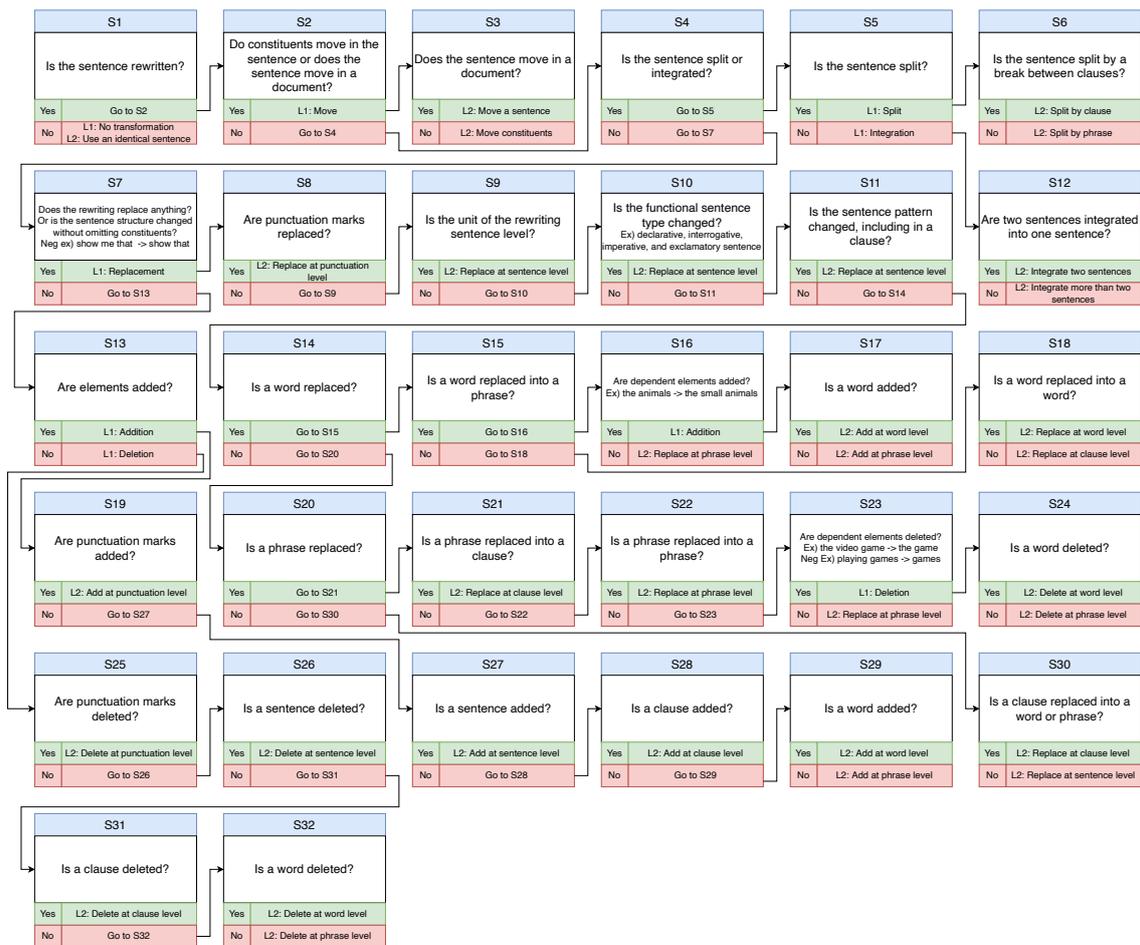


Figure 3: Annotation guidelines for surface strategies.

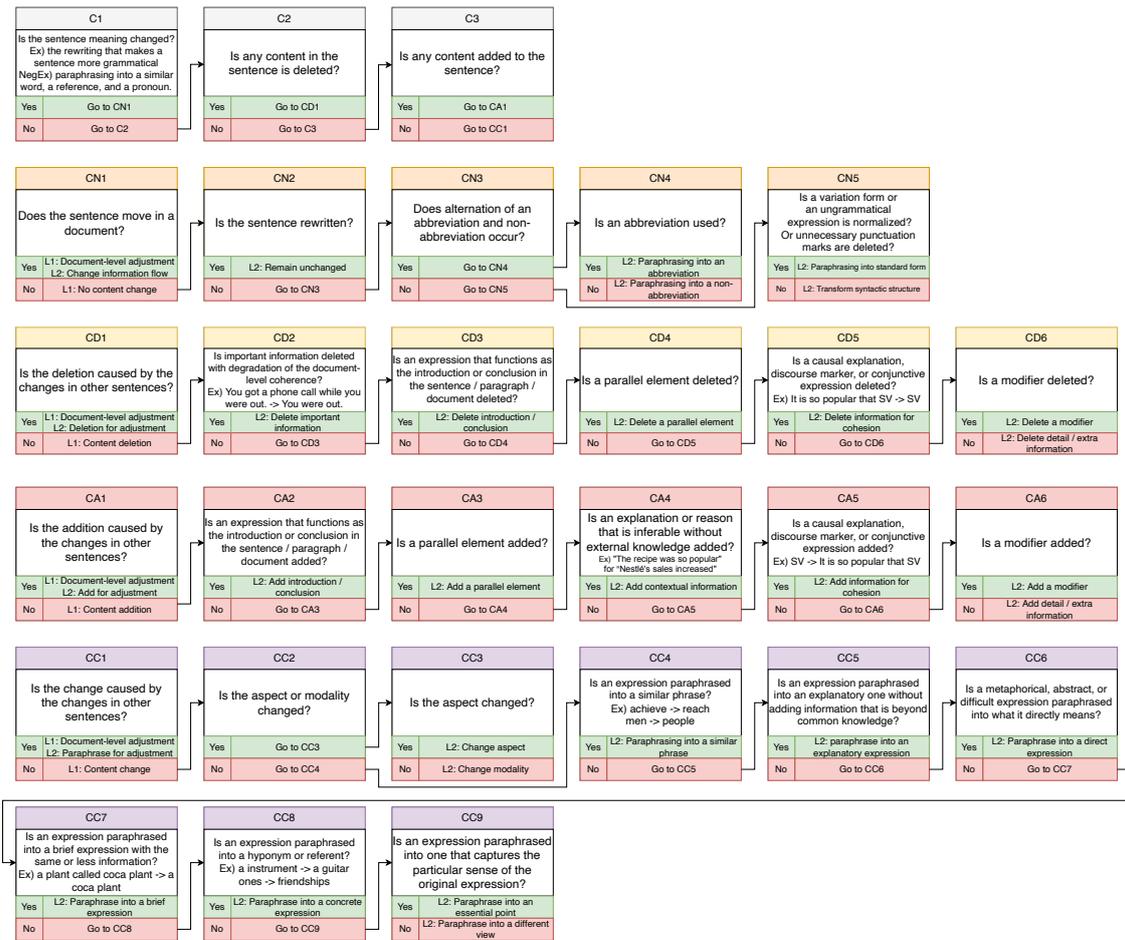


Figure 4: Annotation guidelines for content strategies.

Strategy	Example
Replacement	
(S1) Replace at punctuation level	Comp. ... 10,000 other neurons! Simp. ... 10,000 other neurons.
(S2) Replace at word level	Comp. ... make better surgeons. Simp. ... make good surgeons.
(S3) Replace at phrase level	Comp. ... about playing video games is friendship. Simp. ... about video games is friendship.
(S4) Replace at clause level	Comp. The persistence you use in games Simp. The persistence in games
(S5) Replace at sentence level	Comp. People who tried the syrup liked the taste. Simp. People liked the taste of the syrup.
Deletion	
(S6) Delete at punctuation level	Comp. ... "Toll House Chocolate Crunch Cookies." Simp. ... Toll House Chocolate Crunch Cookies.
(S7) Delete at word level	Comp. ... inside and outside the video game. Simp. ... inside and outside the game.
(S8) Delete at phrase level	Comp. Beating the final boss or another really good player Simp. Beating another really good player
(S9) Delete at clause level	Comp. He licked the ice that was stuck around it. Simp. He licked the ice.
(S10) Delete at sentence level	Comp. So does saving a teammate when they're down. Simp. ϕ
Addition	
(S11) Add at punctuation level	<i>This strategy does not exist in our collected instances.</i>
(S12) Add at word level	Comp. Remember games are Simp. Remember that games are
(S13) Add at phrase level	Comp. You have to be smart. Simp. In video games , you have to be smart.
(S14) Add at clause level	Comp. ... — the monkeys, apes, and gorillas — Simp. ... — the monkeys, apes, and gorillas that are most like human —
(S15) Add at sentence level	Comp. ϕ Simp. Scientists have studied video games.
Integration	
(S16) Integrate two sentences	Comp. Epperson pulled the stick. He licked the frozen juice. Simp. Epperson pulled the stick and licked the frozen juice.
(S17) Integrate more than two sentences	Comp. We got masks. We got gloves. We got all those hand wipes. They're everywhere. Simp. Now masks, gloves, hand wipes, and other material are everywhere
Splitting	
(S18) Split by phrase	Comp. Think about how boring it can be to play an easy game Simp. Think about playing an easy game. It can get boring.
(S19) Split by clause	Comp. Ruth Wakefield was an expert chef, and the inn became famous for its desserts. Simp. Ruth Wakefield was an expert chef. The inn became famous for its desserts.
Move	
(S20) Move constituents	Comp. It can also help fix broken ones. Simp. It also can help fix broken ones.
(S21) Move a sentence	<i>The complex and simplified sentences are identical.</i>
No transformation	
(S22) Use an identical sentence	<i>The complex and simplified sentences are identical.</i>

Table 7: Examples of surface strategies (Comp.: Complex sentence; Simp.: Simplified sentence).

Strategy	Example
No content change	
(C1) Transform syntactic structure	Comp. Some people think ... are waste of time or bad for you. Simp. Some people think ... are a waste of time. Some people think they are bad for you.
(C2) Paraphrase into an abbreviated form	Comp. ... seem like they've been around forever. Simp. ... seem like they have been around forever.
(C3) Paraphrase into a non-abbreviated form	Comp. Helping build Simp. Helping to build
(C4) Paraphrase into a standard form	Comp. But Simp. However,
(C5) Paraphrase into an identical sentence	<i>The complex and simplified sentences are identical.</i>
Content deletion	
(C6) Delete introduction / conclusion	Comp. Think about your favorite games. Simp. ϕ
(C7) Delete a parallel element	Comp. ... feel strong and popular. Simp. ... feel strong.
(C8) Delete information for cohesion	Comp. The treats were so popular that Epperson started Simp. Epperson started
(C9) Delete a modifier	Comp. He licked the ice that was stuck around it. Simp. He licked the ice.
(C10) Delete important information	Comp. Winkler teamed up with another scientist named Greg Bryant, a professor Simp. He is a professor [Transformer IND]
(C11) Delete detail / extra information	Comp. ... created the semi-sweet morsel, or chocolate chip. Simp. ... created chocolate chip.
Content Addition	
(C12) Add introduction / conclusion	Comp. ϕ Simp. Chocolate chip cookies seem like they've been around forever.
(C13) Add a parallel element	Comp. ... a different culture. Simp. ... a different culture or speak a different language.
(C14) Add contextual information	Comp. ϕ Simp. The company was selling more and more chocolate bars.
(C15) Add information for cohesion	Comp. People can recognize it, even if Simp. Laughter is so important to humans that people can recognize it, even if
(C16) Add a modifier	Comp. It can help fix broken ones. Simp. It can also help fix broken ones.
(C17) Add detail / extra information	Comp. ... to connect and bond. Simp. ... to connect and bond with others.
Content change	
(C18) Change aspect	Comp. You might do Simp. You might start doing
(C19) Change modality	Comp. They can teach Simp. They teach
(C20) Paraphrase into a similar phrase	Comp. ... Nestlé's sales soared. Simp. ... Nestlé's sales increased.
(C21) Paraphrase into an explanatory expression	Comp. ... to make a headache medicine. Simp. ... to make a medicine to fix headaches.
(C22) Paraphrase into a direct expression	Comp. ... to shred a guitar in real life. Simp. ... to play a guitar in real life.
(C23) Paraphrase into a brief expression	Comp. ... parts of a plant called the coca plant. Simp. ... parts of the coca plant.
(C24) Paraphrase into a concrete expression	Comp. It also can fix broken ones. Simp. It also can fix broken friendships.
(C25) Paraphrase into an essential point	Comp. It is one of many benefits Simp. It is one of many good things
(C26) Paraphrase into a different view	Comp. The cookies became so popular Simp. The recipe became so popular
Documet-level adjustment	
(C27) Change information flow	<i>The complex and simplified sentences are identical.</i>
(C28) Delete for adjustment	Comp. This makes you see that solving problems can be fun. Simp. Solving problems can be fun.
(C29) Add for adjustment	Comp. You also have to be smart. Simp. In video games, you also have to be smart.
(C30) Paraphrase for adjustment	Comp. It shows you that Simp. They shows you that

Table 8: Examples of content strategies (Comp.: Complex sentence; Simp.: Simplified sentence).

Error		Example
(E1) Inappropriate deletion	Input	When you think, feel, move, or use your senses, signals travel through this network.
	Output	When you think , feel , move , or use your senses . [DRESS IND]
(E2) Inappropriate addition	Input	It's how we tell friends that we find their joke funny, ...
	Output	it's how we tell friends that we find their joke funny funny , ... [Transformer IND]
(E3) Inappropriate paraphrase	Input	... but rats can make a very high-pitched trill.
	Output	... but rats can make a very high-pitched noise. [Transformer IND]
(E4) Non-sentence	Input	The animals that laugh the most include primates like monkeys, rats, and mammals that live in the ocean like dolphins.
	Output	humans, on the other hand, like monkeys, rats and mammals that live in the ocean like dolphins. [Transformer IND]

Table 9: Examples of errors.

	Transformer		DRESS		SUC	
	IND	OOD	IND	OOD	IND	OOD
SARI \uparrow	37.57	30.89	37.08	31.83	31.09	22.24
BLEU \uparrow	32.20	38.22	37.11	39.29	31.92	24.12
FKGL \downarrow	3.00	4.40	3.27	4.02	4.20	2.61

Table 10: Results of automatic evaluation. The upper/down arrow indicates that the higher/lower the score, the better the performance.

	Transformer		DRESS		SUC	
	IND	OOD	IND	OOD	IND	OOD
Grammaticality/Fluency	4.30	4.60	3.55	4.45	3.23	3.76
Meaning preservation/Adequacy	3.52	4.21	3.38	4.25	3.35	4.64
Simplicity	3.74	3.42	3.20	3.15	2.46	2.40

Table 11: Results of human evaluation using a five-point Likert scale.

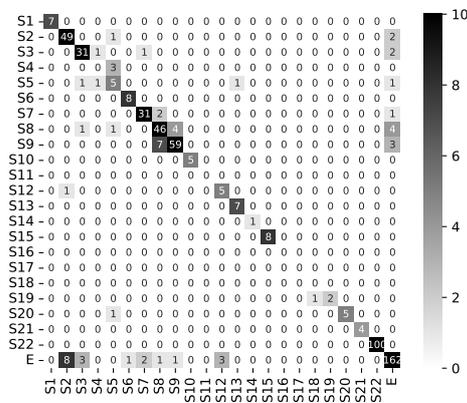


Figure 5: The distribution of annotations for surface strategies.

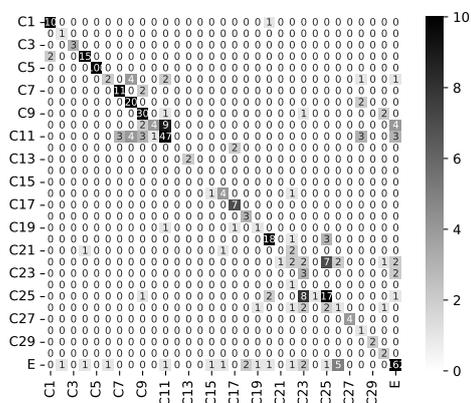


Figure 6: The distribution of annotations for content strategies.

D Distributions of Annotations

Figures 5, 6, and 7 show the distributions of annotations conducted by the two annotators in §4.2. S#, C#, and E# correspond to the surface strategy, content strategy, and error, respectively. When displaying the distributions for the strategies, we aggregated the annotations for the errors into one class and vice versa.



Figure 7: The distribution of annotations for errors.

Bridging the Gap between Pre-Training and Fine-Tuning for Commonsense Generation *

Haoran Yang¹, Yan Wang², Piji Li², Wei Bi², Wai Lam¹, Chen Xu³

¹The Chinese University of Hong Kong

²Tencent AI Lab

³Beijing University of Technology

hryang@se.cuhk.edu.hk

Abstract

Commonsense generation aims to generate a plausible sentence containing all given unordered concept words. Previous methods focusing on this task usually directly concatenate these words as the input of a pre-trained language model (PLM). However, in PLMs' pre-training process, the inputs are often corrupted sentences with correct word order. This input distribution discrepancy between pre-training and fine-tuning makes the model difficult to fully utilize the knowledge of PLMs. In this paper, we propose a two-stage framework to alleviate this issue. Firstly, in pre-training stage, we design a new format of input to endow PLMs the ability to deal with masked sentences with incorrect word order. Secondly, during fine-tuning, we insert the special token [MASK] between two consecutive concept words to make the input distribution more similar to the input distribution in pre-training. We conduct extensive experiments and provide a thorough analysis to demonstrate the effectiveness of our proposed method. The code is available at <https://github.com/LHRYANG/CommonGen>.

1 Introduction

To investigate machines' ability of generating logical sentences, Lin et al. (2020) propose the Commonsense Generation task. Given a set of concept words, this task is designed to generate a sentence which not only contains the given concepts but also can correctly describe the relations between concepts. An example is shown in Table 1.

Existing methods employ the Pre-trained Language Models (PLMs) such as BART (Lewis et al., 2020), GPT-2 (Radford et al., 2019) as the backbone to solve this problem. They (Liu et al., 2021; Fan et al., 2020; Wang et al., 2021; Li et al., 2021) usually take the concatenated concepts words as the inputs. However, such processing of inputs

concept words	{wear, player, field, jersey}
references	The player will wear a jersey while on the field. A soccer player wears a jersey on the field. ...
output of our model	football player wears a jersey on the field.

Table 1: An example of Commonsense Generation task

causes a huge gap between pre-training and fine-tuning. Specifically, these concept words are unordered which means the order of the input words is inconsistent with the order of these words in the references. It seems incompatible to PLMs pre-trained with ordered words (For BART (Lewis et al., 2020), sentence permutation is adopted, nevertheless, the word order within a sentence remains correct.). As studied by Zhao et al. (2022) and Ou et al. (2022), the word order of inputs can hinder the exploitation of knowledge existing in PLMs. Moreover, even if the word order of inputs is correct, for some LMs (e.g., BART (Lewis et al., 2020), T5 (Raffel et al., 2020)), the inputs are masked sentences during pre-training, while in commonsense generation task, the inputs are unconnected word sequences. This kind of discrepancy also degrades the models' performance.

In this paper, we propose a two-stage framework to bridge the gap between pre-training and fine-tuning for this task. Specifically, we firstly propose to introduce a domain-specific pre-training stage using the tasks' training dataset. The pre-training objective is designed to recover original sentences given the *masked* and *shuffled* sentences. Therefore, the PLMs' ability of reasoning out new concepts or relations (mask operation) and processing order-agnostic inputs (shuffle operation) is enhanced. Secondly, in downstream task fine-tuning, we insert the special token [MASK] between two consecutive concept words. This makes the input distribution more similar to the distribution in pre-training. The experimental results shows

*This work was done during an internship at Tencent.

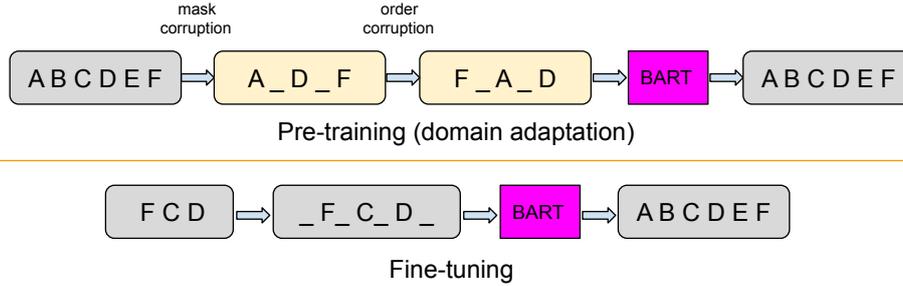


Figure 1: An overview of our model. _ represents the [MASK] token.

that our proposed model can significantly improve the performance of the commonsense generation task. We also conduct experiments to show that our model is superior than baselines in terms of continual learning and few-shot scenarios.

2 Model

We propose a two-stage training framework as shown in Figure 1. We firstly continually pre-train the BART with a newly designed input format. Secondly, we fine-tune the model whose inputs are inserted with the special token [MASK].

Formally, given a concept word set $\mathbf{x} = \{x_1, x_2, \dots, x_n\} \in X$ (n can be different for different inputs), the task aims to generate a fluent, plausible and grammatically correct sentence $\mathbf{y} = (y_1, y_2, \dots, y_m) \in Y$ containing all the words in \mathbf{x} .

2.1 Domain-Specific Pre-training

Continually pre-training the PLMs on the target domain is beneficial to improving the performance of the target task consistently (Gururangan et al., 2020). We adopt this idea and moreover, we design a new sentence corruption strategy considering that the input words order in target task is shuffled. Below is the procedure for constructing the corrupted inputs for each sentence $\mathbf{y} \in Y$ in training dataset:

1. Randomly select a subset of words in \mathbf{y} and each word is selected with a probability p which is also called the mask probability.
2. Replace the selected words with the special token [MASK]. It should be noted that multiple consecutive [MASK] tokens are merged to one [MASK] token. This allows the PLMs to predict a span (multiple words) based on one [MASK] token, which is more similar to the commonsense generation task as we will see below.

3. The unmasked words are shuffled while the positions of the [MASK] tokens remain unchanged. The corrupted input is denoted by $\tilde{\mathbf{y}}$.

An example of the above process is shown in the upper part of Figure 1. We usually choose a large value for the probability p instead of 15% used by BERT (Devlin et al., 2019). We will study the effect of p (0.5 in our experiment) in Section 3.2. Since a part of concept words and non-concept words are masked, this pre-training process can also enhance PLMs’ ability of reasoning out unseen concepts and relations between concepts.

Finally, the pre-training loss function is:

$$\mathcal{L}(\theta) = -\frac{1}{|Y|} \sum_{\mathbf{y} \in Y} \log\left(\prod_{i=1}^m P(y_i | y_{<i}, \tilde{\mathbf{y}}; \theta)\right) \quad (1)$$

2.2 Fine-tuning

Although the domain-specific pre-training can adapt the PLMs to the target domain and alleviate the problem related to word order, the inputs during fine-tuning are still a list of words while in pre-training for many LMs, the inputs are corrupted sentences with [MASK] tokens. Chada and Natarajan (2021) have shown that aligning the input distribution between pre-training and fine-tuning can boost the few-shot performance on QA tasks. Armed with such finding, we transform the inputs by inserting [MASK] tokens. Formally, given an input $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$, we transform \mathbf{x} to ¹:

$$[\text{MASK}], x_1, \dots, [\text{MASK}], x_i, [\text{MASK}], \dots, x_n, [\text{MASK}]$$

Then, we input the transformed \mathbf{x} to the PLM to predict the target \mathbf{y} . Through this way, the input distribution is more similar to that in pre-training (especially domain-specific pre-training). This input format is similar to the text infilling task (Donahue et al., 2020), the main differences are that

¹We send the transformed \mathbf{x} to the tokenizer so that [CLS] and [EOS] will also be added.

Model \ Metrics	ROUGE-2/L		BLEU-3/4		METEOR	CIDEr	SPICE
GPT-2	17.18	39.28	30.70	21.10	26.20	12.15	25.90
UniLM	21.48	43.87	38.30	27.70	29.70	14.85	30.20
T5	22.01	42.97	39.00	28.60	30.10	14.96	31.60
BART	22.23	41.98	36.30	26.30	30.90	13.92	30.60
KG-BART	23.38	44.54	42.10	30.90	32.40	16.83	32.70
NeuroLogic	-	44.70	41.3	30.60	31.00	15.90	31.10
CALM	-	-	-	29.50	31.90	15.61	33.20
EKI-out	24.36	45.42	42.90	32.10	32.00	16.80	32.50
Ours	24.17	44.89	43.31	32.49	32.50	17.10	32.81

Table 2: Automatic Evaluation Results.

the words in commonsense generation task are unordered and masked words also account for a large proportion of sentences.

3 Experiments

3.1 Experimental Settings

Dataset We use the CommonGen dataset collected by Lin et al. (2020). The dataset contains 67389/4018/6042 training/development/testing samples with 32651/993/1497 different concept sets (one concept set has multiple references.). For evaluation metrics, we use BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2016). We also report human evaluation score and coverage score. However, due to the space constraint, regarding these two scores, please refer to Appendix ?? for more details.

Baselines We compare our model with several baselines. For PLMs, we choose GPT-2 (Radford et al., 2019), UniLM (Dong et al., 2019), T5 (Rafael et al., 2020), BART (Lewis et al., 2020). We also compare our model with (1) KG-BART (Liu et al., 2021) which incorporates the knowledge graph to BART. (2) NeuroLogic (Lu et al., 2021) which controls the decoding stage to enforce the satisfaction of the given lexical constraints. (3) CALM (Zhou et al., 2021) which designs several self-supervised tasks to obtain a concept-aware language model. (4) EKI-out (Fan et al., 2020) which augments inputs with retrieved sentences from *out-of-domain* corpus. Generally, EKI-out is stronger than other baselines due to the high informativeness of Wikipedia.

Implementation Details We adopt BART-large as the generation model. The max length of x and

y are set to 48 and 128 respectively. The batch size is set to 32. For Domain-Specific Pre-training, the mask probability p is set to 0.5. The number of training epochs is 10. We use AdamW (Loshchilov and Hutter, 2019) with learning rate $1e-7$ to optimize the model. For fine-tuning, the model is optimized using AdamW with an initial learning rate of $2e-5$. We also employ linear warmup with steps 10000. We save the model with the highest Rouge-L score on development set for testing.

3.2 Results

Main Results As summarized in Table 2, Ours can generally achieve better performance than all the baselines on BLEU, METEOR, CIDEr. On ROUGE, Ours outperforms most of the baselines except EKI-out which facilitates the Wikipedia as the external corpus. On SPICE, Ours is superior than most of the baselines except CALM.

Ablative Results We conduct ablation study with three variants. The results are shown in Table 4. We can see that the performance of -mask (Ours without adding [MASK] during fine-tuning) and -pretraining (Ours without pretraining) are inferior than Ours. -both (Ours with neither) obtains the worst performance. We can also observe that adding mask and adding pretraining have similar degree of improvement compared to -both. Moreover, since there are numerous ways to insert [MASK] to inputs (different positions or different numbers), we compare our model with Random Mask: during pre-training and fine-tuning, *one* mask token is randomly inserted into the corrupted inputs. We can see from Table 4 that Ours outperforms Random Mask. Moreover, we provide some generated examples in Appendix B.

Model	Training on	Evaluation on	ROUGE-L	BLEU-4	METEOR	CIDEr	SPICE
Ours	Size = 3	Size = 3	45.25	18.74	24.06	14.64	34.92
	Size = 4	Size = 3	46.31(+1.06)	19.33(+0.59)	25.76(+1.70)	15.47(+0.83)	36.89(+1.97)
		Size = 4	44.97	31.02	31.25	16.14	31.68
	Size = 5	Size = 3	45.62(-0.69)	19.38(+0.05)	25.50(-0.26)	15.26(-0.21)	36.51(-0.38)
		Size = 4	45.27(+0.30)	32.00(+0.98)	31.63(+0.38)	16.49(+0.35)	31.46(-0.22)
		Size = 5	43.53	30.98	30.93	16.12	31.01
-mask	Size = 3	Size = 3	44.80	17.89	24.24	14.68	34.28
	Size = 4	Size = 3	45.52(+0.72)	17.46(-0.43)	24.71(+0.47)	14.58(-0.10)	35.38(+1.10)
		Size = 4	44.29	31.53	30.98	16.24	32.04
	Size = 5	Size = 3	45.36(-0.16)	17.75(+0.29)	24.77(+0.06)	14.69(+0.11)	35.88(+0.50)
		Size = 4	44.49(+0.20)	30.99(-0.54)	30.88(-0.10)	16.09(-0.15)	31.17(-0.87)
		Size = 5	42.69	29.16	29.63	15.11	30.05

Table 3: Continual Learning Results. The rows with the same color represents the same domain we evaluate the model on. The red number in parentheses is the improvement compared with the previous time step on the same domain. For example, (+1.06) = 46.31 − 45.25, (-0.69) = 45.62 − 46.31, (+0.30) = 45.27 − 44.97.

Model	ROUGE-L	BLEU-4	METEOR	CIDEr	SPICE
Ours	44.89	32.49	32.50	17.10	32.81
-mask	44.67	31.66	32.09	16.51	32.11
-pretraining	44.35	31.60	31.87	16.57	32.33
-both	43.56	29.61	30.87	15.61	30.93
Random Mask	44.43	31.64	32.23	16.69	32.36

Table 4: Variant Analysis Results.

Effects of Hyperparameter p We investigate the effects of the mask probability p . As presented in Table 6, the performance is the best when p equals 0.5. The reason may be that if p is too large, it is hardly possible to recover corrupted sentences during pre-training. However, if p is too small, most of the masked tokens are not concept words, thus the pre-trained model cannot learn the relations between concepts.

Human Evaluation & Coverage To provide more perspective of the generation quality, we report the human evaluation score and coverage score. For human evaluation, we randomly select 30 sentences and each sentence is given a score ranging from one to five to assess the holistic quality. We report the average value of two annotators. The concept coverage score is the average percentage of input concepts that are present in lemmatized outputs. The results are shown in Table 5. We can see that Ours achieves the highest human evaluation score and coverage score and Ours-w/o-pretraining achieves a slightly better performance than Ours-w/o-mask, indicating that inserting [MASK] to the input is more important than adding the pretraining stage.

3.3 Few-Shot Scenario

We investigate the performance of our model under few-shot scenario. We randomly select $n \in$

Model	Ours	Ours-w/o-mask	Ours-w/o-pretraining	Ours-w/o-both
human score	4.534	4.367	4.467	4.084
coverage	97.48	96.03	96.07	93.05

Table 5: Human Evaluation Score and Coverage Score

p	ROUGE-L	BLEU-4	METEOR	CIDEr	SPICE
0.2	44.36	31.21	31.16	16.48	32.33
0.4	44.32	30.99	32.05	16.64	32.48
0.5	44.89	32.49	32.50	17.10	32.81
0.6	44.37	31.95	32.64	16.91	32.72
0.8	44.58	32.25	32.38	16.83	31.90

Table 6: Effects of p .

{16, 32, 64} samples from original training dataset as the new training dataset and the testing dataset remains unchanged. The learning rate is set to $2e-5$. Table 7 shows the results. We can see that inserting [MASK] to the inputs can significantly boost the performance on all the metrics. Combined with the result in Table 2, we can conclude that inserting [MASK] to the inputs is beneficial to the performance on both full-data and few-shot settings.

3.4 Continual Learning Scenario

We investigate the performance of our model under continual learning scenario (Biesialska et al., 2020). We regard concept sets with the same length as a domain. The details of the dataset are described in Appendix A. The model is trained sequentially from the domain with length 3 to the domain with length 5. After the model is trained on a new domain, we also evaluate it on previous domains to measure the backward transfer degree. Backward transfer means that learning a new task may hurt (negative backward transfer) or improve (positive backward transfer) the performance of previously

n	model	ROUGE-L	BLEU-4	METEOR	CIDEr	SPICE
16	Ours	35.04	16.22	21.98	9.02	21.93
	-pretraining	33.49	12.25	19.29	7.46	20.42
	-mask	32.33	7.5	19.23	6.16	17.24
	-both	31.74	6.75	19.83	6.18	16.30
32	Ours	35.66	19.72	23.20	10.00	21.43
	-pretraining	35.92	16.21	21.35	8.88	20.78
	-mask	33.98	15.17	21.34	8.63	18.36
	-both	33.15	11.73	19.36	7.51	19.13
64	Ours	38.94	22.15	25.96	12.31	26.64
	-pretraining	38.17	21.17	25.48	11.6	26.27
	-mask	35.75	18.79	24.73	10.73	24.00
	-both	35.04	15.43	22.58	9.18	21.23

Table 7: Few-Shot Setting.

learned tasks (Lopez-Paz and Ranzato, 2017). The results are shown in Table 3. We can see that Ours generally obtains better performance than -mask. Also, we can see that our model achieves a larger positive backward transfer and a smaller negative backward transfer (forget less) than -mask. For example, ROUGE-L of the domain with concept set size 3 is changed from 45.25 to 46.31 (improved by 1.06) after the model is trained on the domain with concept size 4 for our model. While for -mask, the improvement is only 0.72. Therefore, we can conclude that bridging such a gap is effective under continual learning setting.

4 Conclusion

We study the gap issue between pre-training and fine-tuning for commonsense generation task. We propose a two-stage training framework which is composed of a domain-specific pre-training stage and a fine-tuning stage. Pre-training stage aims to recover the masked and shuffled sentences which could enhance the models’ ability of processing unordered inputs and reasoning out the relations and concepts. Inserting [MASK] to the inputs during fine-tuning have also been demonstrated very useful. Experimental results show that our model is superior than many baselines, especially under few-shot setting.

Acknowledgement

The work described in this paper is supported by Tencent AI Lab Rhino-Bird Gift Fund (YD4200722).

Limitations

In this work, we study the gap between pre-training and fine-tuning for commonsense generation task. Despite the promising experimental results, there are still several limitations of our work:

1. The order issue is still not fully solved since the original pre-training stage uses ordered sentences. Our proposed domain-specific training stage can only alleviate this issue instead of completely solving it.
2. During fine-tuning, the optimal positions and an optimal number of the [MASK] tokens are not well solved.

References

- Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*.
- Satanjeev Banerjee and Alon Lavie. 2005. **METEOR: An automatic metric for MT evaluation with improved correlation with human judgments**. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. **Continual lifelong learning in natural language processing: A survey**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Rakesh Chada and Pradeep Natarajan. 2021. **Few-shotQA: A simple framework for few-shot learning of question answering tasks using pre-trained text-to-text models**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6081–6090, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Chris Donahue, Mina Lee, and Percy Liang. 2020. **Enabling language models to fill in the blanks**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501, Online. Association for Computational Linguistics.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.
- Zhihao Fan, Yeyun Gong, Zhongyu Wei, Siyuan Wang, Yameng Huang, Jian Jiao, Xuanjing Huang, Nan Duan, and Ruofei Zhang. 2020. **An enhanced knowledge injection model for commonsense generation**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2014–2025, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. **Don't stop pretraining: Adapt language models to domains and tasks**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. **BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Haonan Li, Yeyun Gong, Jian Jiao, Ruofei Zhang, Timothy Baldwin, and Nan Duan. 2021. **KFCNet: Knowledge filtering and contrastive learning for generative commonsense reasoning**. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2918–2928, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. **CommonGen: A constrained text generation challenge for generative commonsense reasoning**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries.
- Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. **Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6418–6425.
- David Lopez-Paz and Marc' Aurelio Ranzato. 2017. **Gradient episodic memory for continual learning**. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Ilya Loshchilov and Frank Hutter. 2019. **Decoupled weight decay regularization**. In *International Conference on Learning Representations*.
- Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. **NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints**. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299, Online. Association for Computational Linguistics.
- Zebin Ou, Meishan Zhang, and Yue Zhang. 2022. **On the role of pre-trained language models in word ordering: A case study with bart**.

- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. [Cider: Consensus-based image description evaluation](#). In *CVPR*, pages 4566–4575. IEEE Computer Society.
- Han Wang, Yang Liu, Chenguang Zhu, Linjun Shou, Ming Gong, Yichong Xu, and Michael Zeng. 2021. [Retrieval enhanced model for commonsense generation](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3056–3062, Online. Association for Computational Linguistics.
- Chao Zhao, Faeze Brahman, Tenghao Huang, and Snigdha Chaturvedi. 2022. [Revisiting generative commonsense reasoning: A pre-ordering approach](#).
- Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, and Xiang Ren. 2021. [Pre-training text-to-text transformers for concept-centric common sense](#). In *International Conference on Learning Representations*.

A Continual Learning

We introduce how to construct the dataset used for continual learning. Table 8 shows the distribution of the original dataset. Since there is no testing instances whose concept set size is 3. We randomly sample a number of instances with concept size 3 from the training dataset. Also, since the dataset is unbalanced (the number of instances belonging to the domain with concept size 3 is far larger than that in other domains.) We re-sample the instances to make the dataset more balanced. The statistic of the continual learning setting dataset is shown in Table 9.

Statistics	Train	Dev	Test
Sentences	67,389	4,018	6,042
Concept-Sets	32,651	993	1,497
-Size = 3	25,020	493	-
-Size = 4	4,240	250	747
-Size = 5	3,391	250	750

Table 8: Statistics of Original Dataset.

Statistics	Train	Dev	Test
-Size = 3	5,867	1,819	2,170
-Size = 4	5,352	1,137	2,993
-Size = 5	3,436	1,062	3,049

Table 9: Statistics of Continual Learning Dataset.

B Generated Examples

We list some examples generated by our proposed model and ablative models, which are shown in Table 10.

concept words	<i>{sheep, wool, shave, hold}</i>
Ours	<i>A man is holding a sheep and shaving its wool.</i>
Ours-w/o-pretraining	<i>A woman holds a sheep and shaves its wool.</i>
Our-w/o-mask	<i>A man is holding a sheep and shaving it with wool.</i>
Our-w/o-both	<i>sheep holding their wool in their beaks as they shave.</i>
concept words	<i>{stand, fence, feed, goat}</i>
Ours	<i>A goat stands at the fence to be fed.</i>
Ours-w/o-pretraining	<i>goats standing next to a fence to feed.</i>
Our-w/o-mask	<i>A goat standing next to a fence to feed.</i>
Our-w/o-both	<i>A goat stands at the fence to feed a goat.</i>
concept words	<i>{hold, bag, popsicle, eat, chip}</i>
Ours	<i>A boy is eating a popsicle while holding a bag of chips.</i>
Ours-w/o-pretraining	<i>A girl is holding a bag of chips and eating a popsicle.</i>
Our-w/o-mask	<i>A man holding a bag of chips and a popsicle to eat.</i>
Our-w/o-both	<i>A man holding a bag of chips and a popsicle eats a chip.</i>

Table 10: Generated Examples

LED: A Dataset for Life Event Extraction from Dialogs

Yi-Pei Chen¹, An-Zi Yen², Hen-Hsen Huang³, Hideki Nakayama¹, Hsin-Hsi Chen⁴

¹ The University of Tokyo, Japan

² National Yang Ming Chiao Tung University, Taiwan

³ Academia Sinica, Taiwan ⁴ National Taiwan University, Taiwan

¹ypc@g.ecc.u-tokyo.ac.jp, ²azyen@nycu.edu.tw, ³hhhuang@iis.sinica.edu.tw

¹nakayama@ci.i.u-tokyo.ac.jp, ⁴hhchen@ntu.edu.tw

Abstract

Lifelogging has gained more attention due to its wide applications, such as personalized recommendations or memory assistance. The issues of collecting and extracting personal life events have emerged. People often share their life experiences with others through conversations. However, extracting life events from conversations is rarely explored. In this paper, we present Life Event Dialog, a dataset containing fine-grained life event annotations on conversational data. In addition, we initiate a novel conversational life event extraction task and differentiate the task from the public event extraction or the life event extraction from other sources like microblogs. We explore three information extraction (IE) frameworks to address the conversational life event extraction task: OpenIE, relation extraction, and event extraction. A comprehensive empirical analysis of the three baselines is established. The results suggest that the current event extraction model still struggles with extracting life events from human daily conversations. Our proposed life event dialog dataset and in-depth analysis of IE frameworks will facilitate future research on life event extraction from conversations.

1 Introduction

Daily conversation, as a means of communication and switching information, is full of personal information, including personal background, interests and hobbies, connections to other people, and various life events. Mining life events lets us better understand a person. The extracted life events can be used to construct the personal knowledge base and benefit a variety of downstream tasks, such as lifestyle understanding (Doherty et al., 2011) and memory assistance (Rahman et al., 2018).

Previous research on life event extraction mainly focuses on life events from microblogs or social

media platforms such as Twitter (Li et al., 2014; Yen et al., 2018, 2019). However, these events from a given fixed passage are static. In contrast, an event mentioned in a conversation might change its status dynamically throughout the chat. Besides, conversations allow participants to interact with each other and gather the information which stimulates participants' interests, revealing people's general interests in different aspects of information about a life event and expanding additional event information. For example, when a person talks about a travel event only with the destination mentioned, the other interlocutor might ask additional information about who they are traveling with, how much the trip cost, and the period and timing of the travel. Nevertheless, life event extraction from conversations is rarely explored and existing works only detect course or ambiguous event types (Eisenberg and Sheriff, 2020; Kao et al., 2021). The participants and status of events are not recognized, preventing more fine-grained life events analysis and limiting the applications.

We present Life Event Dialog (LED), a dataset with refined life event annotations in English.¹ We define life events as activities in a person's daily life. Following previous works, our life event definition is verb-centered. For each event, we annotate three levels of event type from fine-grained to coarse: *Verb*, *Class*, and *Frame*. Unlike formal writing and social network posts, dialogue is usually in a more flexible and more abstruse style, where the event type is often omitted. For example, "S1: Can I get you some coffee? S2: De-caff." indicates an "order" event, where the verb "order" does not appear in the dialogue. Therefore, we also introduce *Explicitness* of an event. When the event type cannot be extracted from the dialogue, we manually

¹<https://github.com/ntunlp/LifeEventDialog>

assign a verb to denote the activity and label the event as an implicit event. Besides event types, we annotate *Subject* and *Object* of each event as event participants. Furthermore, based on the interactive nature of a conversation, more detailed event information is likely to be revealed as the conversation continues. People might ask follow-up questions or clarifications in a response that specify the status or attributes of a known event. We consider the new supplemental information as the event status change instead of a new event. To be more specific, we record three aspects of event status: *Polarity*, *Modality*, and *Time*. These detailed annotations provide more comprehensive information about life events and allow us to track the dynamic event status changes throughout the conversation.

Moving forward from previous research on classifying the types of life events, we introduce the Conversational Life Event Extraction task, which classifies the event type and identifies event participants simultaneously from conversations. Classifying the event type of a life event is much harder than conventional public event extraction because of the high diversity of life events. The form of conversation further adds up to the difficulty of this task. For instance, event participants are challenging to identify because they are often in free form, and mentions of the same entity are easily changed throughout the dialogue. Due to the uniqueness of conversational life event extraction, there has not been a model that specifically tackles this problem.

In this paper, we examine multiple information extraction (IE) frameworks, including OpenIE, event extraction (EE), and end-to-end relation extraction (RE) models, for this task. Experimental results show that the existing information extraction models, even the recent models on top of their tasks, still perform poorly in extracting life events from conversations. We analyze the strengths and limitations of each model, and urge the development of a better model for Conversational Life Event Extraction. The contributions of this work are threefold as follows:

- We introduce Life Event Dialog (LED) dataset, the first dataset annotated with fine-grained life events in conversations.
- We propose a novel task of Conversational Life Event Extraction, stepping forward the event type classification task from previous works.

- We explore several IE frameworks on the conversational life event extraction task and offer a thorough analysis of the baselines.

2 Related Work

2.1 Life Event Extraction

With the rise of social media platforms, people increasingly document their lives online. A large amount of personal data is beneficial for applying to lifelogging tasks. Most life event research collects data from Twitter and contains limited event types. Li et al. (2014) gathered tweets with congratulations or condolences replies and proposed a pipeline system to extract 42 major life events like “getting a job”, “graduation”, or “marriage”. Yen et al. (2018) constructed a multi-labeled Chinese tweets dataset with 12 life event types and proposed multiple LSTM models for life events extraction. Yen et al. (2019) built a life event corpus on Chinese tweets focusing on general life events such as dining or visiting a local place, transforming the extracted events into personal knowledge-based facts. Other than social media posts, the NTCIR14 Lifelog dataset (Gurrin et al., 2019) consists of multimodal lifelogs of images and their metadata. They assorted daily activities into 16 categories, but targeted visual lifelog retrieval instead of life event extraction. Although all concentrate on life events, Conversational Life Event Extraction is distinct from social media or multimodal sources.

2.2 Conversational Event Extraction

Li et al. (2021) designed a task-oriented dialogue system especially for the event extraction task, which differs from our goal of extracting life events from an existing open-domain conversation. Imani (2014) studied the performance of OpenIE systems on conversations collected from reviews, emails, meetings, blogs, forums, and Twitter. Besides the small data size of only a hundred sentences and the dataset not being publically available, their dataset lacks of auxiliary event information such as the event status.

2.3 Life Event Extraction from Conversation

Works by Eisenberg and Sherif (2020) and Kao et al. (2021) are the most related works to ours. Eisenberg and Sherif (2020) collected conversations from a podcast and classified event features by SVM. Their event annotations only include the event tokens and lack other event information. Kao

D	Dialogue	i	Event Types	Participants	P	M	T
1	S1: Bill , I must tell you the truth. You <u>failed</u> the English exam again.	1	[Explicit] Verb: failed Class: fail Frame: Success Act	[S] You [O] English exam	+	○	before
	S2: Ah? Really? That stinks!						
	S1: Haha. April Fool’s! Did you forget what day it is today?				-	○	before
	S1: Excuse me. I would like to <u>purchase</u> some travelers’ checks.	1	[Explicit] Verb: purchase Class: purchase Frame: Buy	[S] I [O] some travelers’ checks	+	△	now
	S2: Sure. How much do you want?						
2	S1: \$5000 and I want them all in fifties.	2	[Explicit] Verb: purchase Class: purchase Frame: Buy	[S] you [O] \$5000	+	○	now
	S2: OK, here you are. Please <u>sign</u> your name here.	3	[Implicit] Verb: give Class: give Frame: Giving	[S] S2 [O] S1 [O] \$5000	+	○	now
	S1: Thank you .	4	[Explicit] Verb: sign Class: sign Frame: Text Creation	[S] S1 [O] your name	+	△	after

Table 1: Two example dialogues with 1 and 4 events, respectively. D: Dialogue ID, i: Event ID. We display the coreference cluster in **red** for S1 and in **blue** for S2. *Verb* of explicit events (extractive) are underlined. For each event, we show the event types, participants, and status (*Polarity* (P), *Modality* (M), and *Time* (T)). +: positive event, -: negative event, ○: actual event, △: hypothetical event.

et al. (2021) also constructed a dataset from Daily-Dialog (Li et al., 2017), but they only annotated the frame name for each event. Both works also aimed at extracting personal life events from conversations, yet their proposed datasets only contain plain event annotations. In contrast, our LED dataset has more comprehensive annotations, including participants, status, event category, and the coreference clusters of participants.

3 Life Event Dialog

In this paper, we define life events as daily life activities, personal habits, life experiences, or personal information of the interlocutors or related people. On the other hand, personal feelings or preference, public issues, and general knowledge are not considered life events in our dataset.

3.1 Event Schema

Event Type: We define three granularities of event type: *Verb*, *Class*, and *Frame*. We also labeled the *Explicitness* based on whether *Verb* can be extracted from the dialogue.

- *Explicitness* (E) is determined by whether a verb exists in the dialogue that triggers an event. If no explicit verb exists in the dialogue, but an event is recognized and labeled

by annotators, we consider it as an implicit event. See Dialogue 2 Event 3 in Table 1 for an example.

- *Verb* is a verb event trigger, which might be a span extracted from the dialogue (explicit event) or abstractly written by annotators (implicit event).
- *Class* is the fine-grained event type determined by the lemma of *Verb*.
- *Frame* is the coarse event type selected from FrameNet (Fillmore et al., 2002) by annotators. This event type is also used in previous works (Yen et al., 2019; Eisenberg and Sheriff, 2020; Kao et al., 2021; Wang et al., 2020).

Note that *Frame* and *Class* are not one-to-one mappings. For example, *Class* “get” could belong to *Frame* “Possession”, “Receiving”, or “Giving”. In LED, each *Class* belongs to 1.25 *Frame* on average. **Participant:** We label the span for *Subject* (S) and *Object* (O). In a conversation, the same S/O entity might appear recurrently in different mentions, therefore, we also include the coreference cluster ID for S/O as their entity ID.

Status: Three event properties that might change dynamically throughout the dialogue are recorded,

	# Dialogs	U	Evt	Unique Evt
Train	858	3,823	5,529	1,856
Valid	75	349	593	179
Test	70	313	426	151
Total	1,003	4,485	6,548	2,186

Table 2: Dataset Statistics. The number of utterances (U) is the number of training instances (a training instance is an utterance with its dialogue history), and the number of events (Evt) is the cumulative number of events of a training instance. Also, we consider events with same event types and participants as the same event (Unique Evt), which might have different event status.

including *Polarity*, *Modality*, and *Time*.

- *Polarity* (P) is a binary class of whether an event happens (positive) or does not happen (negative). In some conversations, a life event is specifically expressed in a negative form. Given an utterance, “You did not invite me to the party.” We consider the negativity in this sentence as a strong indication of a particular event rather than a random event that doesn’t happen. Moreover, an event might change its *Polarity* as the conversation continues. As shown in Dialogue 1 Event 1 in Table 1, (You, failed, English exam again) is a positive event in the first two utterances, but after the speaker S1 says it’s an April Fool’s joke, *Polarity* becomes negative. Therefore, we especially mark the negative event status to keep track of the polarity changes of a life event in the conversation.
- *Modality* (M) refers to whether an event has happened/is happening (actual), or is mentioned in the dialogue that it will happen in the future (hypothetical), as illustrated in Dialogue 2 Event 1. Note that an event is hypothetical only when indicated in an affirmative sentence and not in a question. For example, (We, have, meeting) in “We will have a meeting at 9 a.m. tomorrow.” is a hypothetical event, but (she, call, you) in “Can she call you back?” is not.
- *Time* (T) is labeled as one of “before”, “now”, “after”, “continuously”, or a specified time span if the time information is explicitly mentioned in the dialogue. *Time* might be related to *Modality*. For instance, one hypothetical

event might have *Time* “after”, waiting for confirmation. After the next utterance reply, the event status would become an actual event with time labeled “now”. Dialogue 2 Event 4 is an example that changes its status after the last turn is given.

The default event status is positive, actual, and happens at now.

3.2 Annotation Details

We recruited three annotators with a linguistic degree to annotate the data. The dialogue is augmented by one turn at each time, and annotators are asked to label life events for the whole conversation up to the given turns. To calculate the agreement, we sampled 40 dialogues and asked all annotators to annotate them. We calculate the agreement on the *Frame* of all positive and actual events in the last turn of each dialogue (the accumulated events in one dialogue). The total number of annotated events are 550. The annotation agreement is 0.81, measured by Krippendorff’s alpha (Krippendorff, 2011). For the disagreed cases, we conducted the majority vote or discussed with annotators to re-annotate the event. The annotation guideline and more annotation details are provided in Appendix A.

3.3 Dataset Construction

We sample 1,003 dialogues from the DailyDialog dataset (Li et al., 2017) as the material for life event annotation. DailyDialog is a multi-turn English dialogue dataset, which contains daily life conversations from various English learning websites. The conversations usually focus on a certain topic and under a certain situation, such as a customer finding some goods in a shop. We take the five most frequent topics, including Relationship (35%), Ordinary Life (28%), Work (20%), Tourism (9%), and Attitude & Emotion (8%), and annotate four to six utterances of each conversation. We include conversations with (73.5%) and without (26.5%) events to reflect the real world scenario that not all conversations contain life events. Overall, we annotate 2,186 unique life events (Unique Evt) from 4,485 utterances. Note that one training instance is an utterance (U) with its dialogue history, and the events of an instance (Evt) would be the cumulative events from the utterance and its dialogue history. The statistics of our dataset is shown in Table 2.

For every unique event, the event status might

Unique Event Types			Status Change		
<i>Verb</i>	<i>Class</i>	<i>Frame</i>	P	M	T
695	371	175	26	58	117

Table 3: The number of unique categories in each event type and the number of times when an event changes one of its status.

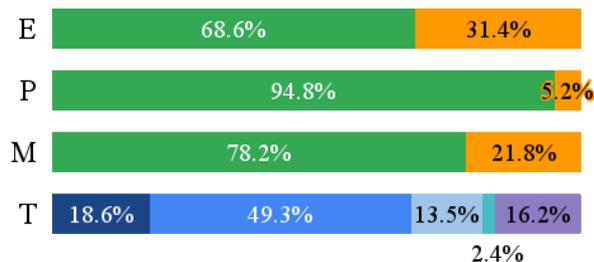


Figure 1: Statistics of *Explicitness* (E) and event status. Green and orange colors stand for explicit/implicit, positive/negative, and actual/hypothetical, for E, P, and M, respectively. Colors of T from left to right are “before”, “now”, “after”, “continuously”, and the specified time.

change throughout the conversation. We list the number of event status change for P, M, and T, as well as the number of unique event types for *Verb*, *Class*, and *Frame* in Table 3. The ratio of explicit vs. implicit, positive vs. negative, actual vs. hypothetical events, and the distribution of the T labels are shown in Figure 1.

4 Dataset Analysis

4.1 Life Events Distribution

We list the top five most frequent *Class* and *Frame* among 371 classes and 175 frames in Table 4, from which we can see that either *Class* or *Frame* is sparsely distributed. Even the most frequent *Class* accounts for only 3.9% of all, and the dominant *Frame* makes up only 6.1%. The majority event status change is the change of *Time*, which usually happens when people specify the event time. The top five implicit event classes are: “receive”, “hear”, “give”, “invite”, and “pay”. In contrast, the top explicit event classes are: “have”, “tell”, “go”, “see”, and “be”. Three classes (“go”, “hear”, and “bring”) are overlapped in top 10 explicit and implicit events classes.

4.2 Comparison with Event Extraction and Relation Extraction Benchmarks

Both event extraction (EE) and relation extraction (RE) aim to predict the event type and participant

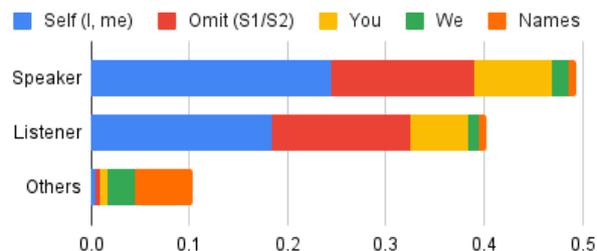


Figure 2: Subject analysis. When S is the speaker, the listener, or others, the mention of S usually belongs to one of the five categories: Self, We, You, Omit, Names.

information. For EE, each event has a event type (subtype) and argument roles. We regard *Frame* and *Class* in LED as the type and sub-types and map *S*, *O*, and event status (*Polarity*, *Modality*, and *Time*) as the argument roles. The RE output is a (head, relation, tail) triple. We consider (S, event type, O) in LED as the mapping of a RE triple. The major difference between the life events from our LED dataset and the public events from EE/RE benchmarks is the event domain and the distribution of event types. Life events in LED belong to a wide variety of categories that are sparsely distributed. In contrast, current EE and RE benchmarks are often from news reports and focus on certain limited event types. We compare two EE benchmarks (ACE2005 (Walker et al., 2006) and MAVEN (Wang et al., 2020)) and one RE benchmark (CONLL04 (Roth and Yih, 2004)) in Table 4, demonstrating the distinguishable event type discrepancy on domain and distribution.

Further, the arguments in EE benchmarks are often a single entity or the head word of a noun phrase, but we often want to keep the informative descriptions of life events, especially for objects. The average object length in LED is 2.95, which is 2.5 times of argument length in ACE2005. In addition, a quarter of life events are implicit events, which means 25% of the event trigger (*Verb*) cannot be found in the text input, whereas all event triggers and arguments are extractable from the given text in EE benchmarks.

4.3 Comparison with Life Event Datasets

LiveKB (Yen et al., 2019) is a large-scale life event dataset crawled from Chinese Twitter with an event schema similar to ours. The major difference between LiveKB and Life Event Dialog derives from the characteristics of a single-person narrative versus interactions between two people. In a tweet, the event subject is almost always the author of

LED (Frame)	%	LED (Class)	%	ACE2005	%	MAVEN	%	CONLL04	LiveKB
Statement	6.1	have	3.9	Attack	28.8	Action	46.9	kill	Perception
Perception	5.3	go	3.8	Transport	13.5	Change	27.5	work for	Presence
Motion	3.8	tell	3.5	Die	11.2	Scenario	13.4	organization based on	Using
Request	3.2	hear	2.8	Meet	5.2	Sentiment	6.4	live in	Motion
Ingestion	3.2	see	2.8	End-Position	4.0	Possession	5.7	located in	Ingestion

Table 4: Top 5 event types of our LED dataset compared to other datasets.

Dataset	Task	Source	# Docs	# Events	# Types (Subtypes)	# Arg Roles	Coref
ACE2005 (2006)	EE	News	599	5,349	8 (33)	35	
CONLL04 (2004)	RE	News	1,437	2,041	5	4	
LiveKB (2019)	Life EE	Twitter	25,344	15,525	137	6	
PEDC (2020)	Life EE	Podcast	1,038	3,664	278	0	
DiaLog (2021)	Life EE	DailyDialog	600	780	21	0	
Life Event Dialog	Life EE	DailyDialog	1,003	2,186	175 (371)	5	✓

Table 5: Datasets comparison. EE: Event Extraction, RE: Relation Extraction.

Framework	Original Output	LED Output
OpenIE	(head, relation, tail)	(S, <i>Verb</i> (explicit), O)
RE	(head, relation, tail)	(S, <i>Verb/Class/Frame</i> , O)
EE	[T span, T type, A ₁ span, A ₁ type, A ₂ span, A ₂ type, ...]	[<i>Verb</i> (explicit), <i>Class/Frame</i> , S/O, “subject”/“object”]

Table 6: Outputs from OpenIE, RE, and EE frameworks and their mapping to LED output. For EE framework, original output is the span and type of event trigger (T) and the span and type of arguments (A). The T span maps to the span of *Verb* of explicit events; T type maps to *Class* or *Frame* of that event; A span maps to the span of S or O with corresponding “subject” or “object” string as their A type.

the tweet if not mentioned. In contrast, the event subject in a dialogue is half time the speaker, 40% the listener, and 10% the others, as shown in Fig. 2. The case of the subject being the listener happens when the event of the listener is told by the speaker, such as “You are hired by our compan”, “You get high marks in the exam”, or “I’m Jame, your neighbor when you lived here last year (indicating the event of the listener living here last year)”. Also, besides the case when the speaker themselves being the subject (when the mention is self-referred), the mention of the subject is often omitted (and annotated as S1/S2) or being “you”. It usually happens when the speaker is confirming an event. For example, S1: “Could you please sign this memo?” S2: “No problem.” The event (S2, sign, memo) becomes positive after S2’s confirmation. These kinds of events that happen after user interactions only appear in our Life Event Dialog data. There is sometimes an ambiguity regarding the event subject, e.g., S mention “we” might refer to only the speaker or

both participants in the dialogue. Further, comparing the top 10 *Frame* in LED and LiveKB, we find that LED has more interactive activities, such as “Statement”, “Request”, and “Acquaintance”. In contrast, LiveKB activities are more self-centered, like “Presence”, “Create”, and “Buy”.

Both conversational event extraction datasets, PEDC (Eisenberg and Sheriff, 2020) and DiaLog (Kao et al., 2021), only annotate event type labels. The former is collected from podcast transcripts and focuses on event from life stories told by first-person narrators. The latter classifies events by FrameNet and is also from the DailyDialog. Our LED has more data, more event types, and additional annotations of argument roles, event status, and coreference clusters, compared with them.

5 Conversational Life Event Extraction

We define Conversational Life Event Extraction as the combination of two subtasks: (1) Event Type Classification and (2) Participants Identification.

Given a dialogue $D_u = \{T_1, T_2, \dots, T_u\}$ of u turns utterances, we extract i events $E_u^i = \{e_u^1, \dots, e_u^i\}$ from D_u , where an event e comprises an event type from either *Verb*, *Class*, or *Frame* and spans of participants (S and O). We consider an input instance as the concatenation of turns T_1 to T_u .

5.1 Frameworks

We aim to identify the event type and participants simultaneously. By contrast, previous works on life event extraction only dealt with event type prediction. Hence no model specifically tackles our proposed task of conversational life event extraction. As a result, we examine different information extraction frameworks, including (1) OpenIE, (2) Event Extraction (EE), and (3) Relation Extraction (RE), for this task. We transform our data schema to fit the original schema of each framework, as shown in Table 6. Both OpenIE and RE output (head, relation, tail) triples. We consider the head and tail to be S and O and relation to be an event type. EE outputs the span and type of an event trigger, as well as the span and type of arguments. When converting to our LED schema, the event trigger can be seen as the event type and arguments as participants. Due to limitations of each framework, the output from each framework is slightly different when adapting to our dataset. The major constraint is that OpenIE and EE frameworks can only predict explicit events because both output spans from the input dialogue.

OpenIE: OpenIE requires each element in the triplet to be a span from the input, therefore, it is not able to predict event types of *Class* and *Frame*, nor the implicit event which *Verb* is written by annotators. Also, OpenIE always outputs the whole event triplet, so it can never correctly predict the events without object. We use Stanford Open IE system (Angeli et al., 2015) as the OpenIE baseline to extract life event triples.

Relation Extraction: RE framework also generates triples as output. REBEL (Huguet Cabot and Navigli, 2021) is selected as the relation extraction baseline, which is based on an autoregressive model BART-large (Lewis et al., 2019). Since REBEL is a generation model, it can generate tokens not in the given dialogue and avoid the limitations of OpenIE framework.

Event Extraction: Event Extraction framework predicts both spans and their type; thus, the implicit events without trigger span can never be pre-

dicted. We choose DyGIE++ (Wadden et al., 2019) as our event extraction baseline. DyGIE++ is a span-based model with RoBERTa-base (Liu et al., 2019) backbone, which can perform multi-tasks training on entity recognition, relation extraction, event extraction, and coreference resolution.

5.2 Evaluation

Evaluation metrics vary between frameworks. We evaluate the output triples from OpenIE and RE using precision (P), recall (R), and micro-F1, following previous works (Huguet Cabot and Navigli, 2021). We adapt the strict evaluation (Taillé et al., 2020), that is, a triple is considered as correct only if the whole triple is exactly the same as the ground truth triplet. EE results are evaluated by P, R, and F1 of span identification and type classification. An event trigger is correctly identified if the span is correct and is correctly classified if the event type is correct. An event argument is correctly identified if both the event type and the argument span are correct, and is correctly classified if the argument type is correct.

We unite evaluation metrics for all frameworks using a lenient evaluation metric. For each life event, we first evaluate the event type classification (ET-C) by P, R, and F1. Then, for those events with correct event type, we evaluate the participants identification by P, R, and F1 of S (S-ID) and O (O-ID F1). We also compute BERT Score (Zhang et al., 2020) for the object (O-ID BS), because O in LED are often longer than a single token, unlike in EE/RE datasets (as discussed in Sec 4.2).

5.3 Analysis

Table 7 presents the result of employing each framework on explicit life event extraction, suggesting that the EE framework works the best on event type classification (ET-C) and subject identification (S-ID) over different granularities of event type. We think the graph-based EE model (DyGIE++) can better capture critical entities and their interactions for event type and S. The other thing we can benefit from DyGIE++ is that it is compatible with the coreference training, so we can make use of our annotations on participants’ coreference clusters. However, we are surprised to find that the additional coreference training does not help. We suspect that a large amount of examples of the same mention referring to different entities in a dialogue confuse the coreference training. For example, the

Event Type Granularity	Framework	ET-C			S-ID			O-ID			
		P	R	F1	P	R	F1	P	R	F1	BS
<i>Verb</i>	OpenIE	18.5	29.1	22.6	17.3	27.2	21.1	6.5	10.2	7.9	33.5
	RE	28.5	49.8	36.2	23.6	41.3	30.1	15.4	26.9	19.6	66.2
	EE	79.0	30.0	43.5	64.2	24.4	35.4	28.4	10.8	15.6	42.0
	EE + coref	84.1	24.9	38.4	63.5	18.8	29.0	30.2	8.9	13.8	19.7
<i>Class</i>	RE	27.6	49.3	35.4	22.9	40.8	29.3	14.7	26.3	18.9	64.4
	EE	67.8	27.7	39.3	55.2	22.5	32.0	26.4	10.8	15.3	42.0
	EE+coref	59.2	19.7	29.6	40.8	13.6	20.4	26.8	8.9	13.4	19.7
<i>Frame</i>	RE	23.4	40.4	29.6	16.3	28.2	20.7	12.0	20.7	15.1	61.1
	EE	58.6	23.9	34.0	46.0	18.8	26.7	26.4	10.8	15.3	40.2
	EE+coref	57.4	12.7	20.8	57.4	12.7	20.8	21.3	4.7	7.7	64.0

Table 7: Result on explicit events across different frameworks evaluated by our lenient evaluation. ET-C: Event Type Classification, S-ID: Subject Identification, O-ID: Object Identification, BS: BERT Score.

Event Type Granularity	Data	ET-C		S-ID		O-ID	
		(F1)	(Δ)	(F1)	(Δ)	(F1)	(BS)
<i>Verb</i>	E	36.2		30.1		19.6	66.2
	E+I	29.9	-6.3	20.7	-9.4	13.9	57.7
<i>Class</i>	E	35.4		29.3		18.9	64.4
	E+I	28.4	-7.0	20.9	-8.4	12.0	57.9
<i>Frame</i>	E	29.6		20.7		15.1	61.1
	E+I	24.3	-5.3	16.4	-4.3	12.4	58.3

Table 8: Event extraction with (E+I) and without (E) implicit events by RE framework.

same subject mention ‘‘I’’ might refer to S1 or S2 in different events.

As for object identification (O-ID), the RE framework gets the top. We can see from Table 7 that the bottleneck of Conversational Life Event Extraction is on O-ID, whose F1 score is much lower than the ET-C and S-ID. The reason might be the high variance of object mentions. We think the best performing RE model (REBEL), an autoregressive model based on a large pretrained language model, is better at copying a sequence of input for O, therefore, can get the best result on O-ID. We also found that REBEL often generates repeated output and has higher recall (R) than precision (P), in contrast to DyGIE++, which gets a higher P than R.

For the three event type granularities, *Verb* is the easiest to predict, and *Frame* is the most challenging. The result in Table 7 shows a consistent decreasing trend from *Verb*, *Class*, to *Frame* across all frameworks. For ET-C, the gap from *Verb* to *Class* (RE: -0.8, EE: -4.2) is smaller than from *Class* to *Frame* (RE: -5.8, EE: -5.3). This is intuitive because *Verb* and *Class* are more similar. The drastic drop on *Frame* demonstrates the difficulty of inferring the frame name from the dialogue.

RE is the only framework among the three that

Event Type Granularity	Framework	ET-C	S-ID	O-ID	
		(F1)	(F1)	(F1)	(BS)
<i>Verb</i>	OpenIE	37.6	37.6	12.5	36.8
	RE	25.6	22.8	15.9	50.0
	EE	21.7	21.7	0.0	8.0
<i>Class</i>	RE	22.5	22.5	12.5	44.3
	EE	21.7	21.7	0.0	8.0
<i>Frame</i>	RE	0.0	0.0	0.0	51.6
	EE	0.0	0.0	0.0	0.0

Table 9: Zero-shot result on explicit events across different frameworks.

can deal with implicit events. The implicit events account for 31.4% of all events; hence, we further analyze their impact. Table 8 shows the results from the RE framework with and without implicit events. Despite event type granularities, the results drop after adding implicit events. Particularly, when the event type is of *Verb* or *Class*, the negative effect of implicit events is significant (see the Δ column). The results with implicit events are almost the same as the result of explicit events’ frame name prediction. In other words, predicting an event type that is not in the input dialogue is extremely difficult, and current models cannot achieve promising results.

We examine the zero-shot result over the three frameworks, when the testing event types are not seen in training time. The result is shown in Table 9. OpenIE performs the best for ET-C and S-ID on the setting of *Verb* zero-shot. Since OpenIE is a rule-based model and does not need any training, it is better than the models required training for unseen event types. In addition, for the trained models of RE and EE frameworks, they cannot infer any unseen frame name. Since life events are broad and not fully covered in our dataset, developing

models that can extract unseen event types remains an essential research question.

6 Conclusion

This work presents Life Event Dialog: a comprehensive life event dataset annotated on DailyDialog conversations. The main differences between our dataset and previous datasets on personal life event extraction are: (1) Life Event Dialog is built on top of conversations instead of microblogs like Twitter. The interaction between speakers adds dynamics to events, such as information expansion or status modification, and indicates people’s general interests in multiple aspects of other’s life events. (2) Life Event Dialog contains more data, more types, and more fine-grained event annotations compared to other conversational life event datasets.

We propose the Conversational Life Event Extraction task, extending life event extraction tasks from social media to the conversation domain and from event type detection to predicting both event type and participants simultaneously. We then carefully examine three information extraction frameworks: OpenIE, relation extraction (RE), and event extraction (EE), for the pilot study on this task. The result suggests that current top models on three closely related fields cannot perform well in the Conversational Life Event Extraction task. Improving object identification and implicit event extraction, detecting unseen life events, and keeping track of event status, constitute our future work.

Limitations

Our LED dataset is annotated on DailyDialog. While annotating on another dataset brings some benefits, it also constrains our dataset. For instance, our dataset is limited to the top five frequent topics in DailyDialog, which might not be enough to cover all life events in various scenarios. Also, DailyDialog only contains conversations between two interlocutors. For a multi-party conversation, the conversational life events extraction would be much more complicated and interesting.

The other limitation of LED is the size of the dataset. Although with more comprehensive annotations of life events, the number of events in our dataset might not be enough for today’s data-hungry models. There is always room for larger datasets and more annotations. Compared to the entity types in RE like “person”, “organization”, “location”, to name a few. We do not label such so-

phisticated argument roles but only “subject” and “object”. We leave this part to our future work. Besides, we only consider up to 6 turn utterances, yet a dialogue might be much longer in real life.

Lastly, the definition of life events varies from individual to individual, and our definition of life events might not suit everyone’s needs. However, our exploration of the zero-shot experiment shows that it is still possible to find unseen events, and a better model for zero-shot life extraction is needed.

Ethics Statement

Our Life Event Dialogs dataset is an extension of an existing public dataset DailyDialog, with all speakers being anonymized in the original release. In other words, our dataset does not contain any personally identifiable information that would infringe on someone’s privacy. In this work, we will only release the life event annotations for research purposes. The dialogues in DailyDialog will not be included in LED, but one can access the full DailyDialog dataset from the author’s website.²

Our dataset is constructed upon a considerable amount of human annotation. We recruited three annotators and paid them a local hourly wage for the time they spent. The annotation period spanned 1.5 months and resulted in 1,003 annotated conversations (including conversations without events).

Acknowledgements

This research was supported by the commissioned research (No. 225) by National Institute of Information and Communications Technology (NICT), Japan, and JSPS/MEXT KAKENHI Grant Numbers JP19H04166 and JP22H05015. Chen is supported by JST SPRING, Grant Number JPMJSP2108. This research was also supported by National Science and Technology Council, Taiwan, under grants MOST 110-2221-E-002-128-MY3 and NSTC 111-2634-F-002-023-.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354.

²<http://yanran.li/dailydialog>

- Aiden R Doherty, Niamh Caprani, Vaiva Kalnikaite, Cathal Gurrin, Alan F Smeaton, Noel E O'Connor, et al. 2011. Passively recognising human activities through lifelogging. *Computers in Human Behavior*, 27(5):1948–1958.
- Joshua Eisenberg and Michael Sheriff. 2020. Automatic extraction of personal events from dialogue. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 63–71.
- Charles J Fillmore, Collin F Baker, and Hiroaki Sato. 2002. The framenet database and software tools. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, pages 1157–1160.
- Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, V-T Ninh, T-K Le, Rami Albatat, D-T Dang-Nguyen, and Graham Healy. 2019. Overview of the ntcir-14 lifelog-3 task. In *NTCIR-14*, pages 14–26. NII.
- Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. **REBEL: Relation extraction by end-to-end language generation**. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Mahsa Imani. 2014. *Evaluating open relation extraction over conversational texts*. Ph.D. thesis, University of British Columbia.
- Pei-Wei Kao, An-Zi Yen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Convlogminer: A real-time conversational lifelog miner. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*.
- Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Jiwei Li, Alan Ritter, Claire Cardie, and Eduard Hovy. 2014. Major life event extraction from twitter based on congratulations/condolences speech acts. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1997–2007.
- Qian Li, Hao Peng, Jianxin Li, Jia Wu, Yuanxing Ning, Lihong Wang, S Yu Philip, and Zheng Wang. 2021. Reinforcement learning-based dialogue guided event extraction to exploit argument relations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:520–533.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Md Abed Rahman, AM Esfar E Alam, Md Hasan Mahmud, and Md Kamrul Hasan. 2018. Towards a smartphone based lifelogging system for reminiscence. *Journal of Engineering and Technology*, 14(1).
- Dan Roth and Wen-tau Yih. 2004. **A linear programming formulation for global inference in natural language tasks**. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. **Let's Stop Incorrect Comparisons in End-to-end Relation Extraction!** In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3689–3701, Online. Association for Computational Linguistics.
- David Wadden, Ulme Wennberg, Yi Luan, and Hananeh Hajishirzi. 2019. **Entity, relation, and event extraction with contextualized span representations**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.
- Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.
- Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. **MAVEN: A Massive General Domain Event Detection Dataset**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.
- An-Zi Yen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2018. Detecting personal life events from twitter by multi-task lstm. In *Companion Proceedings of the The Web Conference 2018*, pages 21–22.
- An-Zi Yen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2019. Personal knowledge base construction from text-based lifelogs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and*

Development in Information Retrieval, pages 185–194.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.

A Annotation Guideline

A.1 Goal

We want to extract personal life events related to the speaker according to their dialogue, so that we can construct a personal life knowledge base and benefit other downstream tasks.

A.2 What **are** personal life events?

1. The event happens or might happen in the future to the interlocutor themselves or their relatives and friends.
 - Example: “I went to Salt Lake City on business with Mr. Wang.”
2. The event must occur before the dialog or before the dialog ends.
3. When expressing personal thoughts or feelings, the context implies life events.
 - Example: “These cookies taste delicious.” may imply an event that the speaker has eaten cookies.
4. The life history or personal information of the interlocutor.
 - Example: summer vacation, school start, graduation, “I skipped fourth grade.”, etc, all belong to life experiences.
 - Example: “I live in Taiwan.”, “I was born in 1980.” are personal information.
5. Interlocutor’s personal habits.
 - Example: “I usually look at English language websites every day and go to my local English Corner twice a week.”
6. If there is no clear sentence describing an event in the conversation, use the context to see if a life event occurred before the conversation completes.
 - Example: “S1: What’s for supper? S2: Red cooked carp and rape with fresh mushrooms.” When the dialogue is completed, it can be deduced that the event

“S2 cooked Red cooked carp and rape with fresh mushrooms for dinner” occurred.

- Example: “S1: I ran a red light? S2: Yes, you did.”, S1 was originally a question, and the answer of S2 affirmed the occurrence of S1 running a red light.

A.3 What **are not** personal life events?

1. Public issues or general knowledge
 - Examples: news, knowledge, company business related events.
 - Examples: “We run a spotless and cockroach-less hotel.” Events that represent the company’s position are not counted.
2. Only expressing personal feelings and preferences (related to emotions)
 - Examples: “I feel tired,” “I think you are cute,” “I like Chinese food,” “I’m worried about his condition,” “I’m tired of going to school,” etc.
3. Expressing personal abilities
 - Example: “I can type 80 words a minute.”
4. Things that are not guaranteed to happen don’t need to be marked as possible future events
 - Examples: “Can you wait a little while?” “You should go to school tomorrow.”
5. “Ask questions” and “express opinions” are not considered life events of themselves (unless there is an answer response to judge that an event has occurred)
 - Example: “S1: Did you go to school yesterday? S2: No, I didn’t.” Only need to mark the event “S2 did not go to school yesterday”, and do not need to mark the event “S1 asked S2 a question”.
6. A simple description of the environment, people, things, and things is not considered a life event (unless there is an implied life event)
 - Example: “That girl standing there is pretty.”

A.4 Event Explicitness

Events can be classified into *Explicit* or *Implicit* events, depending on whether there is a clear action in the sentence to indicate the occurrence of the event.

Explicit Event: There exists an explicit action describing a life event.

- As long as the Predicate appears in the dialogue that clearly represents the action of the event, it belongs to the *Explicit* event. If there is a verb but it is not clear, please deduce the explicit verb and mark it as *Implicit*.
 - Example: “S1: I ran a red light? S2: Yes.” → Explicit Event: (Subject= S1, Predicate= ran, Object= red light, Time= BEFORE, Polarity= POS, Modality= ACTUAL)
- Object can be missing, for example: “We’ll wait.” with a clear action (wait).
- If the life event has been explicitly described, it is not necessary to extend the label to other possible events.
 - Example: “Today I played basketball.” There is no need to mark the event of “I went to the basketball court.”

Implicit Event: Contexts and situations are required to infer a life event. (As long as the Predicate needs to be deduced, it is considered *Implicit*).

- Please infer the action most relevant to your life experience based on the dialogue context.
- A sentence with an ambiguous verb.
 - Example: “I want a fillet steak, medium.” In the context of ordering food, please deduce that the Predicate is “order”, and mark the event as *Implicit*. → Implicit Event: (Subject= I, Predicate= order, Object= fillet steak, medium, Time= NOW, Polarity= POS, Modality= ACTUAL)
- Events implicit in the dialogue.
 - Example: “S1 : Can I get you some coffee? S2 : De-caff.” → Implicit Event: (Subject= S2, Predicate= order, Object= De-caff, Time= NOW, Polarity= POS, Modality= ACTUAL)

- Implicit event in a sentence.
 - “S1 : You must be exhausted after your long trip from Canada.” → Implicit Event: (Subject= You, Predicate= travel from, Object= Canada, Time= BEFORE, Polarity= POS, Modality= ACTUAL)
- The situations of the dialogue, such as order meals, make phone calls, send things, job interviews, etc.
 - Example: “S1 : This is John speaking. S2 : Hi, this is Mary.” → Implicit Event: (Subject= S2(Mary), Predicate= call, Object= S1(John))
- Note: Except for the Predicate of Implicit Event, please use the vocabulary in the sentence for Subject, Predicate, Object, and Time of Explicit Event, and do not create your own vocabulary.

A.5 Format Description

The annotation for an event includes the following fields: Subject, Predicate, Object, Time, Polarity, Modality.

Subject: The subject is the word that performs the action. Most subjects are nouns, pronouns, noun phrases or noun clauses. Subjects are mainly the two interlocutors, but may also be people or things related to life events.

Predicate: The action of a life event, expressing what the subject did or what happened. Usually a verb, but may also be a preposition (please refer to the example label below).

- Predicate needs to indicate a clear action.
 - Example: “I’d like to take the apartment I looked at yesterday.”, take means accept, but we know from the above that the interlocutor wants to rent a house, so please mark the more specific action rent as a Predicate.
 - Example: “I need a double and three triples.”, need means need, but it can be inferred in the dialogue that the interlocutor wants to book a room, so please mark the action book as Predicate.
 - Example: “I’ll be right there.” This sentence means that I will go to a certain store immediately, please do not directly mark (I, be, there), please deduce a more precise action go to from the predicate

- When Predicate is a preposition, please mark it according to the following example:
 - "with" means "and", which means an event involving more than two people.
 - * Example: "I went shopping with her." or "I went shopping ... with her."
 - Explicit Event 1: (Subject= I, Predicate= went , Object= shopping, Time= BEFORE, Polarity= POS, Modality=ACTUAL,)
 - Explicit Event 2: (Subject= I, Predicate= went with / with , Object= her, Time= BEFORE, Polarity= POS, Modality= ACTUAL)
 - Modifies verbs, such as prepositions denoting the destination and means of movement.
 - * Example: "I went to San Francisco by plane."
 - Event 1: (Subject= I, Predicate= went to , Object= San Francisco, Time= BEFORE, Polarity= POS, Modality= ACTUAL)
 - Event 2: (Subject= I, Predicate= went by , Object= plane, Time= BEFORE, Polarity= POS, Modality= ACTUAL)
 - * Example: "He is on the school volleyball team."
 - Event: (Subject= He, Predicate= is on , Object= school volleyball team, Time= CONTINUOUSLY, Polarity= POS, Modality= ACTUAL)
 - * Example: "S1 : Did you hear it on the radio? S2 : Yes."
 - Event 1: (Subject= S2, Predicate= hear , Object= it, Time= BEFORE, Polarity= POS, Modality= ACTUAL)
 - Event 2: (Subject= S2, Predicate= hear on , Object= radio, Time= BEFORE, Polarity= POS, Modality= ACTUAL)
 - If the preposition refers to the time, please mark the time directly in the field of Time.
 - * Example: "We ate dinner at 8 pm"
 - Event: (Subject= We, Predicate= ate, Object= dinner, Time= 8 pm, Polarity= POS, Modality= ACTUAL)
 - Nested events.
 - Example: "I'm planning to sing a song in front of everybody."
 - Event 1: (Subject= I , Predicate= 'm planning to , Object= sing a song in front of everybody , Time= NOW, Polarity= POS, Modality= ACTUAL)
 - Event 2: (Subject= I , Predicate= sing , Object= song , Time= AFTER, Polarity= POS, Modality= HYPOTHETICAL)
 - Event 3: (Subject= I , Predicate= in front of , Object= everybody , Time= AFTER, Polarity= POS, Modality= HYPOTHETICAL)
 - Sentences that describe situations where no event occurred.
 - Example: "John didn't go to the party tonight." Predicate does not need to mark negative words (didn't), please mark positive or negative marks in Polarity.
 - Sentences describe possible future events.
 - Example: "We will have a meeting at 9 am tomorrow." Predicate does not need to mark auxiliary verbs that indicate future occurrences (for example: will, is going to), please mark the form of event occurrence in Modality.
 - Not a predicate of personal life events: think, know, need, want, hope, trust, like, feel.
- Object:** The object may be a person, thing, or object, expressing the relationship with the Subject through the Predicate. Most are nouns, pronouns, noun phrases or noun clauses.
- Please use words that appear in the dialogue as much as possible, and only mark words that are meaningful to the event.
- Example: "I have a hat." Do not need to annotate articles (such as "a", "the").
 - Example: "I made this delicious dinner." Do not need to annotate the adjective.
 - Example: "I have a problem with my room." Supplemental words such as "with my room" need to be annotated.

Time: Express the time information of the life event, such as the time or frequency of the event.

If there is a clear description of the time information in the dialogue, for example: yesterday, last week, directly fill in the time information in the sentence.

If there is no clear description, the default time mark can be filled in as follows:

- **BEFORE** : Indicates that the event occurs before the dialog occurs.
- **NOW** : Indicates that the event occurred during the period from the beginning of the conversation to the end of the conversation
- **CONTINUOUSLY** : Indicates that the event has continued to occur from the past to the present (longer duration).
- **AFTER** : Indicates that the event (possibly) happens after the conversation ends.

Please infer which label is suitable for filling in according to the dialogue.

If there is a vague description in the sentence, please fill in the mark that matches the meaning of the adverb of time.

- Example: “I just finished my homework.”
Please fill in NOW for Time.

If people use “after...” or “before...” to describe the occurrence time in the sentence, you can fill in it directly.

Polarity: Indicates that the life event is positive or negative. The default is POS for positive and NEG for negative.

- Example: “You did not invite me to the party.”
→ Event 1: (Subject=You, Predicate=invite, Object=me, Time=BEFORE, Polarity= NEG , Modality=ACTUAL)
→ Event 2: (Subject=You, Predicate=invite to, Object=party, Time=BEFORE, Polarity= NEG , Modality=ACTUAL)
- Example: “I have no money with me.”
→ Event: (Subject=I, Predicate=have, Object=money, Time=NOW, Polarity= NEG , Modality=ACTUAL)

Modality: Indicates the form of life events, with the following symbols:

- **ACTUAL:** Indicates that the event has occurred before or at the moment when the sentence is spoken.
- **HYPOTHETICAL:** Indicates that the event may happen in the future, but only if there is a clear sentence in the dialogue to affirm or deny that the future will do. Even if the next moment of speaking may happen but has not happened yet, please mark it as HYPOTHETICAL. After adding the next sentence of dialogue, the situation can be deduced that it has happened, and then changed to ACTUAL.

A.6 Coreference Annotation

Mark all words in the dialogue that point to pronouns in the Event. Mark all the words representing the same thing into the same mention.

- Example: “S1 : Did you eat the cake on the table? S2 : Yes, I ate that.”
→ Explicit Event: (Subject= I, Predicate= ate, Object= that, ...)
→ Coref tag: (Subject: (I, S2), Object: (that, cake on the table))

B Annotation Interface

Figure 3 shows the annotation interface. The annotator was first shown the topic of the conversation, the number of turns to annotate, and the full dialogue. Then, the utterances of the dialogue are displayed turn by turn cumulatively. The example in Fig 3 is the second instance of the dialogue. The annotators should decide whether the cumulative turns contain life events of the speakers. If answering “Yes”, they will add the index of “Subject”, “Predicate” (if it’s an explicit event), and “Object”, and select the event status.

Annotate Life Events

Topic : Work

Turns : 4

S1 : May I help you ?
 S2 : Yes , I 'm looking for Bob .
 S1 : He 's in a meeting with Phil .
 S2 : No problem , I can wait .

Please annotate life events according to the order of the utterances

S1 : May I help you ?
 (0) (1) (2) (3) (4) (5) (6)
 S2 : Yes , I 'm looking for Bob .
 (7) (8) (9) (10) (11) (12) (13) (14) (15) (16)

Does the conversation contain life events of "S1" or "S2"? No Yes

Explicitness	Subject	Predicate	Object	Time	Polarity	Modality
<input checked="" type="radio"/> Explicit <input type="radio"/> Implicit	7 Paste 11 Paste + Coreference	12-14 Paste	15 Paste + Coreference	<input checked="" type="radio"/> NOW <input type="radio"/> BEFORE <input type="radio"/> AFTER <input type="radio"/> CONTINUOUSLY <input type="radio"/> Paste	<input checked="" type="radio"/> POS <input type="radio"/> NEG	<input checked="" type="radio"/> Actual <input type="radio"/> Hypothetical

+ Add an event

Pass

Figure 3: The annotation interface.

Reading and Reasoning over Chart Images for Evidence-based Automated Fact-Checking

Mubashara Akhtar, Oana Cocarascu and Elena Simperl

Department of Informatics, King’s College London

{mubashara.akhtar,oana.cocarascu,elena.simperl}@kcl.ac.uk

Abstract

Evidence data for automated fact-checking (AFC) can be in multiple modalities such as text, tables, images, audio, or video. While there is increasing interest in using images for AFC, previous works mostly focus on detecting manipulated or fake images. We propose a novel task, chart-based fact-checking, and introduce ChartBERT as the first model for AFC against chart evidence. ChartBERT leverages textual, structural and visual information of charts to determine the veracity of textual claims. For evaluation, we create ChartFC, a new dataset of 15,886 charts. We systematically evaluate 75 different vision-language (VL) baselines and show that ChartBERT outperforms VL models, achieving 63.8% accuracy. Our results suggest that the task is complex yet feasible, with many challenges ahead.

1 Introduction

Charts are often used to present data in news articles, reports, scientific publications, and across social media posts (Lo et al., 2022; Zhang et al., 2021). For example, in recent years, charts have been widely used to guide policymakers in deciding health policies and to communicate COVID information with the general public; a popular example is the coronavirus dashboard by Johns Hopkins University,¹ which was integrated in several websites (Perkel, 2020).

Misinformation can spread through charts in various ways. Previous works in data visualization have discussed how misleading chart design can cause misinformation (Lo et al., 2022). However, a more subtle form of misinformation occurs during chart interpretation (e.g. through invalid comparisons, framing correlation as causation, or spreading of misleading claims). To identify these misinformation types not only the stand-alone chart but the chart together with its message need to be

¹<https://coronavirus.jhu.edu/map.html>

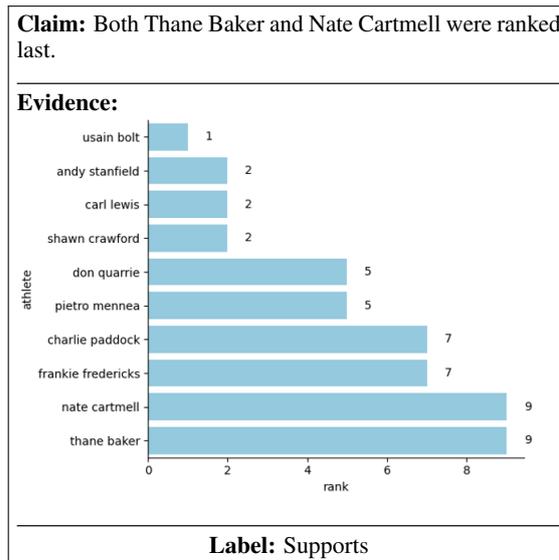


Figure 1: An example from the ChartFC dataset where the claim is supported by the evidence chart.

considered jointly (Lo et al., 2022). In this work, we focus on verifying whether charts support or refute claims about them.

There has been substantial progress in automated fact-checking (AFC) in recent years, with a focus on verifying claims against text (Wang, 2017; Thorne et al., 2018; Schuster et al., 2021; Thorne et al., 2021; Diggelmann et al., 2020), table (Aly et al., 2021; Diggelmann et al., 2020; Chen et al., 2020a; Akhtar et al., 2022), and image (Yao et al., 2022; Zlatkova et al., 2019; Qu et al., 2022) evidence. Previous work has widely ignored claim verification against chart images. There are several challenges related to chart fact-checking as opposed to other evidence modalities: the structural information, text in charts, and location of text need to be considered jointly for chart understanding. Text plays a key role and is used, for example, as bar labels, chart titles, or in legends to explain the use of colors. Moreover, verifying claims against charts requires different reasoning

types, e.g. retrieving values, finding extremes, or calculating a sum.

To address these challenges, we propose the chart fact-checking task where, given a text claim and a chart, the goal is to classify if it *supports* or *refutes* the claim. We introduce ChartBERT as the first model for AFC against chart evidence comprising (i) an OCR-based reading component to extract text and structural information from chart images; (ii) a sequence generation component to process the extracted information; and (iii) an encoding component that extends the BERT architecture (Devlin et al., 2019) with three additional structural embeddings to jointly learn textual and structural representations of chart images.

Moreover, we release ChartFC as the first benchmark for chart-based AFC, created using TabFact (Chen et al., 2020a) as a seed dataset. Our dataset contains 15.9k human-written claims and bars of different colors, orientations, and backgrounds (see Figure 1 for an example). Our highest-performing ChartBERT model achieves 63.8% accuracy on ChartFC. We compare ChartBERT to 75 vision-language (VL) baselines, combining five vision encoders, three language encoders, and five fusion methods. The best-performing VL model is a transformer-based (Vaswani et al., 2017), dual encoder architecture that uses a simple, yet effective fusion block: concatenation and gated recurrent units (GRUs) (Bahdanau et al., 2015). Our results suggest that state-of-the-art VL approaches struggle with the proposed task, calling for more research.

Our **contributions** are as follows: 1) we propose the chart fact-checking task and build ChartBERT as the first chart fact-checking model; 2) we introduce ChartFC, the first dataset for AFC with chart evidence; 3) we systematically evaluate state-of-the-art language/vision encoders and fusion methods on the proposed task, highlighting challenges and providing an analysis of common reasoning types that contribute to failures.²

2 Related Work

2.1 Verifying Claims against Evidence

Evidence-based fact-checking aims to predict claims’ veracity given evidence data. While many datasets focus on text (Thorne et al., 2018; Kotonya and Toni, 2020; Schuster et al., 2021; Wang, 2017)

²The ChartFC dataset and our code are available at https://github.com/mubasharaak/ChartFC_chartBERT.

and table evidence (Chen et al., 2020a; Gupta et al., 2020; Aly et al., 2021; Wang et al., 2021a; Akhtar et al., 2022), human fact-checkers use a wider range of modalities for verification (Nakov et al., 2021b; Alam et al., 2021). They consult experts and extract information from databases, text, tables, graphics, and audio/video material from numerous sources.³

Charts influence how messages are perceived (Pandey et al., 2014). For example, Lee et al. (2021) use the term “counter-visualization” to describe data visualizations by the anti-vaccination communities in the US who created charts from publicly available data and interpreted them in a way that challenged the narrative of the pandemic, leading to disinformation.

2.2 Automated Fact-Checking with Images

Given that claims and evidence can be conveyed through different modalities, interest in AFC with images has increased recently (Nakov et al., 2021a; Cao et al., 2020; Alam et al., 2021; Yao et al., 2022; Sharma et al., 2022). Previous tasks focus mainly on detecting manipulated or fake images rather than on evidence-based claim verification (Blaier et al., 2021; Kiela et al., 2020; Alam et al., 2021; Sharma et al., 2022; Abdali, 2022). Whilst manipulated or fake images can be detected using the image only, claim verification requires understanding the claim and evidence jointly.

2.3 Chart Images in other NLP Tasks

Two tasks related to chart fact-checking are chart question answering and chart summarization. Given a chart image, the summarization task requires to generate a summary of the chart in natural language text (Kantharaj et al., 2022; Tan et al., 2022). For question answering (chartQA) the answer to natural language questions is extracted from chart images. However, different to claim verification, questions typically provide strong indicators for the correct answers. Existing chartQA datasets are either small (Kim et al., 2020) or comprise automatically-generated, template-based questions (Chaudhry et al., 2020; Kahou et al., 2018; Kaffle et al., 2018).

3 ChartBERT Model

We introduce ChartBERT, a first BERT-based chart fact-checking model. Our model consists of (i) a

³https://ballotpedia.org/The_methodologies_of_fact-checking

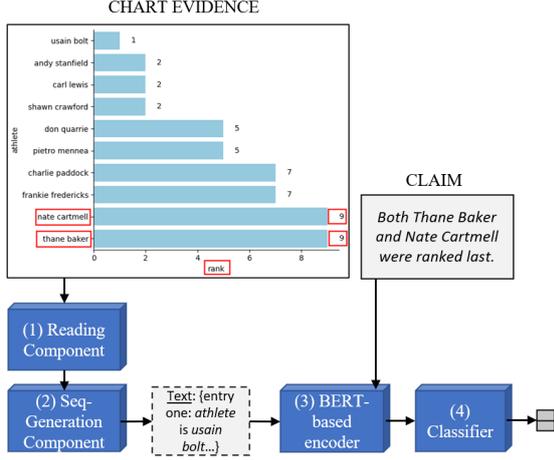


Figure 2: The ChartBERT architecture.

reading component which extracts text and structural information from charts (Section 3.2); (ii) a component for generating textual sequences from the information previously extracted (Section 3.3); and (iii) a BERT-based encoder with additional structural embeddings for the text extracted from charts (Section 3.4). The model architecture is shown in Figure 2.

3.1 Task Formulation

Following previous AFC work (Chen et al., 2020a; Aly et al., 2021; Thorne et al., 2018; Wang et al., 2021b), we view chart fact-checking as a classification task where, given a natural language claim and a piece of evidence (i.e. the chart image), the goal is to decide if the evidence *supports* or *refutes* the claim. We use support/refute as labels for claim classification instead of true/false as we only assess the claim veracity given the provided evidence rather than claiming universal statements.

Each ChartFC sample $i = (c_i, img_i, y_i)$ comprises a natural language claim c_i , a chart image img_i (see Figure 1 for an example), and a label $y_i \in \{supports, refutes\}$.

3.2 Reading Text from Chart Images

Given an image img_i , the reading component extracts text and structural information. First, we detect text regions in the chart using a Tesseract OCR model (Kay, 2007). Specifically, for each image, the model extracts n text regions $T_i = \{t_1, t_2, \dots, t_n\}_{j=1}^n$, where each region t_j consists of $text_j$, a sequence of m tokens, and a rectangular bounding box b_j that surrounds the text region in the chart. The bounding box is a tuple $b_j = (x_j, y_j, w_j, h_j)$ where x_j and y_j are the pixel

coordinates of the top left point of the box, and w_j and h_j represent the width and height of the box in pixels. Thus, for each image img_i we obtain the following output o_i :

$$o_i = f_R(img_i) = \{(text_j, x_j, y_j, w_j, h_j)\}_{j=1}^n$$

3.3 Text Sequence Generation

Next, we process the reading component’s output into a text sequence s_i consisting of m tokens:

$$s_i = f_{SeqGen}(o_i) = [s_1, s_2, \dots, s_m]$$

We compare two approaches as follows.

Concatenation: The concatenation method processes the text regions (i.e. $t_j \in T_i$) based on their coordinates x_j and y_j so that texts that are close in the chart are also close in the generated sequence. The chart text is concatenated into one sequence and tokens that belong to different text regions are separated using a $[\ ;]$ token. Thus, for the chart Figure 1 we obtain a text sequence starting with “usain bolt ; 1 ; andy stanfield ; 2 ; [...]”

Template: We use the structural information (i.e. x, y, w_j, h_j) to fill templates and generate text sequences. We evaluate three templates (an example for each template, extracted from Figure 1, is provided in brackets):

tmp_1 : entry $[num]$: $[l_x]$ is $[text_x]$; $[l_y]$ is $[text_y]$ (entry one: athlete is usain bolt ; rank is 1);
 tmp_2 : “row $[num]$: $[l_x]$ is $[text_x]$; $[l_y]$ is $[text_y]$ ” (“row 0: athlete is usain bolt ; rank is 1”);
 tmp_3 : “[l_x] is $[text_x]$ when $[l_y]$ is $[text_y]$ ” (“athlete is usain bolt when rank is 1”).

The placeholder $[l_x]$ is replaced with the x-axis label from the chart (e.g. “rank” in Figure 1). Similarly, the y-axis label (e.g. “athlete”) replaces $[l_y]$. Based on the coordinates, we classify a bounding boxes that contain axes labels (i.e. the boxes with the largest y coordinates).

A counter starting from *one* replaces $[num]$ and numbers the bars in the chart. We fill $[text_y]$ and $[text_x]$ with text regions detected as bar labels and axis ticks given their positions.

3.4 Encoding and Classification

ChartBERT captures the structure of charts through three learned embeddings: the x coordinate embedding which captures the horizontal location of the text in the chart, the y coordinate embedding which captures the vertical location, and the label embedding which takes value 1 if the text region is part

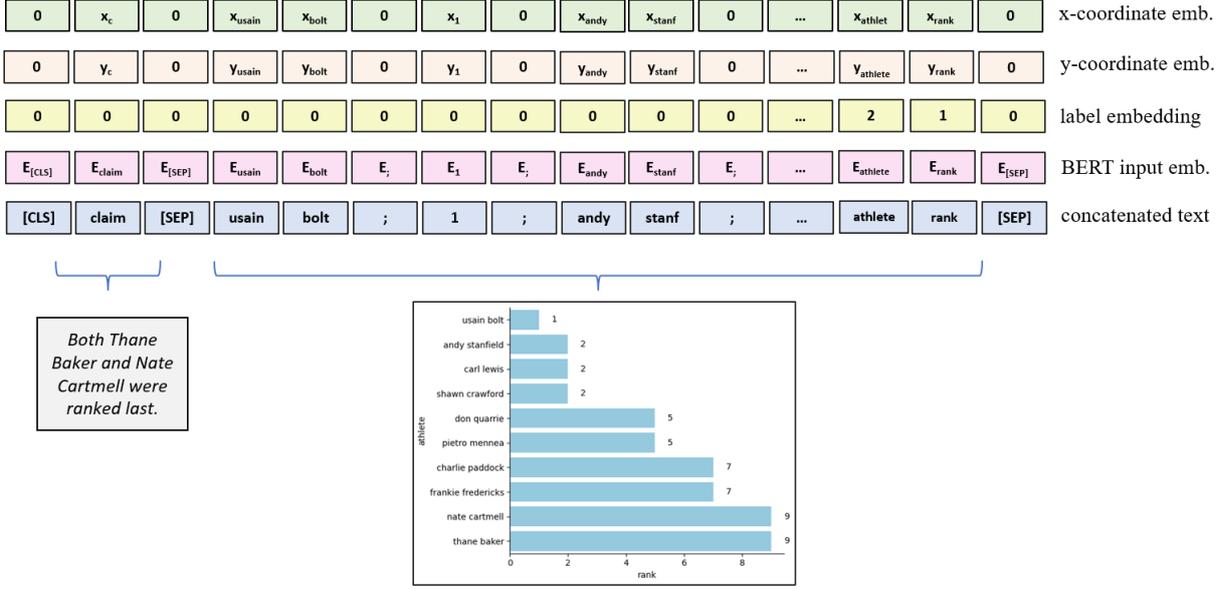


Figure 3: ChartBERT input representation with the text extracted from the chart and concatenated following the approach in Section 3.3. We include additional structural embeddings (i.e. x and y coordinates and label embeddings) to the BERT input embeddings (i.e. token, segment and position embeddings).

of the x -axis label (l_x), 2 if the text region is part of the for y -axis label (l_y) and 0 otherwise.

Figure 3 shows an example of the encoder with the structural embeddings. We concatenate claim c_i and sequence s_i , separate them with a $[SEP]$ token, add $[CLS]$ as the first input token, and feed the resulting vector as input to ChartBERT which generates 768-dimensional representations $h_i \in \mathbb{R}_{768}$. Finally, we pass h_i through a fully connected layer and determine the predicted label using sigmoid. ChartBERT uses binary cross entropy to minimize loss on the training set.

$$inp_i = (c_i, s_i, \{x_j, y_j, l_j^x, l_j^y\}_{j=1}^n)$$

$$h_i = f_{Encoder}(inp_i)$$

$$p_i = \sigma(f_{FC}(h_i))$$

4 Evaluation

For evaluation, we first create a new dataset, ChartFC. We compare ChartBERT with several VL baselines, each comprising three components: a vision encoder, a language encoder, and a fusion block to obtain joint representations. We evaluate the dataset size and potential biases, discuss results obtained with ChartBERT and the baselines, and analyse reasoning types the models fail on.

4.1 ChartFC Dataset

This section provides an overview of the ChartFC dataset and its creation process. Each dataset entry comprises a natural language claim, a chart image, and a label $\in \{supports, refutes\}$.

4.1.1 The TabFact Dataset

We use TabFact (Chen et al., 2020a) as a seed dataset. TabFact is a table fact-checking dataset of natural language claims and tables extracted from Wikipedia as evidence, where the veracity of the claim is decided based on the accompanying table. Claims were written and evaluated by human crowdworkers with at least 95% approval rates for prior tasks and more than 500 accepted HITs on Amazon Mechanical Turk. The inter-annotator agreement for the claim verification task is *Fleiss* $\kappa = 0.75$.

4.1.2 Creation Pipeline

Figure 4 shows the dataset creation process.⁴ Starting with 117,784 claims and 16,000 Wikipedia tables from TabFact, we first generate sub-tables. To link the claim text to table columns, we (i) lemmatize and tokenize the claim and the table content, (ii) link claim tokens to column headers and cells using string matching and heuristic rules, and (iii) decide if a claim token is linked to multiple columns using the minimum *Levenshtein distance*

⁴Figure 8 in the Appendix A illustrates the pipeline.

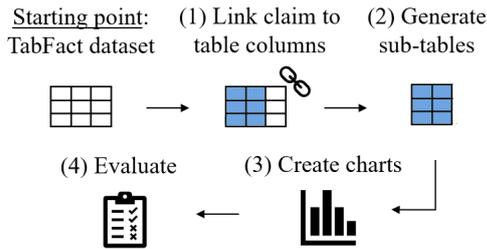


Figure 4: Dataset creation process.

	Train	Valid	Test	Sum
Support	7,048	896	885	8,829
Refute	5,654	697	706	7,057
Sum	12,702	1,593	1,591	15,886

Table 1: Class distribution across dataset split.

(Levenshtein, 1966), and finally, (iv) filter sub-tables with a maximum of twenty rows and two linked columns. This results in a total of 15,886 pairs of claims and sub-tables.

Finally, we generate charts using the Python libraries *seaborn* and *matplotlib*. The charts vary across the dimensions (i) orientation (horizontal, vertical); (ii) bar colors (green, blue, pink); and (iii) background (no/white grid lines, white/gray background color). We show an example in Figure 1. We partition the dataset into training, validation, and test sets using 8:1:1 ratio and show statistics in Table 1.

4.1.3 Dataset Evaluation

To assess the data quality, we apply human and automated evaluation. We evaluate the sub-table generation step (step 2 in Figure 4) by checking the verifiability of claims against the extracted sub-tables with TableBERT (Chen et al., 2020a). We obtain 69.3% accuracy on our test set, comparable to 65.1% accuracy reported by Chen et al. (2020a) on their test set.

For human validation, we extract 100 random dataset entries and manually evaluate the claims against sub-tables and charts. Of the 100 claims, 92 were successfully verifiable against their sub-tables and chart images, six claims were not verifiable because a relevant column was missing in the sub-table, and two claims were already mislabelled in the TabFact dataset.

4.1.4 Chart Reasoning Types

We label 100 random test samples with *chart reasoning types*, using a taxonomy of common reasoning types humans apply while interacting with data

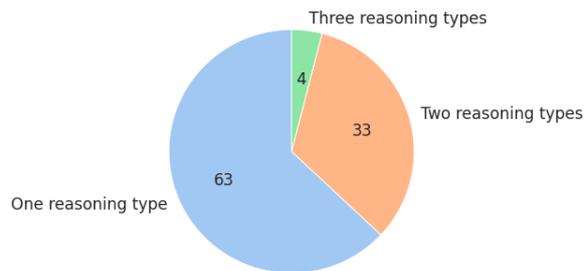


Figure 5: Number of chart reasoning types found in 100 dataset entries.

visualisations (Amar et al., 2005). We find seven reasoning types present in our data: *retrieve value*, *filter*, *comparison*, *compute derived value*, *find extremum*, *determine range*, and *find anomalies*.⁵ On average, we find 1.4 different types per claim with most claims including either one or two different reasoning types (see Figure 5). The reasoning type *retrieve value*, which requires extracting a value from the chart image given certain criteria, occurs most frequently (51%), followed by *find extremum*, i.e. highest or lowest values in the chart, and *filter*, which occur in approximately a quarter of all labelled claims. More complex types such as *compute derived value* or extracting all values in a given *range* are less frequent.

4.2 Vision-Language Baselines

We evaluate our task with several VL baselines, which jointly use claim text and visual information from images for claim verification. We also assess the top-3 VL baselines with OCR-extracted chart text as additional input. Each baseline consists of a language encoder, a vision encoder, and a fusion component to obtain joint representations. We systematically evaluate various state-of-the-art encoders and fusion techniques: we use shallow (e.g. BERT Embedder (Chen et al., 2020b)) and deep encoders (e.g. DenseNet (Huang et al., 2017)), as well as model-agnostic (e.g. concatenation) and model-based (e.g. transformer layers) fusion methods.

Language encoders: Given a claim c_i , we use a language encoder to obtain a feature vector:

$$h_i^{\text{text}} = f_{\text{LangEncoder}}(c_i)$$

We experiment with three language encoders:

BERT Embedder: Following Chen et al. (2020b),

⁵We describe the chart reasoning types in detail and give examples in Appendix B.

we tokenize the claim text into sub-words. For each token, we add the word and position embeddings to obtain the text representation which we then pass through a normalization (Ba et al., 2016) layer.

LSTM: We encode the text with 32-dimensional word embeddings and pass them through two LSTMs (Hochreiter and Schmidhuber, 1997) with 768-dimensional hidden states in each layer. We use the hidden states of the second layer as text representations.

BERT: We use a twelve-layer BERT encoder, initialized with weights from a pretrained BERT-base model.

Vision encoders: We use a vision encoder to extract representations for the chart images:

$$h_i^{\text{img}} = f_{VisEncoder}(img_i)$$

We evaluate five vision encoders:

Fully connected layer: We use a fully connected layer to extract 768-dimensional representations per image $h_i^{\text{img}} \in \mathbb{R}_{768}$.

AlexNet: Using AlexNet (Krizhevsky et al., 2012), for each image, we obtain a representation vector $h_i^{\text{img}} \in \mathbb{R}_{1024}$ by extracting the model output after the third max pooling layer.

ResNet: We use ResNet-152 (He et al., 2016) to obtain 2048-dimensional image representations by extracting the model output before the two final layers of ResNet-152, i.e. before the average pooling layer.

DenseNet: We use a DenseNet (DN) (Huang et al., 2017) comprising three dense blocks, with 6, 12, and 24 layers, respectively. We extract and concatenate the output of the first and third dense block as low- and high-level feature vectors: $h_i^{\text{img}} = f_{concat}(f_{DN[block1]}(img_i); f_{DN[block3]}(img_i))$.

Vision Transformer (ViT): We split images into sequences of n 16x16 patches before using them as input to a pretrained base-ViT model (Dosovitskiy et al., 2021).⁶ We extract the hidden states from the model’s final layer and use them as image representations, resulting in 768-dimensional vectors for each patch: $h_i^{\text{img}} = [h \in \mathbb{R}_{768}]_n$.

Fusion methods: We then fuse the text and image representations:

$$h_i^{\text{joint}} = f_{Fusion}(h_i^{\text{img}}; h_i^{\text{text}})$$

We experiment with five fusion methods:

Concatenation and multiplication: Concatena-

⁶<https://huggingface.co/google/vit-base-patch16-224>

tion and multiplication are common baseline approaches for multimodal fusion (Baltrušaitis et al., 2018). We reshape the text and image representations and either (i) concatenate both vectors, or (ii) perform element-wise multiplication.

Concatenation with GRUs: Inspired by Kafle et al. (2020), we concatenate the text and image representations and pass the resulting vector through m 1x1 convolutional layers and two GRUs. The first GRU takes the input in a forward direction, while the second GRU processes the input vector in a backwards direction to incorporate contextual information:

$$h_i^{\text{concat}} = f_{conv}(f_{concat}\{h_i^{\text{img}}; h_i^{\text{text}}\})$$

$$h_i^{\text{joint}} = f_{concat}\{f_{GRU}(h_i^{\text{concat}}); f_{GRU}(h_i^{\text{concat}})\}$$

Multimodal Compact Bilinear Pooling (MCB):

MCB is an efficient and popular baseline for multimodal fusion (Fukui et al., 2016). The text and image representations are each projected to a higher dimensional space using the projection function Count Sketch (Charikar et al., 2004). The outer product of the projected vectors is then calculated in Fast Fourier Transform space to obtain a joint representation for both modalities and thus reduce the amount of learnable parameters during model training.

Transformer layers: Given the recent popularity of transformer layers used for joining text and visual representations (Tan and Bansal, 2019; Chen et al., 2020b; Yang et al., 2021), we use a three-layer transformer to get cross-modal embeddings.

The representation h_i^{joint} is passed through two fully-connected layers and sigmoid to obtain the classification. We use binary cross entropy loss and stratified sampling in each training batch to minimize the loss on the training set.

4.3 Experimental Setup

We perform hyper-parameter search on the validation set and select the best-performing combination from the following values: $\{8, 16, 32\}$ for batch size, $\{1e^{-3}, 7e^{-4}, 5e^{-5}, 5e^{-6}, 5e^{-7}\}$ for learning rate, $\{1, \dots, 50\}$ for training epochs with early stopping. We also experimented with different learning rates for the language and vision encoders. Ultimately, we used one learning rate for the entire VL model as the modality-specific learning rates did

SeqGen	Val Acc	Val F ₁	Test Acc	Test F ₁
concat.	59.2	55.1	60.6	57.0
temp. <i>tmp</i> ₁	62.4	59.1	63.3	61.0
temp. <i>tmp</i> ₂	62.0	59.4	61.9	58.7
temp. <i>tmp</i> ₃	62.1	59.7	63.8	61.1

Table 2: Results for ChartBERT with different sequence generation (SeqGen) approaches: **concatenation** and **template**.

V-Encoder	Fusion	no OCR	text concat
ViT	concat GRU	59.8	60.5
ResNet	mult	60.1	61.3
ResNet	concat	59.8	62.7

Table 3: Test accuracy of top-3 VL baselines: without (**no OCR**) chart text and chart **text concatenated**. All models use BERT as language encoder.

not provide any performance gains.⁷

We run all experiments on a single NVIDIA Tesla V100 GPU with 32GB RAM. We measure model performance with prediction accuracy and (macro) F_1 on the test dataset.

4.4 Results & Discussion

How does ChartBERT perform on the task? How do different approaches for sequence generation influence model performance?

Table 2 gives an overview of the results obtained by ChartBERT. The best ChartBERT variant yields 63.8% test accuracy and processes chart text into text sequences using the template *tmp*₃. Compared to the concatenation approach, using *tmp*₃ increases the accuracy by +3.2%.

Interestingly, the choice of template design impacts the model performance only slightly. While template *tmp*₃ might seem more “natural” to humans, it does not yield much higher performance compared to *tmp*₂.

How do VL baselines perform on ChartFC? How does the selection of encoder or fusion method impact model performance?

In contrast to many state-of-the-art VL approaches that use simple vision encoders and attention-based fusion (Chen et al., 2020b; Kim et al., 2021; Xia et al., 2021), the three best-performing VL models on ChartFC use BERT as language encoder, ViT or ResNet to obtain image representations, and either concatenation, multiplication, or concatenation with GRUs as a fusion method. Using only the claim and chart as input

⁷The hyper-parameters for each VL baseline can be found in our GitHub repo.

(i.e. without the OCR-extracted chart text), the highest test accuracy we obtain is 60.1% with the model consisting of BERT, ResNet, and multiplication fusion (see Table 3).

Regarding the language encoder,⁸ models that use BERT perform best, irrespectively of the vision encoder and fusion method: the best LSTM-based model achieves 56.1% test accuracy and the best model with BERT embedder yields 56.5% accuracy, both lower than the best BERT-based VL model with 60.1% accuracy. In contrast, we obtain similar accuracy scores across different vision encoder: for example, replacing ResNet in Table 3 row two with a fully connected layer reduces the accuracy slightly by 0.6% to 59.7%. The choice of fusion method does not impact performance strongly: while using multiplication mostly outperforms other methods by a small margin, no fusion method stands out across all vision and language encoders. We also evaluate the chartQA model PReFIL (Kafle et al., 2020), which uses LSTM as language encoder, DenseNet for image representations, and concatenation with GRUs for fusion, and obtain on ChartFC a low test accuracy of 55.6%.

How does OCR-extracted chart text influence performance of VL models?

In addition to claim text and chart images used in VL baselines, we also include the text extracted from the charts through OCR as input (see Sections 3.2 and 3.3 for details). Table 3 shows that using the concatenated chart text as input improves accuracy compared to the models that do not use the chart text (e.g. from 59.8% to 62.7%). The highest accuracy 62.7% is obtained with the BERT-ResNet-concatenation baseline.

Do models fail on particular chart reasoning types?

We evaluate the best VL baseline, consisting of BERT, ViT, and concatenation with GRUs, on the chart reasoning types present in ChartFC and described in Section 4.1.4. We find that the model performs best on the reasoning types *retrieve value*, *filter*, and *finding extremum*, while struggling particularly with *compute derived values*. Figure 6 shows that the model classifies correctly 65% (i.e. 33 out of 51) of claims that require *retrieval* and 61% of claims that require *filtering*. However, only 50% of *comparison* claims and 38% of claims required to *compute derived values* are correctly predicted.

⁸The complete set of results obtained with different encoders and fusion methods can be found in Tables 5, 6, and 7 in the Appendix.

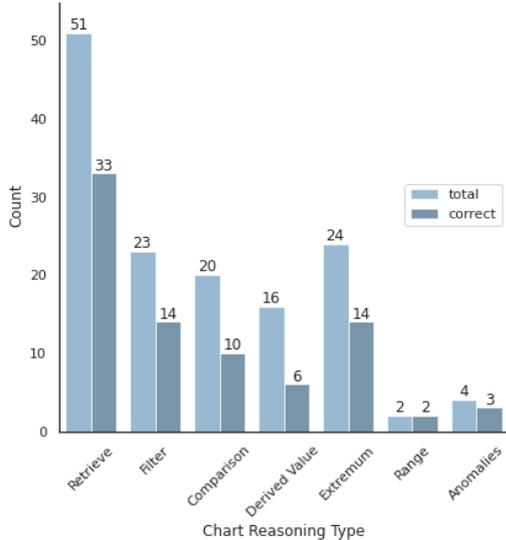


Figure 6: Chart reasoning types: total count and correct predictions of manually annotated test samples.

These results are in line with previous works that discuss limitations of state-of-the-art models in tasks requiring numerical reasoning capabilities (Thawani et al., 2021).

Is the dataset size sufficient for our proposed task? Do ChartFC claims contain biases?

We evaluate the size of the dataset by training our VL baseline (i.e. using BERT, ViT, and concatenation with GRUs) on various subsets of the training data as shown in Table 4 and report the accuracy on the test set. The performance on the test set improves as the number of training samples increases. While the performance gain is high when increasing the training set from 1% to 25% (51.6% accuracy compared to 57%), the difference in accuracy between the baseline trained on half of the training data and the entire training data is only 2.6%, indicating that our training set has a reasonable size.

We also train a claim-only BERT model to determine whether claims contain biases that allow the model to correctly predict the label while ignoring the evidence charts. Trained on the claim text only, the model achieves 52% accuracy on the test set, compared to ChartBERT’s accuracy of (63.8%). We conclude that the claim text itself is not sufficient for correct classification.

What are the dis-/advantages of an automated dataset pipeline for chart fact-checking?

We automatically create ChartFC using a table fact-checking dataset as seed by identifying subtables relevant to the claims and then building the

Training Samples	Test Accuracy
127 (1%)	51.6
3,175 (25%)	57.0
6,351 (50%)	57.1
9,526 (75%)	58.0
12,702 (100%)	59.8

Table 4: Performance of VL baseline (BERT, ViT, and concatenation with GRUs) with different training set sizes.

charts. ChartFC includes common stylistic variations: bars of different colors, horizontal/vertical orientations, different backgrounds (light/dark, grid lines/no grid lines). While natural charts come with large stylistic variation, using them results in reduced control over task complexity and dataset. In future work, we plan to explore two alternative dataset creation pipelines: first, automated pipelines for other charts types to extend the current dataset, and second, a pipeline with natural charts where we would create claims for charts.

Using natural charts would require a multi-step annotation process: selecting and separating charts from other images (Vougiouklis et al., 2020); writing claims which support/refute them; evaluating the claims to check for correctness, typos, etc. We would require annotators with proficiency in interpreting charts, and with basic mathematical and language skills to create claims with different reasoning types (see Figure 5).

5 Conclusion and Future work

We propose the chart fact-checking task and introduce ChartBERT, a novel model for fact-checking claims against chart images comprising three main components: a reading component, a sequence generation component, and an encoder that extends BERT’s encoder with structural embeddings. We also introduce ChartFC as the first dataset for fact-checking against chart images, consisting of 15, 886 claims and chart images.

ChartBERT achieves 63.8% accuracy on ChartFC. We systematically evaluate 75 different VL baselines, using various language encoders, vision encoders, and fusion methods. The highest-performing VL baseline uses BERT as language encoder, ResNet to extract image representations, and concatenation to obtain joint representations for both modalities. The model achieves 62.7% test accuracy. Our results indicate that chart fact-checking, which requires extracting and reasoning over text and structural information from charts, is

a challenging task for future research on AFC and VL methods.

Limitations

The TabFact dataset (Chen et al., 2020a) has been a valuable resource for creating ChartFC. However, using it as (the sole) seed dataset has limitations.

ChartFC consists of bar charts only; indeed, given the claims and tables found in TabFact, the bar chart was deemed the most appropriate chart type. Various types of charts exist (e.g. pie charts, line charts) and their effectiveness in different data contexts and tasks has been investigated in the literature. For example, Saket et al. (2019) evaluated the effectiveness of chart types using crowdsourcing experiments across the chart reasoning types we discussed in Section 4.1.4. In the context of small datasets, i.e. up to 34 rows and two columns which is similar to our setting, Saket et al. (2019) found bar charts to be the most accurate visualization type for the given chart reasoning types. In addition to bar charts, other types of charts used as evidence for fact-checking tasks ought to be investigated. Behrisch et al. (2018) studied visualization methods for different data types (i.e. multi- and high-dimensional data, relational data, geo-spatial data, sequential and temporal data, and text data). For example, they found that scatter plots were appropriate visualization types for queries regarding data distribution (e.g. correlations and clusters), while line charts were more appropriate for queries about temporal aspects of data. To extend ChartFC with other chart types, we require more diverse data types (e.g. sequential and temporal data) and appropriate claims.

Moreover, ChartFC claims are restricted to English, whereas misinformation is commonly spread in different languages. Future work is necessary to address the limited availability of non-English fact-checking datasets and to contribute to the efforts done in this space (Gupta and Srikumar, 2021).

References

Sara Abdali. 2022. [Multi-modal misinformation detection: Approaches, challenges and opportunities](#). *CoRR*, abs/2203.13883.

Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. 2022. [PubHealthTab: A public health table-based dataset for evidence-based fact checking](#). In *Findings of the Association for Computational Linguistics*:

NAACL 2022, pages 1–16, Seattle, United States. Association for Computational Linguistics.

- Firoj Alam, Stefano Cresci, Tanmoy Chakraborty, Fabrizio Silvestri, Dimiter Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2021. [A survey on multimodal disinformation detection](#). *CoRR*, abs/2103.12541.
- Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. [FEVEROUS: fact extraction and verification over unstructured and structured information](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS*.
- Robert A. Amar, James Eagan, and John T. Stasko. 2005. [Low-level components of analytic activity in information visualization](#). In *IEEE Symposium on Information Visualization (InfoVis)*, pages 111–117. IEEE Computer Society.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. [Layer normalization](#). *arXiv preprint arXiv:1607.06450*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. [Neural machine translation by jointly learning to align and translate](#). In *3rd International Conference on Learning Representations, ICLR*.
- Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. [Multimodal machine learning: A survey and taxonomy](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.
- Michael Behrisch, Michael Blumenschein, Nam Wook Kim, Lin Shao, Mennatallah El-Assady, Johannes Fuchs, Daniel Seebacher, Alexandra Diehl, Ulrik Brandes, Hanspeter Pfister, Tobias Schreck, Daniel Weiskopf, and Daniel A. Keim. 2018. [Quality metrics for information visualization](#). *Comput. Graph. Forum*, 37(3):625–662.
- Efrat Blaier, Itzik Malkiel, and Lior Wolf. 2021. [Caption enriched samples for improving hateful memes detection](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9350–9358, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Juan Cao, Peng Qi, Qiang Sheng, Tianyun Yang, Junbo Guo, and Jintao Li. 2020. [Exploring the role of visual content in fake news detection](#). *CoRR*, abs/2003.05096.
- Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. 2004. [Finding frequent items in data streams](#). *Theor. Comput. Sci.*, 312(1):3–15.
- Ritwick Chaudhry, Sumit Shekhar, Utkarsh Gupta, Pranav Maneriker, Prann Bansal, and Ajay Joshi.

2020. **LEAF-QA: locate, encode & attend for figure question answering**. In *IEEE Winter Conference on Applications of Computer Vision, WACV*, pages 3501–3510. IEEE.
- Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyong Zhou, and William Yang Wang. 2020a. **TabFact: A large-scale dataset for table-based fact verification**. In *8th International Conference on Learning Representations, ICLR*. OpenReview.net.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020b. **UNITER: universal image-text representation learning**. In *Computer Vision - ECCV - 16th European Conference*, volume 12375 of *Lecture Notes in Computer Science*, pages 104–120. Springer.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Thomas Diggelmann, Jordan L. Boyd-Graber, Janis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. **CLIMATE-FEVER: A dataset for verification of real-world climate claims**. *CoRR*, abs/2012.00614.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. **An image is worth 16x16 words: Transformers for image recognition at scale**. In *9th International Conference on Learning Representations, ICLR*. OpenReview.net.
- Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. **Multimodal compact bilinear pooling for visual question answering and visual grounding**. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468, Austin, Texas. Association for Computational Linguistics.
- Ashim Gupta and Vivek Srikumar. 2021. **X-fact: A new benchmark dataset for multilingual fact checking**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 675–682, Online. Association for Computational Linguistics.
- Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. **INFOTABS: Inference on tables as semi-structured data**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. **Deep residual learning for image recognition**. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 770–778. IEEE Computer Society.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. **Long short-term memory**. *Neural Computation*, 9(8):1735–1780.
- Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. **Densely connected convolutional networks**. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2261–2269. IEEE Computer Society.
- Kushal Kafle, Brian L. Price, Scott Cohen, and Christopher Kanan. 2018. **DVQA: understanding data visualizations via question answering**. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5648–5656. Computer Vision Foundation / IEEE Computer Society.
- Kushal Kafle, Robik Shrestha, Brian L. Price, Scott Cohen, and Christopher Kanan. 2020. **Answering questions about data visualizations using efficient bimodal fusion**. In *IEEE Winter Conference on Applications of Computer Vision, WACV*, pages 1487–1496. IEEE.
- Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2018. **Figureqa: An annotated figure dataset for visual reasoning**. In *6th International Conference on Learning Representations, ICLR*. OpenReview.net.
- Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. **Chart-to-text: A large-scale benchmark for chart summarization**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4005–4023, Dublin, Ireland. Association for Computational Linguistics.
- Anthony Kay. 2007. Tesseract: an open-source optical character recognition engine. *Linux Journal*, 2007(159):2.
- Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. **The hateful memes challenge: Detecting hate speech in multimodal memes**. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*.
- Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. 2020. **Answering questions about charts and generating visual explanations**. In *CHI '20: CHI Conference on Human Factors in Computing Systems*, pages 1–13. ACM.

- Wonjae Kim, Bokyoung Son, and Ildoo Kim. 2021. [Vilt: Vision-and-language transformer without convolution or region supervision](#). In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR.
- Neema Kotonya and Francesca Toni. 2020. [Explainable automated fact-checking for public health claims](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. [Imagenet classification with deep convolutional neural networks](#). In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems*, pages 1106–1114.
- Crystal Lee, Tanya Yang, Gabrielle D. Inchoco, Graham M. Jones, and Arvind Satyanarayan. 2021. [Viral visualizations: How coronavirus skeptics use orthodox data practices to promote unorthodox science online](#). In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pages 607:1–607:18. ACM.
- VI Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.
- Leo Yu-Ho Lo, Ayush Gupta, Kento Shigyo, Aoyu Wu, Enrico Bertini, and Huamin Qu. 2022. [Mis-informed by visualization: What do we learn from misinformative visualizations?](#) *Comput. Graph. Forum*, 41(3):515–525.
- Preslav Nakov, David P. A. Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021a. [Automated fact-checking for assisting human fact-checkers](#). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI*, pages 4551–4558. ijcai.org.
- Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Nikolay Babulkov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, and Thomas Mandl. 2021b. [The CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news](#). In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR*, volume 12657 of *Lecture Notes in Computer Science*, pages 639–649. Springer.
- Anshul Vikram Pandey, Anjali Manivannan, Oded Nov, Margaret Satterthwaite, and Enrico Bertini. 2014. [The persuasive power of data visualization](#). *IEEE Trans. Vis. Comput. Graph.*, 20(12):2211–2220.
- J Perkel. 2020. Behind the Johns Hopkins University coronavirus dashboard. *Nature Index*, 7.
- Jingnong Qu, Liunian Harold Li, Jieyu Zhao, Sunipa Dev, and Kai-Wei Chang. 2022. [Disinformeme: A multimodal dataset for detecting meme intentionally spreading out disinformation](#). *CoRR*, abs/2205.12617.
- Bahador Saket, Alex Endert, and Çagatay Demiralp. 2019. [Task-based effectiveness of basic visualizations](#). *IEEE Trans. Vis. Comput. Graph.*, 25(7):2505–2512.
- Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. [Get your vitamin C! robust fact verification with contrastive evidence](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.
- Shivam Sharma, Firoj Alam, Md. Shad Akhtar, Dimitar Dimitrov, Giovanni Da San Martino, Hamed Firooz, Alon Y. Halevy, Fabrizio Silvestri, Preslav Nakov, and Tanmoy Chakraborty. 2022. [Detecting and understanding harmful memes: A survey](#). In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5597–5606. ijcai.org.
- Hao Tan and Mohit Bansal. 2019. [LXMERT: Learning cross-modality encoder representations from transformers](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.
- Hao Tan, Chen-Tse Tsai, Yujie He, and Mohit Bansal. 2022. [Scientific chart summarization: Datasets and improved text modeling](#).
- Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro Szekely. 2021. [Representing numbers in NLP: a survey and a vision](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–656, Online. Association for Computational Linguistics.
- James Thorne, Max Glockner, Gisela Vallejo, Andreas Vlachos, and Iryna Gurevych. 2021. [Evidence-based verification for real world information needs](#). *CoRR*, abs/2104.00640.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. [FEVER: a large-scale dataset for fact extraction and VERification](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5998–6008.
- Pavlos Vougiouklis, Leslie Carr, and Elena Simperl. 2020. [Pie chart or pizza: Identifying chart types and their virality on twitter](#). In *Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020, Held Virtually, Original Venue: Atlanta, Georgia, USA, June 8-11, 2020*, pages 694–704. AAAI Press.
- Nancy X. R. Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021a. [SemEval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents \(SEM-TAB-FACTS\)](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 317–326, Online. Association for Computational Linguistics.
- Nancy Xin Ru Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021b. [Semeval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents \(SEM-TAB-FACTS\)](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP*, pages 317–326. Association for Computational Linguistics.
- William Yang Wang. 2017. [“liar, liar pants on fire”: A new benchmark dataset for fake news detection](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.
- Qiaolin Xia, Haoyang Huang, Nan Duan, Dongdong Zhang, Lei Ji, Zhifang Sui, Edward Cui, Taroon Bharti, and Ming Zhou. 2021. [XGPT: cross-modal generative pre-training for image captioning](#). In *Natural Language Processing and Chinese Computing - 10th CCF International Conference, NLPCC 2021, Qingdao, China, October 13-17, 2021, Proceedings, Part I*, volume 13028 of *Lecture Notes in Computer Science*, pages 786–797. Springer.
- Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florêncio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. 2021. [TAP: text-aware pre-training for text-vqa and text-caption](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8751–8761. Computer Vision Foundation / IEEE.
- Barry Menglong Yao, Aditya Shah, Lichao Sun, Jin-Hee Cho, and Lifu Huang. 2022. [End-to-end multimodal fact-checking and explanation generation: A challenging dataset and models](#). *CoRR*, abs/2205.12487.
- Yixuan Zhang, Yifan Sun, Lacey M. K. Padilla, Sumit Barua, Enrico Bertini, and Andrea G. Parker. 2021. [Mapping the landscape of COVID-19 crisis visualizations](#). In *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pages 608:1–608:23. ACM.
- Dimitrina Zlatkova, Preslav Nakov, and Ivan Koychev. 2019. [Fact-checking meets fauxtography: Verifying claims about images](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2099–2108, Hong Kong, China. Association for Computational Linguistics.

A Dataset Pipeline

ChartFC charts vary across the dimensions (*i*) orientation (horizontal, vertical); (*ii*) bar colors (green, blue, pink); and (*iii*) background (no/white grid lines, white/gray background color). Figure 7 shows multiple chart examples.

In Figure 8, we give an example of the dataset creation pipeline. Starting with the claim and initial TabFact table, we first filter columns required to decide the claims veracity label: “age at appointment” and “prior occupation”. This sub-table is used to create the evidence chart (bottom right).

B Chart Reasoning Types

We label 100 random test set samples with chart reasoning types. Next, we briefly describe each type, for more details we refer to the taxonomy by [Amar et al. \(2005\)](#):

- **Retrieve Value:** Given some conditions, retrieve a single value from the chart image.
- **Filter:** Find all data points in the chart that fulfill some specified conditions.
- **Compute Derived Value:** Calculate an aggregated value (e.g. average or count) using data points extracted from the chart.
- **Find Extremum:** Extract the top-*n* data points given some conditions.
- **Determine Range:** Based on some conditions, find a span of values such that all extracted data points fulfil the conditions.
- **Find Anomalies:** Find any anomalies in a specified set of data points.
- **Compare:** Compare the values of different data points to each other.

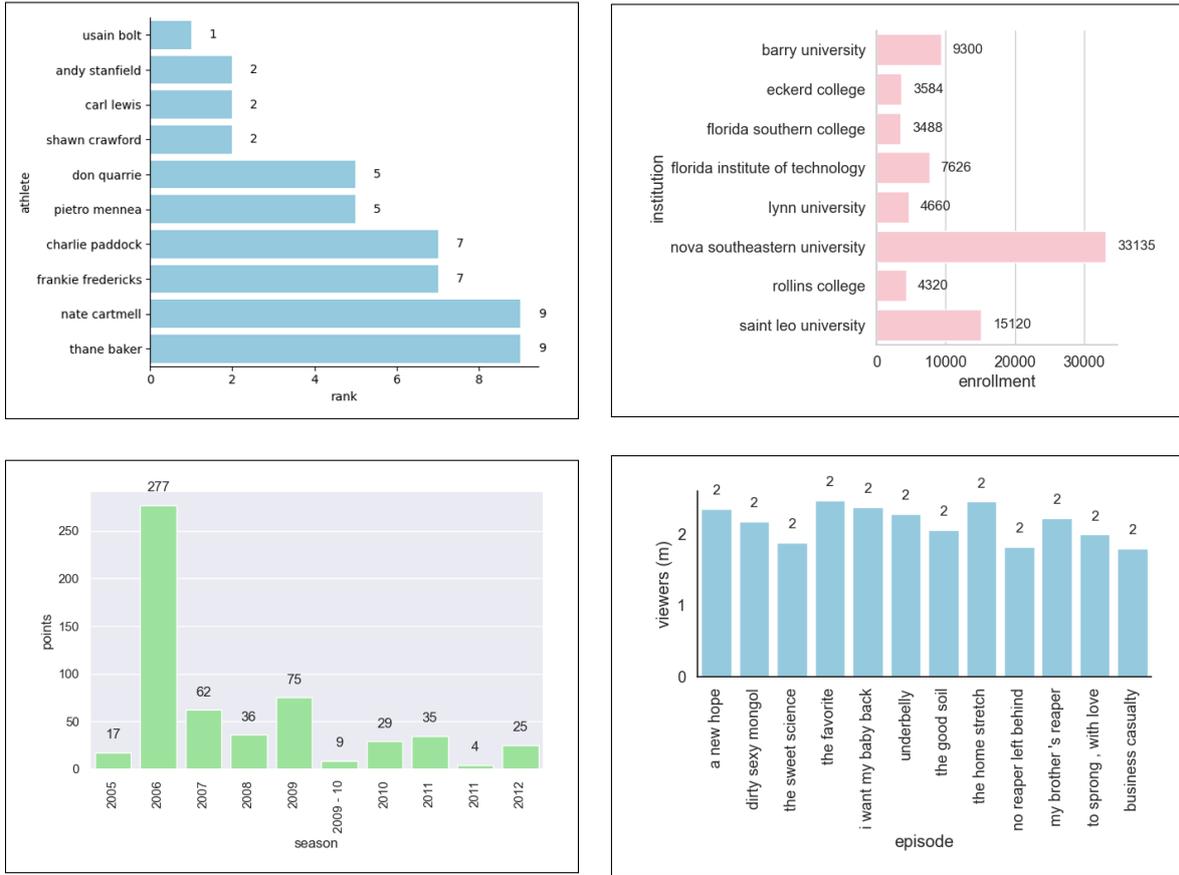


Figure 7: Examples from the ChartFC dataset.

Claim: There are four people who were appointed at secretary at the age of 50.

1. Initial table

	romanised name	chinese name	age at appointment	portfolio	prior occupation
0	donald tsang yam - kuen	曾蔭權	58	chief secretary for administration (cs)	chief secretary for administration (cs)
1	anthony leung kam - chung	梁錦松	50	financial secretary (fs)	financial secretary (fs)
2	elsie leung oi - see	梁愛詩	63	secretary for justice (sj)	secretary for justice (sj)
3	joseph wong wing - ping	王永平	54	secretary for civil service	secretary for civil service
4	henry tang ying - yen	唐英年	50	secretary for commerce , industry and technology	chairman , federation of hong kong industries

2. Subtable

	age at appointment	prior occupation
0	58	chief secretary for administration (cs)
1	50	financial secretary (fs)
2	63	secretary for justice (sj)
3	54	secretary for civil service
4	50	chairman , federation of hong kong industries

3. Chart

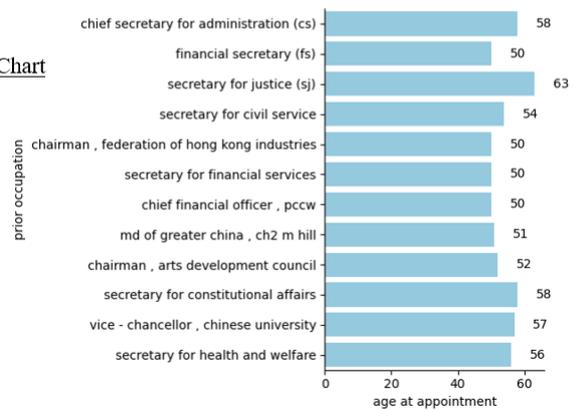


Figure 8: Example for dataset creation process.

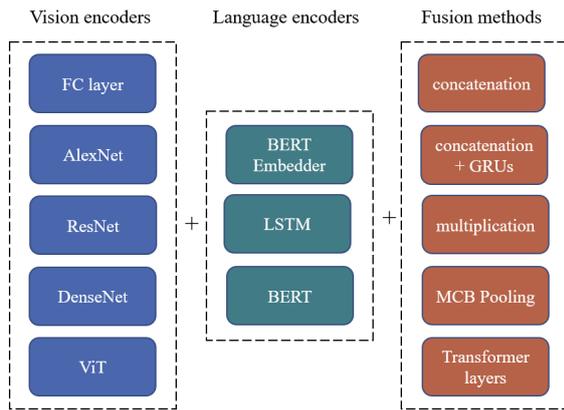


Figure 9: Encoders and fusion methods used in VL baselines.

C VL Baselines

Figure 9 provides an overview of all encoders and fusion methods we use in our evaluation.

Table 5, 6, and 7 provide an overview of all VL baselines we evaluated on ChartFC.

Lang Encoder	Vis Encoder	Fusion	Val Acc	Val F_1	Test Acc	Test F_1
BERT Emb	FC	concatenation	56.7	37.8	55.6	36.6
BERT Emb	FC	concatenation, biGRU	56.2	36.0	55.6	35.7
BERT Emb	FC	multiplication	56.6	52.8	56.5	52.3
BERT Emb	FC	MCB	56.2	36.1	55.6	35.7
BERT Emb	FC	Transformer layers	56.2	36.0	55.6	35.7
BERT Emb	AlexNet	concatenation	56.5	40.2	55.1	38.1
BERT Emb	AlexNet	concatenation, biGRU	56.2	36.0	55.6	35.7
BERT Emb	AlexNet	multiplication	57.0	41.4	55.9	39.9
BERT Emb	AlexNet	MCB	56.2	36.0	55.6	35.7
BERT Emb	AlexNet	Transformer layers	56.2	36.0	55.6	35.7
BERT Emb	ResNet 152	concatenation	56.5	45.4	56.2	45.5
BERT Emb	ResNet 152	concatenation, biGRU	56.2	36.0	55.6	35.7
BERT Emb	ResNet 152	multiplication	56.6	38.3	56.3	38.8
BERT Emb	ResNet 152	MCB	56.2	36.0	55.6	35.7
BERT Emb	ResNet 152	Transformer layers	56.2	36.0	55.6	35.7
BERT Emb	DenseNet (6, 12, 24)	concatenation	56.5	43.7	54.0	40.7
BERT Emb	DenseNet (6, 12, 24)	concatenation, biGRU	56.6	45.3	54.1	42.2
BERT Emb	DenseNet (6, 12, 24)	multiplication	56.5	37.1	55.6	36.4
BERT Emb	DenseNet (6, 12, 24)	MCB	56.2	36.1	55.6	35.7
BERT Emb	DenseNet (6, 12, 24)	Transformer layers	56.2	36.0	55.6	35.7
BERT Emb	ViT	concatenation	56.2	36.0	55.6	35.7
BERT Emb	ViT	concatenation, biGRU	56.2	36.0	55.6	35.7
BERT Emb	ViT	multiplication	57.1	42.1	54.8	37.6
BERT Emb	ViT	MCB	56.2	36.0	55.6	35.7
BERT Emb	ViT	Transformer layers	56.2	36.0	55.6	35.7

Table 5: VL baselines using BERT embedder for text encoding, different vision encoders, and fusion methods

Lang Encoder	Vis Encoder	Fusion	Val Acc	Val F_1	Test Acc	Test F_1
LSTM	FC	concatenation	56.6	36.9	55.5	35.8
LSTM	FC	concatenation, biGRU	56.2	36.0	55.6	35.7
LSTM	FC	multiplication	56.2	36.0	55.6	35.7
LSTM	FC	MCB	56.2	36.0	55.6	35.7
LSTM	FC	Transformer layers	56.2	36.0	55.6	35.7
LSTM	AlexNet	concatenation	56.3	39.6	56.1	39.8
LSTM	AlexNet	concatenation, biGRU	56.2	36.0	55.6	35.7
LSTM	AlexNet	multiplication	56.2	36.0	55.6	35.7
LSTM	AlexNet	MCB	56.2	36.0	55.6	35.7
LSTM	AlexNet	Transformer layers	56.2	36.0	55.6	35.7
LSTM	ResNet 152	concatenation	56.2	36.0	55.6	35.7
LSTM	ResNet 152	concatenation, biGRU	56.2	36.0	55.6	35.7
LSTM	ResNet 152	multiplication	56.2	36.0	55.6	35.7
LSTM	ResNet 152	MCB	56.4	36.3	56.0	35.9
LSTM	ResNet 152	Transformer layers	56.2	36.0	55.6	35.7
LSTM	DenseNet (6, 12, 24)	concatenation	56.2	36.0	55.6	35.7
LSTM	DenseNet (6, 12, 24)	concatenation, biGRU	56.2	36.0	55.6	35.7
LSTM	DenseNet (6, 12, 24)	multiplication	56.2	36.0	55.6	35.7
LSTM	DenseNet (6, 12, 24)	MCB	56.2	36.0	55.6	35.7
LSTM	DenseNet (6, 12, 24)	Transformer layers	56.2	36.0	55.6	35.7
LSTM	ViT	concatenation	56.2	36.0	55.6	35.7
LSTM	ViT	concatenation, biGRU	56.2	36.0	55.6	35.7
LSTM	ViT	multiplication	56.2	36.0	55.6	35.7
LSTM	ViT	MCB	56.3	36.7	55.7	36.5
LSTM	ViT	Transformer layers	56.2	36.0	55.6	35.7

Table 6: VL baselines with LSTM as language encoder, different vision encoders, and fusion methods

Lang Encoder	Vis Encoder	Fusion	Val Acc	Val F_1	Test Acc	Test F_1
BERT	FC	concatenation	59.3	50.7	59.6	51.0
BERT	FC	concatenation, biGRU	58.8	51.1	58.5	50.2
BERT	FC	multiplication	59.4	54.5	59.7	54.9
BERT	FC	MCB	59.7	49.6	59.1	49.3
BERT	FC	Transformer layers	56.2	36.0	55.6	35.7
BERT	AlexNet	concatenation	59.5	47.9	59.1	47.6
BERT	AlexNet	concatenation, biGRU	59.2	48.2	58.0	47.0
BERT	AlexNet	multiplication	59.0	56.2	59.6	57.0
BERT	AlexNet	MCB	58.8	45.2	57.4	43.9
BERT	AlexNet	Transformer layers	57.6	50.8	59.5	52.6
BERT	ResNet 152	concatenation	59.8	50.9	59.8	50.8
BERT	ResNet 152	concatenation, biGRU	59.1	47.0	58.8	46.7
BERT	ResNet 152	multiplication	59.3	52.2	60.1	53.6
BERT	ResNet 152	MCB	58.2	47.0	58.7	48.9
BERT	ResNet 152	Transformer layers	56.2	36.0	55.6	35.7
BERT	DenseNet (6, 12, 24)	concatenation	59.1	51.4	59.1	52.4
BERT	DenseNet (6, 12, 24)	concatenation, biGRU	60.2	53.0	59.0	51.0
BERT	DenseNet (6, 12, 24)	multiplication	59.4	49.2	58.7	48.7
BERT	DenseNet (6, 12, 24)	MCB	59.9	49.6	58.8	48.6
BERT	DenseNet (6, 12, 24)	Transformer layers	58.7	48.0	58.1	46.8
BERT	ViT	concatenation	56.2	36.0	55.6	35.7
BERT	ViT	concatenation, biGRU	59.0	51.2	59.8	51.7
BERT	ViT	multiplication	58.0	42.7	56.6	41.1
BERT	ViT	MCB	59.2	49.5	59.2	49.6
BERT	ViT	Transformer layers	57.1	40.8	55.9	39.1

Table 7: VL baselines with BERT as language encoder, different vision encoders, and fusion methods

Causal Reasoning About Entities and Events in Procedural Texts

Li Zhang^{*,*}, Hainiu Xu^{*,*}, Yue Yang^{*,*}, Shuyan Zhou[◇],
Weiqiu You^{*,*}, Manni Arora^{*,*}, Chris Callison-Burch^{*,*}

^{*,*}University of Pennsylvania [◇]Carnegie Mellon University

{zharry, seacow, yueyang1, weiqiuy, manni, ccb}@seas.upenn.edu
{shuyanzh}@cs.cmu.edu

Abstract

Entities and events are crucial to natural language reasoning and common in procedural texts. Existing work has focused either exclusively on entity state tracking (e.g., *whether a pan is hot*) or on event reasoning (e.g., *whether one would burn themselves by touching the pan*), while these two tasks are often causally related. We propose CREPE, the first benchmark on causal reasoning of event plausibility and entity states. We show that most language models, including GPT-3, perform close to chance at .35 F1, lagging far behind human at .87 F1. We boost model performance to .59 F1 by creatively representing events as programming languages while prompting language models pretrained on code. By injecting the causal relations between entities and events as intermediate reasoning steps in our representation, we further boost the performance to .67 F1. Our findings indicate not only the challenge that CREPE brings for language models, but also the efficacy of code-like prompting combined with chain-of-thought prompting for multihop event reasoning.¹

1 Introduction

Event-centric natural language processing (Chen et al., 2021b) is one of the leading paradigms in machine understanding of texts. This line of work focuses on first extracting entities and events from texts (Yang et al., 2019; Du and Cardie, 2020) and then making inferences about them (Li et al., 2020; Du et al., 2021). Even with the recent advances of large language models (LLMs), reasoning about events remains challenging as it requires highly contextual information and ample common-sense knowledge. For example, the event “*adding water to a pan containing hot oil*” causes the event “*there is a sizzling sound*” to happen, while “*heat up an*

^{*}Equal contribution.

¹Data and code can be found at https://github.com/zharry29/causal_reasoning_of_entities_and_events.

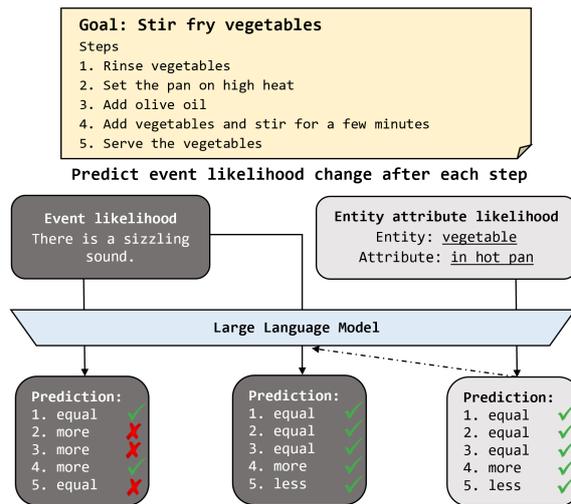


Figure 1: Example of our task CREPE. A procedure including a goal and some steps are provided. A model needs to predict the change in the likelihood of an event throughout the procedure. We show that predicting entity states as an intermediate step improves performance.

empty pan” does not. Any model that can draw the correct conclusion given these contexts is expected to have access to some implicit knowledge about these entities and events.

One type of text which demonstrates these challenges is procedural text, namely sequences of events, such as how-to instructions, recipes, natural processes, scientific protocols, etc. Procedural texts describe an environment that changes dynamically through a sequence of steps. Therefore, the exact environment configuration is often implicit. In the previous cooking example, whether “*there is a sizzling sound*” depends on what steps have taken place. With these interesting challenges coupled with the added benefit of application to robotics (Brohan et al., 2022) and household smart assistants such as Alexa (Panagopoulou et al., 2022), reasoning about procedures attracts great attention from the NLP community (Zhang, 2022).

Most work on reasoning about procedural texts has focused solely on either predicting the proper-

ties of events (e.g., which event is more likely to happen) (Zhang et al., 2020c; Yang et al., 2021b; Tandon et al., 2019) or tracking entity states (e.g., what is some property of an entity after some step) (Dalvi et al., 2018; Tandon et al., 2020), while the causal relation between events and entities is largely underexplored – for example, whether “*there is a sizzling sound*” is determined by the state of “*water*” and “*oil*.” Therefore, we claim that many event prediction tasks are multihop reasoning tasks that require the knowledge of intermediate entity states. Causal reasoning about events and entities differs from existing multihop reasoning tasks, such as Yang et al. (2018); Dua et al. (2019) whose reasoning process is explicitly formulated by a direct question (e.g., *how old is the previous US president*); and Geva et al. (2021) whose supporting evidence is factual and static. In contrast, causal reasoning in procedures requires models to first figure out the relevant entity attributes, then infer their states based on the current context, and finally predict the event.

To this end, we propose the task of **Causal Reasoning of Entities and Events in Procedural Texts (CREPE)**, with an overview in Figure 1. Given a procedure consisting of a goal (“*stir fry vegetables*”) and some steps (“*rinse vegetable*”...), a model is to predict the likelihood of some unobserved events (“*there is a sizzling sound*”) after the execution of each step. We provide a handcrafted, high-quality benchmark containing 183 procedures, 1219 steps, and 324 changes in the likelihood of events along with the corresponding underlying entity state changes. In an in-context learning setting, we show that most LLMs including GPT-3 (Brown et al., 2020) perform no better (.350 F1) than chance (.297 F1), greatly underperforming the human performance of .868 F1, on the development set. Providing ground-truth entity state changes to the prompt of GPT-3 shows no performance gain, indicating that it cannot leverage this causal signal. Instead, we draw inspiration from Madaan et al. (2022) who represented texts as programming languages as the prompt to code language model Codex (Chen et al., 2021a) to perform event reasoning. We propose a novel Python code representation of procedures that achieves .585 F1. Furthermore, our code-like representation allows us to effectively encode and leverage predicted or labeled entity state changes by generating them as an intermediate reasoning step (namely, chain-of-

thought), boosting the performance to .667 using predicted entity state changes and .715 F1 using labeled entity state changes.

Our contributions are summarized as follows:

- We propose a novel task, a dataset, and several strong baselines for causal reasoning about events and entities in procedural texts.
- We devise an effective code-like representation of procedures, leading to superior performance and allowing the injection of structured knowledge for reasoning.
- We are among the first to show that code language models can apply chain-of-thought to tackle multihop reasoning.

2 Task and Hypothesis

A procedure P of length n consists of a goal G and some steps $s_1 \dots s_n \in S$, each represented as a short sentence. Each procedure is associated with a set of hypothetical events $e_1 \dots e_m \in E$ whose likelihood of happening changes throughout the procedure. The task is to predict the change of likelihood of a hypothetical event e_j from step s_{i-1} (the previous step) to step s_i (the current step):

$$\delta_i = p(e_j | s_i, \dots, s_1, G) - p(e_j | s_{i-1}, \dots, s_1, G)$$

The likelihood change δ_i is positive if the label is “more likely”, negative if “less likely”, or zero if “equally likely”.

Predicting the likelihood of hypothetical events, also known as counterfactual reasoning, is extremely important for machine reasoning (Pearl and Mackenzie, 2018) (see more in Section 7). In our work, we hypothesize that **the causal relation between entity changes and events** can be leveraged by LLMs to better perform counterfactual reasoning. In other words, any change of the likelihood of a hypothetical event is given rise to by changes of some entity attributes $a_1 \dots a_m \in A$.

$$\delta_i = p(a_j | s_i, \dots, s_1, G) - p(a_j | s_{i-1}, \dots, s_1, G)$$

3 Dataset

Our CREPE benchmark dataset has two portions. The first is handcrafted and cross-validated by six authors of this paper. The annotation happens in 3 phases: (1) we first write down or acquire a procedure from the web; (2) we then annotate some hypothetical events whose likelihood of happening changes throughout the procedure, and how their likelihood change after each step; (3) for each

Data Statistics			
	Dev	Test	Total
Num. procedures	42	141	183
Num. steps	295	924	1219
Num. event changes	144	180	324
Avg. step per procedure	7.0	6.6	6.7
Avg. token per step	6.8	6.8	6.8
Procedure Topics			
	Dev	Test	Total
Recipe	10	33	43
Household	12	40	52
Craft	4	17	21
Technology	5	19	24
Travel	4	4	8
Sports	2	13	15
Others	5	15	20

Table 1: Statistics of the CREPE dataset.

event, we annotate a tuple of entity, attribute, and change that causes the event likelihood change. To obtain interesting and challenging data, we require annotators to write procedures covering a diverse range of topics and to prioritize events that undergo multiple likelihood changes, and those that involve information implicit from the steps. In our work, we strictly use this portion as the development set to inform all our experimental designs.

The second portion, designed to be drawn from a different distribution to minimize bias, was annotated by students in an Artificial Intelligence class at the University of Pennsylvania who participated in an extra-credit assignment. The students were given an overview of the project and some guidelines to annotate data with the aforementioned criteria. We carefully validated all resulting annotations by discarding or editing erroneous and inappropriate examples. In our work, we strictly use this portion as the test set to evaluate the generalization ability of our final models. The complete dataset and annotation instructions can be found in our public repository containing no personally identifiable information of any annotator.

The statistics of CREPE are in Table 1. In this work, we consciously focus on few-shot and in-context settings because our data annotation inevitably contains bias and limitation, and thus cannot be truly representative of counterfactual reasoning in every scenario. In such cases, we believe having a sizeable training set aggravates such biases and induces spurious artifacts.

4 Event Likelihood Prediction

The task of CREPE is essentially ternary classification, where the likelihood change of each event after each step is labeled as one of “more likely”, “less likely”, or “equally likely”. In this section, all models have no access to the annotated entity state changes until later sections.

4.1 Baselines

To show the challenge CREPE brings to existing models, we first introduce some naive baselines.

- The **chance** baseline assigns random labels.
- The **majority** baseline always assigns the majority label “equally likely”.

Next, we consider the following state-of-the-art LLMs as strong baselines, where all models are given exactly three examples in their prompt:

- **T5** (Raffel et al., 2020) is one of the state-of-the-art LLMs. Given the goal, steps, and question formatted by a prompt template, we compare the probability of generating “the answer is no|yes.” We use T0-3B² with 3 billion parameters.
- **T0** (Sanh et al., 2022) is a variant of T5, finetuned on a large set of downstream tasks with natural language prompts. We adopt the same inference process as T5 described above. We use T0pp³ with 11 billion parameters.
- **GPT-3** (Brown et al., 2020) is a series of LLMs that excels at few-shot learning using the prompting mechanism. We consider text-curie-001 (7B parameters), text-davinci-002, text-davinci-003, and ChatGPT (all 175B parameters). We use default parameters with a temperature of 0 for deterministic predictions. An example of the prompt is shown in Figure 2.
- **GPT-3 finetuned on StrategyQA** is a GPT-3 curie model finetuned with StrategyQA (Geva et al., 2021), a dataset of factual multihop questions and their decomposition. StrategyQA is similar to our task in that estimating the change of event likelihood can also be decomposed into sub-tasks of estimating the change of state of related entities (Section 5.1).

Table 2 shows that all state-of-the-art LLMs we have attempted achieve close-to-chance performance on CREPE around 0.350 F1, whereas text-davinci-003 and ChatGPT which are

²<https://huggingface.co/t5-3b>

³<https://huggingface.co/bigscience/T0pp>

Goal: Wash sneakers
 Context: I remove shoelaces. I rinse.
 Question: What is the likelihood that my feet get wet by wearing the sneakers?
 Answer: likely

Figure 2: Our GPT-3 prompt, which is typical for a QA task. Each likelihood label is compared with the previous one to get the label for the change.

known to be stronger at reasoning perform better. Details about prompt formulation and experimental results on prompt sensitivity are shown in Appendix B and A.

4.2 Representing Procedures as Python Code

Codex (Chen et al., 2021a) is a variation of GPT-3 that was designed to be prompted with and to generate code, in addition to natural language texts. Recently, Madaan et al. (2022) found that prompting Codex with some structured representation such as Python code. Inspired by this observation, we propose novel code representations of procedures and hypothetical events. Among many possibilities we experimented with, the representation with the best empirical performance is described below, later shown to greatly outperform all baseline models. The representation is exemplified in Figure 3.

The procedure is represented as a class where the goal G is the class name, followed by the steps s_i as comments. Then, each step is defined as a member function, in which the hypothetical events e_j are represented as objects with comments. Each event object has an attribute “change” whose value describes the change of the likelihood. During inference, Codex is provided with the prompt including three in-context examples and the current procedure up to the definition of the “init” function and predicts the definition of all step functions. Finally, we extract the assigned value of the “change” attribute as the event likelihood change δ_i .

This prompt design effectively leverages the semantic similarity between procedures with entity states and functions with variables, by representing texts as function identifiers and comments. We use code-davinci-002⁴ with 175B parameters and default hyperparameters with a temperature of 0.

⁴While OpenAI announced that text-davinci-002 is based on code-davinci-002 (<https://platform.openai.com/docs/model-index-for-researchers>), we empirically find the former to perform worse with our code prompt and thus only consider the latter with code prompt.

```
class Wash_Sneakers:
    # Init
    # Remove shoelaces
    # Rinse
    def __init__(self, event0):
        self.event0 = event0 # My feet get
            wet by wearing the sneakers.
    def remove_shoelaces(self):
        self.event0.change = "equally likely
            " # My feet get wet by wearing
                the sneakers.
    def rinse(self):
        self.event0.change = "more likely" #
            My feet get wet by wearing the
                sneakers.
```

Figure 3: Our best-performing Python code representation of a procedure and hypothetical events, for Codex.

4.3 Results

As CREPE is a ternary classification task, we report the macro F1 score across the three classes. As shown in Table 2, T5 and T0 perform only slightly better (.343 and .336 F1) than chance (.297 F1). GPT-3, one of the most dominant models across a variety of NLP tasks, is no better (.336 F1), whereas finetuning it on another multihop reasoning dataset StrategyQA does not bring about any improvement (.341 F1). The latest GPT-3 models, text-davinci-003 (.424 F1) and ChatGPT (.470 F1) which were released contemporarily with this paper, greatly outperform their predecessors.

On the other hand, our code-representation of events as the prompt to Codex greatly outperforms all other models with .585 F1. As Codex is trained on public Github code in addition to the internet texts that GPT-3 is trained on, it is noteworthy that Codex can effectively reason about texts with code-like structures, for a procedure has many analogies to a class in object-oriented programming.

4.4 Ablation Studies

To understand why the representation in our Codex prompt is effective, we perform an ablation study with various changes of the format to the representation, including:

- Remove steps comments in the beginning
- Remove event comments in step functions
- Use nested functions instead of a class
- Use flat variables to encode goals, steps, and events (no hierarchical class functions)

Examples of these empirically inferior representations are shown in Appendix B. As seen in Table 3, the hierarchical representation of procedures, steps,

Params	Naive		Large Language Models								Human
	Cha.	Maj.	T5	T0	GPT3C	GPT3C+S	GPT3D2	GPT3D3	ChatGPT	Codex (ours)	
	-	-	3B	11B	13B	13B	175B	175B	175B	175B	-
Dev	.262	.297	.343	.336	.346	.341	.350	.424	.470	.585	.868
Test	.251	.296	.343	.337	.356	.346	.533	.423	.462	.591	-

Table 2: Macro F1 of baseline models on the CREPE dataset. Human performance is not benchmarked on the test set as we strictly hold out its labels during all experiments. GPT3C represents the text-curie-001 model. GPT3D2 represents the text-davinci-002 model with an abnormal performance on the test set that we have confirmed but regrettably cannot explain. GPT3D3 represents the text-davinci-003 model. GPT3C+S represents the GPT-3 curie model finetuned on StrategyQA. All of the above models work with textual prompts. Codex represents the code-davinci-002 model and works with our proposed code-like prompts.

	Dev	Test
Codex	.585	.591
no step comments	.377	.352
no event comments	.576	.555
nested function	.568	.572
flat variables	.338	.341

Table 3: Macro F1 of the ablations of our Codex prompt.

and events as classes or nested functions is critical. Besides, listing all the steps as comments helps, mimicking a programmer’s textual explanation of a class or a function.

5 Causal Reasoning with Entities

When a human tries to predict whether the event “one would get burnt by touching a pan” is likely, their reasoning process would first focus on some entities in the question (e.g., “the pan”), then attend to some attributes and states of that entity (e.g., the temperature of the pan is hot), and finally draw a logical conclusion (e.g., “the pan being hot means one would get burnt by touching it.”) CREPE is constructed precisely with this thought process in mind. An entity-attribute-change tuple is annotated along with each event likelihood change. In this section, we study how to explicitly leverage the intermediate information to assist the prediction of event likelihood prediction.

5.1 Predicted Entity States as CoT

In CREPE, the task of predicting event likelihood change can be seen as a case of multihop reasoning, where a model first decomposes the question into some open-ended sub-questions, answer these sub-questions, and aggregate them as a final answer. LLMs can be prompted to perform chain-of-thought (CoT) style reasoning (Nye et al., 2021; Wei et al., 2022). Thus, we ask the question:

Q1. Can LLMs benefit from first **predict-**

Goal: Wash sneakers
Context: I remove shoelaces. I rinse.
Question: What is the likelihood that my feet get wet by wearing the sneakers?
Answer: To get feet wet by wearing the sneakers, the sneakers must be wet. In the given context, the sneakers are wet. Therefore, comparing to the previous step, the likelihood change is “more likely”.

Goal: Wash sneakers
Context: I remove shoelaces. I rinse.
Question: What is the likelihood that my feet get wet by wearing the sneakers?
Follow up: Are the sneakers wet?
Intermediate answer: Yes
Follow up: Will my feet get wet by wearing wet sneakers?
Intermediate answer: Yes
Answer: likely

Figure 4: Our GPT-3 prompt with intermediate questions, mimicking the CoT prompt (top) and the Self-Ask prompt (bottom).

ing entity state changes, as a CoT, before predicting event likelihood changes?

CoT with GPT-3. First, we prompt GPT-3 with Wei et al. (2022)’s CoT paradigm and Press et al. (2022)’s self-ask paradigm, both of which are shown in Figure 4. While self-ask relies on search engines for fact retrieval, we use LM generation instead as most of our entity state tracking questions are heavily context-dependent and unanswerable by any search engine. When writing demonstrations for few-shot learning, we impose the following logic progression for the follow-up questions: (1) initial followups shall ask questions on the state of entities that are directly related to the event; (2) followups following the entity state questions shall ask for the logical relationship between the entity states and the original event.

	Naive	LLMs		CoT Large Language Models				Human
	Majority	GPT-3	Codex	GPT-3 + CoT	GPT-3+self-ask	Codex soft (ours)	Codex hard (ours)	
Dev	.297	.346	.585	0.359	.342	.624	.667	.868
Test	.296	.356	.591	0.379	.345	.626	.609	-

Table 4: Macro F1 of chain-of-thought models on the CREPE dataset. GPT-3 + CoT|self-ask represents the text-davinci-002 model prompted with the CoT or self-ask style prompt.

CoT Codex with Soft Entity Representation.

We modify our Codex prompt in Figure 3, so that a sub-event is represented as a string variable whose declaration and value assignments are right before those of the hypothetical event. We refer to this as a *soft representation* of entities (Figure 5). During inference, Codex is provided with the code up to the step function header and predicts the entity and event changes for every step function. Our Codex model achieves the new best performance of .624 F1, outperforming the same model without predicted entities as CoT by .039 F1.

```
class Wash_Sneakers():
    # Init
    # Remove shoelaces
    # Rinse
    def init(self, event0, subevent0):
        self.event0 = event0 # My feet get
            wet by wearing the sneakers.
        self.event0.subevent = subevent0 #
            The sneakers are wet
    def remove_shoelaces(self):
        self.event0.subevent.change =
            "equally likely" # The sneakers
                are wet
        self.event0.change = "equally likely
            " # My feet get wet by wearing
                the sneakers.
    def rinse(self):
        self.event0.subevent.change =
            "more likely" # The sneakers are
                wet
        self.event0.change = "more likely" #
            My feet get wet by wearing the
                sneakers.
```

Figure 5: Our Codex prompt with a soft representation of entity state changes as strings.

CoT Codex with Hard Entity Representation.

The two approaches above both *softly* represent the intermediate entity state changes as texts, either questions or statements. Here, LLMs are not enforced to generate intermediate reasoning steps that contain entities and attributes. To answer Q1 more precisely, we experiment with a *hard entity representation* where the entity-attribute-change tuple is explicitly baked into the Codex prompt as shown in Figure 6. Here, each entity is represented as an

	Dev	Test
Majority	.297	.296
GPT-3 CoT	.342	.345
w/ gold entity changes	.351	.380
Codex CoT	.667	.609
w/ gold entity changes	.715	.722
Human	.868	-

Table 5: Macro F1 of GPT-3 and Codex with chain-of-thought provided with gold entity state changes.

object with an attribute and assigned value. The hard entity representation leads to a far superior performance of .667 F1 on the development set but generalizes worse on the test set with .609 F1.

```
class Wash_Sneakers():
    # Init
    # Remove shoelaces
    # Rinse
    def init(self, event0):
        self.sneakers = Sneakers()
        self.event0 = event0 # My feet get
            wet by wearing the sneakers.
    def remove_shoelaces(self):
        self.event0.change = "equally likely
            " # My feet get wet by wearing
                the sneakers.
    def rinse(self):
        self.sneakers.wet = True
        self.event0.change = "more likely" #
            My feet get wet by wearing the
                sneakers.
```

Figure 6: Our Codex prompt with a hard representation of entity states as variables, attributes, and values.

To recap, we have shown that LLMs can be prompted to exhibit a CoT that first predicts entity state changes and then event likelihood changes. Hence, our answer to Q1 raised at the beginning of this subsection is ‘yes.’

5.2 Annotated Entity States as CoT

In the above section, we have shown how event likelihood prediction can be improved by first having the LLMs predict entity states as a CoT. These experiments mimic a realistic setting where infor-

mation about entities is unavailable. However, in some scenarios, the entity states may be provided. For example, an embodied agent or a robot might have a reliable component that tracks entities; some practitioners might care about a small set of procedures in a narrow domain with annotated entity changes; or, some event schemata containing entity information could be used to predict unseen events. Here, we try to answer the following question:

Q2. Can LLMs effectively leverage **annotated** entity state changes to better predict event likelihood changes?

Instead of having LLMs predict entity state changes, we provide the annotated entity state changes in the CREPE dataset to GPT-3 and Codex. Doing so has the additional benefit of verifying that entity state changes indeed causally benefit LLMs in predicting events.

As shown in Table 5, our Codex representation with access to gold entity changes leads to improved performance of .715 F1 on the development set. In contrast, GPT-3 does not see any gain. Hence, the answer to **Q2** is ‘yes’ for the code-trained LLMs but ‘no’ for standard LLMs.

5.3 Externally Predicted Entity States

As we will discuss further in Section 7, entity state tracking is an established task in NLP with existing datasets and models. We have now predicted entity state changes using LLMs in a few-shot learning setting. It is then natural to pose the question:

Q3. Do existing entity state tracking models make predictions that lead to better performance on CREPE?

Our definition of causal reasoning of events is directional since we consider entity state changes as the cause of the change in event likelihoods. To this extent, we incorporate OpenPI (Tandon et al., 2020), the only open-domain entity state tracking dataset in procedural texts, as a part of the pipeline. In OpenPI, the input is a goal, a step, and the output is tuples of an entity, a feature, and two attributes before and after the execution of the step. For example, after “heat the pan [step]”, “the temperature [feature] of the pan [entity] is cool [attribute] before and hot [attribute] afterward.” While the original paper proposed a GPT2 model (Radford et al., 2019), we opt to finetune the superior GPT-3 Curie model on its data. After the model makes a prediction, we post-process it into the format of CREPE

by discarding the feature and producing two entity-attribute-change pairs (e.g., pan-hot-“more likely” and pan-cold-“less likely”). We provide Codex with only the entity changes when the entity is mentioned in the event. Further, to fit our prompt in the context window of Codex, we provide Codex with 5 entity state changes uniformly drawn from a pool of candidate choices at every step. The resulting OpenPI-prompted Codex gives a degraded macro F1 score of 0.553 on the development set and 0.496 on the testing set. Hence, our answer to **Q3** is ‘no,’ suggesting that existing entity state tracking datasets may be insufficient for our causal reasoning task.

6 Performance Analysis

In this section, we analyze potential factors that play a role in our Codex model’s performance. We investigate three factors: (1) the number of steps in a procedure; (2) explicit mentions of event-related entity-of-interest (EoI) in a given step; and (3) the logical relation (entailment or contradiction) between the event likelihood change and its related entity state change. To study factor (1), we dichotomize procedures from the development set by the average length of the procedure. To investigate factors (2) and (3), we manually labeled the ground truth EoI mentioning and logical relation for the development dataset. Intuitively, estimating event likelihood in lengthy procedures and in steps where EoI is not explicitly mentioned would be difficult. Rather surprisingly, Codex shows no significant performance discrepancy under factors (2) and (3), and only a slight performance difference in factor (1) (see Appendix C).

Further, the task of CREPE can be divided into two sub-tasks, first to identify whether an event likelihood change occurred at all, and then to classify the change as either more or less likely. We observe that CoT Codex outperforms Codex on both sub-tasks. For the classification task, in particular, CoT Codex obtained a .149 increase in macro F1 score from .805 to .954. This shows not only that CoT Codex is effective, but also that its bottleneck is identifying event likelihood change.

7 Related Work

Event & Entity Extraction and Representation
Event-centric NLP has been a dominant strand of approaches to machine reasoning. Myriad work has focused on extracting events from the news

or web data (Liu et al., 2018; Yang et al., 2019; Du and Cardie, 2020). The effort of structurally representing scripts, groups of events in certain scenarios including procedures, started decades ago (Abelson and Schank, 1977) and is receiving revived attention in present years (Li et al., 2020; Wang et al., 2022a). While this line of work mostly focuses on the representation as relations (e.g., temporal, hierarchical) among events, we recognize entities as a cause of event relations and thus propose a more granular representation. Furthermore, structured representations of events typically cannot take advantage of the power of textual LLMs for challenging downstream tasks. In contrast, we advance towards the best of two worlds by working with code language models.

Besides, existing work on jointly extracting and representing events and entities (Lee et al., 2012; Wadden et al., 2019; Barhom et al., 2019) neglects the causal relation therein and treats entities and events simply as two related tasks to be tackled simultaneously. We causally bridge the two.

Entity State Tracking Prior work on entity state tracking spans various disciplines of AI. For instance, object tracking, a sub-task of entity state tracking, has led to much work in both robotics (Wang et al., 2007) and computer vision (Comaniciu et al., 2003). In NLP, early efforts focus on synthetic, closed-domain data (Weston et al., 2015; Long et al., 2016) and more recent ones shift attention to real-world procedures (Bosselut et al., 2017; Dalvi et al., 2018; Gupta and Durrett, 2019; Du et al., 2019; Mysore et al., 2019) with a closed set of entities and attributes or an open-ended set (Tandon et al., 2020). In all prior work, entity state track is treated as an end-task, whereas we treat it as a critical intermediate step for event reasoning, a more practical application.

Counterfactual Reasoning In this work, we hope to provide evidence that signals of entities effectively help models reason about events. We specifically focus on hypothetical event reasoning because it is a high-level cognitive ability beyond pattern recognition and a manifestation of complex reasoning ability (Pearl and Mackenzie, 2018; Pearl, 2019). Counterfactual reasoning has a long history with formal methods (Forbus, 1984; Lewis, 2013). Less modern work exists in commonsense (Feng et al., 2021), procedural texts (Tandon et al., 2019), and even computer vision (Yue et al., 2021).

Multihop Reasoning Prior studies on multihop reasoning mainly focus on question answering from a passage (Welbl et al., 2018; Talmor and Berant, 2018; Yang et al., 2018; Kočiský et al., 2018; Mihaylov et al., 2018; Khot et al., 2020) and representing and utilizing multihop information in the form of structured data (De Cao et al., 2019; Ding et al., 2019; Qiu et al., 2019; Cao et al., 2019; Fang et al., 2020; Thayaparan et al., 2019; Zhang et al., 2020d, 2021; Huang and Yang, 2021).

There are also efforts such as Decomprc, StrategyQA, and CGDe-FGIn that attempt to conduct multihop reasoning by decomposing the original task to a series of logically related sub-tasks (Min et al., 2019; Geva et al., 2021; Cao and Liu, 2022). Such an approach has recently seen great success with the Chain-of-Thought (CoT) prompting of GPT-3, which significantly improves numerous multihop reasoning tasks (Nye et al., 2021; Kojima et al., 2022; Wei et al., 2022; Wang et al., 2022c). Following CoT prompting, Self-Ask further elicits CoT by demanding GPT-3 to explicitly generate the reasoning questions raised during its chain-of-thought process (Press et al., 2022).

Code-Based Language Models and Prompts Recent work has shown that LLMs trained on programs or code (PLMs) have an augmented ability of reasoning over natural language texts. Notably, Suzgun et al. (2022); Liang et al. (2022) showed that PLMs outperforms only-text-trained LMs on certain reasoning tasks even though the prompts are purely natural language and contain no code. Moreover, there has been speculation that multihop reasoning is an emergent ability exclusive to PLMs and absent in their only-text-trained predecessors (Fu and Khot, 2022).

Even more interestingly, a line of contemporary work found that, for some reasoning tasks, prompting PLMs with certain structured programs (e.g., Python code, JSON, PDDL) that represent the originally textual data outperforms doing so simply with natural language prompts. These tasks include math questions (Chen et al., 2022; Lyu et al., 2023; Mishra et al., 2022) and event reasoning (Madaan et al., 2022; Wang et al., 2022b) like our work.

Procedural Texts Procedural texts are an attractive data source to reason about events and entities which undergo frequent changes. There has been steady efforts in computer vision (Miech et al., 2019), robotics (Ahn et al., 2022), and language (Mujtaba and Mahapatra, 2019; Zhang, 2022). In

NLP specifically, work on procedures includes extracting them from instructional texts (Paris et al., 2002; Delpech and Saint-Dizier, 2008; Zhang et al., 2012), reasoning about events (Takechi et al., 2003; Tandon et al., 2019; Rajagopal et al., 2020; Zhang et al., 2020c), knowledge-base construction (Jung et al., 2010; Chu et al., 2017; Park and Motahari Nezhad, 2018), or applying them to downstream applications (Yang et al., 2021b,a; Zhang et al., 2020a; Lyu et al., 2021; Dalvi et al., 2019; Zhang et al., 2020b; Chen et al., 2020). Our work is scoped in procedural texts due to the outstanding causal relations between entities and events in a dynamic environment.

8 Conclusion and Future Work

We present CREPE, a benchmark for causal reasoning about events and entities in procedural texts. We show that mainstream LLMs such as GPT-3 perform close to chance on CREPE, while using code-like event representation as a prompt to code language model Codex greatly improves the performance. Further, we experiment with various ways to encode entity information into this representation and find that eliciting chain-of-thought reasoning from Codex further improves performance while existing CoT approaches with GPT-3 are ineffective. We clearly show that LLMs benefit from lower-level entity information when making predictions about higher-level events. Future work should explore related tasks such as next-event prediction, event temporal ordering, etc., by injecting relevant information about entities into our representation. Our code-representation of events allows more powerful expressions than simply entailment and negation considered in this work. Future work may explore other forms of code chain-of-thought such as first-order logic. These expressions generated by LLMs can be computed objectively, thus ameliorating LLMs’ hallucinations and improving the interpretability and faithfulness of predictions.

9 Limitations

Despite our best efforts, our CREPE dataset has inherent limitations. First, the choice of studying procedure texts, despite many discussed advantages, limits the domain, writing style, and other semantic features of the texts. As a result, porting our methods and findings to other text styles such as stories or news might require domain adaptation. Second, we prioritize quality over quantity when creating

this benchmark, which suffers from small size and contains biases from the annotators, even though we address the latter by having different annotators label a test set.

When annotating the hypothetical events, our intention is that they represent a wild variety that doers of the procedures, humans or machines, would care about. However, we also have to ensure these events are unambiguously bound to some entities in order to challenge models for their causal reasoning ability. While we do our utmost to balance these two conflicting objectives, the issue might still persist.

In CREPE, each event likelihood change is caused by exactly one entity state change. This is an over-simplification made to facilitate evaluation. In real life, many complex events require many entity states to be reasoned about, which in turn may have complex logical relations among them. We leave this for future work.

While we intend our representation of events and entities to be a general and effective one, we have only shown that it works well empirically using Codex, which is one of the only code language models at present. Whether the idea of our structured representation applies to other models remains to be explored.

10 Acknowledgements

This research is based upon work supported in part by the DARPA KAIROS Program (contract FA8750-19-2-1004), the DARPA LwLL Program (contract FA8750-19-2-0201), the IARPA BETTER Program (contract 2019-19051600004), and the NSF (Award 1928631). Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, IARPA, NSF, or the U.S. Government.

We thank the students in the Artificial Intelligence class at the University of Pennsylvania in Fall 2022 who participated in the annotation of the test set of CREPE. We thank Niket Tandon and Qing Lyu for valuable discussions about this work.

References

Robert Abelson and Roger C Schank. 1977. Scripts, plans, goals and understanding. *An inquiry into human knowledge structures New Jersey*, 10.

- Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.
- Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189, Florence, Italy. Association for Computational Linguistics.
- Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2017. Simulating action dynamics with neural process networks. *arXiv preprint arXiv:1711.05313*.
- Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. In *6th Annual Conference on Robot Learning*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Xing Cao and Yun Liu. 2022. Coarse-grained decomposition and fine-grained interaction for multi-hop question answering. *Journal of Intelligent Information Systems*, 58(1):21–41.
- Yu Cao, Meng Fang, and Dacheng Tao. 2019. BAG: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 357–362, Minneapolis, Minnesota. Association for Computational Linguistics.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021a. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Muhao Chen, Hongming Zhang, Qiang Ning, Manling Li, Heng Ji, Kathleen McKeown, and Dan Roth. 2021b. Event-centric natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Tutorial Abstracts*, pages 6–14, Online. Association for Computational Linguistics.
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.
- Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.
- Cuong Xuan Chu, Niket Tandon, and Gerhard Weikum. 2017. Distilling task knowledge from how-to communities. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 805–814. ACM.
- Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. 2003. Kernel-based object tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5):564–577.
- Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1595–1604, New Orleans, Louisiana. Association for Computational Linguistics.
- Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4496–4505, Hong Kong, China. Association for Computational Linguistics.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317, Minneapolis, Minnesota. Association for Computational Linguistics.
- Estelle Delpéch and Patrick Saint-Dizier. 2008. Investigating the structure of procedural texts for answering how-to questions. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).
- Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of*

- the 57th Annual Meeting of the Association for Computational Linguistics, pages 2694–2703, Florence, Italy. Association for Computational Linguistics.
- Li Du, Xiao Ding, Ting Liu, and Bing Qin. 2021. [Learning event graph knowledge for abductive reasoning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5181–5190, Online. Association for Computational Linguistics.
- Xinya Du and Claire Cardie. 2020. [Event extraction by answering \(almost\) natural questions](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.
- Xinya Du, Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, Peter Clark, and Claire Cardie. 2019. [Be consistent! improving procedural text comprehension using label consistency](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2347–2356, Minneapolis, Minnesota. Association for Computational Linguistics.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. [DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuo-hang Wang, and Jingjing Liu. 2020. [Hierarchical graph network for multi-hop question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online. Association for Computational Linguistics.
- Fuli Feng, Jizhi Zhang, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. [Empowering language understanding with counterfactual reasoning](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2226–2236, Online. Association for Computational Linguistics.
- Kenneth D Forbus. 1984. Qualitative process theory. *Artificial intelligence*, 24(1-3):85–168.
- Hao Fu, Yao Peng and Tushar Khot. 2022. [How does gpt obtain its ability? tracing emergent abilities of language models to their sources](#). *Yao Fu's Notion*.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. [Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies](#). *Transactions of the Association for Computational Linguistics*, 9:346–361.
- Aditya Gupta and Greg Durrett. 2019. [Tracking discrete and continuous entity state for process understanding](#). In *Proceedings of the Third Workshop on Structured Prediction for NLP*, pages 7–12, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yongjie Huang and Meng Yang. 2021. [Breadth first reasoning graph for multi-hop question answering](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5810–5821.
- Yuchul Jung, Jihee Ryu, Kyung-min Kim, and Sung-Hyon Myaeng. 2010. Automatic construction of a large-scale situation ontology by mining how-to instructions from the web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(2-3):110–124.
- Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.
- Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.
- Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. [Joint entity and event coreference resolution across documents](#). In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500, Jeju Island, Korea. Association for Computational Linguistics.
- David Lewis. 2013. *Counterfactuals*. John Wiley & Sons.
- Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020. [Connecting the dots: Event graph schema induction with path language modeling](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–695, Online. Association for Computational Linguistics.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian

- Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.
- Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.
- Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1456–1465, Berlin, Germany. Association for Computational Linguistics.
- Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning.
- Qing Lyu, Li Zhang, and Chris Callison-Burch. 2021. Goal-oriented script construction. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 184–200, Aberdeen, Scotland, UK. Association for Computational Linguistics.
- Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Abu Dhabi, UAE.
- Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.
- Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hananeh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy. Association for Computational Linguistics.
- Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, et al. 2022. Lila: A unified benchmark for mathematical reasoning. *arXiv preprint arXiv:2210.17517*.
- Dena Mujtaba and Nihar Mahapatra. 2019. Recent trends in natural language understanding for procedural knowledge. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 420–424.
- Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64, Florence, Italy. Association for Computational Linguistics.
- Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.
- Artemis Panagopoulou, Manni Arora, Li Zhang, Dimitri Cugini, Weiqiu You, Yue Yang, Liyang Zhou, Yuxuan Wang, Zhaoyi Hou, Alyssa Hwang, Lara Martin, Sherry Shi, Chris Callison-Burch, and Mark Yatskar. 2022. Quakerbot: A household dialog system powered by large language models. In *Alexa Prize TaskBot Challenge Proceedings*.
- Cécile Paris, Keith Vander Linden, and Shijian Lu. 2002. Automated knowledge acquisition for instructional text generation. In *Proceedings of the 20th Annual International Conference on Computer Documentation, SIGDOC '02*, page 142–151, New York, NY, USA. Association for Computing Machinery.
- Hogun Park and Hamid Reza Motahari Nezhad. 2018. Learning procedures from text: Codifying how-to procedures in deep neural networks. In *Companion Proceedings of the The Web Conference 2018*, pages 351–358.
- Judea Pearl. 2019. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60.
- Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect*, 1st edition. Basic Books, Inc., USA.
- Ofir Press, Sewon Min, Muru Zhang, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models.
- Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Dheeraj Rajagopal, Niket Tandon, Peter Clark, Bhavana Dalvi, and Eduard Hovy. 2020. [What-if I ask you to explain: Explaining the effects of perturbations in procedural text](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3345–3355, Online. Association for Computational Linguistics.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. [Multi-task prompted training enables zero-shot task generalization](#). In *International Conference on Learning Representations*.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.
- Mineki Takechi, Takenobu Tokunaga, Yuji Matsumoto, and Hozumi Tanaka. 2003. [Feature selection in categorizing procedural expressions](#). In *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages*, pages 49–56, Sapporo, Japan. Association for Computational Linguistics.
- Alon Talmor and Jonathan Berant. 2018. [The web as a knowledge-base for answering complex questions](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.
- Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. [WIQA: A dataset for “what if...” reasoning over procedural text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6076–6085, Hong Kong, China. Association for Computational Linguistics.
- Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. [A dataset for tracking entities in open domain procedural text](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.
- Mokanarangan Thayaparan, Marco Valentino, Viktor Schlegel, and André Freitas. 2019. [Identifying supporting facts for multi-hop question answering with document graph networks](#). In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 42–51, Hong Kong. Association for Computational Linguistics.
- David Wadden, Ulme Wennberg, Yi Luan, and Hananeh Hajishirzi. 2019. [Entity, relation, and event extraction with contextualized span representations](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.
- Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. 2007. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9):889–916.
- Hongwei Wang, Zixuan Zhang, Sha Li, Jiawei Han, Yizhou Sun, Hanghang Tong, Joseph P Olive, and Heng Ji. 2022a. [Schema-guided event graph completion](#). *arXiv preprint arXiv:2206.02921*.
- Xingyao Wang, Sha Li, and Heng Ji. 2022b. [Code4struct: Code generation for few-shot structured prediction from natural language](#). *arXiv preprint arXiv:2210.12810*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022c. [Self-consistency improves chain of thought reasoning in language models](#). *arXiv preprint arXiv:2203.11171*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). *arXiv preprint arXiv:2201.11903*.
- Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. [Constructing datasets for multi-hop reading comprehension across documents](#). *Transactions of the Association for Computational Linguistics*, 6:287–302.
- Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. [Towards ai-complete question answering: A set of prerequisite toy tasks](#). *arXiv preprint arXiv:1502.05698*.

- Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. [Exploring pre-trained language models for event extraction and generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.
- Yue Yang, Joongwon Kim, Artemis Panagopoulou, Mark Yatskar, and Chris Callison-Burch. 2021a. [Induce, edit, retrieve: Language grounded multi-modal schema for instructional video retrieval](#). *ArXiv preprint*, abs/2111.09276.
- Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. 2021b. [Visual goal-step inference using wikiHow](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2167–2179, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. 2021. Counterfactual zero-shot and open-set visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15404–15414.
- Hongming Zhang, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. 2020a. [Analogous process structure induction for sub-event sequence prediction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1541–1550, Online. Association for Computational Linguistics.
- Li Zhang. 2022. [Reasoning about procedures with natural language processing: A tutorial](#).
- Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. [Intent detection with WikiHow](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 328–333, Suzhou, China. Association for Computational Linguistics.
- Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020c. [Reasoning about goals, steps, and temporal ordering with WikiHow](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639, Online. Association for Computational Linguistics.
- Min Zhang, Feng Li, Yang Wang, Zequn Zhang, Yanhai Zhou, and Xiaoyu Li. 2020d. Coarse and fine granularity graph reasoning for interpretable multi-hop question answering. *IEEE Access*, 8:56755–56765.
- Yuyu Zhang, Ping Nie, Arun Ramamurthy, and Le Song. 2021. Answering any-hop open-domain questions with iterative document reranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 481–490.
- Ziqi Zhang, Philip Webster, Victoria Uren, Andrea Varga, and Fabio Ciravegna. 2012. [Automatically extracting procedural knowledge from instructional texts using natural language processing](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 520–527, Istanbul, Turkey. European Language Resources Association (ELRA).

A Prompt Sensitivity

In addition to the results reported in Table 2, we also investigated the effect of the number and choice of in-context examples.

Number of in-context examples The context window of text-davinci-002 maximally fits 3 shots. We experiment with 1-shot (0.245 f1), 2-shots (0.348 f1), and 3-shots (0.359 f1) learning using text-002 with CoT prompting. We see that having more context provides limited improvements in model performance.

Prompt sensitivity with random examples We tested the text-davinci-002 model with CoT prompt on the dev set using randomly chosen examples from our example bank. The F1 scores for 5 runs with randomly chosen in-context examples are 0.333, 0.327, 0.359, 0.336, and 0.331. The mean score is 0.337, and the standard deviation is 0.011, implying low sensitivity of in-context example selection.

B Prompt Engineering

B.1 Code Prompts for Codex

In Section 4 and 5, we have discussed our best-performing prompts for GPT-3 and Codex. Here, we elaborate on inferior Codex prompts and shed light on why they do not work well empirically.

Best prompt As discussed, our best-performing prompt represents procedures as classes and steps as functions.

```
class Wash_Sneakers:
    # Init
    # Remove shoelaces
    # Rinse
    def __init__(self, event0):
        self.event0 = event0 # My feet get wet by
        wearing the sneakers.
    def remove_shoelaces(self):
        self.event0.change = "equally likely" # My
        feet get wet by wearing the sneakers.
    def rinse(self):
        self.event0.change = "more likely" # My
        feet get wet by wearing the sneakers.
```

Nested functions Instead of representing procedures as classes as in our best-performing prompt, we can also represent them as nested functions.

```
def wash_sneakers(event0):
    # Init
    # Remove shoelaces
    # Rinse
    event0 = event0 # My feet get wet by
    wearing the sneakers.
    def remove_shoelaces(self):
        event0.change = "equally likely" # My
        feet get wet by wearing the sneakers.
    def rinse(self):
        event0.change = "more likely" # My
        feet get wet by wearing the sneakers.
```

No step comments The comments displaying the steps immediately after the class declaration are removed.

```
class Wash_Sneakers:
    def __init__(self, event0):
        self.event0 = event0 # My feet get wet by
        wearing the sneakers.
    def remove_shoelaces(self):
        self.event0.change = "equally likely" # My
        feet get wet by wearing the sneakers.
    def rinse(self):
        self.event0.change = "more likely" # My
        feet get wet by wearing the sneakers.
```

No event comments The comments displaying the events in step functions except init are removed.

```
class Wash_Sneakers:
    def __init__(self, event0):
        self.event0 = event0 # My feet get wet by
        wearing the sneakers.
    def remove_shoelaces(self):
        self.event0.change = "equally likely"
    def rinse(self):
        self.event0.change = "more likely"
```

Two-step In this approach, we hypothesize that providing entity state change at every step is helpful. To do this, we first prompt Codex to generate entity states corresponding to a specific event:

```
class Wash_Sneakers:
    def remove_shoelaces(self):
        event = "My feet get wet by wearing
        the sneakers."
        event.precondition = \
            ("sneakers", "wet")
    def rinse(self):
        event = "My feet get wet by wearing
        the sneakers."
        event.precondition = \
            ("sneakers", "wet")
```

We select event-related entities by majority vote. The resulting entity state bank is used to prompt Codex to first deduce entity state at every step and then answer the likelihood of the event.

Flat variables Instead of defining functions using def or creating class with class, we use only

variables to define relevant information.

```
Goal = "Wash Sneakers"

Context = "Remove shoelaces. After this,
the shoelaces are removed"
Question = "What is the likelihood that my feet
get wet by wearing the sneakers?"
Options = [
    "more likely",
    "less likely",
    "equally likely",
]
Answer = Options[2]

Context = "Rinse the sneakers. After this,
the sneakers are damp."
Question = "What is the likelihood that my feet
get wet by wearing the sneakers?"
Options = [
    "more likely",
    "less likely",
    "equally likely",
]
Answer = Options[0]
```

B.2 Textual Prompts for GPT-3

For GPT-3, we attempted a dozen of prompt formulations in our preliminary experiments which we found to differ minimally in performance. Here, we show one example:

```
"Wash hands" involves the followings steps:
1. Turn on the tap water.
2. Put hands under running water.
3. Apply soap and rub hands.
4. Turn off the tap water.
5. Dry my hands using a towel.

For every step, find out how likely it is that
water streaming sound can be heard. Answer as
(A) very likely (B) likely (C) not very likely
(D) unlikely.

Step 1: (A) very likely
Step 2: (A) very likely
Step 3: (A) very likely
Step 4: (D) unlikely
Step 5: (D) unlikely
```

For GPT-3 finetuned with StrategyQA, we ask two questions regarding the likelihood of the events, namely whether it is more/less likely that some event occurs. After obtaining the result, we conduct a consistency check. For consistent likelihood estimates, where only one of the two questions gives a positive answer, or both questions give negative answers, we assign the corresponding label to the event state change. For inconsistent estimates, where both questions give positive answers, we assign the event change likelihood to the majority label, which is "equally likely". An example of

a finetuning prompt-completion pair is shown as follows

```
Prompt:
Context: Julius Caesar had three children.
Genghis Khan had sixteen children.
Modern geneticists have determined
that out of every 200 men today
has DNA that can be traced to
Genghis Khan.
Question: Are more people today
related to Genghis Khan than Julius Caesar?
Take it step by step:

Completion:
#1 How many kids did Julius Caesar have?
two
#2 How many kids did Genghis Khan have?
fourth
#3 Is fourth greater than two?
no
Therefore, the answer to the original
question is True
```

An example of our StrategyQA GPT-3 prompt on the CREPE task is as follows:

```
Context: Remove shoelaces. Rinse. Scrub the
shoes with cleaning solution. Rinse the shoes
again. Air dry the shoes and put the shoelaces
back on.
Question: Is it more likely that my feet get
wet by wearing the sneakers?
Take it step by step:

Completion:
#1 Is the sneaker wet?
Yes
#2 Will my feet get wet by wearing wet shoes?
Yes
Therefore, the answer to the original question
is True.
```

B.3 Textual Prompts for ChatGPT

As of the time of camera-ready submission of this paper (February 1, 2023), OpenAI has not released the API for ChatGPT. Thus, we use an unofficial API⁵ which is believed to behave the same as the official web playground. Because ChatGPT is designed to only work with a zero-shot and multi-turn dialog setting, we tweak our prompt as follows:

⁵<https://github.com/acheong08/ChatGPT>

```

I'm trying to wash hands.
First, I turn on the tap water.
At this point, is it likely that
water streaming sound can be heard?
Answer with yes or no.
[answer]
Then, I put hands under running water.
At this point, is it likely that
water streaming sound can be heard?
Answer with yes or no.
[answer]
...

```

B.4 Textual Prompts for T5/T0

We design the following prompt for T5 and T0 to perform our task:

```

Goal: [The name of the goal]
Step: [The list of steps]
Question: Is that okay that [question]?
Answer: [yes or no, generated by the model]

```

C Error Analysis

In Section 6, we conclude that the performance of Codex is not influenced by (1) the number of steps in a procedure; (2) explicit mentions of event-related entity-of-interest (EoI) in a given step; and (3) the logical relation (entailment or contradiction) between the event likelihood change and its related entity state change.

Factors	Dev
Procedure Length > 7	.629
Procedure Length ≤ 7	.700
EoI Mentioned	.481
EoI NOT Mentioned	.496
Entailment	.482
Contradiction	.461

Table 6: Macro F1 Score of error analysis. The scores for EoI and Logical relation are lower since we do not consider the majority label, "equally likely", in the error analysis.

Few-Shot Structured Policy Learning for Multi-Domain and Multi-Task Dialogues

Thibault Cordier^{1,2} and Tanguy Urvoy² and Fabrice Lefèvre¹ and Lina M. Rojas-Barahona²

¹LIA - Avignon University, France

²Orange Innovation, Lannion, France

thibault.cordier@alumni.univ-avignon.fr

fabrice.lefevre@univ-avignon.fr

{linamaria.rojasbarahona, tanguy.urvoy}@orange.com

Abstract

Reinforcement learning has been widely adopted to model *dialogue managers* in task-oriented dialogues. However, the user simulator provided by state-of-the-art dialogue frameworks are only rough approximations of human behaviour. The ability to learn from a small number of human interactions is hence crucial, especially on multi-domain and multi-task environments where the action space is large. We therefore propose to use *structured policies* to improve sample efficiency when learning on these kinds of environments. We also evaluate the impact of *learning from human vs simulated experts*. Among the different levels of structure that we tested, the graph neural networks (GNNs) show a remarkable superiority by reaching a success rate above 80% with only 50 dialogues, when learning from simulated experts. They also show superiority when learning from human experts, although a performance drop was observed, indicating a possible difficulty in capturing the variability of human strategies. We therefore suggest to concentrate future research efforts on bridging the gap between human data, simulators and automatic evaluators in dialogue frameworks.

1 Introduction

Multi-domain multi-task dialogue systems are designed to complete specific *tasks* in distinct *domains* such as finding and booking a hotel or a restaurant (Zhu et al., 2020). A domain is formally defined as a list of *slots* with their valid values. The most common task, the information-seeking task, is usually modelled as a slot-filling data-query problem in which the system requests constraints to the user and proposes items that fulfil those constraints.

The design of a *dialogue manager* (DMs) is costly: *hand-crafted* policies require a lot of engineering, *pure supervised learning* (or *behaviour cloning*) requires a lot of expert demonstrations, and *pure reinforcement learning* requires a lot of

user interactions to converge. The simulators provided with frameworks, such as PYDIAL (Ultes et al., 2017) or CONVLAB (Zhu et al., 2020), are only rough approximations of human behaviour and the ability to learn from a small number of human interactions remains crucial. This is especially true on multi-domain and multi-task environments where the action space is large (Gao et al., 2018).

A popular approach to reduce these costs is to wire some knowledge about the problem into the policy model, namely: *few shot learning* (Wang et al., 2020). In particular, structured policies like *graph neural networks* (GNNs) are known to be well suited to handle a variable number of slots and domains for the information-seeking task (Chen et al. 2018; Chen et al. 2020). In this paper, we explore structured policies based on GNN. A graph in a GNN is *fully connected* and *directed*. Each *node* represents a sub-policy associated with a slot, while a directed *edge* between two nodes represents a message passing.

For studying sample efficiency, we analyse the dialogue success rate of structured policies once trained in a supervised way from expert demonstrations. We consider two types of demonstrations: *human experts* extracted from the MULTIWOZ dataset (Budzianowski et al., 2018), and *simulated experts* generated by letting the CONVLAB’s *hand-crafted* policy interact with a simulated user.

We perform large scale experiments. We study the impact of different levels of structure (see them in Figure 2) on policy success rate after a limited number of dialogue demonstrations. For each level of structure, we also compare two sources of demonstrations: simulated and human dialogues. We show a notable result: our structured policies are able to reach a success rate above 80% with only 50 when following a simulated expert in CONVLAB. To the best of our knowledge there are not previous works that studied the impact of structure for dialogue policy in a few-shot setting.

Another important finding is that few-shot learning from human demonstrations is harder, producing a lower success rate. This can be explained first by the large variability of human strategies that is not covered by simulated users which stick to more repetitive – easy to learn – dialogue patterns. Another explanation could be an evaluation bias, simulated dialogues are more in line with artificial evaluators.

The remainder of this paper is structured as follows. We present the related work in Section 2. Section 3 presents the proposed GNNs from demonstrations. The experiments and evaluation are described in Sections 4 and 5 respectively. Finally, we conclude in Section 6.

2 Related Work

Few shot learning takes advantage of prior knowledge to avoid overloading the empirical risk minimiser when the number of available examples is small. In particular, prior knowledge can be used to constrain hypothesis space (i.e. model parameters) with parameter sharing or tying in order to reduce reliance on data acquisition and on data annotation (Wang et al., 2020).

Prior knowledge can be built into dialogue systems by imposing a structure in the neural network architecture. A first approach is to use *hierarchical reinforcement learning* that divides a main problem into several simpler sub-problems. We refer to Sutton et al. (1999) that introduces *semi-Markov decision process* using temporal abstraction and to Wen et al. (2020) that introduces *sub-Markov decision process* using state partition. In the scope of the paper, a *hierarchical policy* corresponds to a meta-controller that chooses to activate a domain and we have one sub-policy per domain (Budzianowski et al., 2017; Casanueva et al., 2018; Le et al., 2018).

In the same vein, *graph neural networks* (GNNs) have been explored in a wide range of domains because of their empirical success and their theoretical properties which explains its efficiency: the abilities of generalisation, stability and expressiveness (Garcia and Bruna, 2018). GNNs are suitable for applications where the data have a graph structure i.e where the graph outputs are supposed to be permutation-invariant or equivariant to the input features (Zhou et al., 2020; Wu et al., 2020).

In single-domain dialogue environments, this architecture has been adapted to model the DM in Chen et al. (2018) and Chen et al. (2020). They

have shown that GNNs generalise between similar dialogue slots, manage a variable number of slots and transfer to different domains that perform similar tasks. We thus adopt in this work the *domain independent parametrisation* (DIP) (Wang et al., 2015), which standardises the slots representation into a common feature space.

In this work, as in Chen et al. (2018) and Chen et al. (2020), we propose to improve multi-domain covering by learning a generic policy based on GNN. But unlike them, (i) we use a multi-domain multi-task setting, in which several domains and tasks can be evoked in a dialogue; (ii) the *dialogue state tracker* (DST) output is not discarded when activating the domain; and (iii) we adapt the GNN structure to each domain by keeping the relevant nodes while sharing the edge’s weights.

3 Structured Policies with Expert Demonstrations

In order to investigate the impact of structured policies with behaviour cloning in improving sample efficiency in multi-domain multi-task dialogue environments, we introduce the dialogue state and action spaces for structured policies and we present the different policies and the experts’ nature.

3.1 Dialogue State / Action Representations

In multi-domain multi-task dialogues, the *domain* refers to the set of concepts and values speakers can talk about. Examples of domains are restaurants, attractions, hotels, trains, etc. A *dialogue act* is a predicate that refers to the performative actions of speakers in conversations (Austin, 1975). These actions are formalised as predicates like INFORM (i.e., affirm) or REQUEST with slots or slot-values pairs as arguments. Examples of system actions are: REQUEST(food), or INFORM(address). These structured actions are used to frame a message to the user. We adopt here the multi-task setting as presented in CONVLAB (Zhu et al., 2020), in which a single dialogue can have the following tasks: (i) *find*, in which the system requests information in order to query a database and make an offer; (ii) *book*, in which the system requests information in order to book the item.

We adopt the DIP state and action representations, which are not reduced to a flat vector but to a set of sub-vectors: one corresponding to the domain parametrisation (or *slot-independent representation*), the others to the slots parametrisation

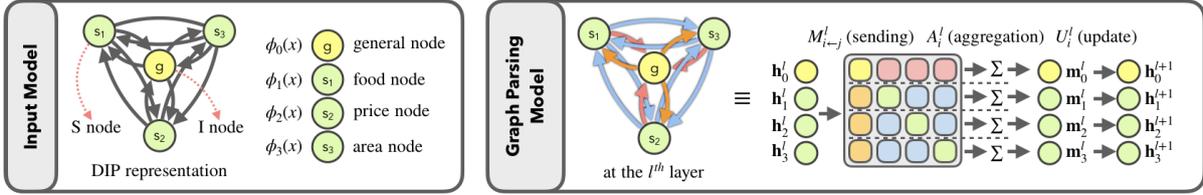


Figure 1: Structure of the input and graph parsing model in restaurant domain example. The input is a fully-connected graph with two kinds of nodes and three kinds of edges. The I-NODE are depicted in yellow; the S-NODE in green. The structured policy is described by successive graph convolutions composed of the shared weights $\mathbf{W}_{i,j}^l$.

(or *slot-dependent representations*). For any active domain, the input to the *slot-independent representation* is the concatenation of the previous *slot-independent* user and system actions (see examples of the output below, and a formal definition in Section 3.2), the number of entities fulfilling the user’s constraints in the database, the booleans indicating if the dialogue is terminated and whether an offer has been found / booked. The output corresponds to action scores such as REQMORE, OFFER, BOOK, GREAT, etc. Regarding the *slot-dependent representation*, its input is composed of the previous *slot-dependent* user and system actions (see output below), the booleans indicating if a value is known and whether the slot is needed for the *find / book* tasks. Its output are actions scores such as INFORM, REQUEST and SELECT. The parameterisation used in CONVLAB does not depend on the probabilistic representation of the states, *i.e.* does not consider the uncertainty in the predictions made by the *natural language understanding* (NLU) module.

3.2 Graph Neural Network

Prior knowledge can be integrated in our models by constraining the layer structure imposing symmetries in the neural dialogue policies. Without prior knowledge, the standard structure used is the *feed-forward neural network* layer (FNN). This unconstrained structure does not assume any symmetry in the network.

Assuming that sub-policies associated with the slots are the same, a better alternative is to use the *graph neural network* layer (GNN). This structure assumes that the state and action representations have a graph structure that are identically parameterised by DIP. The GNN structure is a fully connected and directed graph, in which each *node* represents a sub-policy associated with a slot and a directed *edge* between two sub-policies represents a message passing. We identify two roles for sub-policies: the general node as I-NODE associated

to the *slot-independent representation* and the slot nodes denoted as S-NODE associated to the *slot-dependent representations*. Both representations were introduced in Section 3.1. We also identify the relations: I2S for I-NODE to S-NODE, S2I and S2S respectively¹ (as presented in Figure 1).

We formally define the GNN structure as follows. Let n be the number of slots and L the number of layers. Let be x the dialogue state, $\mathbf{x}_0 = \phi_0(x)$, $\mathbf{h}_0^l \forall l \in [0, L - 1]$ and \mathbf{y}_0 be respectively the input, hidden and output I-NODE representations. Let the input, hidden and output S-NODES representations be respectively $\forall i \in [1, n]$, $\mathbf{x}_i = \phi_i(x)$, $\mathbf{h}_i^l \forall l \in [0, L - 1]$ and \mathbf{y}_i . First, the GNN transforms inputs:

$$\forall i \in [0, n], \quad \mathbf{h}_i^0 = \sigma^0(\mathbf{W}_i^0 \phi_i(\mathbf{x}) + \mathbf{b}_i^0) \quad (1)$$

Then, at the l -th layer, it computes the hidden nodes representations by following message sending² (Eq. 2), message aggregation (Eq. 3) and representation update (Eq. 4). $\forall i, j \in [0, n]$ ²:

$$\mathbf{m}_{i \leftarrow j}^l = M_{i \leftarrow j}^l(\mathbf{h}_j^{l-1}) = \mathbf{W}_{i,j}^l \mathbf{h}_j^{l-1} + \mathbf{b}_{i,j}^l \quad (2)$$

$$\mathbf{m}_i^l = A_i^l(\mathbf{m}_{i \leftarrow *}) = \frac{1}{n} \sum_{j=0}^n \mathbf{m}_{i \leftarrow j}^l \quad (3)$$

$$\mathbf{h}_i^l = U_i^l(\mathbf{m}_i^l) = \sigma^l(\mathbf{m}_i^l) \quad (4)$$

The message sending function $M_{i \leftarrow j}^l$ is a linear transformation with bias. The message aggregation function A_i^l is the average pooling function. The representation update function U_i^l compute the new hidden representation with RELU activation function and dropout technique during learning stage. Finally, the GNN concatenates (\oplus symbol) all final nodes representations and computes the policy function with the Softmax activation function.

$$\mathbf{y} = \sigma^L(\bigoplus_{i=0}^n \mathbf{W}_i^L \mathbf{h}_i^{L-1} + \mathbf{b}_i^L) \quad (5)$$

¹We omit the I2I relation because there is only one I-node.

²The notation $i \leftarrow j$ denotes a message sending from slot j to slot i . It also corresponds to the directed relation between the slots j and i . The notation $i \leftarrow *$ denotes all messages sending to slot i .

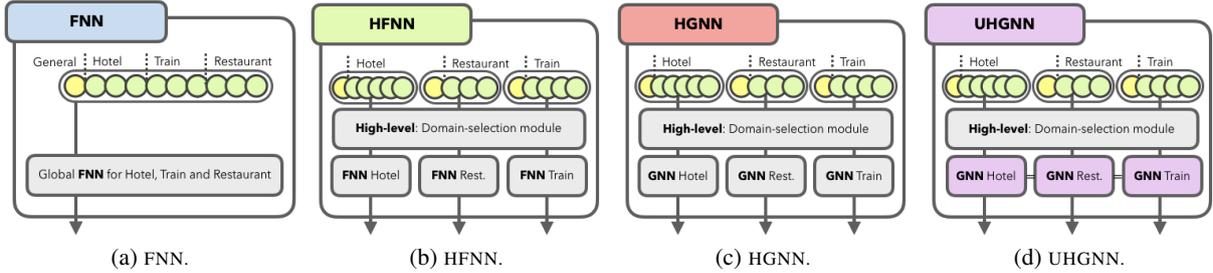


Figure 2: Policy and input data structures. Different levels of structure are presented from classical *feed-forward neural network* (FNN) to *graph neural network* (GNN). The prefix H- corresponds to a hierarchical policy and UH- corresponds to a unique sub-policy for all domains. For a FNN layer, the input data is the concatenation of all DIP slot representations. For a GNN layer, the input keeps its structure.

3.3 Structured Policies

We propose a wide range of dialogue policies to study the impact of the structure in sample efficiency. An ablation study progressively adds some notion of hierarchy to FNNs to approximate the structure of GNNs. Similarly, we analyse the advantage of sharing a generic GNN among several domains versus specialising a GNN to each domain. Therefore, we propose from the least to the most constrained:

- **Feed-forward Neural Network (FNN)** that is a classical feed-forward neural network with DIP parametrisation (Figure 2a).
- **Hierarchy of Feed-forward Neural Networks (HFNN)** that is a hierarchical policy with hand-crafted domain-selection and FNNs for each domain. Each domain has one corresponding FNN model (Figure 2b).
- **Hierarchy of Graph Neural Networks (HGNN)** that is a hierarchical policy with hand-crafted domain-selection and GNNs. Each domain has one corresponding GNN model (Figure 2c).
- **Hierarchy with Unique Graph Neural Network (UHGNN)** that is a HGNN with a unique GNN for all domains. Each domain shares the same GNN model (Figure 2d).

3.4 The Expert’s Nature

Since our goal is to learn on observed demonstrations delivered by an expert, we propose to focus on policies that learn from both simulated and human experts. For this purpose, we use the dataset MULTIWOZ (Budzianowski et al., 2018) to follow

human experts and the hand-crafted policy of CONVLAB (Zhu et al., 2020) as the simulated expert.

Human expert The MULTIWOZ dataset is a large annotated and open-sourced collection of human-human chats that covers multiple domains and tasks. Nearly 10k dialogues have been collected by a *Wizard-of-Oz* set-up at relatively low cost and with a small time effort. However, different versions of this dataset corrected and improved the annotations (Eric et al., 2020; Zang et al., 2020; Han et al., 2021; Ye et al., 2021). In this work, we use the MULTIWOZ dataset integrated in CONVLAB with extended user dialogue act annotations.

Simulated expert The CONVLAB framework has been proposed to automatically build, train and evaluate multi-domain multi-task oriented dialogue systems based on MULTIWOZ features. It implements both hand-crafted simulated user and policy. The latter has been shown to be nearly the optimal policy according to the CONVLAB evaluation setup of (Takanobu et al., 2020). Therefore we use it as the simulated expert.

4 Experiments

In this section we explain the experimental setup, the proposed models and the evaluation metrics.

4.1 Experiment Setup

We performed an ablation study by gradually adding different levels of structure from a baseline FNN to the proposed GNN (Subsection 4.2). On the one hand, we analyse the learning efficiency of our models in small training steps. On the other hand, we compare their generalisation ability in few shot learning.

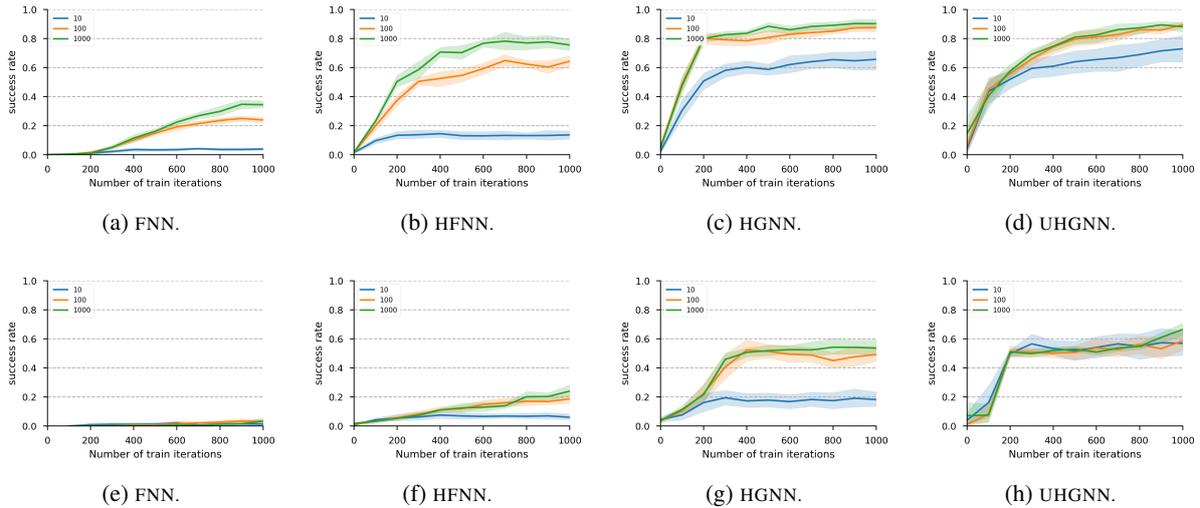


Figure 3: Dialogue manager evaluation with simulated users. We present the success rate on 10 / 100 / 1 000 training dialogues as a function of the number of gradient descent steps in a short training scenario. Learning is based on simulated experts (Figures (a) up to (d)) or on human experts (Figures (e) up to (h)). The line plot represents the mean and the coloured area represents the 95% confidence interval over a sample of 10 runs.

To analyse the learning efficiency, we measure performance with respect to the number of gradient descent steps up to 1 000 iterations with a step size of 100 iterations. We compare learning curves based on randomly chosen 10, 100 and 1 000 training dialogues³. We also measure performance as a function of the number of training dialogues available (randomly chosen) namely 10, 50, 100, 500 and 1000 when each training is performed up to 10 000 gradient descent steps. All the experiments were run on CONVLAB, restarted 10 times with random initialisation and the results estimated on 500 new dialogues.

4.2 Models

The FNN models have two hidden layers, both with 128 neurons. The GNN models have one first hidden layer with 64 neurons for both nodes (S-NODE and I-NODE). Then the second hidden layer is composed of 64 neurons for each relation (S2S, S2I and I2S). For training stage, we use the ADAM optimiser with a learning rate $lr = 0.001$, a dropout rate $dr = 0.1$ and a batch size $bs = 64$.

4.3 Metrics

We evaluate the performance of the policies for all tasks as in CONVLAB. Precision, recall and F-score, namely the **inform rates**, are used for the

³These values were chosen arbitrarily to give us an insight into the impact of the number of dialogues on the performance.

find task. Inform recall evaluates whether all the requested information has been informed while inform precision evaluates whether only the requested information has been informed. For the *book* task, the accuracy, namely the **book rate**, is used. It assesses whether the offered entity meets all the constraints specified in the user goal. The dialogue is marked as **successful** if and only if both inform recall and book rate are equal to 1. The dialogue is considered **completed** if it is successful from the user’s point of view⁴.

5 Evaluation

First, we evaluate the dialogue manager performance when talking to a simulated user. Second, we evaluate the learned policies within the entire dialogue system both with simulated and with real users. The evaluations have been done within CONVLAB.

5.1 Dialogue Manager Evaluation

We analyse our models on the learning efficiency in small training steps and on the ability to generalise in a few-shot setting.

Efficiency We report in Figure 3 the results of the ablation study showing the ability of the models to succeed in a short training stage. First, when

⁴A dialogue can be completed without being successful if the information provided is not the one objectively expected by the simulator.

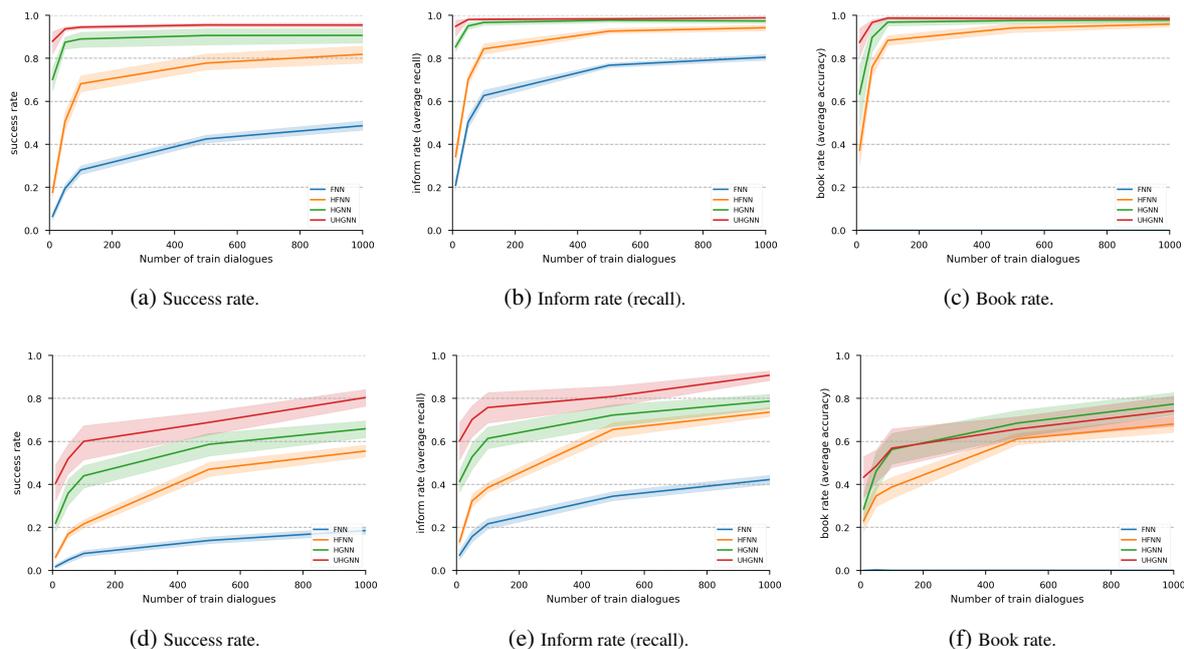


Figure 4: Dialogue manager evaluation with simulated user presenting the success rate based on 10 000 training iterations as a function of the number of training dialogues in a long learning scenario. Learning is based on a simulated expert (Figures (a), (b) and (c)) or human experts (Figures (d), (e) and (f)). The line plot represents the mean and the coloured area represents the 95% confidence interval over a sample of 10 runs.

learning from simulated demonstrations we notice in Figure 3a that the baseline (FNN) needs a large number of training dialogues (more than 100) to achieve a moderate performance (less than 40%). We show then in Figure 3b that hierarchical networks (HFNN) do improve learning efficiency up to 60% with 100 dialogues, up to 80% with 1 000 dialogues. Finally we show that graph neural network (HGNN in Figure 3c) and generic policy (UHGNN in Figure 3d) drastically improve the efficiency with few dialogues, more than 60% with 10 dialogues, and achieve remarkable performance above 80% with only 100 dialogues in 1 000 training steps. These observations confirm that hierarchical and generic GNNs allow efficient learning and collaborative gradient update in a short training stage.

Although standard or hierarchical policies (FNN in Figure 3e and HFNN in Figure 3f) are less efficient when learning from human demonstrations, they are still above baselines. It is worth noting that structured or generic GNN policies HGNN in Figure 3g and UHGNN in Figure 3h are able to reach more than 50% success rate.

Few-Shot We extended the ablation study in a few-shot scenario focusing on the ability of the

models to succeed on specific dialogue tasks as reported in Figure 4. In particular, we show the success rate in Figure 4a, the inform rate (recall) in Figure 4b and the book rate in Figure 4c when using simulated demonstrations and respectively in Figure 4d, Figure 4e and Figure 4f when using human demonstrations. The more structured the model, the greater the learning efficiency and the greater the data efficiency. Likewise, we notice that learning is more data-intensive when imitating human strategies. It appears that the booking task is more difficult to perform according to human demonstrations (when comparing Figure 4c and Figure 4f) or using a flat architecture (FNN gets null results). We therefore foresee that more high quality data is needed to learn on human dialogues.

5.2 Dialogue System Evaluation

We continue our analysis on the robustness of the studied models with the entire dialogue system facing both simulated and human users. The dialogue system utilises a BERT NLU (Devlin et al., 2019) and a hand-crafted NLG.

Simulated User Evaluation As in the previous subsection, we study the robustness of the models in a few-shot scenario as presented in Figure 5.

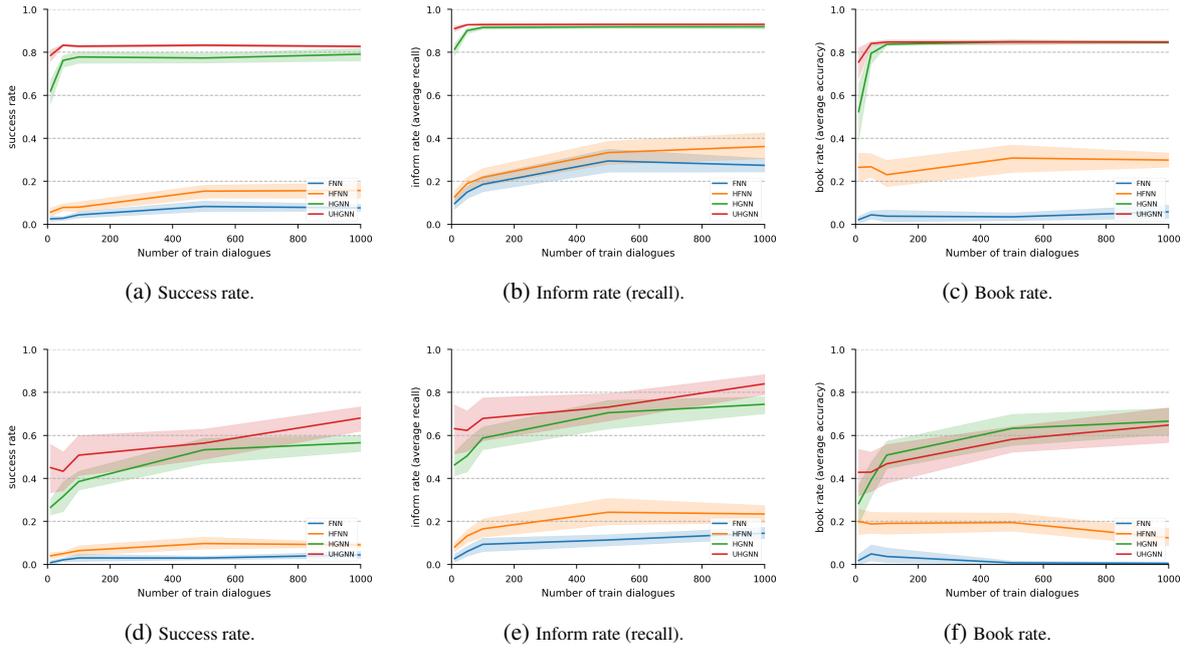


Figure 5: Dialogue system performance with simulated user based on 10 000 training iterations as a function of the number of training dialogues in a long training scenario. The supervised DM is based on simulated demonstrations (Figures (a),(b),(c)) or on human demonstrations (Figures (d),(e),(f)). The line plot represents the mean and the coloured area represents the 95% confidence interval over a sample of 10 runs.

We observe that FNN (in blue) and HFNN (in orange) learning is collapsing when using simulated dialogues (see Figures 5a, 5b and 5c). On the opposite, HGNN (in green) and UHGNN (in red) performance appears more stable in the entire dialogue system even when using real dialogues (see Figures 5d, 5e and 5f). Therefore, these results confirm that behaviour cloning is easier from simulated than human experts. As observed before in Subsection 5.2, this can be explained by a large variability of human strategies (hence the need for more data to improve performance). Another explanation is that simulated dialogues are more in line with the artificial evaluator provided in the CONVLAB. In addition, it is important not to neglect the side effects of cascading errors due to successive NLU, DST, DM and NLG modules. In particular, the NLU BERT proposed by CONVLAB was pre-trained and evaluated on 7 372 user utterances with 14% of errors (F1 86.4%, precision 85.1%, recall 87.8%). This problem can therefore be exacerbated by cascading human errors, as confirmed in the next paragraph.

Finally, we present a detailed comparison table with the best structured policies UHGNN trained on simulated dialogues of CONVLAB noted MLE-

UHGNN-HDC (HDC for *hand-crafted policy*) and trained on real dialogues of MULTIWOZ noted MLE-UHGNN-MW and the baselines of CONVLAB (see Table 1). In particular, the *maximum likelihood estimator* (MLE) proposed by CONVLAB is an implementation of FNN model trained on MULTIWOZ corpus in a very long training scenario (multiple passes on all 10k dialogues)⁵. Our models show competitive results against CONVLAB’s baselines, confirming that the structured with supervised learning in few-shot settings is adapted to address the difficulties in multi-task multi-domain dialogues.

Human Evaluation We organised preliminary evaluation sessions, in which volunteers were invited to chat on-line with three dialogue systems that were randomly assigned⁶. Subjects do not know which system they are evaluating. Each sys-

⁵Another difference is that our models returns one unique action per turn instead of a group of actions.

⁶Crowdsourcing was not used because of ethical concerns regarding the work conditions of collaborators. Volunteers from our research institution were invited to participate and they were aware of the scientific motivations behind the evaluation. In this sense, they were motivated to participate without any economic reward implying no pressure and without knowing the nature of the models they were evaluating, avoiding in this way any evaluation bias.

Configuration	Avg Turn (succ/all)	Inform rate (%) Prec. / Rec. / F1	Book Rate (%)	Complete Rate (%)	Success Rate (%)		
Dialogue Management							
HDC	10.6/10.6	87.2 / 98.6 / 90.9	98.6	97.9	-	97.3	-
MLE-UHGNN-HDC (ours)	12.8/13.0	95.3 / 98.8 / 96.4	98.5	97.3	(-0.6)	95.4	(-1.9)
MLE-UHGNN-MW (ours)	16.5/20.7	94.3 / 90.7 / 91.6	76.7	81.4	(-16.5)	81.0	(-6.3)
Dialogue System (BERT NLU + hand-crafted NLG)							
HDC	11.4/12.0	82.8 / 94.1 / 86.2	91.5	92.7	-	83.8	-
HDC [†]	11.6/12.3	79.7 / 92.6 / 83.5	91.1	90.5	(-2.2)	81.3	(-2.5)
MLE [†]	12.1/24.1	62.8 / 69.8 / 62.9	17.6	42.7	(-50.0)	35.9	(-47.9)
PG [†]	11.0/25.3	57.4 / 63.7 / 56.9	17.4	37.4	(-55.3)	31.7	(-52.1)
GDPL [†]	11.5/21.3	64.5 / 73.8 / 65.6	20.1	49.4	(-43.3)	38.4	(-45.4)
PPO [†]	13.1/17.8	69.4 / 85.8 / 74.1	86.6	75.5	(-17.2)	71.7	(-12.1)
MLE-UHGNN-HDC (ours)	14.0/15.4	89.3 / 93.0 / 90.2	84.8	90.0	(-2.7)	82.7	(-1.1)
MLE-UHGNN-MW (ours)	17.0/23.0	84.0 / 87.6 / 84.5	64.8	72.1	(-20.6)	68.1	(-15.7)

Table 1: Dialogue manager and system evaluations with simulated users. When evaluating the dialogue manager, the simulated user passes directly dialogue acts and vice-versa. Our tested configurations are evaluated and averaged on 10 run each with 250 dialogues. Configurations with [†] are taken from the [GitHub of CONVLAB](#).

Dialogue System (BERT NLU + Rule NLG)	Avg Turn	Satisfaction Rate (%)	Nb of Dial.
HDC	22.6	92.6 ± 9.87	27
MLE-UHGNN-HDC	25.6	50.0 ± 14.8	44
MLE-UHGNN-MW	17.3	36.7 ± 17.2	30

Table 2: Dialogue system evaluation with real users with a 95% confidence level for satisfaction rate.

tem has a different DM model: HDC (*hand-crafted policy*), MLE-UHGNN-HDC (based on simulated demonstrations with HDC policy) and MLE-UHGNN-MW (based on MULTIWOZ demonstrations) combined with the BERT NLU and the hand-crafted NLG provided by CONVLAB. At the end of the chat, evaluators were asked whether or not they reach the goal and were satisfied with the performance of the system. The **satisfaction rate** is then the proportion of dialogues in which the system solved the task at the end of the dialogue according to the human evaluator. We reported results on roughly 30 dialogues for each method. The results of this experimentation are presented in Table 2. Although test is small-sized and not highly statistically significant, these preliminary results are disconcerting with respect to the simulated ones. The HDC does very well whereas MLE-UHGNN-HDC gets by in half the cases, MLE-UHGNN-MW fails in most cases.

These results can be explained by the limitations of the NLU facing impatient evaluators, short and ambiguous sentences where the active domain is

unclear (as in this example of the user saying "What is the name?") or typographical errors. Moreover, it is important to underline that CONVLAB does not natively propose the management of uncertainties in the state representation which can strongly restrict the performance of the learning methods in noisy environments. Another limitation is that the HDC is more adapted to conventional dialogues whereas MLE-UHGNNs were trained only on winning dialogues. This implies that learning methods are more sensitive to dialogues that break out of the learned patterns. Similarly, the strategies of simulated and real users do not seem to be well aligned with each other and even more strongly with the expectations of human evaluators.

6 Conclusion

We investigated in this work the impact of policy structure and experts on success rate in few-shot learning for multi-domain multi-task dialogues. Promising results were obtained: hierarchical and generic GNN policies are able to achieve remarkable performance with few dialogues and few training iterations when following a simulated expert. This confirms the growing interest for these neural structures. We also present an important finding: the policy performance degrades in few-shot learning when using human demonstrations. This fact questions the alignment between dialogue evaluators and human strategies in state-of-the-art dialogue frameworks.

Limitations

The reduced performance when learning from human experts suggests that we shall concentrate the efforts in bridging the gap between automatic evaluators and high-quality human-human datasets. We also devise the use of *curriculum learning* (Bengio et al., 2009) strategies: starting from simple – simulated – dialogues then adding progressively more complex, human dialogues demonstrations.

It is also necessary to analyse the impact of GNN policies with neural NLU/NLG modules to study how to integrate such structures in end-to-end architectures.

We point out some limitations of CONVLAB. The detection of the active domain is sensitive to the output of the NLU and thus sensitive to ambiguous statements. Data representation restricts the DST to a deterministic view and must be adapted to a probabilistic representation to capture the uncertainties in the user’s input. Similarly, it may be worthwhile to improve the action space by adding more possibilities for human users, for instance to CONFIRM or DENY in a more flexible way.

Finally, the human evaluation was performed on a small scale and on models trained in a context with few training iterations. A more in-depth or supervised study could shed more light on the raised issues.

Acknowledgements

This work was performed using HPC resources from GENCI-IDRIS (Grant 20XX-[AD011011407]).

References

- John Langshaw Austin. 1975. *How to do things with words*. Oxford university press.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.
- Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Inigo Casanueva, Lina Rojas-Barahona, and Milica Gašić. 2017. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. *arXiv preprint arXiv:1706.06210*.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*.
- Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Stefan Ultes, Lina M Rojas Barahona, Bo-Hsiang Tseng, and Milica Gasic. 2018. Feudal reinforcement learning for dialogue management in large domains. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 714–719.
- Lu Chen, Bowen Tan, Sishan Long, and Kai Yu. 2018. Structured dialogue policy with graph neural networks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1257–1268.
- Zhi Chen, Lu Chen, Xiaoyuan Liu, and Kai Yu. 2020. Distributed structured actor-critic reinforcement learning for universal dialogue management. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2400–2411.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tür. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*, page 422–428. Marseille, France. European Language Resources Association.
- Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374.
- Victor Garcia and Joan Bruna. 2018. Few-shot learning with graph neural networks. In *6th International Conference on Learning Representations, ICLR 2018*.
- Ting Han, Ximing Liu, Ryuichi Takanabu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2021. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 206–218. Springer.
- Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudik, Yisong Yue, and Hal Daumé III. 2018. Hierarchical imitation and reinforcement learning. In *International conference on machine learning*, pages 2917–2926. PMLR.

- Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211.
- Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao, and Minlie Huang. 2020. Is your goal-oriented dialog model performing really well? empirical analysis of system-wise evaluation. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 297–310.
- Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, and Milica Gasic. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78.
- Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34.
- Zhuoran Wang, Tsung-Hsien Wen, Pei-Hao Su, and Yannis Stylianou. 2015. Learning domain-independent dialogue policies via ontology parameterisation. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 412–416.
- Zheng Wen, Doina Precup, Morteza Ibrahimi, Andre Barreto, Benjamin Van Roy, and Satinder Singh. 2020. On efficiency in hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 33:6708–6718.
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.
- Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2021. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. *arXiv preprint arXiv:2104.00773*.
- Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. [MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines](#). In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.
- Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.
- Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149.

Transfer Knowledge from Natural Language to Electrocardiography: Can We Detect Cardiovascular Disease Through Language Models?

Jielin Qiu^{1*}, William Han^{1*}, Jiacheng Zhu¹, Mengdi Xu¹,
Michael Rosenberg³, Emerson Liu², Douglas Weber¹, Ding Zhao¹

¹Carnegie Mellon University, ²Allegheny General Hospital, ³University of Colorado

Abstract

Recent advancements in Large Language Models (LLMs) have drawn increasing attention since the learned embeddings pretrained on large-scale datasets have shown powerful ability in various downstream applications. However, whether the learned knowledge by LLMs can be transferred to clinical cardiology remains unknown. In this work, we aim to bridge this gap by transferring the knowledge of LLMs to clinical Electrocardiography (ECG). To address this problem, we propose an approach for cardiovascular disease diagnosis and automatic ECG diagnosis report generation. We also introduce an additional loss function by Optimal Transport (OT) to align the distribution between ECG and language embeddings. The learned embeddings are evaluated on two downstream tasks: (1) automatic ECG diagnosis report generation, and (2) zero-shot cardiovascular disease detection. Our approach is able to generate high-quality cardiac diagnosis reports and also achieves competitive zero-shot classification performance even compared with supervised baselines, which proves the feasibility of transferring knowledge from LLMs to the cardiac domain.

1 Introduction

Heart and cardiovascular diseases are the leading global cause of death, with 80% of cardiovascular disease-related deaths due to heart attacks and strokes. The clinical 12-lead ECG, when correctly interpreted, is the primary tool to detect cardiac abnormalities and heart-related issues. ECG provides unique information about the structure and electrical activity of the heart and systemic conditions through changes in the timing and morphology of the recorded waveforms. Achievements of ECG interpretation, such that critical and timely ECG interpretations of cardiac conditions, will lead to efficient and cost-effective intervention.

* marked as equal contribution

LLM starts from the Transformer model (Vaswani et al., 2017) and grows quickly with a wide range of applications (Devlin et al., 2019; Liu et al., 2019b; Brown et al., 2020). Recently, LLM has shown great potential for accelerating learning in many other domains since the learned embeddings can provide meaningful representation for downstream tasks. Examples include transferring the knowledge of LLM to, i.e., robotics control (Liang et al., 2022; Ahn et al., 2022), multimodal reasoning and interaction (Zeng et al., 2022; Zellers et al., 2021), robotics planning (Shah et al., 2022; Kant et al., 2022; Jain et al., 2022), decision-making (Li et al., 2022; Huang et al., 2022), robotics manipulation (Shridhar et al., 2022; Ren et al., 2022; Cui et al., 2022; Tam et al., 2022; Khandelwal et al., 2022), code generation (Fried et al., 2022), laws (Kaplan et al., 2020), computer vision (Radford et al., 2021), and so on.

Some previous works explored LLM and biological protein (Rives et al., 2021), or health records (Yang et al., 2022). However, the medical or health-care domains contain so much domain knowledge that different sources preserve unique data characteristics without a unified paradigm. To the best of our knowledge, no previous work explores the knowledge transfer from LLM to cardiovascular disease with ECG signals.

In this work, we bridge the gap between LLM and clinical ECG by investigating the feasibility of transferring knowledge of LLM to the cardiology domain. Our contributions are listed as follows:

- To the best of our knowledge, our work is the first attempt to bridge the gap between LLM and clinical cardiovascular ECG by leveraging the knowledge from pretrained LLM.
- We propose a cardiovascular disease diagnosis and automatic ECG diagnosis report generation approach by transferring the knowledge from LLM to the cardiac ECG domain.
- We introduce an additional learning objective

based on Optimal Transport distance, which empowers the model to learn the distribution between ECG and language embedding.

- Our method can generate high-quality cardiac diagnosis reports and achieve competitive zero-shot classification performance even compared with supervised baselines, proving the feasibility of using LLM to enhance research and applications in the cardiac domain.

2 Related Work

Cardiovascular diagnosis via ECG The 12-lead ECG is derived from 10 electrodes placed on the surface of the skin (Cadogan, 2020). An ECG works by recording electrical activity corresponding to the heartbeat muscle contractions (Bonow et al., 2011). Although computerized interpretations of ECGs are widely used, automated approaches have not yet matched the quality of expert cardiologists, leading to poor patient outcomes or even fatality (Breen et al., 2019).

Deep learning in ECG Deep learning approaches have been rapidly adopted in many fields for their accuracy and flexibility, including ECG domain (Kiranyaz et al., 2015; Nonaka and Seita, 2021; Khurshid et al., 2021; Raghunath et al., 2021; Giudicessi et al., 2021; Strodthoff et al., 2021; Al-Zaiti et al., 2020; Acharya et al., 2017; Shanmugam et al., 2019; Śmigiel et al., 2021). Transformer (Vaswani et al., 2017) has recently been adopted in several ECG applications, i.e., arrhythmia classification, abnormalities detection, stress detection, etc (Yan et al., 2019; Che et al., 2021; Natarajan et al., 2020; Behinaein et al., 2021; Song et al., 2021; Weimann and Conrad, 2021).

LLM in healthcare Zhou et al. (2021) reviewed existing studies concerning NLP for smart healthcare. Yang et al. (2022) developed a large pre-trained clinical language model using transformer architecture. Steinberg et al. (2021) showed that using patient representation schemes inspired by techniques in LLM can increase the accuracy of clinical prediction models. More related work can be found in Appendix B.

3 Methods

Problem Formulation We formulate the problem as generating cardiovascular diagnosis reports through pretrained LLMs. Given ECG signals $x = [x_1, x_2, \dots, x_t]$, our goal is to take advantage

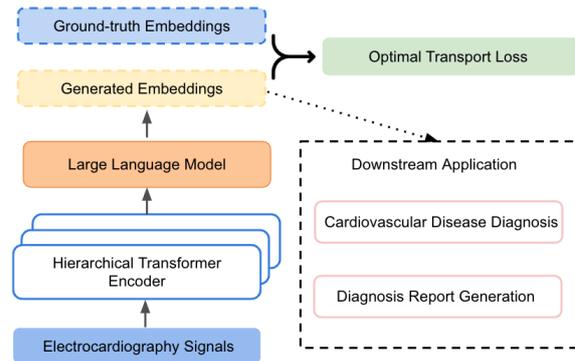


Figure 1: The architecture of our model. The Transformer encoder takes input ECG to generate ECG features as the input to LLM, where LLM transforms it into generated embeddings. An optimal transport based loss objective is formulated on generated embeddings and ground-truth embeddings for the model update.

of the knowledge from LLM and learn a generated text embedding $L = [L_1, L_2, \dots, L_m]$, which can then be decoded into natural language as reports or directly used for disease classification.

Model Architecture The model architecture is shown in Fig. 1, The ECG inputs are processed by hierarchical transformer encoders (Vaswani et al., 2017) to obtain transformed ECG embeddings $X = [X_1, X_2, \dots, X_n]$. Then we adopt a pretrained LLM to transform the ECG embeddings into language embeddings $L = [L_1, L_2, \dots, L_m]$. For the learning objective, we use expert reports to formalize the learning loss, which includes a new loss based on Optimal Transport (OT) in addition to the traditional cross-entropy loss. The learning objective is to update the transformer encoders, which can be interpreted as a sequence-to-sequence mapping from ECG embeddings X to sentence embeddings L . After the learning process, the learned embedding L should be capable of conducting downstream applications.

Downstream Applications For the downstream applications, we first consider a classification problem that uses the embeddings L for cardiovascular disease diagnosis. In addition, we consider a text generation task by decoding the output embeddings L into a cardiovascular report.

Transformer Encoders The transformer is based on the attention mechanism (Vaswani et al., 2017). The original transformer model is composed of an encoder and a decoder. The encoder maps an input sequence into a latent representation, and the

decoder uses the representation with other inputs to generate a target sequence. Our model only adopts the encoder since the target is to learn the representations of ECG features. More details can be found in Appendix D.

Optimal Transport Loss OT is the problem of transporting mass between two discrete distributions supported on latent feature space \mathcal{X} . Let $\boldsymbol{\mu} = \{\mathbf{x}_i, \mu_i\}_{i=1}^n$ and $\boldsymbol{v} = \{\mathbf{y}_j, v_j\}_{j=1}^m$ be the distributions of generated embeddings and ground-truth embeddings, where $\mathbf{x}_i, \mathbf{y}_j \in \mathcal{X}$ denotes the spatial locations and μ_i, v_j , respectively, denoting the non-negative masses. Without loss of generality, we assume $\sum_i \mu_i = \sum_j v_j = 1$. $\pi \in \mathbb{R}_+^{n \times m}$ is a valid transport plan if its row and column marginals match $\boldsymbol{\mu}$ and \boldsymbol{v} , respectively, which is $\sum_i \pi_{ij} = v_j$ and $\sum_j \pi_{ij} = \mu_i$. Intuitively, π transports π_{ij} units of mass at location \mathbf{x}_i to new location \mathbf{y}_j . Such transport plans are not unique, and one often seeks a solution $\pi^* \in \Pi(\boldsymbol{\mu}, \boldsymbol{v})$ that is most preferable in other ways, where $\Pi(\boldsymbol{\mu}, \boldsymbol{v})$ denotes the set of all viable transport plans. OT finds a solution that is most cost-effective w.r.t. cost function $C(\mathbf{x}, \mathbf{y})$:

$$\mathcal{D}(\boldsymbol{\mu}, \boldsymbol{v}) = \sum_{ij} \pi_{ij}^* C(\mathbf{x}_i, \mathbf{y}_j) = \inf_{\pi \in \Pi(\boldsymbol{\mu}, \boldsymbol{v})} \sum_{ij} \pi_{ij} C(\mathbf{x}_i, \mathbf{y}_j) \quad (1)$$

where $\mathcal{D}(\boldsymbol{\mu}, \boldsymbol{v})$ is known as OT distance. $\mathcal{D}(\boldsymbol{\mu}, \boldsymbol{v})$ minimizes the transport cost from $\boldsymbol{\mu}$ to \boldsymbol{v} w.r.t. $C(\mathbf{x}, \mathbf{y})$. When $C(\mathbf{x}, \mathbf{y})$ defines a distance metric on \mathcal{X} , and $\mathcal{D}(\boldsymbol{\mu}, \boldsymbol{v})$ induces a distance metric on the space of probability distributions supported on \mathcal{X} , it becomes the Wasserstein Distance (WD). We use WD as one loss objective, in addition to the standard cross-entropy loss, for the model update.

4 Dataset and Preprocessing

Dataset We conducted the experiments on the PTB-XL dataset (Wagner et al., 2020), which contains clinical 12-lead ECG signals of 10-second length. There are five conditions in total, including Normal ECG (NORM), Myocardial Infarction (MI), ST/T Change (STTC), Conduction Disturbance (CD), and Hypertrophy (HYP). The waveform files are stored in WaveForm DataBase (WFDB) format with 16-bit precision at a resolution of $1\mu\text{V}/\text{LSB}$ and a sampling frequency of 100Hz. The ECG statements conform to the SCP-ECG standard and cover diagnostic, form, and rhythm statements.

Preprocessing The raw ECG signals are first processed by the WFDB library (Xie et al., 2022) and Fast Fourier transform (FFT) to process the time series data into the spectrum, which is shown in Fig. 2. Then we perform n-points window filtering to filter the noise within the original ECG signals and adopt notch processing to filter power frequency interference (noise frequency: 50Hz, quality factor: 30). The ECG signals are segmented by dividing the 10-second ECG signals into individual ECG beats. We first detect the R peaks of each signal by ECG detectors (Porr et al., 2022), and then slice the signal at a fixed-sized interval on both sides of the R peaks to obtain individual beats. More details can be found in Appendix C.

Feature Extraction Instead of directly using the time-series signals, we extract time domain and frequency domain features to better represent ECG signals. The time-domain features include: maximum, minimum, range, mean, median, mode, standard deviation, root mean square, mean square, k-order moment and skewness, kurtosis, kurtosis factor, waveform factor, pulse factor, and margin factor. The frequency-domain features include: FFT mean, FFT variance, FFT entropy, FFT energy, FFT skew, FFT kurt, FFT shape mean, FFT shape std, FFT shape skew, FFT kurt. More details can be found in Appendix C. An analysis of the statistics of the processed ECG data can also be found in Table 1.

Table 1: Statistics of the processed ECG data.

Category	Patients	Percentage	Beats	Percentage
NORM	9528	34.2%	28419	36.6%
MI	5486	19.7%	10959	14.1%
STTC	5250	18.9%	8906	11.5%
CD	4907	17.6%	20955	27.0%
HYP	2655	9.5%	8342	10.8%

5 Experiments

5.1 Experimental Settings

Data and Model The dimension of the processed ECG is 864, including 600 ECG signals and 264 time & frequency domain features. Experiments are conducted on two NVIDIA A6000 GPUs. All the models’ parameters are listed in Appendix A.

Tasks To evaluate the learned embeddings from ECG signals, we tested the performance on two downstream applications: automatic cardiac report generation as a text generation (TG) task, and

Table 2: Comparisons of different backbones on Text generation (TG) and Disease detection (DD). (BERT as LLM)

Different backbones + BERT as LLM	Text generation (TG)						Disease detection (DD)		
	BLEU-1(%)	ROUGE-1(%)			Meteor(%)	BertScore(%)	Acc	AUCROC	F-1
		P	R	F					
MLP (Rumelhart et al., 1986)	22.24	17.68	22.63	18.11	14.27	84.68	0.71	0.89	0.57
LSTM (Hochreiter and Schmidhuber, 1997)	19.74	19.76	18.83	17.99	19.54	84.74	0.73	0.89	0.55
ResNet (He et al., 2016)	21.14	20.35	30.67	25.08	19.55	86.88	0.70	0.86	0.59
Transformer (Vaswani et al., 2017)	26.93	25.35	35.67	28.08	21.23	88.90	0.77	0.92	0.68

Table 3: Comparisons of different LLMs on Text generation (TG) and Disease detection (DD). (Transformer as the encoder).

Different LLMs	Text generation (TG)						Disease detection (DD)		
	BLEU-1(%)	ROUGE-1(%)			Meteor(%)	BertScore(%)	Acc	AUCROC	F-1
		P	R	F					
BERT (Devlin et al., 2019)	26.93	25.35	35.67	28.08	21.23	88.90	0.77	0.92	0.68
BART (Lewis et al., 2020)	27.21	26.12	35.71	29.56	24.51	89.61	0.75	0.88	0.68
RoBERTa (Liu et al., 2019b)	27.01	25.31	36.01	27.88	22.41	89.72	0.77	0.89	0.70
BioClinical BERT (Alsentzer et al., 2019)	27.91	25.41	36.33	28.42	23.54	87.21	0.78	0.89	0.71
PubMed BERT (Gu et al., 2022)	27.89	25.21	35.97	27.70	24.00	88.56	0.77	0.88	0.69
BioDischargeSummary BERT (Alsentzer et al., 2019)	26.81	25.32	35.66	28.10	21.19	88.90	0.73	0.85	0.66

Table 4: Comparisons with supervised baselines (DD).

Supervised learning baselines	Acc	AUROC	F-1
Transformer (Zhu et al., 2022)	0.75	0.843	0.575
CNN (Śmigiel et al., 2021)	0.72	0.877	0.611
SincNet (Ravanelli and Bengio, 2018)	0.73	0.84	0.6
Contrastive Learning (Lan et al., 2022)	-	0.722	-
CNN + Entropy (Śmigiel et al., 2021)	0.76	0.910	0.68
Ours _{BERT}	0.77	0.92	0.68

zero-shot cardiac disease detection (DD) as a multi-class classification task.

Evaluation For text generation evaluation, we adopted the BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), Meteor (Banerjee and Lavie, 2005), and BertScore (Zhang et al., 2020) as evaluation metrics. We report the standard classification evaluation metrics for zero-shot cardiac disease detection: accuracy, AUCROC, and F-1 score.

5.2 Results

In Table 2, we showed the performance of both text generation and disease detection tasks with different backbone models as baselines. We found that the Transformer encoder outperforms other backbones, i.e., MLP, LSTM, and ResNet, showing Transformer encoder could be a good selection as the feature extractor.

In Table 4, we showed the performance of our zero-shot disease detection approach, compared with supervised baselines. Even though our method is in the zero-shot setting, we can already achieve the same performance with state-of-the-art supervised learning methods, demonstrating that the transferred ECG representation from LLM is al-

Table 5: Examples of comparison on generated reports (marked as Predicted-X) and ground-truth reports (marked as GT-X).

Backbone	Reports
GT-1	“sinus rhythm left type peripheral low voltage”
Predicted-1	“ventricular arrhythmia flatfar arrhythmia”
GT-2	“sinus rhythm incomplete right block otherwise normal ekg”
Predicted-2	“ventricularrear extrasystole block sinus rhythm or normal.”

ready good for practical usage. We also showed some examples of generated reports compared with ground-truth reports in Table 5.

5.3 Ablation Study

Different LLM To further analyze the components, we conduct ablation studies on different LLMs and the number of transformer layers (with BERT as LLM). Table 3 shows the results of different LLMs for the text generation and disease detection tasks. We found that all LLMs showed good performance in both tasks, demonstrating that knowledge can be transferred from the language domain to the cardiac domain without constraints. BART shows good performance in the text generation task, while BioClinical BERT shows better performance in the disease detection task, though the variation between different LLMs is not large.

Transformer Layers To evaluate the impact of the number of transformer layers, we conducted additional experiments with different transformer layers, and the results are shown in Table 6. We

Table 6: Ablation study of different transformer layers.

Layers	Text generation (TG)					Disease detection (DD)			
	BLEU-1(%)	ROUGE-1(%)			Meteor(%)	BertScore(%)	Acc	AUCROC	F-1
		P	R	F					
1	25.81	20.36	30.72	23.12	21.38	83.58	0.69	0.83	0.59
2	24.77	19.22	28.55	24.51	20.44	82.89	0.72	0.81	0.61
3	25.44	20.44	27.21	24.81	19.99	84.63	0.75	0.80	0.62
4	25.12	21.36	30.88	25.76	22.68	86.35	0.74	0.80	0.64
5	26.93	25.35	35.67	28.08	21.23	88.90	0.77	0.92	0.68

Table 7: Comparisons with different backbones on the text generation task, where BERT is used as LLM.

Backbone	BLEU-1(%)	ROUGE-1(%)			Meteor(%)	BertScore
		P	R	F		
MLP	18.16	16.19	13.71	14.48	12.11	80.77
LSTM	19.72	19.67	18.83	17.99	19.54	84.73
Resnet	21.15	20.35	20.67	24.08	19.55	85.22
Transformer	24.51	23.22	30.81	26.19	20.02	85.44

Table 8: Comparisons with different backbones on the disease detection task, where BERT is used as LLM.

Backbone	Acc	AUCROC	F-1
MLP	0.69	0.77	0.49
LSTM	0.71	0.82	0.59
Resnet	0.70	0.83	0.55
Transformer	0.75	0.81	0.60

found that more layers could lead to better representations, achieving better performance for downstream applications.

ECG Time Series Signals Only For the results above, we used ECG signals along with ECG time & frequency domain features as inputs. To compare the performance, we also conducted the experiments by only using ECG signals as inputs, with no time & frequency domain features. This set of experiments can be considered an additional ablation study for the inputs. The results are shown in Tables 7, 8, 9, 10.

Compare Table 7 & 8 with Table 2, we can find that the performance of only using ECG signals as inputs is lower than combining time & frequency features as inputs in both text generation and disease detection tasks, which demonstrates that incorporating time & frequency features is useful for capturing the characteristics of ECG and can lead to better representations through LLM.

In Tables 9, 10, the transformer backbone performs the best compared to others in both disease detection and text generation tasks, which is in consistent with the findings in the paper, showing that more layers could lead to better representations,

Table 9: Comparisons of different number of transformer layers on the text generation task, where BERT is used as LLM.

LLM	BLEU-1(%)	ROUGE-1(%)			Meteor(%)	BertScore(%)
		P	R	F		
1	25.52	19.10	27.65	21.43	20.11	86.52
2	24.21	20.00	28.75	23.90	20.32	84.66
3	23.44	20.44	27.21	24.81	19.99	84.63
4	23.17	20.99	28.01	24.44	20.18	87.65
5	25.69	24.75	34.81	27.59	21.03	87.33

Table 10: Comparisons of different numbers of transformer layers on the disease detection task, where BERT is used as LLM.

Num of Layers	Acc	AUCROC	F-1
1	0.62	0.79	0.51
2	0.74	0.80	0.60
3	0.71	0.82	0.59
4	0.72	0.83	0.61
5	0.75	0.88	0.64

achieving better performance for downstream applications. In addition, compared with Table 6 in the paper, we can find that the performance in Tables 9 and 10 are lower than the ones in Table 6, which also proved the same findings that adding time & frequency features is useful for learning the cardiac ECGs.

6 Conclusion

In this paper, we bridge the gap between LLMs and cardiovascular ECG by transferring knowledge of LLMs into the cardiovascular domain. The transferred knowledge embeddings can be used for downstream applications, including cardiovascular disease diagnosis and automatic ECG diagnosis report generation. Our results demonstrate the effectiveness of knowledge transfer, as the proposed method shows excellent performance in both downstream tasks, where our zero-shot classification approach even achieved competitive performance with supervised learning baselines, showing the feasibility of using LLM to enhance applications in the cardiovascular domain.

7 Acknowledgements

The research is partially supported by the DARPA ADAPTER program, and partially supported by the Allegheny Health Network and Mario Lemieux Center for Innovation and Research in EP.

8 Limitations

Due to the constrain of the available datasets, we only conducted experiments on the PTB-XL dataset, which is the current largest ECG dataset that contains high-quality clinical ECG signals and cardiac reports by experienced cardiologists.

We understand that collecting high-quality clinical data is much more complicated and time-consuming than collecting other data from online resources, like images, since it requires expert domain knowledge and is limited by many privacy regulations. We are working with cardiologists, hospitals, and clinical research labs, hope we can release a new dataset to provide additional materials for this research direction.

9 Ethics Statement

In this work, the data used as experimental materials are from publicly available databases, where the patients' information is anonymized. To the best of our knowledge, we do not foresee any harmful uses of this study.

References

- U. Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, Muhammad Adam, Arkadiusz Gertych, and Ru San Tan. 2017. A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine*, 89:389–396.
- Michael Ahn et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *ArXiv*, abs/2204.01691.
- Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical bert embeddings. *ArXiv*, abs/1904.03323.
- Salah Al-Zaiti, Lucas Besomi, Zeineb Bouzid, Ziad Faramand, Stephanie O. Frisch, Christian Martin-Gill, Richard E. Gregg, Samir F. Saba, Clifton Callaway, and Ervin Sejdić. 2020. Machine learning-based prediction of acute coronary syndrome using only the pre-hospital 12-lead electrocardiogram. *Nature Communications*, 11.
- Ezra A. Amsterdam, J. Douglas Kirk, David A. Bluemke, Deborah B. Diercks, Michael E. Farkouh, J. Lee Garvey, Michael C Kontos, James McCord, Todd D. Miller, Anthony P Morise, L. Kristin Newby, Frederick L. Ruberg, Kristine Anne Scordo, and Paul D. Thompson. 2010. Testing of low-risk patients presenting to the emergency department with chest pain: a scientific statement from the american heart association. *Circulation*, 122 17:1756–76.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEevaluation@ACL*.
- Lanqing Bao, Jielin Qiu, Hao Tang, Wei-Long Zheng, and Bao-Liang Lu. 2019. Investigating sex differences in classification of five emotions from eeg and eye movement signals. *EMBC*, pages 6746–6749.
- Behnam Behinaein, Anubha Bhatti, Dirk Rodenburg, Paul C. Hungler, and Ali Etemad. 2021. A transformer architecture for stress detection from eeg. *2021 International Symposium on Wearable Computers*.
- Robert O Bonow, Douglas L Mann, Douglas P Zipes, and Peter Libby. 2011. *Braunwald's heart disease e-book: A textbook of cardiovascular medicine*. Elsevier Health Sciences.
- C.J. Breen, G.P. Kelly, and W.G. Kernohan. 2019. [Ecg interpretation skill acquisition: A review of learning, teaching and assessment](#). *Journal of Electrocardiology*.
- Tom B. Brown et al. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.
- Mike Cadogan. 2020. ECG Lead positioning.
- Chao Che, Peiliang Zhang, Min Zhu, Yue Qu, and Bo Jin. 2021. Constrained transformer network for ecg signal processing and arrhythmia classification. *BMC Medical Informatics and Decision Making*, 21.
- Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. 2020. Fine-grained video-text retrieval with hierarchical graph reasoning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10635–10644.
- Yuchen Cui, Scott Niekum, Abhi Gupta, Vikash Kumar, and Aravind Rajeswaran. 2022. Can foundation models perform zero-shot task specification for robot manipulation? *ArXiv*, abs/2204.11134.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.
- Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. 2021. Dual encoding for video retrieval by text. *IEEE transactions on pattern analysis and machine intelligence*, PP.

- Daniel Fried et al. 2022. InCoder: A generative model for code infilling and synthesis. *ArXiv*, abs/2204.05999.
- John R. Giudicessi, Matthew Schram, J. Martijn Bos, Conner Galloway, Jacqueline Baras Shreibati, Patrick W. Johnson, Rickey E. Carter, Levi W. Disrud, Robert B. Kleiman, Zach I. Attia, Peter A. Noseworthy, Paul A. Friedman, David E. Albert, and Michael J. Ackerman. 2021. Artificial intelligence-enabled assessment of the heart rate corrected qt interval using a mobile electrocardiogram device. *Circulation*.
- Yuxian Gu, Robert Tinn, Hao Cheng, Michael R. Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3:1 – 23.
- William Han, Jieli Qiu, Jiacheng Zhu, Mengdi Xu, Douglas Weber, Bo Li, and Ding Zhao. 2022. An empirical exploration of cross-domain alignment between language and electroencephalogram. *ArXiv*, abs/2208.06348.
- Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.
- Nora Hollenstein, Cédric Renggli, Benjamin James Glaus, Maria Barrett, Marius Troendle, Nicolas Langer, and Ce Zhang. 2021. Decoding eeg brain activity for multi-modal natural language processing. *Frontiers in Human Neuroscience*, 15.
- Renee Y. Hsia, Zachariah Hale, and Jeffrey A. Tabas. 2016. A national study of the prevalence of life-threatening diagnoses in patients with chest pain. *JAMA internal medicine*, 176 7:1029–32.
- Wenlong Huang, P. Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *ICML*.
- Vidhi Jain, Yixin Lin, Eric Undersander, Yonatan Bisk, and Akshara Rai. 2022. Transformers are adaptable task planners. *ArXiv*, abs/2207.02442.
- Yash Kant, Arun Ramachandran, Sriram Yenamandra, Igor Gilitschenski, Dhruv Batra, Andrew Szot, and Harsh Agrawal. 2022. Housekeep: Tidying virtual households using commonsense reasoning. *ArXiv*, abs/2205.10712.
- Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv*, abs/2001.08361.
- Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. 2022. Simple but effective: Clip embeddings for embodied ai. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14809–14818.
- Shaan Khurshid, Samuel N. Friedman, Christopher Reeder, Paolo Di Achille, Nathaniel Diamant, Pulkit Singh, Lia X. Harrington, Xin Wang, Mostafa A. Al-Alusi, Gopal Sarma, Andrea S. Foulkes, Patrick T. Ellinor, Christopher D Anderson, Jennifer E. Ho, Anthony A. Philippakis, Puneet Batra, and Steven A. Lubitz. 2021. Electrocardiogram-based deep learning and clinical risk factors to predict atrial fibrillation. *Circulation*.
- Serkan Kiranyaz, Turker Ince, Ridha Hamila, and M. Gabbouj. 2015. Convolutional neural networks for patient-specific eeg classification. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2608–2611.
- Xiang Lan, Dianwen Ng, Linda Qiao, and Mengling Feng. 2022. Intra-inter subject self-supervised learning for multivariate cardiac signals. In *AAAI*.
- Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. *ArXiv*, abs/1803.08024.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.
- Shuang Li, Xavier Puig, Yilun Du, Clinton Jia Wang, Ekin Akyürek, Antonio Torralba, Jacob Andreas, and Igor Mordatch. 2022. Pre-trained language models for interactive decision-making. *ArXiv*, abs/2202.01771.
- J. Liang, Wenlong Huang, F. Xia, Peng Xu, Karol Hausman, Brian Ichter, Peter R. Florence, and Andy Zeng. 2022. Code as policies: Language model programs for embodied control. *ArXiv*, abs/2209.07753.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*.
- Wei Liu, Jie-Lin Qiu, Wei-Long Zheng, and Bao-Liang Lu. 2019a. Multimodal emotion recognition using deep canonical correlation analysis. *ArXiv*, abs/1908.05349.
- Wei Liu, Jieli Qiu, Wei-Long Zheng, and Bao-Liang Lu. 2021. Comparing recognition performance and robustness of multimodal deep learning models for multimodal emotion recognition. *IEEE Transactions on Cognitive and Developmental Systems*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b.

- Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.
- Harry McGurk and John MacDonald. 1976. Hearing lips and seeing voices. *Nature*, 264:746–748.
- George B. Moody and Roger G. Mark. 2001. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20:45–50.
- Annamalai Natarajan, Yale Chang, Sara Mariani, Asif Rahman, Gregory Boverman, Shruti Gopal Viji, and Jonathan Rubin. 2020. A wide and deep transformer neural network for 12-lead ecg classification. *2020 Computing in Cardiology*, pages 1–4.
- Naoki Nonaka and Jun Seita. 2021. In-depth benchmarking of deep neural network architectures for ecg diagnosis. In *Proceedings of the 6th Machine Learning for Healthcare Conference*, Proceedings of Machine Learning Research, pages 414–439. PMLR.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.
- Bernd Porr, Luis Howell, Ioannis Stourmaras, and Yoav Nir. 2022. [Popular ecg r peak detectors written in python](#).
- Jielin Qiu, Franck Dernoncourt, Trung Bui, Zhaowen Wang, D. Zhao, and Hailin Jin. 2022a. Liveseg: Unsupervised multimodal temporal segmentation of long livestream videos. *ArXiv*, abs/2210.05840.
- Jielin Qiu, Ge Huang, and Tai Sing Lee. 2019. Visual sequence learning in hierarchical prediction networks and primate visual cortex. *Advances in neural information processing systems*.
- Jielin Qiu, W. Liu, and Bao-Liang Lu. 2018a. Multi-view emotion recognition using deep canonical correlation analysis. *International Conference on Neural Information Processing*.
- Jielin Qiu, Xin-Yi Qiu, and Kai Hu. 2018b. Emotion recognition based on gramian encoding visualization. *Brain Informatics*.
- Jielin Qiu and Wei-Ye Zhao. 2018. Data encoding visualization based cognitive emotion recognition with ac-gan applied for denoising. *ICCI*CC*, pages 222–227.
- Jielin Qiu, Jiacheng Zhu, Michael Rosenberg, Emerson Liu, and D. Zhao. 2022b. Optimal transport based data augmentation for heart disease diagnosis and prediction. *ArXiv*, abs/2202.00567.
- Jielin Qiu, Jiacheng Zhu, Mengdi Xu, Franck Dernoncourt, Trung Bui, Zhaowen Wang, Bo Li, Ding Zhao, and Hailin Jin. 2022c. Mhms: Multimodal hierarchical multimedia summarization. *ArXiv*, abs/2204.03734.
- Jielin Qiu, Jiacheng Zhu, Mengdi Xu, Franck Dernoncourt, Trung Bui, Zhaowen Wang, Bo Li, Ding Zhao, and Hailin Jin. 2022d. Semantics-consistent cross-domain summarization via optimal transport alignment. *ArXiv*, abs/2210.04722.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *ICML*.
- Sushravya Raghunath et al. 2021. Deep neural networks can predict new-onset atrial fibrillation from the 12-lead ecg and help identify those at risk of atrial fibrillation-related stroke. *Circulation*, 143:1287–1298.
- Mirco Ravanelli and Yoshua Bengio. 2018. Speaker recognition from raw waveform with sincnet. *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028.
- Allen Z. Ren, Bharat Govil, Tsung-Yen Yang, Karthik Narasimhan, and Anirudha Majumdar. 2022. Leveraging language for accelerated learning of tool manipulation. *ArXiv*, abs/2206.13074.
- Alexander Rives, Siddharth Goyal, Joshua Meier, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. 2021. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 118.
- David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning internal representations by error propagation.
- Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. 2022. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. *ArXiv*, abs/2207.04429.
- Divya Shanmugam, Davis Blalock, and John Guttag. 2019. Multiple instance learning for ecg risk stratification. In *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pages 124–139. PMLR.
- Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2022. Perceiver-actor: A multi-task transformer for robotic manipulation. *ArXiv*, abs/2209.05451.
- Sandra Śmigiel, Krzysztof Pałczyński, and Damian Ledziński. 2021. Ecg signal classification using deep learning techniques based on the ptb-xl dataset. *Entropy*, 23(9):1121.
- Yonghao Song, Xueyu Jia, Lie Yang, and Longhan Xie. 2021. Transformer-based spatial-temporal feature learning for ecg decoding. *ArXiv*, abs/2106.11170.

- Ethan H. Steinberg, Kenneth Jung, Jason A. Fries, Conor K. Corbin, Stephen R. Pfohl, and Nigam Haresh Shah. 2021. Language models are an effective representation learning technique for electronic health record data. *Journal of biomedical informatics*, page 103637.
- Nils Strodthoff, Patrick Wagner, Tobias Schaeffter, and Wojciech Samek. 2021. Deep learning for ecg analysis: Benchmarks and insights from ptb-xl. *IEEE Journal of Biomedical and Health Informatics*, 25:1519–1528.
- Allison C. Tam et al. 2022. Semantic exploration from language abstractions and pretrained representations. *ArXiv*, abs/2204.05080.
- Kaisa Tiippana. 2014. [What is the mcgurk effect?](#) *Frontiers in Psychology*, 5.
- Atousa Torabi, Niket Tandon, and Leonid Sigal. 2016. Learning language-visual embedding for movie understanding with natural-language. *ArXiv*, abs/1609.08124.
- Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.
- Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio', and Yoshua Bengio. 2018. Graph attention networks. *ArXiv*, abs/1710.10903.
- Patrick Wagner, Nils Strodthoff, R. Bousseljot, D. Kreiseler, F. Lunze, W. Samek, and T. Schaeffter. 2020. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific Data*, 7.
- Qinxin Wang, Haochen Tan, Sheng Shen, Michael W. Mahoney, and Zhewei Yao. 2020. An effective framework for weakly-supervised phrase grounding. *ArXiv*, abs/2010.05379.
- Kuba Weimann and Tim O. F. Conrad. 2021. Transfer learning for ecg classification. *Scientific Reports*, 11.
- Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. 2019. Fine-grained action retrieval through multiple parts-of-speech embeddings. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 450–459.
- Hao Wu et al. 2019. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. *CVPR*, pages 6602–6611.
- Chen Xie, Lucas McCullum, Alistair Johnson, Tom Pollard, Brian Gow, and Benjamin Moody. 2022. Waveform database software package (wfdb) for python (version 4.0.0). *PhysioNet*.
- Genshen Yan, Shen Liang, Yanchun Zhang, and Fan Liu. 2019. Fusing transformer model with temporal features for ecg heartbeat classification. *BIBM*, pages 898–905.
- Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. 2021. Taco: Token-aware cascade contrastive learning for video-text alignment. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11542–11552.
- Xi Yang, Nima M. Pournejatian, Hoo-Chang Shin, Kaleb E. Smith, Christopher Parisien, Colin B. Compas, Cheryl Martin, Mona G. Flores, Ying Zhang, Tanja Magoc, Christopher A. Harle, Gloria P. Lipori, Duane A. Mitchell, William R. Hogan, Elizabeth A. Shenkman, Jiang Bian, and Yonghui Wu. 2022. Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records. *ArXiv*, abs/2203.03540.
- Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2018. Exploring visual relationship for image captioning. In *ECCV*.
- Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. 2017. End-to-end concept word detection for video captioning, retrieval, and question answering. *CVPR*, pages 3261–3269.
- B.P. Yuhas, M.H. Goldstein, and T.J. Sejnowski. 1989. [Integration of acoustic and visual speech signals using neural networks](#). *IEEE Communications Magazine*, 27:65–71.
- Rowan Zellers, Ari Holtzman, Matthew E. Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. 2021. Piglet: Language grounding through neuro-symbolic interaction in a 3d world. In *ACL*.
- Andy Zeng, Adrian S. Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael S. Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Peter R. Florence. 2022. Socratic models: Composing zero-shot multimodal reasoning with language. *ArXiv*, abs/2204.00598.
- Bowen Zhang, Hexiang Hu, and Fei Sha. 2018. Cross-modal and hierarchical modeling of video and text. In *ECCV*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.
- Bingui Zhou, Guanghua Yang, Zheng Shi, and Shao-dan Ma. 2021. Natural language processing for smart healthcare. *ArXiv*, abs/2110.15803.
- Jiacheng Zhu, Jieli Qiu, Zhuolin Yang, Douglas Weber, Michael A. Rosenberg, Emerson Liu, Bo Li, and Ding Zhao. 2022. Geoecg: Data augmentation via wasserstein geodesic perturbation for robust electrocardiogram prediction. *ArXiv*, abs/2208.01220.
- Sandra Śmigiel, Krzysztof Pałczyński, and Damian Ledziński. 2021. Ecg signal classification using deep learning techniques based on the ptb-xl dataset. *Entropy*, 23.

A Experiment Parameters

We provide the experimental parameters of the models in the paper in Table 11 and Table 12.

B More Related Work

Cardiovascular Disease in Current Practice

Patients presenting with chest pain to the emergency department (ED) constitute a diagnostic and logistic challenge as chest pain can be caused by an extensive variety of disorders (Amsterdam et al., 2010). Diagnostic tests and decision algorithms play a critical role in speeding up the appropriate triage of chest pain patients in the ED, facilitating further (often more invasive) testing if warranted, and preventing unnecessary hospitalization of patients with non-critical disorders. In current practice, about half of the patients presenting with chest pain can be discharged from the ED, and only 5.5 percent of all ED visits lead to serious diagnoses (Hsia et al., 2016). However, research suggests the diagnosis of chest pain in the ED now costs an estimated \$10 to \$12 billion per year in the U.S. So a automatic cardiovascular disease diagnosis system is essential to provide cost-efficient patient care.

Deep learning in ECG Deep learning approaches have been rapidly adopted across a wide range of fields due to their accuracy and flexibility but require large labeled training sets. With the development in machine learning, many models have been applied to ECG disease detection (Kiranyaz et al., 2015; Nonaka and Seita, 2021; Khurshid et al., 2021; Raghunath et al., 2021; Giudicessi et al., 2021; Strodtzoff et al., 2021; Qiu et al., 2022b; Zhu et al., 2022). Al-Zaiti et al. (2020) predicted acute myocardial ischemia in patients with chest pain with a fusion voting method. Acharya et al. (2017); Moody and Mark (2001) proposed a nine-layer deep convolutional neural network (CNN) to classify heartbeats in the MIT-BIH Arrhythmia database. Shanmugam et al. (2019) estimate a patient’s risk of cardiovascular death after an acute coronary syndrome by a multiple instance learning framework. Recently, Śmigiel et al. (2021) proposed models based on SincNet (Ravanelli and Bengio, 2018) and used entropy-based features for cardiovascular diseases classification. The transformer model has also recently been adopted in several ECG applications, i.e., arrhythmia classification, abnormalities detection, stress detection, etc (Yan et al., 2019; Che et al., 2021; Natarajan

et al., 2020; Behinaein et al., 2021; Song et al., 2021; Weimann and Conrad, 2021).

Multimodal Learning Formalized multimodal learning research dates back to 1989, when Yuhua et al. (1989) conducted an experiment that built off the McGurk Effect for audio-visual speech recognition using neural networks (Tiippana, 2014; McGurk and MacDonald, 1976). Aligning representations from different modalities is an important step in multimodal learning. With the recent advancement in computer vision and natural language processing, multimodal learning, which aims to explore the explicit relationship between vision and language, has drawn significant attention (Wang et al., 2020). There are many methods proposed for exploring the multimodal alignment objective. Torabi et al. (2016); Yu et al. (2017) adopted attention mechanisms, Dong et al. (2021); Qiu et al. (2022a,d,c) composed pairwise joint representation, Chen et al. (2020); Wray et al. (2019); Zhang et al. (2018) learned fine-grained or hierarchical alignment, Lee et al. (2018); Wu et al. (2019) decomposed the images and texts into sub-tokens, Velickovic et al. (2018); Yao et al. (2018) adopted graph attention for reasoning, and Yang et al. (2021) applied contrastive learning algorithms for video-text alignment.

Multimodal Learning in Healthcare Applications

Many previous works have explored multimodal learning to boost performance in clinical healthcare applications, i.e., affective computing for depression disease detection and so on (Liu et al., 2021; Qiu et al., 2018a; Liu et al., 2019a; Qiu and Zhao, 2018; Qiu et al., 2018b, 2019; Han et al., 2022). Liu et al. (2021); Qiu et al. (2018a); Liu et al. (2019a); Qiu and Zhao (2018); Qiu et al. (2018b) explored the inner correlation between different modalities. Bao et al. (2019) investigated the demographics, showing that the subject’s individual characteristics can also be involved in robustness and personalized design. Qiu et al. (2019) investigated the relationship between computational vision models and computational neuroscience. Holenstein et al. (2021); Han et al. (2022) explored the connectivity between natural language and EEG signals.

C Preprocessing

The raw ECG signals are first processed by the WFDB library (Xie et al., 2022) and Fast Fourier

Table 11: Experiment parameters (best ones marked in bold).

Task	Batch Size	Encoder Layers	Att. Heads	Dropout	Epochs	Warmup Steps
Text Generation	[8, 16 , 32, 64]	[1, 2, 3, 4, 5]	[1, 2, 3, 4, 5]	[0.1, 0.2, 0.3]	[10, 20, 50 , 100, 200]	[1000, 2000]
Disease Detection	[8, 16 , 32, 64]	[1, 2, 3, 4, 5]	[1, 2, 3, 4, 5]	[0.1, 0.2, 0.3]	[10, 20, 50 , 100, 200]	[1000, 2000]

Table 12: Baseline parameters (best ones marked in bold).

Models	Batch Size	Layers	In Channel Size	Kernel Sizes	Dropout	Epochs	Warmup Steps
MLP	[8, 16 , 32, 64]	[2, 3, 4]	[128 , 256, 512, 1024]	[1,3]	[0.1, 0.2, 0.3]	[10, 20, 50 , 100, 200]	[1000, 2000]
LSTM	[8, 16 , 32, 64]	[1, 2, 3, 4]	[128, 256 , 512, 1024]	[1,3]	[0.1, 0.2, 0.3]	[10, 20, 50 , 100, 200]	[1000, 2000]
Resnet	[8, 16 , 32, 64]	[1, 2, 3, 4]	[128, 256 , 512, 1024]	[1,3]	[0.1, 0.2, 0.3]	[10, 20, 50 , 100, 200]	[1000, 2000]
Transformer	[8, 16 , 32, 64]	[1, 2, 3, 4, 5]	[128, 256 , 512, 1024]	[1,3]	[0.1, 0.2, 0.3]	[10, 20, 50 , 100, 200]	[1000, 2000]

transform (FFT) to process the time series data into the spectrum, which is shown in Fig. 2. Then we perform n-points window filtering to filter the noise within the original ECG signals and adopt notch processing to filter power frequency interference (noise frequency: 50Hz, quality factor: 30). The ECG signals are segmented by dividing the 10-second ECG signals into individual ECG beats. We first detect the R peaks of each signal by ECG detectors (Porr et al., 2022), and then slice the signal at a fixed-sized interval on both sides of the R peaks to obtain individual beats. Examples of the filtered ECG signal results after n-points window filtering, notch processing, R peak detection, and segmented ECG beats are shown in Figures. 3,4,5.

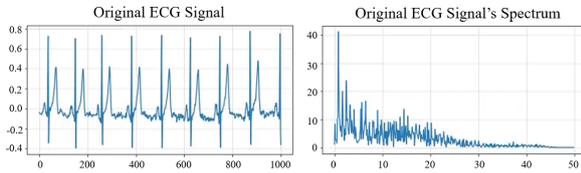


Figure 2: ECG data in the format of time series and spectrum.

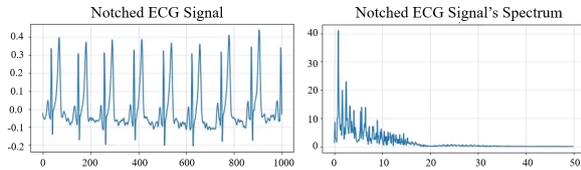


Figure 3: Filtered ECG data in the format of time series and spectrum.

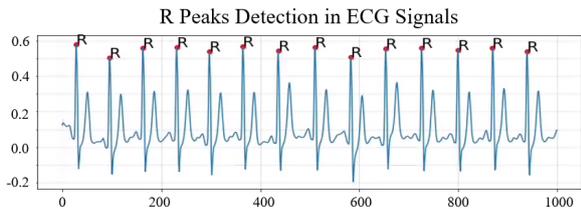


Figure 4: Detecting R peaks in the ECG signals.

Table 13: ECG statistical features in the frequency domain.

Feature Symbol	Formula
Z_1	$\frac{1}{N} \sum_{k=1}^N F(k)$
Z_2	$\frac{1}{N-1} \sum_{k=1}^N (F(k) - Z_1)^2$
Z_3	$-1 \times \sum_{k=1}^N \left(\frac{F(k)}{Z_1 N} \log_2 \frac{F(k)}{Z_1 N} \right)$
Z_4	$\frac{1}{N} \sum_{k=1}^N (F(k))^2$
Z_5	$\frac{1}{N} \sum_{k=1}^N \left(\frac{F(k) - Z_1}{\sqrt{Z_2}} \right)^3$
Z_6	$\frac{1}{N} \sum_{k=1}^N \left(\frac{F(k) - Z_1}{\sqrt{Z_2}} \right)^4$
Z_7	$\frac{\sum_{k=1}^N (f(k) - F(k))}{\sum_{k=1}^N F(k)}$
Z_8	$\sqrt{\frac{\sum_{k=1}^N [(f(k) - Z_6)^2 F(k)]}{\sum_{k=1}^N F(k)}}$
Z_9	$\frac{\sum_{k=1}^N [(f(k) - F(k))^3 F(k)]}{\sum_{k=1}^N F(k)}$
Z_{10}	$\frac{\sum_{k=1}^N [(f(k) - F(k))^4 F(k)]}{\sum_{k=1}^N F(k)}$

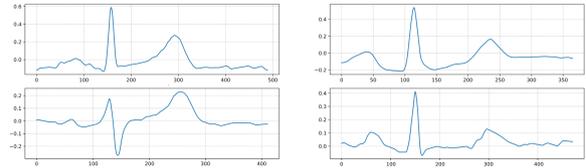


Figure 5: Extracted ECG beats divided by R peaks.

Feature Extraction Instead of directly using the time-series signals, we extract time domain and frequency domain features to better represent ECG signals. The time-domain features include: maximum, minimum, range, mean, median, mode, standard deviation, root mean square, mean square, k-order moment and skewness, kurtosis, kurtosis factor, waveform factor, pulse factor, and margin factor. The frequency-domain features include: FFT mean, FFT variance, FFT entropy, FFT energy, FFT skew, FFT kurt, FFT shape mean, FFT shape std, FFT shape skew, FFT kurt. The function of each component is shown in Table 13. An analysis of the statistics of the processed ECG data can also be found in Table 1.

D Transformer Encoders

The input for the Transformer is the ECG signal. First, we feed out the input into an embedding layer, which is a learned vector representation of each ECG feature by mapping each ECG feature to a vector with continuous values. Then we inject positional information into the embeddings by:

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \end{aligned} \quad (2)$$

The attention model contains two sub-modules, a multi-headed attention model and a fully connected network. The multi-headed attention computes the attention weights for the input and produces an output vector with encoded information on how each feature should attend to all other features in the sequence. There are residual connections around each of the two sub-layers followed by a layer normalization, where the residual connection means adding the multi-headed attention output vector to the original positional input embedding, which helps the network train by allowing gradients to flow through the networks directly.

In our model, our attention model contains N same layers, and each layer contains two sub-layers, which are a multi-head self-attention model and a fully connected feed-forward network. Residual connection and normalization are added in each sub-layer. So the output of the sub-layer can be expressed as: $\text{Output} = \text{LayerNorm}(x + (\text{SubLayer}(x)))$ For the Multi-head self-attention module, the attention can be expressed as: $\text{attention} = \text{Attention}(Q, K, V)$, where multi-head attention uses h different linear transformations to project query, key, and value, which are Q , K , and V , respectively, and finally concatenate different attention results:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (3)$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (4)$$

where the projections are parameter matrices:

$$\begin{aligned} W_i^Q &\in \mathbb{R}^{d_{\text{model}} \times d_k}, & W_i^K &\in \mathbb{R}^{d_{\text{model}} \times d_k} \\ W_i^V &\in \mathbb{R}^{d_{\text{model}} \times d_v}, & W_i^O &\in \mathbb{R}^{h d_v \times d_{\text{model}}} \end{aligned} \quad (5)$$

where the computation of attention adopted scaled dot-product:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (6)$$

For the output, we use a 1D convolutional layer and softmax layer to calculate the final output.

Practical Takes on Federated Learning with Pretrained Language Models

Ankur Agarwal Mehdi Rezagholizadeh Prasanna Parthasarathi

Huawei Noah’s Ark Lab, Montréal

{ankur.agarwal1,mehdi.rezagholizadeh,prasanna.parthasarathi}@huawei.com

Abstract

Real-world applications of language models entail data privacy constraints when learning from diverse data domains. Federated learning with pretrained language models for language tasks has been gaining attention lately but there are definite confounders that warrants a careful study. Specifically, understanding the limits of federated NLP applications through varying the effects of different aspects (such as data heterogeneity, the trade-off between training time and performance, the effect of different data, and client distributions and sensitivity of the shared model to learning local distributions) is necessary to evaluate whether language models indeed learn to generalize by adapting to the different domains. Towards that, we elaborate different hypotheses over the components in federated NLP architectures and study them in detail with relevant experiments over three tasks: Stanford Sentiment Treebank-2, OntoNotes-5.0 and GigaWord. The experiments with different Transformer inductive biases on the variety of tasks provide a glimpse at the understanding of federated learning at NLP tasks. Specifically, the analysis suggests that regularization due to the ensembling effect may be masquerading as domain adaptation of federated learning in NLP with pre-trained language models.

1 Introduction

The success of large pretrained language models (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019; Lewis et al., 2019) have demonstrated their applicability in consumer-based natural language processing (NLP) applications (Otter et al., 2020). While there are such massive datasets (Kiela et al., 2021; Wang et al., 2021), making models trained on these datasets to reflect the data diversity is an important challenge towards building equitable NLP systems. Hence, treating the distribution of data over the users as non-IID (McMahan and Ramage, 2017; Xu et al., 2018; Liu and Mazumder, 2021) to

better emphasize the preferences of users as personalization gets naturally extended to the consumer NLP applications.

But, recent studies highlight that pretrained language models (PLMs) (Devlin et al., 2019) tend to get their predictions skewed by the frequency effects of tokens in the data distribution (Wei et al., 2021), this is concerning from a privacy and personalization standpoint. Han and Eisenstein (2019); Ramponi and Plank (2020); Carlini et al. (2022) show that neural language models (even the large pretrained architectures) have challenges in adapting to different data distributions on generative and classification tasks alike.

Federated learning (FL) (McMahan and Ramage, 2017; Konečný et al., 2016) has been gaining popularity in machine learning as a practical way to mitigate domain adaptation with the promise of data privacy. FL as a learning paradigm focuses on learning a shared model through training data distributed over several clients. Such approaches have only recently begun to focus on NLP applications (Mammen, 2021; Lin et al., 2021). Lin et al. (2021) suggest that the success of federated algorithms can be improved through adapting over the client distribution that improves the generalization performance across the client distributions.

However, the opaqueness of the pretraining routine — primarily, quantifying what, and how much of that a language model has learnt from the pre-training corpora (Zhu et al., 2015; Devlin et al., 2019; Gao et al., 2020) cast a shadow on evaluating the effectiveness of these architectures in learning from diverse domains. Understanding the roles of different inductive biases not limited to the architecture, loss functions and data distribution becomes imperative to carefully look at claims of “domain adaptation” (Kouw and Loog, 2018). The representation of texts guided by syntax and semantic elements makes generalization to non-IID distributions in NLP more challenging if not the same when

compared to domains such as computer vision (Liu et al., 2020; Luo et al., 2021; Zhuang et al., 2021; Yang et al., 2021). Then, it becomes imperative to understand the role of different constituents in the federated learning in NLP setup to take steps in the right direction. In that regard, we investigate four major hypotheses detailing the confounding variables in federated NLP systems: (1) role of pretrained weights in FL’s domain adaptation, (2) distribution of clients, (3) Data homogeneity, and (4) robustness of personalizing to local distribution. Although the primary focus of the paper is to generate discussions along these questions, we support the discussions with relevant experiments on 3 different NLP tasks on 2 large Transformer architectures.

2 Background

Federated Learning Federated learning assumes the set up of K clients $\{C_k\}_{k=1}^K$ with different data distributions and a single server model S . Each client model, θ_{C_k} , is initialized by the server model θ_S a dedicated copy of the server model and then updated locally on the k^{th} client data distribution using an optimizer Opt_{C_k} . This distributed learning is repeated iteratively over R rounds. At each round r , the client models, $\theta_{C_k}^r$, are initialized with the aggregated weights of all the client models from the previous round ($r-1$), referred to as the central server model, θ_S^r . The rounds end with accumulation and aggregation of gradients from all the client models to update the server model with optimizer Opt_S and continued until convergence on an unseen set (D_{test}). FedOpt, a popular federated learning algorithm is shown in Algorithm 1.

Algorithm 1 FedOpt Algorithm (Asad et al., 2020)

Input: $\theta_S^0, \text{Opt}_{C_k}, \text{Opt}_S$
for $r = 1$ **to** R **do**
 for $k = 1$ **to** K **in parallel do**
 $\theta_{C_k}^r \leftarrow \theta_S^{r-1}$
 for $e = 1$ **to** E **do**
 $g_{C_k}^r \leftarrow \nabla_{\theta}(\theta_{C_k}^r | D_k)$
 $\theta_{C_k}^r \leftarrow \text{Opt}_{C_k}(\theta_{C_k}^r, g_{C_k}^r)$
 end for
 $\Delta_k^r \leftarrow \theta_{C_k}^r - \theta_S^{r-1}$
 end for
 $\Delta^r \leftarrow \frac{1}{K} \sum_{k=0}^K \Delta_k^r$
 $\theta_S^r \leftarrow \text{Opt}_S(\theta_S^{r-1}, \Delta^r)$
end for

The Performance Gap Like in (Lin et al., 2021), we compare the performance of federated server, θ_S , with θ_{central} over a common unseen set, D_{test} as in Equation 1,

$$\Delta \text{Perf} = \text{Perf}_S - \text{Perf}_{\text{central}} \quad (1)$$

where θ_{central} is trained over $\{D_k\}_{k=1}^K$ until convergence. Accuracy, F1, or Rouge score (Lin, 2004) can be used for measuring Perf.

Generalization Personalization trade-off Let the best server generalization performance is measured over D_{test} be $\mathcal{P}_{\text{server}}^*$. The generalization performance of the client model trained in r^* is measured on D_{test} be $\mathcal{P}_{\text{client}_k}^*$. The performance of the client model on D_k in round r^* measured be $\tilde{\mathcal{P}}_{\text{client}_k}$, which acts as a proxy to the personalization on D_k . Then, we measure the difference in the test loss for every client, k , between $\theta_{C_k}^*$ and θ_S^{*-1} as ΔP .

$$\Delta \mathcal{P}_k(\mathbf{x}_i) = \mathcal{P}_{\text{client}_k}^*(\mathbf{x}_i) - \mathcal{P}_{\text{server}}^{*-1}(\mathbf{x}_i) \quad (2)$$

For every $\mathbf{x}_i \in D_{\text{test}}$, the correlation between $\Delta \mathcal{P}_k(\mathbf{x}_i)$ to that of $\tilde{\mathcal{P}}_{\text{client}_k}$ measures the mutual cost of personalization to D_k on the generalization performance on D_{test} . Also, we measure the average the empirical risk of $\theta_{C_k}^*$ over D_k as $\tilde{\mathcal{P}}_{\text{client}_k}$:

$$\tilde{\mathcal{P}}_{\text{client}_k} = \frac{1}{|D_k|} \sum_{j=0}^{|D_k|} \mathcal{P}_{\text{client}_k}(\mathbf{x}_j) \quad (3)$$

We now define the trade-off metric as the slope between $\Delta \mathcal{P}_k(\mathbf{x}_i)$ and $\tilde{\mathcal{P}}_{\text{client}_k}$ over all $\mathbf{x} \sim D_{\text{test}}$ and K . $m_{\Delta P}$ measures the unit increase in generalization performance for adapting to D_k . Or $m_{\Delta P}$ estimates the cost of personalizing over the client distribution. Consequently, we make the interpretations for the metric $m_{\Delta P}$ — (a) positive slope (\nearrow) indicates that learning on local distribution aids in better generalization, (b) negative slope (\searrow) indicates that generalization inhibits the learning from local distributions, or (c) neutral (\longrightarrow) shows that the model is unaffected by learning.

3 Related Work

Multi-Domain Learning Realtime applications of most tasks have shown diverse distribution of datapoints requiring domain adaptation strategies (Daumé III, 2009; Dredze and Crammer, 2008). The effect of such domain shift in NLP has been

a topic of study for a while (Blitzer et al., 2006; Quiñonero-Candela et al., 2008; Blitzer, 2008; Ben-David et al., 2010; Cui and Bollegala, 2019). The general topic of domain adaptation in NLP shares similarity with the topics of continual learning (Sun et al., 2019), transfer learning (Devlin et al., 2019; Radford et al., 2019), multi-task learning (Collobert and Weston, 2008), and federated learning (Lin et al., 2021). Federated learning however, is different from the other paradigms since it emphasizes on the notion of preserving privacy of different local data distributions. To that, sophisticated approaches to aggregate the gradients to transfer learning from clients to the shared model (e.g. FedProx (Li et al., 2020), FedAvg (McMahan et al., 2017a), and FedOpt (Asad et al., 2020)) have showcased improvements in the generalization of the shared parameters. On the other hand, due to the many interactions of the clients with the server, communication overhead is an important aspect, and FedOpt (Asad et al., 2020) has been shown to better address it over existing federated algorithms.

Overview of Federated Learning Federated learning (McMahan and Ramage, 2017; Mammen, 2021) addresses the challenge of learning from private data spanning over multiple clients. Although the evaluation of such architectures prioritizes the generalization of the shared model, Mendieta et al. (2022) highlight that learning from the local distributions is critical towards that. The key to such efficient learning in federated architectures has been shaped by homogeneous and heterogeneous data or model (Li and Wang, 2019) distribution in clients (model architectures across clients have similar or different parameters). Further, the emphasis on privacy of client data has also been mitigated through the recent progress in knowledge distillation. However, systematic studies (Kairouz et al., 2021; Li et al., 2021) over federated architectures have identified potential biases due to unbalanced data of clients or diversity in the label distribution among others. Chen et al. (2018) propose a meta learning approach for federated learning that improves personalizing to the non-IID client distributions. Also, constraints on data privacy makes it difficult to import approaches (Kirkpatrick et al., 2017; Rolnick et al., 2019) that avoid catastrophic forgetting of distributions in continual learning tasks.

Federated Learning for language tasks Distributed training on language tasks with federated

learning has been gaining some attention. McMahan et al. (2017b) trained a differentially private language model over non-IID data distributions while Ge et al. (2020) trained a recurrent + convolutional architecture for medical named entity recognition task. Recently, Lin et al. (2021) proposed a framework that enables using modern pretrained language models on different language understanding tasks. Lin et al. (2021) discuss and hypothesizes a gap between the performance of Transformer architectures between the federated and centralized setting with data heterogeneity. Dupuy et al. (2022) analyze the effect of having clients with different amounts of data gathered from Alexa devices and suggest that non-uniform selection of devices improves the performance of the shared model.

In this work, we attempt to investigate different possible confounders for domain adaptation claims of federated systems in NLP and elaborately analyze them in the language tasks of classification, sequence tagging and sequence generation tasks that are popularly used with Transformer architectures.

4 Experiments

Models and Tasks We experiment with a focus on the *pretrain-finetune* setup that is popular with Transformer architectures on many language tasks. Of the many, we pick three tasks—Stanford Sentiment Treebank 2 (SST-2) (Socher et al., 2013), OntoNotes (v5.0) (Weischedel et al., 2013) and Gigaword (Graff et al., 2003) that fall into the broad categories of text classification, sequence tagging, text-generation respectively. The data splits are as used in (Lin et al., 2021), please refer §B for details. As for the models¹, we use BART-Base (Lewis et al., 2019) for text generation and DistilBERT (Sanh et al., 2019) for the other two tasks. In the experiments we use the models *with* (—) and *without* (----) pretrained weights²³.

Centralized Training We use batch-wise gradient descent with AdamW (Loshchilov and Hutter, 2017) as the optimizer along with a linear learning rate scheduler. We use cross-entropy for model selection across the tasks. Also, in our analysis, we

¹We use the transformer weights shared in huggingface: ‘distilbert-base-uncased’ and ‘facebook/bart-base’.

²Across the experiments the line style and the colour is used to denote the corresponding model performances.

³The without pretrained weights setting trains the models from scratch.

Dataset	Model	Metric	Pretrained	Cent.	Fed.	$\Delta(\text{Perf})$	$\Delta(\text{Rel.}\%)$
SST2	DistilBERT	Accuracy	✓	89.0 \pm 0.8	87.8 \pm 0.4	1.3	–
			✗	69.2 \pm 3.2	67.8 \pm 1.0	1.4	7.7 ▲
OntoNotes	DistilBERT	F1	✓	85.9 \pm 0.1	84.4 \pm 0.1	1.5	–
			✗	65.1 \pm 0.3	55.3 \pm 0.3	9.8	550 ▲
Gigaword	BART	Rouge1	✓	34.6 \pm 0.7	32.5 \pm 0.2	2.1	–
			✗	6.1 \pm 0.5	2.9 \pm 0.6	3.2	50 ▲

Table 1: Comparison between the $\Delta(\text{Perf})$ of federated and corresponding centralized set up when using (✓) and *not* using (✗) pretrained transformer weights. Across the 3 tasks it can be seen that the gap increases when not using pretrained weights (▲) suggesting that the pretrained weights of transformer are possibly doing the heavy lifting in domain adaptation of federated learning in language tasks.

use cross-entropy of samples in the test set to evaluate the relative performance of models compared. The complete results of the experiments are in §F.

Federated Training For the federated experiments, we partition the training dataset for the clients and train them using the FedOpt algorithm (Asad et al., 2020) to estimate the server parameter updates. Further, we use AdamW with a linear learning rate scheduler to estimate the gradients in our experiments. The round with the best server test loss is selected as the best round (For the complete results please refer to §F; for their run-time refer to §E).

Evaluation Metrics The metric of evaluation (*Perf*) is accuracy for sentence classification, F1-span for sequence tagging and ROUGE1 score for text summarization.

4.1 Motivation

Federated NLP considers two powerful learning paradigms— Federated algorithms aggregate the gradient updates over clients trained with non-identical data distributions while maintaining privacy, and PLMs trained over large corpora with a generic objective that gives a better downstream performance. Stickland and Murray (2019) and Peng et al. (2020) show that PLMs are successful in tasks that require domain adaptation. The motivation primarily relies on verifying if federated algorithms and PLMs share a synergy in the extreme domain adaptation scenario.

To that, we first study the role of pretrained weights as a confounding variable in the federated setup. The null hypothesis being the federated learning not affected much by the pretrained weights should be supported with $\Delta(\text{Perf})$ remaining similar in both cases. But, in Table 1 across different tasks we see that $\Delta(\text{Perf})$ increases when

not using pretrained weights. This suggests that the pretrained weights may be supporting the performance of federated learning in NLP applications. The corollary to this observation could be that the learning in a federated setting may not be happening from adapting to the client distributions. This raises concerns on personalization that if at all the federated NLP setup with PLM learns anything from the client distribution.

4.2 Estimating the Confounding Variables

Pretraining and Federated Learning Towards understanding the essence of the pretrained weight’s semantic prior as a confounding role in the success of FL for NLP, we continue to control for it in the remainder of the experiments. Disentangling such observations is necessary to objectively analyze the federated algorithm for language tasks.

Contribution by Client size One major challenge in the realistic setting of federated learning for data-driven tasks is the data imbalances that naturally occur among the clients (Lin et al., 2021; Dupuy et al., 2022). The distribution of number of data samples that each client has creates two distinct classes of clients— *major* and *minor* players— whose updates may affect the parameters of the shared server model differently. Generalization aside, ensuring personalization to the local distribution of data in the clients also becomes necessary in different scenarios arisen from the diverse data distributions. While extreme distributions may provide a regularization effect due to the ensemble learning (Balaji et al., 2018; Kumar et al., 2020; Stanton et al., 2021), the objective being able to better generalize through adapting to different local distributions require careful consideration of the distribution of clients. Towards understanding the limits of learning under the influence of clients

of different sizes, we evaluate the role of minor clients by ablating clients smaller than a *threshold* (τ) number of samples.

Client Personalization and Server Generalization Personalization emphasizes the shared model’s capacity to be representative of all clients’ distributions alike. But, training on the local distribution may affect the generalized representation of the shared model similar to the catastrophic forgetting in continual learning (Kirkpatrick et al., 2017) or mode collapse in generative modeling (Salimans et al., 2016). Using PLMs due to their robust semantic representations could alleviate some of these challenges. More specifically, we formulate our questions as ablation experiments: (a) How are personalization and generalization related across different tasks and client distributions (b) Does removing updates from minor players affect this relation? We study these questions in experiments with the help of $m_{\Delta P}$ metric.

Client data partitioning distribution Generalization to unseen distribution, and how well the local client distributions are personalized over the federated learning rounds is also affected by the distribution of samples over the clients. Towards understanding the effect of the sample distribution on the personalization-generalization relation we compare the learning of the server model between non-uniform distribution that is closer to real world scenario, and a more controlled uniform distribution of samples among the clients to (a) evaluate the effect of generalization by the shared model and the personalization to local distribution by varying the number of samples per client uniformly over all the clients, and (b) we perform ablations on the updates by thresholding the clients on heterogeneous data distribution set up to understand the ideal scenarios in different language tasks.

4.3 Additional Setup

Dataset Partitioning Strategies For the study, we use data selection for the clients using two different strategies. Please refer to §C for the choice of hyper-parameters for the two methods.

Random partitioning samples data over $\{C\}_{k=1}^K$ by sampling from a Dirichlet distribution over the K clients with $\alpha = 0.1$ (Lin et al., 2021). For ablation on random distribution, we use hyperparameter τ that denotes the minimum

number of samples in C_i for its parameters to be aggregated.

Uniform partitioning distributes the data uniformly over $\{C\}_{i=1}^K$, which is controlled by a hyperparameter γ . Specifically, uniform distribution is constructed by sampling *at least* γ samples from each class for each client in the classification tasks. For the text summarization dataset, we cluster the SentenceBERT embedding (Reimers and Gurevych, 2019) and use KMeans++⁴ (Arthur and Vassilvitskii, 2006) with 8 as the number of clusters, and sample uniformly over them.

Distributional Similarity We use MAUVE score (Pillutla et al., 2021) to measure the similarity between different text distributions in the experiments. The score uses GPT-2 (Radford et al., 2019) estimating the distributional similarity. Also, we use `mauve_scaling_parameter` set to 20. The score ranges between 0 and 1, where 1 indicates significant overlap.

5 Results

Following our motivation in §4.1 and questions raised for the confounders to the federated system for NLP tasks in §4.2, we structure the results to our investigation in this section.

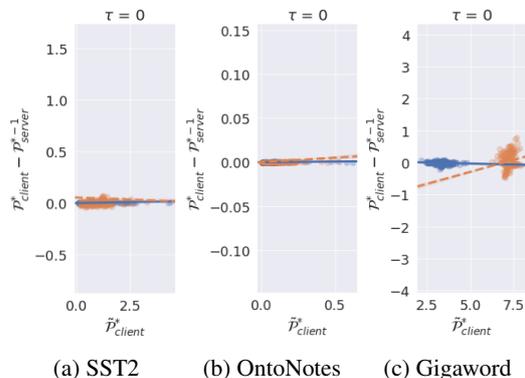


Figure 1: We observe the effect of domain shift by measuring the δ change in the loss over every sample in the test set ($\mathcal{P}_{client} - \mathcal{P}_{server}^{*-1}$) and drawing a correlation with the average loss over the train distribution in the clients ($\tilde{\mathcal{P}}_{client}$).

5.1 Effectiveness of Pretrained weights in adapting to client data distributions

An ideal model is expected to not discount the learning on the client distributions for better gener-

⁴We use the implementation in www.scikit-learn.org.

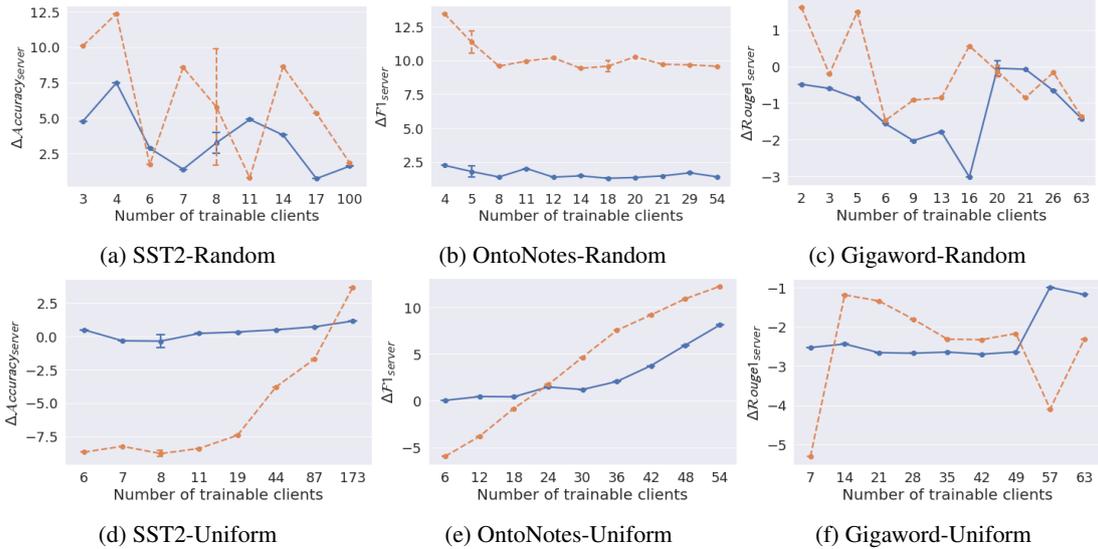


Figure 2: By varying the number of trainable clients in the random setting across the tasks, we measure the sensitivity of the shared model’s performance (Cent. – Fed.: ∇ indicates federated model performing better) to data removal as high as $\sim 55\%$ of training data (SST-2: $\sim 55\%$, OntoNotes: $\sim 33\%$ and Gigaword: $\sim 20\%$ with the smallest number of clients).

alization performance. To verify whether such discounting happens, we analyze the personalization-generalization trade-off with the $m_{\Delta P}$ metric. In Figure 1, across the datasets we observe that the correlation stayed more neutral than positive, suggesting that the pretrained model may not be learning much from the local distributions. We observe a relatively positive $m_{\Delta P}$ value when trained without the pretrained weights. This could be an anticipated behaviour, as the model relies on the information in the client distribution for generalization, unlike the pretrained weights that come with a semantic prior.

5.2 Client contributions to the metrics

We set up additional experiments to understand the contribution of clients in more detail. Particularly, we begin by studying our experiment of ablating updates from the clients that are below threshold (τ) on the *random* experiments. We hypothesize that as the threshold value is increased, the server model is restricted to learning from fewer client local distributions and the generalization performance should also decline as a result. On the contrary, we note in Figure 2 (top row) that the performance of the with pretrained model remained relatively unchanged across the datasets. However, the MAUVE scores estimated over the Ablated, Unablated and Test (Unseen) distributions of data in Table 2 suggest that the test distribution are not close to the ablated

Dataset	$U - A$	$U - T$	$A - T$
SST2	0.52 ± 0.06	0.48 ± 0.07	0.47 ± 0.07
OntoNotes	0.64 ± 0.06	0.53 ± 0.07	0.51 ± 0.06
Gigaword	0.58 ± 0.07	0.45 ± 0.07	0.45 ± 0.07

Table 2: The MAUVE score between the distribution of data in ablated (A), unablated (U) and unseen (T) splits with maximum value of τ in the random distribution setting. The values indicate the maximal distance between the distributions across the tasks.

or unablated clients’ distributions⁵. The performance of the pretrained model in the federated setup, still being little affected only suggests that pretrained models may not be learning from the local distributions that hurt the claims of personalization to these distributions.

If not adapting to the local distributions, we further investigate whether the pretrained models use the updates from client distributions as regularization. To that, we repeat the same experiments on the same datasets with the more controlled uniform distribution setting.

Do client sizes affect the gap To have a clearer picture of the client size affecting the learning contribution, we use the *uniform* distribution with γ controlling the data partitioning size uniformly over all the clients. In Figure 2 (bottom row), we observe a trend showing that the models’ performance

⁵Scores closer to 1 indicate significant overlap

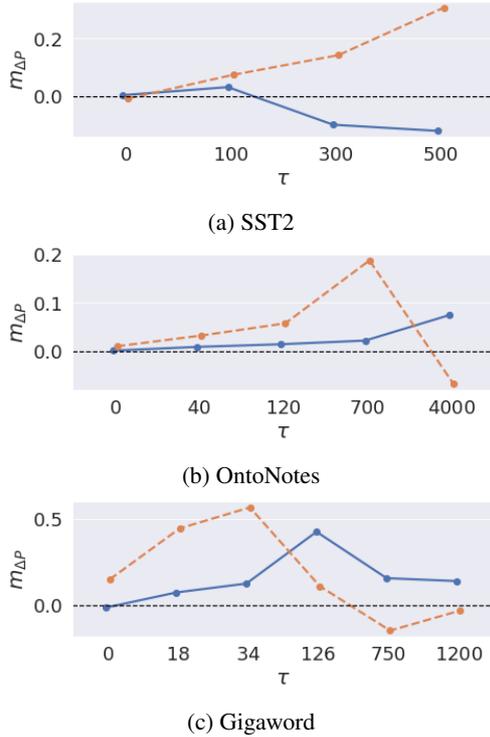


Figure 3: Impact of personalization of clients on generalization of server— $m_{\Delta P}$ values depict the impact in Random distribution strategy when ablating minor clients over different τ values.

(pretrained or not) decreases drastically as the number of clients increases and the client partitioning sizes progressively become smaller. Similar observation across the different tasks suggests that the model requires a quorum of samples to minimize the gap across (Equation 1) the different tasks in a federated setup.

Do client sizes affect the trade-off Again here, we adjust for τ in *random* setting, where the clients with less than τ number of samples are restricted from updating the server parameters. We measure $m_{\Delta P}$ values for the varying τ values in Figure 3. We observe that with updates primarily from more minor clients (lower τ value), the generalization is less affected by personalization. But, the gap being lower as shown in Figure 2 suggests that the noisy updates with fewer clients be acting as a regularizer for the server updates. Further, as the value of τ increases (Figure 3), the trade-off remains healthier until a certain value of τ and then it drops. This trend could be attributed to the fact that with a higher τ value the number of clients updating the server becomes lesser with more data points, which provides better generalization but personalization due to variance in the client distribution gets chal-

lenging.

To understand the impact of varying client distributions on the trade-off, we perform the same analysis with *uniform* distribution shown in Figure 4. The pretrained model’s generalization remains unaffected with more minor clients across SST-2 and GigaWord tasks supporting the alternate that updates from minor clients provide a regularization effect as we also see the gap to remain the same in Figure 2. On OntoNotes, while the $m_{\Delta P}$ value stays the same the gap widens as observed in Figure 2. The varying results do not provide conclusive evidence on whether the pretrained models can learn to adapt to different domains in such extreme settings. Answering this is non-trivial which requires a careful consideration of the pretraining datasets and characterization of domains based on—task, topic, syntax, style etc.

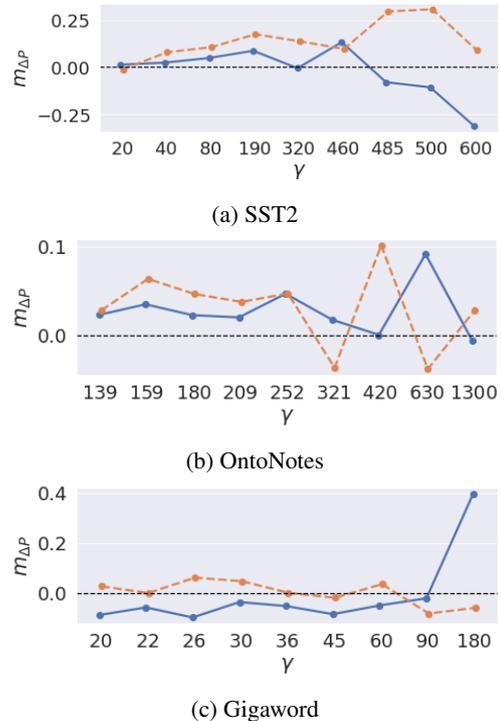


Figure 4: Impact of personalization of clients on generalization of server— $m_{\Delta P}$ values depicting the impact in Uniform distribution strategy with varying sizes of clients.

Time-Performance trade-off with varying client sizes The dropping of updates from the minor clients could also provide acceleration in the number of rounds, R , as federated learning has a communication overhead. We measure the performance of server with pretrained weights, Perf_S , over the different tasks and the number of rounds (R) taken

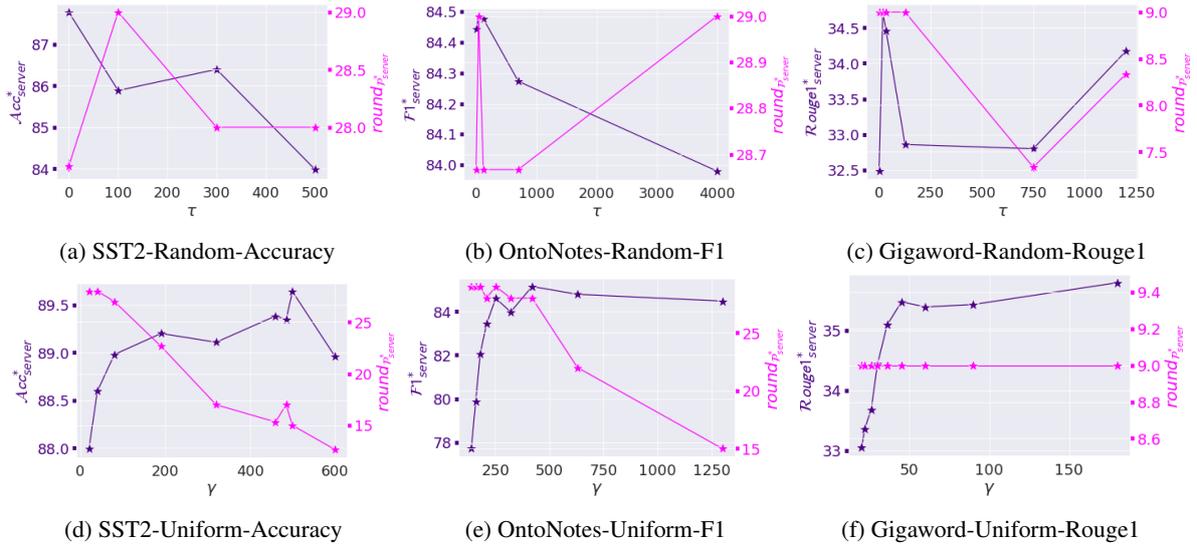


Figure 5: We compare R taken to converge when varying the number of clients in the uniform and in the random distribution settings with pretrained weights (✓). We also measure the corresponding Perf_S of the server model in the task.

by the set up to converge.

In Figure 5, we compare across the tasks with *random* distribution and the *uniform* distribution of samples with pretrained weights by varying the τ and γ respectively. In the random distribution experiments, as we are discarding parameter updates from clients we observe a *not-so-steep* drop in the Perf_S of the server model in SST-2 and OntoNotes. On the other hand, in Gigaword dataset experiments the impact of dropping the clients did not affect Perf_S . With respect to R , τ value being inversely proportional to the number of clients, we did not see a drastic acceleration to the number of rounds as lesser clients also increased the difficulty of the tasks.

In *uniform* setting, by varying the number of clients without the data loss, we make two observations: (1) The Perf_S is always better than when compared with *random* setting, (2) the performance saturates after a certain γ across the tasks, and (3) the number of rounds taken by the models to converge shows drastic decrease as the number of clients decreases. We hypothesize that with only major clients the gradient updates are stable to enable faster convergence. This contradicts with the observation in (Lin et al., 2021) that shows a wider gap in the performance when training pretrained transformer models in a federated set up, which we observe only when *not* using pretrained weights. Collectively, the results hint that the federated set up with PLMs suffer from personalizing

to the client distributions, and the generalization on tasks may be a regularization of the distributed set-up.

6 Conclusion

This work explores pertinent questions that require a closer look at evaluating PLMs in the federated setting. Through empirical observations, we find that in federated learning, where the emphasis is more on personalization while ensuring privacy there could be a risk of pretrained models overlooking the client distributions. We also evaluated the effects of varying the client distributions which suggested that the gap between centralized and federated performance to be reduced when the samples are uniformly distributed over the clients. While that is ideal, the random distribution too does not suffer significant performance loss with pretrained weights. However, the critical aspect of the questions stems from the need to investigate the pretraining routine in identifying the *right* domain adaptation challenges for pretrained models. The gap being minimized while the personalization taking a toll calls for a deeper inspection to explore the limits of domain adaptation in PLMs with an appropriate evaluation framework (datasets, and metrics) that controls for the *leak* in the pretraining corpus.

Acknowledgements

We thank Guojun Zhang, and Xi Chen from the federated learning team at Huawei Noah’s Ark Lab, Montréal for the many interesting discussions. We thank the anonymous reviewers for their insightful comments to our work. We also want to reproduce our results on Mindspore⁶ in future, which is a new deep learning computing framework.

Limitations

The study, though, considers sample tasks from the different language tasks the downstream tasks generally are smaller in the size, and not much diversity with respect to the task complexity is considered. Though there is motivation for using FedOpt for training, the claims could have been further supported by exploring other possible federated algorithms. The scale of the experiments however do not play in favour of such an exhaustive study. Although the distributional similarity is measured with MAUVE, other aspects of texts n -gram, topic modelling could be explored to understand the domain shifts. Further, the study does not consider language models with different other inductive biases. The different transformer models and the effect of their respective pretraining datasets and task remain unexplored for future work. In addition to the above, the behaviour of different federated algorithms in the hypotheses we frame would also become interesting cases to scale our work.

Broader Impact

The trend of fine tuning transformer models for downstream tasks as both time and cost-effective solution for improving performance in downstream tasks has been gaining enough popularity. With federated algorithms giving access to learning from more public data while tackling the privacy concerns, it becomes worthwhile to use pretrained language models for language applications. Thus, understanding the adjustments to this federated language task learning with pretrained transformers on the claims of personalization-generalization trade-off becomes necessary. Knowledge and role of variables like client sizes and their distribution on the federated performance help identifying better decisions on setting up an appropriate domain for learning in downstream NLP tasks.

⁶<https://www.mindspore.cn/>

References

- David Arthur and Sergei Vassilvitskii. 2006. k -means++: The advantages of careful seeding. Technical report, Stanford.
- Muhammad Asad, Ahmed Moustafa, and Takayuki Ito. 2020. Fedopt: Towards communication efficiency and privacy preservation in federated learning. *Applied Sciences*, 10(8):2864.
- Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. 2018. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31.
- Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79(1):151–175.
- John Blitzer. 2008. *Domain adaptation of natural language processing systems*. Ph.D. thesis, University of Pennsylvania.
- John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128.
- Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.
- Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.
- Xia Cui and Danushka Bollegala. 2019. Self-adaptation for unsupervised domain adaptation. *Proceedings-Natural Language Processing in a Deep Learning World*.
- Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

- Mark Dredze and Koby Crammer. 2008. Online methods for multi-domain learning and adaptation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 689–697.
- Christophe Dupuy, Tanya G Roosta, Leo Long, Clement Chung, Rahul Gupta, and Salman Avestimehr. 2022. Learnings from federated learning in the real world. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8767–8771. IEEE.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.
- Suyu Ge, Fangzhao Wu, Chuhan Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2020. Fedner: Privacy-preserving medical named entity recognition with federated learning. *arXiv preprint arXiv:2003.09288*.
- David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.
- Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. *arXiv preprint arXiv:1904.02817*.
- Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. 2021. Dynabench: Rethinking benchmarking in nlp. *arXiv preprint arXiv:2104.14337*.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.
- Wouter M Kouw and Marco Loog. 2018. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*.
- Ananya Kumar, Tengyu Ma, and Percy Liang. 2020. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pages 5468–5479. PMLR.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Daliang Li and Junpu Wang. 2019. Fedmd: Heterogeneous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*.
- Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2021. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*.
- Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450.
- Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. 2021. Fednlp: A research platform for federated learning in natural language processing. *arXiv preprint arXiv:2104.08815*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Bing Liu and Sahisnu Mazumder. 2021. Lifelong and continual learning dialogue systems: learning during conversation. *Proceedings of AAAI-2021*.
- Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. 2020. Fedvision: An online visual object detection platform powered by federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. 2021. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984.
- Priyanka Mary Mammen. 2021. Federated learning: Opportunities and challenges. *arXiv preprint arXiv:2101.05428*.

- Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017a. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.
- Brendan McMahan and Daniel Ramage. 2017. Federated learning: Collaborative machine learning without centralized training data. *Google Blog*.
- H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2017b. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*.
- Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. 2022. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8397–8406.
- Daniel W Otter, Julian R Medina, and Jugal K Kalita. 2020. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624.
- Yifan Peng, Qingyu Chen, and Zhiyong Lu. 2020. An empirical study of multi-task learning on bert for biomedical text mining. *arXiv preprint arXiv:2005.02799*.
- Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.
- Joaquin Quiñero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2008. *Dataset shift in machine learning*. Mit Press.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.
- Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey. *arXiv preprint arXiv:2006.00632*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32.
- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. *Recursive deep models for semantic compositionality over a sentiment treebank*. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.
- Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. 2021. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34:6906–6919.
- Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *International Conference on Machine Learning*, pages 5986–5995. PMLR.
- Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. *arXiv preprint arXiv:1909.03329*.
- Zijie J Wang, Dongjin Choi, Shenyu Xu, and Diyi Yang. 2021. Putting humans in the natural language processing loop: A survey. *arXiv preprint arXiv:2103.04044*.
- Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. *arXiv preprint arXiv:2109.07020*.
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. *OntoNotes Release 5.0*. Abacus Data Network.
- Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Lifelong domain word embedding via meta-learning. *arXiv preprint arXiv:1805.09991*.
- Dong Yang, Ziyue Xu, Wenqi Li, Andriy Myronenko, Holger R Roth, Stephanie Harmon, Sheng Xu, Baris Turkbey, Evrim Turkbey, Xiaosong Wang, et al. 2021. Federated semi-supervised learning for covid region segmentation in chest ct using multi-national data from china, italy, japan. *Medical image analysis*, 70:101992.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. 2021. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4912–4921.

A Reproducibility Checklist

A.1 For all reported experimental results:

1. *A clear description of the mathematical setting, algorithm, and/or model:* We define the details of our experimental setup in §4.
2. *Description of computing infrastructure used:* We use multiple servers equipped with 8 NVIDIA V100 (32 GB) GPUs and 72 cores CPU (754 GB) for running our experiments.
3. *The average runtime for each model or algorithm (e.g., training, inference, etc.), or estimated energy cost and number of parameters in each model:* The details of the runtime costs per experiment and the model have been reported in Table 8. The experiment runs have been tabulated in Table 9, Table 10 & Table 11.
4. *Corresponding validation performance for each reported test result:* Not applicable.
5. *Explanation of evaluation metrics used, with links to code:* This is specified with references in paragraph titled ‘Evaluation’ in §2.

A.2 For all experiments with hyperparameter search:

1. *The exact number of training and evaluation runs:* We run all centralized and random distribution on 3 seeds while the uniform distribution experiments are run on a single seed. The random distribution leads to different client size distributions while the uniform distribution has all clients of similar size.
2. *Bounds for each hyperparameter:* The tunable hyperparameters were batch size and learning rate in both centralized and federated training. For each dataset in both cases of fine-tuning and training from scratch, we find the best learning rate in the range [0.01, 0.000001] for centralized and federated training by tuning on the exponent scale. For federated training we find the best hyperparameters in random distribution setting which we continue to use in other variants of our experiments. For the batch size, we explore in the set 8, 16, 32, 64.
3. *Hyperparameter configurations for best-performing models:* The Table 4, Table 5, Table 6 & Table 7 records the best hyperparameters in use.

4. *Number of hyperparameter search trials:* The best hyperparameters are chosen over 3 seeds.
5. *The method of choosing hyperparameter values (e.g., uniform sampling, manual tuning, etc.) and the criterion used to select among them (e.g., accuracy)* The best test loss resulting combination of hyperparameters is selected. The grid search method is used.
6. *Summary statistics of the results (e.g., mean, variance, error bars, etc.)* The tabulated results show mean and standard deviation results while the line plots are created using median as an estimator. The plots involving test or train loss account the evaluations done on a sample level. The plots using *Perf* evaluations use set of experiment level values.

A.3 For all datasets used:

1. *Relevant details such as languages, and number of examples and label distributions:* The datasets used are SST2, OntoNotes and Gigaword which are all in the English language.
2. *Details of train/validation/test splits:* This can be found tabulated in Table 3.
3. *Explanation of any data that were excluded, and all pre-processing steps:* For the task of text classification we use the complete sentences as samples instead the parsed phrases.
4. *A zip file containing data or link to a downloadable version of the data:* The references to the datasets are provided in §4.
5. *For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control.* Not applicable.

B Dataset Splits

Dataset	Train set	Test set	Labels
SST2	6,920	1,821	2
OntoNotes	59,924	8,262	37
Gigaword	10,000	2000	<i>N.A.</i>

Table 3: Statistics for the 3 different dataset used.

C Client Data Distribution

During the gradient accumulation, we normally use uniform weightage. For sanity check if uniform weighting is the best choice, we made comparison of the random distribution with SST2 dataset using weighted aggregation where the client gradients are weighted to their size proportions. We did not see any advantage and hence continued using the uniform weightage. The comparison in performance can be seen in Table 9.

C.1 Random Distribution

Dataset	Clients	τ
SST2	3-100	{0, 100, 300, 500}
OntoNotes	4-54	{0, 40, 120, 700, 4000}
Gigaword	7-63	{0, 18, 34, 126, 750, 1200}

Table 4: Statistics for random distribution strategy experiments. The number of clients are effected by the ablation threshold for minor clients (τ).

C.2 Uniform Distribution

Dataset	Clients	γ
SST2	5-173	{600, 500, 485, 460, 320, 190, 80, 40, 20}
OntoNotes	6-54	{1300, 630, 420, 321, 252, 209, 180, 159, 139}
Gigaword	7-63	{180, 90, 60, 45, 36, 30, 26, 22, 20}

Table 5: Statistics for uniform distribution strategy experiments. The client count increases as the number of samples per label in a client (γ) decreases.

D Hyperparameters

The experiments on random distribution for all the datasets were carried out with 3 different seeds. However, for the uniform distribution we use only a single seed for OntoNotes and Gigaword datasets. Unlike the random distribution where the sampled client sizes keeps varying dramatically, the uniform distribution has all clients with almost the same number of data samples. Thus, we relax the need for repeating experiments with multiple seeds in the uniform distribution.

Dataset	Pretraining	Epochs	Batch size	L.R.
SST2	✓	10	8	1.00E-05
	✗	10	8	1.00E-05
OntoNotes	✓	5	8	2.00E-05
	✗	5	32	2.00E-04
Gigaword	✓	5	8	3.00E-05
	✗	5	8	3.00E-05

Table 6: The hyperparameters used for the centralized training experiments.

Dataset	Pretraining	Rounds	Batch size	L.R.
SST2	✓	30	64	1.00E-05
	✗	50	64	1.00E-05
SST2 (Weighted Aggregation)	✓	30	64	1.00E-04
	✗	50	64	1.00E-04
OntoNotes	✓	30	64	2.00E-05
	✗	50	64	2.00E-05
Gigaword	✓	10	8	3.00E-05
	✗	15	8	3.00E-05

Table 7: The hyperparameters used for the federated training experiments.

E Runtime of the experiments

Dataset	Model (parameters)	Experiment	Pretrained	GPUs	Runtime (Hrs)
SST2	distilbert-base-uncased (66.9M)	Centralized Training	✓	1	~0.5
			✗		~0.5
		Random Distribution	✓	8	5-6
			✗		8-10
		Uniform Distribution	✓	8	9-12
			✗		15-20
OntoNotes	distilbert-base-uncased (66.4M)	Centralized Training	✓	1	~1
			✗		~1
		Random Distribution	✓	8	4-5
			✗		6.5-7.5
		Uniform Distribution	✓	8	6-27.5
			✗		10-46
Gigaword	facebook/bart-base (139.4M)	Centralized Training	✓	1	~1
			✗		~1
		Random Distribution	✓	8	2-15
			✗		3-22.5
		Uniform Distribution	✓	8	3.5-17.5
			✗		5-26.5

Table 8: Time and resource costs per experiment run for the different datasets. The GPU refers to the NVIDIA V100 (32 GB) in a server having 8 of them.

F Master Results Tables

F.1 SST2

Centralized learning		
Pretraining	Test Accuracy (%)	Epochs
✓	89.02 ± 0.83	4.33 ± 2.87
✗	69.19 ± 3.2	5.33 ± 2.62
Random distribution ($\tau=0$, 100 clients) (Weighted Aggregation)		
Pretraining	Test Accuracy (%)	Rounds
✓	86.84 ± 0.1	30.0 ± 0.0
✗	52.99 ± 2.39	49.67 ± 0.47
Random distribution ($\tau=0$, 100 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	87.77 ± 0.4	28.67 ± 1.89
✗	67.78 ± 1.04	50.0 ± 0.0
Random distribution ($\tau=100$, 100 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	85.89 ± 1.08	30.0 ± 0.0
✗	64.34 ± 1.05	49.67 ± 0.47
Random distribution ($\tau=300$, 100 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	86.44 ± 0.39	30.0 ± 0.0
✗	62.84 ± 0.5	50.0 ± 0.0
Random distribution ($\tau=500$, 100 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	84.79 ± 1.42	30.0 ± 0.0
✗	61.14 ± 1.46	50.0 ± 0.0
Uniform distribution ($\gamma=20$, 173 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	88.05 ± 0.32	29.67 ± 0.47
✗	66.68 ± 1.17	50.0 ± 0.0
Uniform distribution ($\gamma=40$, 87 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	88.61 ± 0.23	30.0 ± 0.0
✗	72.29 ± 0.4	49.33 ± 0.47
Uniform distribution ($\gamma=80$, 44 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	89.07 ± 0.13	29.33 ± 0.94
✗	74.32 ± 0.64	48.67 ± 0.94
Uniform distribution ($\gamma=190$, 19 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	89.2 ± 0.2	23.67 ± 1.7
✗	78.0 ± 0.4	47.67 ± 1.7
Uniform distribution ($\gamma=320$, 11 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	89.11 ± 0.32	18.0 ± 1.41
✗	79.06 ± 0.42	41.0 ± 2.94
Uniform distribution ($\gamma=460$, 8 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	89.38 ± 0.09	16.33 ± 0.47
✗	79.7 ± 0.14	34.33 ± 1.7
Uniform distribution ($\gamma=485$, 8 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	89.35 ± 0.68	18.0 ± 2.94
✗	79.64 ± 1.05	32.33 ± 2.87
Uniform distribution ($\gamma=500$, 7 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	89.64 ± 0.13	16.0 ± 0.82
✗	79.75 ± 0.32	31.33 ± 1.25
Uniform distribution ($\gamma=600$, 6 clients)		
Pretraining	Test Accuracy (%)	Rounds
✓	88.96 ± 0.12	13.67 ± 0.94
✗	79.33 ± 0.39	28.33 ± 2.05

Table 9: Results of all experiments on SST2 dataset after model selection on the best server test loss.

F.2 OntoNotes

Centralized learning		
Pretraining	Test F1 (%)	Epochs
✓	85.93 ± 0.13	3.33 ± 1.7
✗	65.1 ± 0.29	1.0 ± 0.0

Random distribution ($\tau=0$, 54 clients)		
Pretraining	Test F1 (%)	Rounds
✓	84.44 ± 0.05	29.67 ± 0.47
✗	55.31 ± 0.27	49.67 ± 0.47

Random distribution ($\tau=40$, 54 clients)		
Pretraining	Test F1 (%)	Rounds
✓	84.49 ± 0.05	30.0 ± 0.0
✗	55.41 ± 0.26	50.0 ± 0.0

Random distribution ($\tau=120$, 54 clients)		
Pretraining	Test F1 (%)	Rounds
✓	84.48 ± 0.1	29.67 ± 0.47
✗	55.4 ± 0.16	50.0 ± 0.0

Random distribution ($\tau=700$, 54 clients)		
Pretraining	Test F1 (%)	Rounds
✓	84.27 ± 0.25	29.67 ± 0.47
✗	55.08 ± 0.18	49.67 ± 0.47

Random distribution ($\tau=4000$, 54 clients)		
Pretraining	Test F1 (%)	Rounds
✓	83.98 ± 0.32	30.0 ± 0.0
✗	53.07 ± 1.23	50.0 ± 0.0

Uniform distribution ($\gamma=139$, 54 clients)		
Pretraining	Test F1 (%)	Rounds
✓	77.7	30.0
✗	52.67	50.0

Uniform distribution ($\gamma=159$, 48 clients)		
Pretraining	Test F1 (%)	Rounds
✓	79.86	30.0
✗	54.0	50.0

Uniform distribution ($\gamma=180$, 42 clients)		
Pretraining	Test F1 (%)	Rounds
✓	82.05	30.0
✗	55.72	50.0

Uniform distribution ($\gamma=209$, 36 clients)		
Pretraining	Test F1 (%)	Rounds
✓	83.43	29.0
✗	57.38	50.0

Uniform distribution ($\gamma=252$, 30 clients)		
Pretraining	Test F1 (%)	Rounds
✓	84.61	30.0
✗	60.26	50.0

Uniform distribution ($\gamma=321$, 24 clients)		
Pretraining	Test F1 (%)	Rounds
✓	83.99	29.0
✗	63.2	50.0

Uniform distribution ($\gamma=420$, 18 clients)		
Pretraining	Test F1 (%)	Rounds
✓	85.16	29.0
✗	65.22	48.0

Uniform distribution ($\gamma=630$, 12 clients)		
Pretraining	Test F1 (%)	Rounds
✓	84.79	23.0
✗	67.61	45.0

Uniform distribution ($\gamma=1300$, 6 clients)		
Pretraining	Test F1 (%)	Rounds
✓	84.49	16.0
✗	66.14	28.0

Table 10: Results of all experiments on OntoNotes dataset after model selection on the best server test loss.

F.3 Gigaword

Centralized learning				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Epochs
✓	34.57 ± 0.66	15.92 ± 0.36	32.35 ± 0.59	3.0 ± 1.41
✗	6.07 ± 0.51	0.36 ± 0.08	5.86 ± 0.5	3.0 ± 1.63
Random distribution ($\tau=0$, 63 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	32.48 ± 0.22	14.28 ± 0.02	30.61 ± 0.19	10.0 ± 0.0
✗	2.89 ± 0.56	0.05 ± 0.02	2.89 ± 0.54	15.0 ± 0.0
Random distribution ($\tau=18$, 63 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	34.72 ± 0.25	15.57 ± 0.08	32.25 ± 0.16	10.0 ± 0.0
✗	1.87 ± 0.3	0.01 ± 0.01	1.83 ± 0.33	14.33 ± 0.94
Random distribution ($\tau=34$, 63 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	34.46 ± 0.54	15.48 ± 0.22	32.07 ± 0.36	10.0 ± 0.0
✗	2.0 ± 0.3	0.03 ± 0.01	1.96 ± 0.32	14.33 ± 0.94
Random distribution ($\tau=126$, 63 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	32.86 ± 0.97	14.45 ± 0.61	30.88 ± 0.82	10.0 ± 0.0
✗	1.54 ± 0.24	0.02 ± 0.01	1.51 ± 0.26	14.33 ± 0.47
Random distribution ($\tau=750$, 63 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	32.8 ± 1.73	14.54 ± 0.91	30.9 ± 1.39	8.33 ± 1.7
✗	3.29 ± 1.74	0.01 ± 0.01	3.22 ± 1.68	13.67 ± 1.89
Random distribution ($\tau=1200$, 63 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	34.17 ± 0.38	15.41 ± 0.23	32.09 ± 0.26	9.33 ± 0.47
✗	4.71 ± 0.98	0.0 ± 0.0	4.67 ± 0.97	11.33 ± 1.25
Uniform distribution ($\gamma=20$, 63 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	33.05	14.56	31.04	10.0
✗	2.41	0.0	2.4	15.0
Uniform distribution ($\gamma=22$, 57 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	33.36	14.89	31.35	10.0
✗	1.58	0.04	1.56	15.0
Uniform distribution ($\gamma=26$, 49 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	33.68	15.03	31.66	10.0
✗	2.07	0.0	2.09	15.0
Uniform distribution ($\gamma=30$, 42 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	34.44	15.43	32.32	10.0
✗	2.06	0.0	2.06	15.0
Uniform distribution ($\gamma=36$, 35 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	35.09	15.88	32.68	10.0
✗	2.18	0.0	2.17	15.0
Uniform distribution ($\gamma=45$, 28 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	35.48	16.19	33.13	10.0
✗	5.59	0.11	5.38	15.0
Uniform distribution ($\gamma=60$, 21 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	35.39	16.38	33.11	10.0
✗	6.44	0.17	6.23	15.0
Uniform distribution ($\gamma=90$, 14 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	35.43	16.32	33.15	10.0
✗	2.73	0.1	2.69	15.0
Uniform distribution ($\gamma=180$, 7 clients)				
Pretraining	Test Rouge1 (%)	Test Rouge2 (%)	Test RougeL (%)	Rounds
✓	35.8	15.99	33.2	10.0
✗	10.68	1.65	10.2	15.0

Table 11: Results of all experiments on Gigaword dataset after model selection on the best server test loss.

Paper Bullets: Modeling Propaganda with the Help of Metaphor

Daniel Baleato Rodríguez
ILCC, University of Amsterdam
daniel@codealia.com

Verna Dankers
ILCC, University of Edinburgh
vernadankers@gmail.com

Preslav Nakov
MBZUAI
preslav.nakov@mbzuai.ac.ae

Ekaterina Shutova
ILCC, University of Amsterdam
e.shutova@uva.nl

Abstract

Propaganda aims to persuade an audience by appealing to emotions and using faulty reasoning, with the purpose of promoting a particular point of view. Similarly, metaphor modifies the semantic frame, thus eliciting a response that can be used to tune up or down the emotional volume of the message. Given the close relationship between them, we hypothesize that, when modeling them computationally, it can be beneficial to do so jointly. In particular, we perform multi-task learning with propaganda identification as the main task and metaphor detection as an auxiliary task. To the best of our knowledge, this is the first work that models metaphor and propaganda together. We experiment with two datasets for identifying propaganda techniques in news articles and in memes shared on social media. We find that leveraging metaphor improves model performance, particularly for the two most common propaganda techniques: loaded language and name-calling.

1 Introduction

Propaganda aims to influence an audience. It is a type of information that, whether true or false, tries to promote a particular agenda (Cantril, 1938) by appealing to emotions or by using faulty reasoning (Miller, 1939). Although this communication strategy comes in many forms, it is conveyed using specific persuasion techniques that exploit our psychology to sell us an idea or a point of view (Da San Martino et al., 2019b). In Figure 1, we can see an example of such techniques used in a meme shared on social media.

Another rhetorical device at the heart of many successful communication strategies is *metaphor*. Postulated as a primordial mechanism to conceptualize what we think and experience (Lakoff, 1980), metaphor works by mapping a concept in one domain (often a physical domain) to another domain (usually an abstract one) by means of a systematic

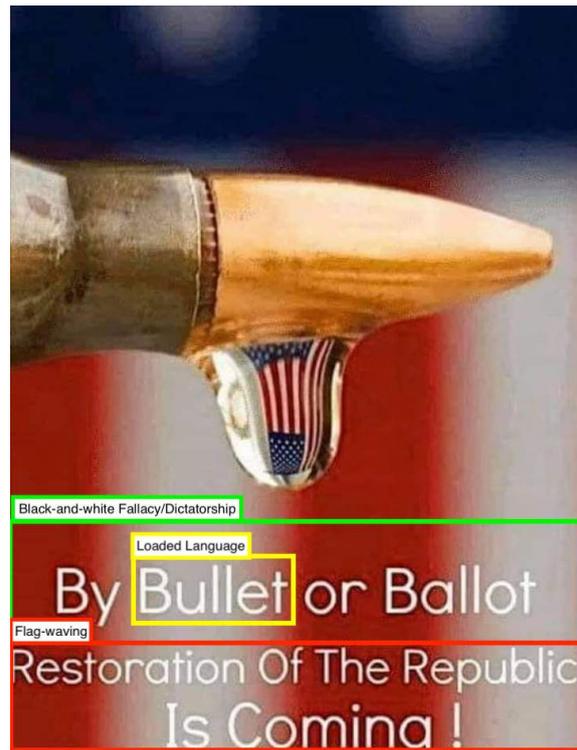


Figure 1: Meme containing propagandistic techniques (Dimitrov et al., 2021). These techniques are highlighted with bounding boxes for illustration purposes.

association. For instance, the term “*paper bullets*”¹ connects the domains of information and war, illustrating the weaponization of information.

In the same way that propaganda can exploit automatic shortcuts our brain uses to process information (e.g., stereotypes) (Tversky and Kahneman, 1974), metaphors can affect how we reason about a particular situation or issue by evoking a different semantic frame (Fillmore et al., 2006). Research shows that characterizing crime as a *beast* delivered more punishment-oriented strategies to *fight* crime (Thibodeau and Boroditsky, 2011). Con-

¹The metaphor “*paper bullets*” was used during World War II, where the Germans used tactical aircrafts to drop anti-Semitic leaflets over American troops (Margolin, 1946) as a way of psychological warfare.

versely, referring to crime as a *virus* gathered a more significant number of preventive measures to *cure* it. As a persuasive device, framing has successfully been used in politics (Howe, 1988; Ana, 1999; Lakoff, 2009) to shift the public opinion about a particular topic. Moreover, the use of metaphors by politicians in their posts on social media increases engagement with their electorate (Prabhakaran et al., 2021).

Some propagandist techniques and metaphors can exhibit a similar intention by the author. For instance, the most common technique is the use of *loaded language* to increase the emotional response of the audience (e.g., “... *disastrous* [nuclear deal]”). Likewise, metaphor can also elicit an emotionally charged reaction (Mohammad et al., 2016). The following example combines both: “the *ruinous* reforms”. Similarly, *name-calling* connects the object of the propaganda campaign with terms the target audience sees positively or negatively (Miller, 1939). This technique seeks a love or hate emotional response, and it could also alter the semantic frame (e.g., “*Crooked* Hillary” or “*Deep* State officials”).

Other salient examples where different propagandist techniques employ metaphor can be found in the Propaganda Techniques Corpus (PTC) (Da San Martino et al., 2019b), including *exaggeration* (“a *tsunami* of lies and smear”), *appeal to fear* (“[bubonic plague in Madagascar] could even *spill over* into neighboring countries and beyond”), *doubt* (“Why is the U.S. *singling out* Iran ...”) and *flag-waving* (“it is time to *take* our government *back* ...”), among others.

We explore how metaphor detection can aid propaganda technique classification under the multi-task learning paradigm. Computational modeling for propaganda detection was initially studied as a document-level classification task in news articles (Rashkin et al., 2017; Barrón-Cedeño et al., 2019; Martino et al., 2020). More recently, annotation efforts produced datasets that identify the text spans where particular forms of propaganda are used. Our work builds upon the most extensive corpus of fragment-level propaganda techniques to date (Da San Martino et al., 2019b) and on shared task 6 from SemEval-2021 (Dimitrov et al., 2021) to identify persuasive techniques in both news articles and internet memes, respectively. We analyze how a multi-task learning approach that leverages metaphor detection can improve results in propa-

ganda identification.

To our knowledge, this is the first study of the role of metaphor in computational propaganda identification. We produce the first models that combine the two phenomena and analyze their predictive capability, both quantitatively and qualitatively.

Our findings show that metaphor detection can increase performance for certain types of propaganda. We see improvements across multiple tasks covering both datasets. The gains are more pronounced for *name-calling*, with significant results for the news domain. Furthermore, our models’ predictions suggest that propagandist content uses figurative language more extensively than non-propagandist text.

2 Related work

2.1 Metaphor detection

NLP applications need to distinguish the particular intent that metaphor plays in context (Veale et al., 2016). Metaphor detection research has studied various approaches: hand-crafted features and word classes (Beigman Klebanov et al., 2016), concreteness and imageability word ratings (Broadwell et al., 2013; Turney et al., 2011), semantic classification making use of lexical databases (e.g., WordNet, VerbNet, ConceptNet) (Wilks et al., 2013; Neuman et al., 2013; Mohler et al., 2013; Tsvetkov et al., 2013), distributional semantic models (Gutierrez et al., 2016; Bulat et al., 2017; Hovy et al., 2013), and even visual (Shutova et al., 2016) or sensorial features (Tekiroglu et al., 2015). More recently, deep learning methods (Mao et al., 2019; Dankers et al., 2020; Gao et al., 2018; Rei et al., 2017; Wu et al., 2018) have been used to detect metaphors.

Current state-of-the-art textual metaphor detection is powered by large pre-trained neural network models (Su et al., 2020; Chen et al., 2020; Gong et al., 2020; Choi et al., 2021) that have been trained using datasets of billions of words. These models can leverage word representations that carry context-sensitive semantic information. As the latest shared task on metaphor detection highlights (ACL 2020) (Leong et al., 2020), more than half of the participants used BERT (Devlin et al., 2019) or its variants, widely successful pre-trained models that perform well on downstream tasks.

In addition, metaphor detection has successfully been used as an auxiliary task in multi-task learning (MTL) (Caruana, 1993) for emotion classification

(Dankers et al., 2019), political perspective, affiliation, and framing (Huguet Cabot et al., 2020); and aspect-based sentiment analysis (Mao and Li, 2021), among others. The MTL approach builds on the idea that the same model can encode valuable features for different tasks that would help each other’s performance. As metaphor is extensively used in everyday language and dramatically influences the expressiveness of the message, it can help in a significant number of semantic tasks.

2.2 Propaganda detection

Propaganda is closely related to political bias and misinformation (colloquially referred to as *fake news*) (Guess and Lyons, 2020). This area of research has gained popularity in the last decade due to concerns regarding the weaponization of social media and how it can negatively affect political discourse (Wardle and Derakhshan, 2017). Work on political bias commonly uses lexicon-based approaches to detect sentiment on political topics, while models to expose fake stories usually rely on publishing patterns and knowledge graphs (Haq et al., 2020).

However, propaganda does not necessarily have to be politically driven or rely on untrue or incorrect information. While some instances of propaganda usually do (e.g., clickbait) (Martino et al., 2020), propagandist content varies in accuracy and the acknowledgment of its sources (Jowett et al., 2012). In essence, propaganda aims to influence an audience to exercise a particular agenda (Cantril, 1938) by appealing to emotions or faulty reasoning (Miller, 1939).

Computational approaches to propaganda detection are relatively recent and were initially directed to the document classification of varying sizes, from news articles to tweets (Barrón-Cedeño et al., 2019; Rashkin et al., 2017; Volkova et al., 2017). Proposed models used BERT, LSTM (Hochreiter and Schmidhuber, 1997), Convolutional Neural Networks (CNN) (LeCun et al., 1995), and Naive Bayes models powered by Glove (Pennington et al., 2014) embeddings. These works rely to different degrees on the labeling of information sources by crowd-sourced groups or non-profit organizations (e.g., MBFC², PropOrNot³). Unfortunately, this categorization approach can introduce noise into the system. Reliable news agencies might occasion-

ally include a propagandist article to fulfill their interest. Conversely, highly propagandist media could publish a non-propagandist piece to boost their credibility.

The latest propaganda detection approaches take advantage of the rhetorical devices that propaganda uses to influence reasoning. Although the literature compiles different accounts of propagandist or persuasive techniques (Miller, 1939; Shah, 2005; Abd Kadir and Abu Hasan, 2014), they are mainly sub-types of the general principles first proposed in Cantril (1938), which share the aim of connecting an idea or propagandist object to an attitude or emotion.

The PTC corpus (Da San Martino et al., 2019b) was the first effort to classify propaganda at a more granular level. It identifies 18 persuasive techniques across 451 news articles, making it the largest of its kind. It annotates the start and end of each propagandist fragment. This corpus, and a later variant, were used in shared tasks on propaganda detection (Da San Martino et al., 2019a, 2020). The best systems used pre-trained Transformer-based models and ensembles (Yoosuf and Yang, 2019; Jurkiewicz et al., 2020; Morio et al., 2020; Chernyavskiy et al., 2020).

More recently, SemEval 2021 Task 6 (Dimitrov et al., 2021) has expanded fragment-level propaganda identification efforts outside the news corpora. It identifies propaganda techniques ingrained in the combination of textual and image data. The task’s dataset consists of 950 internet memes posted on social media with topics related to politics, vaccines, COVID-19, and gender equality. Apart from identifying 20 textual propagandist techniques, it also identifies two that are only present when in combination with the image. The most common and best-performing models used for textual tasks were the transformer-based models BERT and RoBERTa (Kaczyński and Przybyła, 2021; Gupta et al., 2021).

3 Tasks and datasets

In this work, we examine six tasks for fragment-level propagandist technique identification. Half of them use labeled data from news articles, while the others use textual information from memes shared on social media. For each domain, we perform a multi-label classification task — to identify all propagandist techniques in the dataset — and two single-label classification tasks to detect the two

²<https://mediabiasfactcheck.com>

³<http://www.propornot.com/p/the-list.html>

most common persuasive techniques: *loaded language* and *name-calling*. The single-label tasks ignore the rest of the labels in the dataset while using the same textual input as the multi-label tasks.

In addition, MTL models include metaphor detection as an auxiliary task. This task aims to detect all content words used as metaphors in a given text.

3.1 VUA Metaphor Corpus

We use the data from the ACL 2020 shared task on metaphor detection (Leong et al., 2020). Specifically, the all-POS subtask that identifies which content words (i.e., nouns, verbs, adjectives, and adverbs) are used in their metaphorical sense. The data for the task comes from the VU Amsterdam Metaphor Corpus, (Steen et al., 2010) which contains annotations for all words in 117 texts from the British National Corpus (Clear, 1993) and across four different registers: academic text, conversation, fiction, and news. The dataset covers 190K lexical units over 16,189 sentences with a train/test split of 12,109 and 4,080 sentences. The prevalence of metaphorical use for content words is 6.8% for the training set and 7.7% for the test set. We randomly sample 10% of the training split for validation.

3.2 Propaganda Techniques Corpus

The PTC corpus (Da San Martino et al., 2019b) identifies 18 propaganda techniques across 451 articles (350K tokens) from 49 news outlets. The annotations were produced by separate teams of annotators and merged through a consolidation process where all disagreements were discussed before becoming part of the final version. Each annotation identifies the technique used and its start and end within the news article. The dataset contains 20,339 sentences split into training, validation, and test sets with 14,263, 2,034, and 4,042 sentences, respectively.

The number of instances per technique and its length varies widely. The most common classes are *loaded language* with 2,547 occurrences and *name-calling* with 1,294. Those techniques have been used an average of 6.7 and 4.7 times per article, whereas all others appear a maximum of twice per article. We evaluated these two techniques separately as they provide a larger number of positive examples and can relate to metaphor as described in Section 1. Details on the number of annotations per split and their average length are shown in Table 1.

Dataset	#Annotations				Length
	Prop. technique	Total	Train	Val	Test
News					
Loaded language	2,547	1,811	304	432	23.70 \pm 25.30
Name-calling	1,294	931	154	209	26.10 \pm 19.88
All combined	7,480	5,114	927	1,439	46.99 \pm 61.45
Memes					
Loaded language	761	543	68	150	14.87 \pm 18.17
Name-calling	408	301	37	70	17.00 \pm 11.65
All combined	2,083	1,498	182	403	40.43 \pm 48.91

Table 1: Statistics on propaganda technique annotations and their average length in characters.

3.3 Propaganda detection in memes

SemEval-2021 Task 6 (Dimitrov et al., 2021) aims to identify the propagandist technique used in memes shared on social media. The images were collected from 26 public Facebook groups, which provided memes on the following topics: politics, COVID-19, pro-vaccines, anti-vaccines, and gender equality. The annotation process involved a heterogeneous group of annotators and a consolidation step. The text of the images was retrieved automatically using Google Vision API⁴ and manually corrected afterward. We focus on subtask two, which only uses the textual data of the meme to predict where in the text a particular technique is present.

The dataset contains 951 examples (16,840 tokens) divided into 688, 63, and 200 samples for train, validation, and test splits. The average number of sentences per meme is 1.68, with a maximum of 13 sentences in one image alone. Again, the most common techniques are *loaded language* with 761 annotations (36.5%) and *name-calling* 408 occurrences (19.6%) from 2,083 propagandist fragments.

We provide a summary of the textual persuasion techniques in the Appendix Section A.1 and examples in the Appendix Table 12.

4 Methods

4.1 Models

We employ the pre-trained ROBERTA-BASE model (Liu et al., 2019). ROBERTA shares its architecture with its counterpart BERT (Devlin et al., 2019), but it improves performance across many tasks due to its highly optimized training and the use of ten times more data.

⁴<http://cloud.google.com/vision>

ROBERTA tokenizes inputs using byte-pair encodings (Sennrich et al., 2016) and computes contextualised embeddings for these input tokens. We add task-specific classifiers on top of ROBERTA, consisting of a linear layer followed by the sigmoid activation function. During inference, tokens with predicted targets over 0.5 are assigned to the class corresponding to the classifier. We fine-tune all of the model parameters in our respective tasks. Since the datasets provide labels at the word level, we aggregate predictions for words consisting of multiple tokens. If the model predicts any token to belong to a particular class, we assign the label to the whole word.

4.2 Single-task learning

Our main task is to detect the text span of each propaganda technique from news articles and memes. When solely training the model on a propaganda task, we refer to this as *single-task* learning (STL). The standard propaganda task as introduced in Section 3 is *multi label*. Since propagandist fragments can overlap, we perform multi-label classification by predicting the presence of each technique independently at each token, using separate task classifiers per technique as described in Section 4.1.

In addition to the multi-label propaganda technique identification, we generate two *single-label* tasks targeting the most frequent persuasion techniques (i.e., *loaded language* and *name-calling*). Both techniques share common aspects with metaphor discussed in Section 1, making them particularly interesting for experimentation.

4.3 Multi-task learning

In the MTL setup, we train the model jointly on two tasks: one of the propaganda identification tasks and the metaphor detection task. Similar to the STL setup, we do this both for single-label and multi-label classification. As the model learns to identify metaphors, we hypothesize that the metaphor-related features benefit the propaganda technique identification.

We extend the STL models with an additional classifier to predict metaphor as the auxiliary task. All tasks share the pre-trained model (ROBERTA) in a hard parameter sharing fashion. For fine-tuning, we reuse the best configuration from the single-task models to facilitate comparison between the two strategies. We experiment with different MTL regimes attending to the following hyper-parameters:

- Task sampling ratios (r_a, r_m): these ratios are used to select a task at each update step during training. With a probability of $p_m = r_m / (r_a + r_m)$ the main task is selected, and with $p_a = r_a / (r_a + r_m)$ the auxiliary task is selected.
- Epoch sampling coefficients (c_a, c_m): these coefficients are used to update the sampling ratios at every epoch. At epoch n , $r_{m_n} = r_{m_{n-1}} \times c_m$, and $r_{a_n} = r_{a_{n-1}} \times c_a$.
- Loss scaling factors (s_a, s_m): these hyper-parameters are used to scale the losses for the main task (s_m) and the auxiliary task (s_a).

Although the MTL models have access to more data, as they are trained on two datasets, we limit their computational budget to match the one available for STL models. Every epoch, the model is trained in iterations, where the number of iterations is the same as for the STL model. Each iteration randomly selects a task for training according to its sampling probability p . We shuffle all examples in the training set at the start and after exhaustion of the training split. We fill each batch with samples from the selected task at random without replacement.

5 Experiments and results

5.1 Experimental Setup

In the implementation, we use the PyTorch framework and the pre-trained ROBERTA-BASE⁵ model from the *transformers* library (Wolf et al., 2020). We trained all models using a maximum sequence length of 512 tokens, a weight decay of 0.01, and the AdamW optimizer (Loshchilov and Hutter, 2017) with a 10% warm-up period and a cosine-based learning rate decay function. All hyperparameter search trials and the selected configurations for each task are listed in Appendix Table 11. We use the binary cross-entropy loss with modified class weights to account for class imbalances. Hyperparameter search trials are performed over five different random seeds that dictate the order of data presentation and the initialisation of the task-specific classifiers. For the final configuration, performance is computed over ten different random seeds.

To ensure that the MTL models do capture meaningful features for metaphor identification – in spite

⁵<https://huggingface.co/roberta-base>

of it being the auxiliary task – we discard hyperparameter combinations with a median F1-score below 0.6 for metaphor identification, asserting the models exceed the baseline level set out by baselines one and two of [Leong et al. \(2020\)](#).

We evaluate the performance of propaganda detection based on the micro-averaged F1 score using precision (P) and recall (R) metrics defined in [Da San Martino et al. \(2019b\)](#). These metrics give partial credit to imperfect matches to account for overlap between techniques and the significant variation in length between propagandistic fragments. We provide details of these calculations in the Appendix A.2. We use statistical bootstrapping ([Efron, 1979](#)) to test the significance of our results and detail the procedure in Appendix A.3. We detail the system and configurations used for this work in Appendix A.4 for reproducibility.

5.2 Results

5.2.1 Single-label propaganda technique detection

Table 2 shows the performance for single-label propaganda detection tasks. The MTL approach improves results for all single-label tasks. Adding metaphor increases performance in news articles by 1.02 points for *name-calling*, from 28.72 to 29.74. This growth is statistically significant under the paired bootstrap test between learning strategies. The improvement is milder for *loaded language*, with a gain of 0.22 points, although results were more stable, almost halving the standard deviation for the metric.

We observe similar results in the memes dataset. Detection of *name-calling* improves the F1 metric by 1.24 points to 57.77 when training with metaphor as an auxiliary task. This increase is also more stable, lowering the standard deviation from 2.44 to 1.26. *Loaded language* improvements are smaller, adding 0.34 points to a total of 65.5 with lower variability.

5.2.2 Multi-label propaganda technique detection

Table 3 shows the results for the multi-label propaganda identification task in the news dataset. We compare our models to previous work ([Da San Martino et al., 2019b](#)) and achieve better results using a similar pre-trained model with the same number of parameters. The multi-task models obtained the best overall performance with an F1 of 24.32 and

Model	Loaded Language			Name Calling		
	P	R	F1	P	R	F1
News						
STL	32.67	48.04	38.72 ± 1.00	24.48	35.25	28.72 ± 1.14
MTL	33.60	46.88	38.94 ± 0.51	26.30	35.35	29.74 ± 1.64
Memes						
STL	68.88	62.21	65.16 ± 2.16	52.39	61.70	56.53 ± 2.44
MTL	66.29	64.90	65.50 ± 1.62	59.03	57.13	57.77 ± 1.26

Table 2: Propaganda detection performance for single-label models. Statistically significant differences between STL and MTL are underlined ($p < 0.05$).

Model	P	R	F1
Da San Martino et al. (2019b)	24.42	21.05	22.58
Multi-label STL	20.37	30.42	23.78 ± 2.03
Multi-label MTL	21.98	27.46	24.32 ± 0.48

Table 3: Propaganda technique identification results in news articles. The highest performance per model type is shown in bold. Underlined values denote statistical significance ($p < 0.05$) via paired bootstrap test between single-task and multi-task models.

a standard deviation of 0.48. The single-task models averaged 23.78 F1 score with a much higher variation ($\sigma = 2.03$).

Results for multi-label propaganda detection in memes are shown in Table 4. State-of-the-art performance for the task reaches an F1 score of 47.6, ([Gupta et al., 2021](#)) but it uses a model with 340M parameters. This model is three times larger than the ones we used (110M parameters). Comparing performance across same-size models, we see that our STL model performs best with an average F1 score of 46.22 and a standard deviation of 1.82. In contrast, the multi-task model achieves 44.81 ± 1.31 . Both models outperform the value of 43.9 ± 0.9 reported in [Gupta et al. \(2021\)](#).

In the Appendix, Tables 8 and 9 show the performance of multi-label models for all techniques in the news and memes datasets, respectively.

6 Analysis and discussion

Given the shared traits between the use of metaphor and specific propagandist techniques, we hypothesized that it can be beneficial to model them jointly. We split the analysis into two subsections discussing quantitative and qualitative aspects.

6.1 Quantitative analysis

The results show improvements across most propaganda detection tasks when trained in a multi-

Model	P	R	F1
Volta (RoBERTa-Large)	-	-	47.6 ± 1.5
Volta (RoBERTa-Base)	-	-	43.9 ± 0.9
Multi-label STL	46.02	46.51	46.22 ± 1.82
Multi-label MTL	42.62	47.82	44.81 ± 1.31

Table 4: Propaganda technique identification results in memes. We include the winning team for the shared task: *Volta Gupta et al. (2021)*. The highest performance per model type is shown in bold. Underlined values denote statistical significance ($p < 0.05$) via paired bootstrap test between single-task and multi-task models.

task setting with metaphor as the auxiliary task. We hypothesised metaphor detection would benefit the single-label tasks, due to the use of a different semantic frame in *name-calling* and emotionally charged vocabulary in *loaded language*. Improvements were more pronounced for *name-calling* in both datasets, which suggests that, as anticipated, metaphorical framing plays a role in this propaganda technique. The fact that the gain in F1-score is the largest for name calling in both datasets further strengthens this conclusion.

To further consolidate the relationship between propaganda and metaphor our models identify, we investigate the prevalence of metaphors’ predictions in propagandistic text fragments. We use our MTL models to predict metaphors on the propaganda corpora, and observe a higher percentage of metaphors in propagandist fragments than for non-propagandist content, and even higher for *loaded language* and *name-calling*. This is shown in the Appendix, in Figures 2 and 3. These model predictions hint at the likelihood that propagandist content, and some techniques in particular, may resort to metaphor more often than non-propagandist text does. Manual annotation of metaphors in propaganda datasets will allow asserting this with certainty, yet, we leave this for future work.

Although a slight improvement in task performance was observed for multi-label propaganda identification in the news dataset, this was not the case for the memes task. This task was the only one for which the MTL strategy was not superior. The memes dataset is 20 times smaller than the news dataset and includes two more labels. These challenges of size and sparsity could play a role in the utility of the MTL architecture, particularly when imposing on it the best hyper-parameters from the single-task models. We did this to facilitate the

comparison between models, but we risk ending up with a configuration especially harmful to the MTL approach. Further experimentation is needed to investigate this drop in performance.

6.2 Qualitative analysis

To validate the effect of metaphor for the tasks, we pooled the predictions for all ten models of the same type trained with different seeds. We use simple majority voting to harmonize predictions across the different runs. Next, we identify the difference in the predicted spans between single-task and multi-task models. We include gold labels and the predicted metaphors by multi-task models for analysis. Examples of models’ predictions for news articles and memes are shown in Table 5.

MTL models can detect figurative language, which contributes to detecting propaganda techniques that use this device. Idioms such as “*throw out the window*” (ref. LL.N.1) and “*kick the can down the road*” (ref. LL.N.2) are correctly identified, albeit partially, as *loaded language* in the context of the news article. This is also the case for the metaphorical use of the word “*dinosaurs*”, present in an example of *name-calling*, to convey the point of view that current social media platforms will *go extinct* (ref. NC.N.1).

Other instances of non-literal meaning deliver incorrect predictions. However, we believe that some of those instances could be considered correct. In the case of *name-calling*, the models detect “*poor sport*” (ref. NC.N.2), which is alluding to a defeated candidate in an electoral race. Similarly, the phrase “*you can throw us in jail, but you will never defeat us*” (ref. LL.N.3) signals defiance with a considerable degree of emotion which borderlines the *loaded language* category.

Conversely, the label *hardworking* used in “*hardworking Georgians*” (ref. NC.N.3) cannot be attributed to *name-calling* as it does not refer to the propagandist target of the article: Georgia gubernatorial candidate Stacy Abrams. This mislabeled example highlights the task’s difficulty and need for a broader context. Our models received individual sentences for training and inference, which is insufficient in this instance to identify the object of the propaganda campaign.

Looking at predictions on the memes dataset, we observe that the gains in *name-calling* for multi-task models were driven primarily by minimizing incorrect predictions. The examples NC.M.1 and

PT	Reference	Text fragments	
Name-calling	NC.News.1	... we will rise from the ashes of the social media dinosaurs to help build and create new platforms ...	
	NC.News.2	Talk about a poor sport , but Democrats are often like that in these races.	
	NC.News.3	“The election is over and hardworking Georgians are ready to move forward ,” he said.	
	NC.Memes.1	HOLD UP!!! Sleepy Joe broke my record?!?!?!?	
	NC.Memes.2	... the most corrupt, lying and despised member of Congress and the WORST Speaker of the house ...	
	NC.Memes.3	So Don King and Beetlejuice had a baby...	
	NC.Memes.4	WARNING SIGNS OF A CULT // ...	
	NC.Memes.5	ATTENTION PATRIOTS // MEET YOUR CIVIL WAR OPPONENTS	
	Loaded Language	LL.News.1	Political correctness needs to be thrown out the window when dealing with those who...
		LL.News.2	In other words, let’s just kick the can down the road and hope for a more reasonable Iranian regime ...
LL.News.3		You can throw us in jail , but you will never defeat us .	
LL.Memes.1		WHEN TRUMP IS REELECTED THERE WILL BE BLOOD!	
LL.Memes.2		WE ARE AT WAR!	
LL.Memes.3		FAKE WINNER	
LL.Memes.4		... UNDERCOVER FEDS DOCUMENTING THE FRAUD AND THEY’VE STEPPED INTO A TRAP	

Table 5: Example predictions of propaganda techniques. Gold labels in yellow, predictions in blue, and their intersection in green. The underline style identifies predictions only produced by one learning strategy. Predicted metaphors from MTL models are shown in bold.

NC.M.2 were the only ones containing prediction spans singular to the multi-task models. Both instances correctly label parts of the text that do not include predicted metaphors, although they contain metaphors in their vicinity. In contrast, the single-task models produced more mislabels on nouns or noun phrases, see examples NC.M.3, NC.M.4 and NC.M.5. With respect to loaded language, we observe metaphor predictions falling equally into correct and incorrect spans, see examples LL.M.1, LL.M.2, LL.M.3, LL.M.4.

7 Conclusion and Future Work

In this work, we explored the influence of metaphor detection on propaganda technique identification in a multi-task learning setup. Joint modelling of metaphor and propaganda was performed using two propaganda datasets from different domains: news articles and internet memes. We experimented with six different propaganda detection tasks, including multi-label propaganda technique identification and single-label tasks for the two most common propagandist techniques: *name-calling* and *loaded*

language, for each dataset. Incorporating metaphor detection yielded performance improvements in five of the six tasks considered, with the highest improvements observed for the *name-calling* technique. Moreover, the different datasets showed similar patterns in performance changes. We supplemented the task performance results with an analysis of the prevalence of metaphor in the propaganda corpora and qualitatively examined a range of examples of metaphorical language use in propagandist fragments. We are the first to investigate the interaction of these two phenomena and our promising results encourage further research in this direction.

In future work, we plan to extend our analysis to other propaganda techniques. In view of the emergence of datasets for other languages, such as the Arabic propaganda detection shared task at WANLP’2022 and the multilingual SemEval-2023 task 3 subtask 3 on propaganda detection in English, French, German, Italian, Polish, and Russian, we plan future multi/cross-lingual experiments.

Limitations

Although we established a positive influence of metaphor detection on propaganda technique identification, our work also has some limitations. (1) Considering that this work focused on the two most common propagandist techniques, future work could extend this analysis to cover others, although we should note that these analyses are limited by a data scarcity issue (in particular in the memes dataset). (2) While we considered six tasks, these tasks used one MTL architecture. Previous work has experimented with more advanced MTL methods (e.g., soft parameter sharing) and in the future, these methods could also benefit joint learning of metaphor and propaganda. (3) Finally, it should be emphasised that both types of propaganda employed and the types of figurative language used are very specific to cultures and languages. As such, the techniques applied in this study might not deliver the same effect when using data from different geographical locations, or data from languages other than English. Moreover, the prevalence of metaphor varies across different propagandist techniques, meaning that not every propaganda-related task will benefit from joint learning with metaphor.

Ethics and Broader Impact

Intended Use and Misuse Potential Our models can be of interest to the general public, fact-checkers, and journalists. However, they could also be misused by malicious actors. We, therefore, ask researchers to exercise caution.

Environmental Impact We would like to warn that the use of large language models requires a lot of computations and the use of GPUs/TPUs for training, which contributes to global warming (Strubell et al., 2019). This is a bit less of an issue in our case, as we do not train such models from scratch, we just fine-tune them.

References

- Shamsiah Abd Kadir and Ahmad Sauffiyah Abu Hasan. 2014. A content analysis of propaganda in harakah newspaper. *Journal of Media and Information Warfare (JMIW)*, 5:73–116.
- Otto Santa Ana. 1999. ‘Like an Animal I was Treated’: Anti-Immigrant Metaphor in US Public Discourse. *Discourse & Society*, 10(2):191–224.
- Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. **Proppy**:

Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

- Beata Beigman Klebanov, Chee Wee Leong, E. Dario Gutierrez, Ekaterina Shutova, and Michael Flor. 2016. **Semantic classifications for detection of verb metaphors.** In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 101–106, Berlin, Germany. Association for Computational Linguistics.
- George Aaron Broadwell, Umit Boz, Ignacio Cases, Tomek Strzalkowski, Laurie Feldman, Sarah Taylor, Samira Shaikh, Ting Liu, Kit Cho, and Nick Webb. 2013. Using Imageability and Topic Chaining to Locate Metaphors in Linguistic Corpora. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 102–110, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Luana Bulat, Stephen Clark, and Ekaterina Shutova. 2017. **Modelling metaphor with attribute-based semantics.** In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 523–528, Valencia, Spain. Association for Computational Linguistics.
- Hadley Cantril. 1938. Propaganda analysis. *The English Journal*, 27(3):217–221.
- Richard A. Caruana. 1993. **Multitask Learning: A Knowledge-Based Source of Inductive Bias.** In *Machine Learning Proceedings 1993*, pages 41–48. Elsevier.
- Xianyang Chen, Chee Wee (Ben) Leong, Michael Flor, and Beata Beigman Klebanov. 2020. **Go Figure! Multi-task transformer-based architecture for metaphor detection using idioms: ETS team in 2020 metaphor shared task.** In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 235–243, Online. Association for Computational Linguistics.
- Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2020. **Aschern at SemEval-2020 task 11: It takes three to tango: RoBERTa, CRF, and transfer learning.** In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1462–1468, Barcelona (online). International Committee for Computational Linguistics.
- Minjin Choi, Sunkyoung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. **MeiBERT: Metaphor Detection via Contextualized Late Interaction using Metaphorical Identification Theories.** In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1763–1773, Online. Association for Computational Linguistics.

- Jeremy H. Clear. 1993. *The British National Corpus*, page 163–187. MIT Press, Cambridge, MA, USA.
- Giovanni Da San Martino, Alberto Barrón-Cedeño, and Preslav Nakov. 2019a. [Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection](#). In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 162–170, Hong Kong, China. Association for Computational Linguistics.
- Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. [SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.
- Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019b. [Fine-Grained Analysis of Propaganda in News Article](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5635–5645, Hong Kong, China. Association for Computational Linguistics.
- Verna Dankers, Karan Malhotra, Gaurav Kudva, Volodymyr Medentsiy, and Ekaterina Shutova. 2020. [Being neighbourly: Neural metaphor identification in discourse](#). In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 227–234, Online. Association for Computational Linguistics.
- Verna Dankers, Marek Rei, Martha Lewis, and Ekaterina Shutova. 2019. [Modelling the interplay of metaphor and emotion through multitask learning](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2218–2229, Hong Kong, China. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In *Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. [SemEval-2021 Task 6: Detection of Persuasion Techniques in Texts and Images](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.
- Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. [The hitchhiker’s guide to testing statistical significance in natural language processing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392. Association for Computational Linguistics.
- B. Efron. 1979. [Bootstrap methods: Another look at the jackknife](#). *The Annals of Statistics*, 7(1).
- Charles J Fillmore et al. 2006. Frame semantics. *Cognitive linguistics: Basic readings*, 34:373–400.
- Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. [Neural Metaphor Detection in Context](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 607–613, Brussels, Belgium. Association for Computational Linguistics.
- Hongyu Gong, Kshitij Gupta, Akriti Jain, and Suma Bhat. 2020. [IlliniMet: Illinois System for Metaphor Detection with Contextual and Linguistic Information](#). In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 146–153, Online. Association for Computational Linguistics.
- Andrew M. Guess and Benjamin A. Lyons. 2020. *Misinformation, Disinformation, and Online Propaganda*, SSRC Anxieties of Democracy, page 10–33. Cambridge University Press.
- Kshitij Gupta, Devansh Gautam, and Radhika Mamidi. 2021. [Volta at SemEval-2021 Task 6: Towards Detecting Persuasive Texts and Images using Textual and Multimodal Ensemble](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1075–1081, Online. Association for Computational Linguistics.
- E.Dario Gutierrez, Ekaterina Shutova, Tyler Marghetis, and Benjamin Bergen. 2016. [Literal and Metaphorical Senses in Compositional Distributional Semantic Models](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 183–193, Berlin, Germany. Association for Computational Linguistics.
- Ehsan Ul Haq, Tristan Braud, Young D. Kwon, and Pan Hui. 2020. [A Survey on Computational Politics](#). *IEEE Access*, 8:197379–197406. Conference Name: IEEE Access.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Dirk Hovy, Shashank Shrivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huiying Li, Whitney Sanders, and Eduard Hovy. 2013. [Identifying Metaphorical Word Use with Tree Kernels](#). In *Proceedings of the First Workshop on Metaphor in {NLP}*, pages 52–57, Atlanta, Georgia.

- Nicholas Howe. 1988. [Metaphor in Contemporary American Political Discourse](#). *Metaphor and Symbolic Activity*, 3(2):87–104.
- Pere-Lluís Huguet Cabot, Verna Dankers, David Abadi, Agneta Fischer, and Ekaterina Shutova. 2020. [The Pragmatics behind Politics: Modelling Metaphor, Framing and Emotion in Political Discourse](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4479–4488, Online. Association for Computational Linguistics.
- Garth Jowett, Victoria O’Donnell, and Garth Jowett. 2012. *Propaganda & persuasion*, 5th edition. SAGE, Thousand Oaks, Calif. OCLC: ocn674939375.
- Dawid Jurkiewicz, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński. 2020. [ApplicaAI at SemEval-2020 task 11: On RoBERTa-CRF, span CLS and whether self-training helps them](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1415–1424, Barcelona (online). International Committee for Computational Linguistics.
- Konrad Kaczyński and Piotr Przybyła. 2021. [HOMADOS at SemEval-2021 task 6: Multi-task learning for propaganda detection](#). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1027–1031, Online. Association for Computational Linguistics.
- George Lakoff. 1980. *Metaphors we live by*. University of Chicago Press, Chicago [etc].
- George Lakoff. 2009. [Metaphor and War: The Metaphor System Used to Justify War in the Gulf](#). *Cognitive Semiotics*, 4(2).
- Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.
- Chee Wee (Ben) Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xinyang Chen. 2020. [A Report on the 2020 VUA and TOEFL Metaphor Detection Shared Task](#). In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 18–29, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A Robustly Optimized BERT Pretraining Approach](#). *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.
- Ilya Loshchilov and Frank Hutter. 2017. [Decoupled weight decay regularization](#).
- Rui Mao and Xiao Li. 2021. [Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13534–13542.
- Rui Mao, Chenghua Lin, and Frank Guerin. 2019. [End-to-End Sequential Metaphor Identification Inspired by Linguistic Theories](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3888–3898, Florence, Italy. Association for Computational Linguistics.
- Leo Jay Margolin. 1946. *Paper Bullets: A Brief Story of Psychological Warfare in World War II*. New York: Froben Press.
- Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Yu, Roberto Di Pietro, and Preslav Nakov. 2020. [A Survey on Computational Propaganda Detection](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 4826–4832, Yokohama, Japan. International Joint Conferences on Artificial Intelligence Organization.
- Clyde R Miller. 1939. The techniques of propaganda. from “how to detect and analyze propaganda,” an address given at town hall. *The Center for learning*.
- Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 23–33.
- Michael Mohler, David Bracewell, Marc Tomlinson, and David Hinote. 2013. [Semantic Signatures for Example-Based Linguistic Metaphor Detection](#). In *Proceedings of the First Workshop on Metaphor in [NLP]*, pages 27–35, Atlanta, Georgia. Association for Computational Linguistics.
- Gaku Morio, Terufumi Morishita, Hiroaki Ozaki, and Toshinori Miyoshi. 2020. [Hitachi at SemEval-2020 task 11: An empirical study of pre-trained transformer family for propaganda detection](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1739–1748, Barcelona (online). International Committee for Computational Linguistics.
- Yair Neuman, Dan Assaf, Yohai Cohen, Mark Last, Shlomo Argamon, Newton Howard, and Ophir Frieder. 2013. [Metaphor Identification in Large Texts Corpora](#). *PLoS ONE*, 8(4):e62343.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Vinodkumar Prabhakaran, Marek Rei, and Ekaterina Shutova. 2021. [How Metaphors Impact Political Discourse: A Large-Scale Topic-Agnostic Study Using Neural Metaphor Detection](#). *Proceedings of the International AAAI Conference on Web and Social Media*, 15:503–512.

- Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. [Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.
- Marek Rei, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. 2017. [Grasping the Finer Point: A Supervised Similarity Network for Metaphor Detection](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1537–1546, Copenhagen, Denmark. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725. Association for Computational Linguistics (ACL).
- Anup Shah. 2005. War, propaganda and the media. *Global Issues*, 31.
- Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. [Black Holes and White Rabbits: Metaphor Identification with Visual Features](#). In *Proceedings of the 2016 Conference of the North (A)merican Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 160–170, San Diego, California. Association for Computational Linguistics.
- Gerard J Steen, Aletta G Dorst, J Berenike Herrmann, Anna A Kaal, and Tina Krennmayr. 2010. [VU amsterdam metaphor corpus](#). Oxford Text Archive.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and policy considerations for deep learning in NLP](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Chuangdong Su, Fumiyo Fukumoto, Xiaoxi Huang, Jiyi Li, Rongbo Wang, and Zhiqun Chen. 2020. [DeepMet: A Reading Comprehension Paradigm for Token-level Metaphor Detection](#). In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 30–39, Online. Association for Computational Linguistics.
- Serra Sinem Tekiroglu, Gözde Özbal, and Carlo Strapparava. 2015. [Exploring Sensorial Features for Metaphor Identification](#). In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 31–39, Denver, Colorado. Association for Computational Linguistics.
- Paul H. Thibodeau and Lera Boroditsky. 2011. [Metaphors We Think With: The Role of Metaphor in Reasoning](#). *PLoS ONE*, 6(2):e16782.
- Yulia Tsvetkov, Elena Mukomel, and Anatole Gershman. 2013. [Cross-Lingual Metaphor Detection Using Common Semantic Features](#). In *Proceedings of the First Workshop on Metaphor in {NLP}*, pages 45–51, Atlanta, Georgia. Association for Computational Linguistics.
- Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. [Literal and Metaphorical Sense Identification through Concrete and Abstract Context](#). *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690.
- Amos Tversky and Daniel Kahneman. 1974. [Judgment under uncertainty: Heuristics and biases](#). *Science*, 185(4157):1124–1131.
- Tony Veale, Ekaterina Shutova, and Beata Beigman Klebanov. 2016. *Metaphor: A Computational Perspective*. Number Vol. 31 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
- Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. 2017. [Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 647–653, Vancouver, Canada. Association for Computational Linguistics.
- Claire Wardle and Hossein Derakhshan. 2017. Information disorder: Toward an interdisciplinary framework for research and policy making. *Council of Europe report*, 27:1–107.
- Yorick Wilks, Adam Dalton, James Allen, and Lucian Galescu. 2013. [Automatic Metaphor Detection using Large-Scale Lexical Resources and Conventional Metaphor Extraction](#). In *Proceedings of the First Workshop on Metaphor in {NLP}*, pages 36–44. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Chuhan Wu, Fangzhao Wu, Yubo Chen, Sixing Wu, Zhigang Yuan, and Yongfeng Huang. 2018. [THU NGN at NAACL-2018 Metaphor Shared Task: Neural Metaphor Detecting with CNN-LSTM Model](#). In *Proceedings of the Workshop on Figurative Language Processing*, pages 110–114, New Orleans, Louisiana. Association for Computational Linguistics.

Shehel Yoosuf and Yin Yang. 2019. *Fine-grained propaganda detection with fine-tuned BERT*. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 87–91, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Persuasion techniques

The following list compiles the descriptions of propaganda techniques present in the PTC corpus (Da San Martino et al., 2019b) and the dataset used by SemEval-2021 task 6 (Dimitrov et al., 2021).

1. Appeal to authority: stating the validity of a claim because an expert or authority has issued it without providing any other evidence. The datasets include *Testimonials* as part of this technique, although they might not refer to an expert or authority.
2. Appeal to fear/prejudice: building support for an idea by provoking anxiety/panic to the alternative. In some instances, it leverages prejudices to obtain the desired response.
3. Bandwagon: invites the target audience to support an idea or action with the pretext that "*everyone is doing the same*".
4. Black-and-white Fallacy (Dictatorship): introduces two alternatives as the only possible options to weaken or strengthen one of them. In the extreme, it morphs into *dictatorship* when the choice is made for the audience, and all other options are considered impossible.
5. Causal Oversimplification: assuming a single cause for an issue when there might be many factors at play in reality. The data also includes *scapegoating* in this category - moving the blame to a person or group without considering the issue's complexities.
6. Doubt: questioning the credibility of something or someone.
7. Exaggeration/Minimisation: representing something as more extreme/dramatic than it is or, conversely, downplaying its significance.
8. Flag-waving: rally around a solid national sentiment to justify an action or idea.
9. Glittering generalities (Virtue)⁶: words or symbols that produce a positive image of the propagandist object by association with the preferences of the target audience.
10. Loaded Language: the use of emotionally charged words to influence an audience. It often exploits stereotypes and vagueness.
11. Name-calling: referring to the object of the propagandist campaign with a label that connects the target audience with an emotion, either positive (love, praise) or negative (fear, hate).
12. Obfuscation, Intentional Vagueness, Confusion: deliberately use unclear statements forcing the audience to produce their interpretation.
13. Red Herring: presenting irrelevant data to divert attention away from the discussed issue.
14. Reductio ad Hitlerum: seek disapproval of a position by suggesting that it is popular with a group the target audience hates.
15. Repetition: repeating the same message to subdue the audience into acceptance.
16. Slogans: brief and memorable motto or phrase to persuade the audience.
17. Smears⁶: effort to damage or question someone's reputation by propounding negative propaganda.
18. Straw Man: misrepresentation of someone's position to disprove it leaving the original argument unaddressed.
19. Thought-terminating cliché: using expressions to prevent critical thinking and meaningful discussions.
20. Whataboutism: replying with a counter-question or counter-accusation that suggests the rival is hypocritical concerning their position without refuting their argument.

⁶Only present for propaganda in memes, not for propaganda in the news dataset

A.2 Evaluation metrics for propaganda

To evaluate the model’s performance in identifying propagandist instances, we follow the methods used by preceding works. The authors of the PTC corpus (Da San Martino et al., 2019b) propose precision and recall metrics based on the overlaps between the target and predicted spans. These metrics are then used to calculate the F1 score for each technique and all techniques combined.

Should document d be a sequence of characters, we can represent a propaganda technique span by $t = [t_i, \dots, t_j] \subseteq d$. This ground truth will be compared against the predicted model outputs $s = [s_i, \dots, s_j]$. The labeling function $l(x)$ will return the propaganda technique associated with the fragment x . The function $\delta(l_a, l_b)$ will return 1 when l_a equals l_b and 0 otherwise. The groups T and S denote the group of propagandist fragments for gold labels and predictions respectively. Equation 1 calculates the overlapping number of character between two spans and divides it by a given length h .

$$C(s, t, h) = \frac{|(s \cap t)|}{h} \delta(l(s), l(t)) \quad (1)$$

In turn, Equation 2 reuses C to calculate the precision metric as the average proportion of correct prediction spans. Conversely, the Equation 3 defines recall as the average proportion of ground truth fragments covered by the predicted spans. Both metrics are similar, but while precision uses the number and length of the predictions, recall uses the gold label spans instead.

$$P(S, T) = \frac{1}{|S|} \sum_{s \in S, t \in T} C(s, t, |s|) \quad (2)$$

$$R(S, T) = \frac{1}{|T|} \sum_{s \in S, t \in T} C(s, t, |t|) \quad (3)$$

In contrast, precision and recall metrics for metaphor are calculated as a binary classification task at the word level. Only content words are considered for this task.

A.3 Significance testing

To check the statistical significance of our results, we use statistical bootstrapping (Efron, 1979). This powerful non-parametric method is recommended for evaluation metrics such as precision, recall, and F-score in NLP tasks (Dror et al., 2018). The main idea is to assess whether differences in performance

between two models originate from variability in the data rather than from the superiority of one model over the other.

First, we create 100 different bootstrap samples ($B_{1..100}$) from the test data (T) by sampling with replacement (i.e., an example can appear multiple times within the same sample while others might not be present at all). Our examples are either individual sentences from news articles or the textual information of a meme, depending on the dataset used for the task. Each bootstrap sample has the same size as the test set ($|T| = |B_n| \forall n \in \{1, 100\}$). The premise is that being the test set a representative sample from all possible data for the task; we can get a sense of the variability of the task’s data by comparing performance across multiple bootstrap samples.

After randomly generating the samples, we performed a paired bootstrap test as suggested in Dror et al. (2018). We calculate the *p-value* as the proportion of bootstrap samples where one type of model outperforms another. Since we use ten different seeds for each setup, comparing results between single-task and multi-task learning strategies requires calculating the mean across multiple models’ performances. We start by calculating the performance of all models of a particular type by averaging their scores on each bootstrap sample. We do this for single-task and multi-task models. Then, we count the number of times one strategy achieves higher performance than the other. Finally, we calculate the *p-value* as the proportion of samples where that strategy was superior. We use the standard confidence level of 95% ($\alpha = 0.05$).

A.4 Reproducibility

We adapted our source code ⁷ to achieve reproducible results. First, we enabled the use of deterministic algorithms in the *PyTorch* framework. Next, we manually set the seed for all packages involved in random number generation. We use natural numbers for the seeds starting at one and up to the number of runs for each set of hyperparameters tested. Finally, we pinned the versions for all dependencies.

The system we used had the following software: Python/3.8.2, GCCcore/9.3.0, CUDA/11.2, cuDNN/8.2.1.32. Additionally, we assigned the value ":4096:8" to the environment vari-

⁷<https://github.com/baleato/paper-bullets>

Task	Duration	
	STL	MTL
News		
Multi-label	1:31:14	1:31:01
Name Calling	38:53	52:59
Loaded Language	37:06	53:47
Memes		
Multi-label	12:38	36:48
Name Calling	6:33	23:59
Loaded Language	15:06	22:36

Table 6: Average training runtime per task. This includes models discarded in hyper-parameter search trials.

able "CUBLAS_WORKSPACE_CONFIG" as suggested by Nvidia documentation⁸ to avoid non-deterministic behavior. Our models used a single GeForce 1080Ti GPU for training. The average training runtime per task is shown in Appendix Table 6.

A.5 Preprocessing

The PTC dataset used NLTK sentence splitter⁹ to break news articles into individual sentences. We detected duplicates driven primarily by boilerplate content regarding site functionality (e.g., invitation to participate in an online poll or request to subscribe to their newsletter). Duplicates were mainly short sentences that did not include any labels. We removed these instances from the training set.

We observed that the text in 454 examples (47% of the data) for the memes dataset was upper-cased. Since our model is case-sensitive, we true-cased all instances to minimize the number of out-of-vocabulary words by the tokenizer.

⁸<https://docs.nvidia.com/cuda/cublas/index.html>

⁹<https://www.nltk.org/api/nltk.tokenize.html#module-nltk.tokenize.punkt>

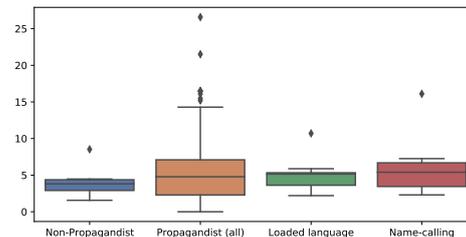


Figure 2: Percentages of metaphorical open-class words predicted by multi-label MTL models in news articles (test set).

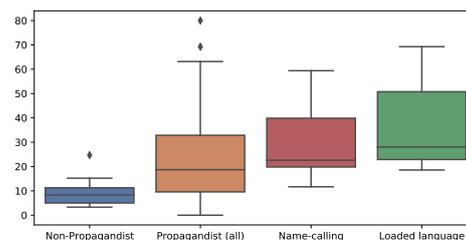


Figure 3: Percentages of metaphorical open-class words as predicted by multi-label MTL models in social memes (test set).

Model	F1 score
Multi-label (news)	66.77 ± 0.61
Name Calling (news)	62.95 ± 2.05
Loaded Language (news)	61.42 ± 1.64
Multi-label (memes)	64.09 ± 4.63
Name Calling (memes)	56.23 ± 4.68*
Loaded Language (memes)	63.05 ± 5.73

Table 7: Metaphor F1 score performance for multi-task models. *The first five runs had a median of 60.79; however, adding five extra seeds brought it down to 56.23.

Model / Propagandist Technique	P	R	F1
Single-task learning			
- Appeal to Authority	7.58	1.14	1.67 ± 0.80
- Appeal to fear-prejudice	23.66	26.87	24.02 ± 2.67
- Bandwagon	0.00	0.00	0.00 ± 0.00
- Black-and-White Fallacy	8.16	14.97	10.20 ± 2.48
- Causal Oversimplification	4.11	8.02	5.14 ± 1.91
- Doubt	7.02	20.89	10.01 ± 1.42
- Exaggeration, Minimisation	18.16	24.25	20.16 ± 2.12
- Flag-Waving	31.66	49.44	37.83 ± 3.44
- Loaded Language	28.26	47.64	34.79 ± 3.16
- Name Calling	23.74	37.70	28.58 ± 1.63
- Obfuscation, Vagueness, Confusion	0.00	0.00	0.00 ± 0.00
- Red Herring	2.26	1.33	1.65 ± 3.32
- Reductio ad hitlerum	16.81	17.73	15.88 ± 4.26
- Repetition	9.82	6.75	7.23 ± 1.76
- Slogans	31.20	32.68	31.42 ± 1.95
- Straw Men	0.00	0.00	0.00 ± 0.00
- Thought-terminating Cliches	3.82	11.07	5.36 ± 2.69
- Whataboutism	13.42	4.73	5.56 ± 3.67
Multi-task learning			
- Appeal to Authority	5.57	0.95	1.50 ± 1.02
- Appeal to fear-prejudice	24.58	24.71	24.44 ± 2.13
- Bandwagon	0.00	0.00	0.00 ± 0.00
- Black-and-White Fallacy	9.05	11.47	9.72 ± 3.44
- Causal Oversimplification	5.39	8.42	6.35 ± 1.88
- Doubt	7.88	15.46	10.31 ± 1.16
- Exaggeration, Minimisation	19.82	21.13	20.29 ± 1.06
- Flag-Waving	34.16	46.40	39.16 ± 1.76
- Loaded Language	29.51	43.40	34.96 ± 0.95
- Name Calling	25.73	34.44	29.36 ± 1.52
- Obfuscation, Vagueness, Confusion	0.00	0.00	0.00 ± 0.00
- Red Herring	1.61	2.00	1.59 ± 2.77
- Reductio ad hitlerum	11.78	16.85	13.59 ± 3.71
- Repetition	10.03	4.54	6.12 ± 1.98
- Slogans	35.35	31.94	32.74 ± 5.45
- Straw Men	0.00	0.00	0.00 ± 0.00
- Thought-terminating Cliches	6.18	13.57	8.32 ± 3.72
- Whataboutism	12.12	3.99	5.89 ± 3.62

Table 8: Performance on propaganda technique identification in news articles by multi-label models on every technique. The highest performance for each metric is in bold.

Model / Propagandist Technique	P	R	F1
Single-task learning			
- Appeal to authority	61.76	45.29	49.67 ± 9.13
- Appeal to fear/prejudice	16.17	6.82	9.05 ± 7.45
- Bandwagon	0.00	0.00	0.00 ± 0.00
- Black-and-white Fallacy	67.70	30.70	41.55 ± 4.01
- Causal Oversimplification	12.20	8.70	8.61 ± 7.63
- Doubt	45.90	13.95	20.98 ± 6.92
- Exaggeration/Minimisation	44.71	35.90	39.44 ± 3.43
- Flag-waving	52.13	35.18	40.88 ± 8.91
- Glittering generalities (Virtue)	33.83	6.74	9.92 ± 7.09
- Loaded Language	60.48	68.93	64.39 ± 1.95
- Name calling	52.90	55.88	54.21 ± 3.23
- Obfuscation, Vagueness, Confusion	0.00	0.00	0.00 ± 0.00
- Red Herring	0.00	0.00	0.00 ± 0.00
- Reductio ad hitlerum	0.00	0.00	0.00 ± 0.00
- Repetition	0.00	0.00	0.00 ± 0.00
- Slogans	32.74	25.58	27.75 ± 6.00
- Smears	30.57	37.19	33.13 ± 2.43
- Straw Man	0.00	0.00	0.00 ± 0.00
- Thought-terminating cliché	23.33	10.16	13.43 ± 10.76
- Whataboutism	21.69	26.25	22.98 ± 6.57
Multi-task learning			
- Appeal to authority	51.53	50.93	49.52 ± 6.90
- Appeal to fear/prejudice	11.81	6.80	7.96 ± 5.79
- Bandwagon	0.00	0.00	0.00 ± 0.00
- Black-and-white Fallacy	42.31	28.17	33.23 ± 5.65
- Causal Oversimplification	13.55	19.10	12.74 ± 7.71
- Doubt	43.46	22.48	28.60 ± 6.31
- Exaggeration/Minimisation	37.45	39.07	37.40 ± 5.86
- Flag-waving	45.99	52.85	48.17 ± 6.56
- Glittering generalities (Virtue)	32.43	11.59	16.07 ± 6.80
- Loaded Language	56.68	68.82	61.51 ± 2.90
- Name calling	52.49	57.49	54.50 ± 1.87
- Obfuscation, Vagueness, Confusion	0.00	0.00	0.00 ± 0.00
- Red Herring	0.00	0.00	0.00 ± 0.00
- Reductio ad hitlerum	0.00	0.00	0.00 ± 0.00
- Repetition	10.42	12.50	11.25 ± 19.65
- Slogans	32.49	27.27	28.87 ± 5.63
- Smears	31.49	34.70	32.28 ± 3.52
- Straw Man	0.00	0.00	0.00 ± 0.00
- Thought-terminating cliché	12.87	5.42	6.72 ± 6.86
- Whataboutism	23.49	25.89	22.49 ± 6.66

Table 9: Performance on propaganda technique identification in memes by multi-label models on every technique. The highest performance for each metric is in bold.

Dataset / Model	P	R	F1
News			
- STL Multi-label	24.92	29.26	26.18 ± 1.87
- MTL Multi-label	26.41	26.59	26.39 ± 0.78
- STL Loaded Language	39.03	50.23	43.80 ± 0.77
- MTL Loaded Language	40.20	48.40	43.70 ± 0.99
- STL Name-Calling	30.32	38.29	33.67 ± 0.88
- MTL Name-Calling	30.96	38.68	34.08 ± 0.62
Memes			
- STL Multi-label	57.84	54.83	56.23 ± 0.79
- MTL Multi-label	54.75	56.68	55.59 ± 1.20
- STL Loaded Language	77.13	73.07	74.84 ± 1.67
- MTL Loaded Language	71.89	75.62	73.64 ± 1.77
- STL Name-Calling	71.71	69.33	70.32 ± 1.91
- MTL Name-Calling	74.56	71.66	72.88 ± 1.65

Table 10: Performance on the validation set for propaganda technique identification.

Task	Parameter	Values
All	dropout	0.0
	LR scheduler	cosine
	warmup	10%
	weight decay	0.01
News - Multi label	batch size	8, 16, 32
	learning rate	1e-5, 3e-5, 4e-5, 5e-5
	max epochs	35
	patience	7
	task sampling ratio *	(1/6, 5/6), (1/5, 4/5), (1/4, 3/4), (1/3, 2/3), (1/2, 1/2), (2/3, 1/3)
epoch factor *	(0.95, 1.0), (0.96, 1.0), (0.97, 1.0), (0.98, 1.0), (0.99, 1.0), (1.0, 1.0)	
News - Name calling	batch size	16, 32
	learning rate	5e-6, 1e-5 , 3e-5, 5e-5
	task sampling ratio *	(1/4, 3/4), (1/3, 2/3), (1/2, 1/2)
	epoch factor *	(0.99, 1.0), (1.0, 1.0)
News - Loaded language	batch size	16, 32
	learning rate	5e-6, 1e-5 , 3e-5, 5e-5
	task sampling ratio *	(1/4, 3/4), (1/3, 2/3), (1/2, 1/2)
	epoch factor *	(0.99, 1.0), (1.0, 1.0)
Memes - Multi label	batch size	8 , 16, 32
	learning rate	1e-5, 3e-5, 4e-5, 5e-5
	max epochs	150
	patience	50
	epoch factor *	(0.98, 1.0), (0.99, 1.0), (0.995, 1.0), (1.0, 1.0)
task sampling ratio *	(1/4, 3/4), (1/3, 2/3), (1/2, 1/2), (6/10, 4/10), (7/10, 3/10)	
loss scaling *	(3/4, 1), (1, 1)	
Memes - Name calling	batch size	8, 16, 32
	learning rate	1e-5, 3e-5, 4e-5 , 5e-5
	task sampling ratio *	(1/5, 4/5), (1/4, 3/4), (1/3, 2/3), (1/2, 1/2), (6/10, 4/10)
	epoch factor *	(0.98, 1), (0.99, 1.0), (0.995, 1), (1.0, 1.0)
	loss scaling *	(3/4, 1), (1, 1), (5/4, 1), (1, 5/4), (1, 3/2)
Memes - Loaded language	batch size	8, 16 , 32
	learning rate	1e-5, 2e-5, 3e-5, 4e-5 , 5e-5
	task sampling ratio *	(1/5, 4/5), (1/4, 3/4), (1/3, 2/3), (1/2, 1/2), (6/10, 4/10), (7/10, 3/10), (4/5, 1/5)
	epoch factor *	(0.98, 1.0), (0.99, 1.0), (0.995, 1.0), (1.0, 1.0)
	loss scaling *	(3/4, 1), (1, 1)

Table 11: Best performance parameters after five runs are in bold. Multi-task parameters are identified with an asterisk, and their values belong to the auxiliary and main tasks.

Technique	Example
Appeal to authority	"... information released by investigative reporter Laura Loomer proves that authorities have directly lied to the American people about the case at least once ...
Appeal to fear	"... students told her daughter that she was going to hell .
Bandwagon	"... the likelihood that this disease will move to other more densely populated regions of the planet has become a huge concern for many .
Black-and-white Fallacy	Either you stand with BDS, Hamas, blood libels and those who want to destroy Israel or with Jews.
Causal Oversimplification	On the other hand, it knows that by seeking continued secrecy, it's essentially an implicit acknowledgment of guilt.
Doubt	What happened during the 6 minutes between Campos being shot and Paddock opening fire, and why weren't the police rushing to the scene immediately?
Exaggeration/Minimisation	Whatever definition that one might put on that nebulous term, no reasonable person can honestly believe that the release of 50-year-old records are going to result in the United States falling into the ocean or even that the communists are going to take over the federal government.
Flag-waving	"I want to get our soldiers out. I want to bring our soldiers back home," Trump said.
Glittering generalities	"... to show the enormous, enthusiastic crowd in front of him .
Loaded Language	On both of their blogs the pair called their bans from entering the UK "a striking blow against freedom" and said the "the nation that gave the world the Magna Carta is dead" .
Straw Man	His opinion is: "Take it seriously, but with a large grain of salt." Which is just Allen's more nuanced way of saying: "Don't believe it."
Name-calling	"It's embarrassing for this so-called land of democracy and freedom of speech ," he said.
Obfuscation	Accordingly, he rushed to the defense of Bergoglio and his corrupt regime against "a radicalization of religious conservatism in the neo-traditionalism sense...
Red Herring	"The jury of six men and six women, including three immigrants , found the Mexican national not guilty ...
Reductio ad Hitlerum	Exactly what this "special need" is that can constitute a Gestapo like police state surveilling its own citizens is a moving target that has already been proven to be abused over and over again.
Repetition	Take notice , Dutch Prime Minister Rutte. Take notice , Mrs. Merkel or President Macron. Take notice : the future is ours and not yours
Slogans	Christianity is Europe's last hope.
Smears	No honor, no integrity, no principles, no morals, ...
Thought-terminating cliché	This whole idea of a two-state solution, it doesn't work.
Whataboutism	"They interpreted the law in my case to say it was criminal," Saucier told Fox News, referring to prosecuting authorities in his case, "but they didn't prosecute Hillary Clinton.

Table 12: Examples of persuasion techniques are in bold.

Lexical Semantics with Large Language Models: A Case Study of English *break**

Erika Petersen
Stanford University
epetsen@stanford.edu

Christopher Potts
Stanford University
cgpotts@stanford.edu

Abstract

Large neural language models (LLMs) can be powerful tools for research in lexical semantics. We illustrate this potential using the English verb *break*, which has numerous senses and appears in a wide range of syntactic frames. We show that LLMs capture known sense distinctions and can be used to identify informative new sense combinations for further analysis. More generally, we argue that LLMs are aligned with lexical semantic theories in providing high-dimensional, contextually modulated representations, but LLMs’ lack of discrete features and dependence on usage-based data offer a genuinely new perspective on traditional problems in lexical semantics.

1 Introduction

[Pater \(2019\)](#) builds a compelling case that linguistic and neural network research have great potential for common ground and common cause. His case has only grown stronger in recent years, with the arrival of large neural language models (LLMs) that provide semantically rich, contextual representations ([McCann et al., 2017](#); [Peters et al., 2018](#); [Radford et al., 2018](#); [Devlin et al., 2019](#)).

In this paper, we argue that LLMs are powerful devices for studying lexical semantics in ways that can deeply inform linguistic theory. We illustrate this with a detailed case study of the lexical semantics of the English verb *break*, building on a richly annotated dataset from [Petersen \(2020\)](#) and drawing on methods from prior work in this area ([Camacho-Collados and Pilehvar, 2018](#); [Tenney et al., 2019](#); [Garí Soler et al., 2019](#); [Reif et al., 2019](#); [Wiedemann et al., 2019](#); [Branco et al., 2020](#); [Nair et al., 2020](#); [Li and Joannis, 2021](#); [Loureiro et al., 2021](#); [Trott and Bergen, 2021](#); [Apidianaki, 2022](#); [McCrae et al., 2022](#)). *Break* has long been

central to theoretical work in lexical semantics because it has a staggering range of senses that appear to be systematically related to its argument structure. Our central empirical finding is that LLM representations capture many of these known sense distinctions and can be used to identify new sense combinations for further analysis.

We use these findings as a chance to reflect on the core theoretical commitments of lexical semantics as they pertain to LLM-based investigations. Our discussion is centered around the three tenets of lexical semantics given in [Table 1](#): lexical representations are *high dimensional*, *contextually modulated*, and include *discrete features*.

The high dimensionality property is not phrased as a direct claim in the literature as far as we know, but it reflects the practice of linguists, who identify numerous interacting features of lexical items. [Section 2](#) offers a summary picture for *break*. Similarly, discreteness is often assumed by linguists working in the broadly generative tradition. For our purposes, the key question is whether there are *any* features that are discrete, since LLMs do not naturally support having such features.

Contextual modulation is a direct claim. We trace the origins to [Dowty \(1976, 1979\)](#), who argues that aspectual analyses need to include at least the entire verb phrase (see also [Kratzer 1996](#)). [Borer \(2005a,b, 2013\)](#) pushes this further, arguing that open-class lexical items are “tantamount to raw material, ‘stuff’ which is poured into the structural mould to be assigned grammatical properties” (2005a, p. 108). On this view, lexical items are mostly unvalued discrete feature representations that are fleshed out and modulated by the environment in which they appear; there may be a stock of identifiable lexical items, but they are highly abstract, with almost unlimited potential to become different items in different contexts.

A similar view is taken by work in the Generative Lexicon ([Pustejovsky, 1991, 1995](#)), which posits

*Data and code available at <https://github.com/epetsen/break-llms>.

	Linguistics	Static vectors	LLMs
High dimensionality: Lexical semantic entries consist of many features.	Yes	Yes	Yes
Contextual modulation: A word sense will be influenced by its immediate morphosyntactic context as well as the broader context of use.	Yes	No	Yes
Discreteness: The features in lexical semantic entries are discrete and highly structured.	Yes	No	No

Table 1: Core tenets. Our focus is in particular on the relationship between ‘Linguistics’ and ‘LLMs’ in this table.

an extensible lexicon that is “open-ended in nature and accounts for the novel, creative, uses of words in a variety of contexts by positing procedures for generating semantic expressions for words on the basis of particular contexts” (Pustejovsky, 2006). This also aligns with Clark’s (1997) rejection of the “Dogma of Sense Selection”, which says “Listeners determine an enumerable set of senses for each expression, and in understanding what a speaker means, they select the appropriate sense from that set.” For Clark, lexical items are highly malleable and constrained mainly by what the discourse participants can reliably communicate with each other (see also Clark and Clark 1979; Searle 1980). On all these views, lexical items are highly abstract objects that can be realized in very diverse ways.

The field of NLP has a complex relation to our tenets. Early work on symbolic grammars in NLP was clearly aligned with all the tenets. The Generative Lexicon is a prominent example and proved influential in linguistics and NLP. When distributional methods first became central to NLP, the dominant mode of lexical representation involved static vector representations. These representations align with the consensus in linguistics only regarding high dimensionality, as we discuss in Section 4.

LLMs have changed NLP’s relationship to lexical semantics considerably. With LLMs, we have a strong commitment to high dimensionality and contextual modulation and a denial of discreteness (Section 5). The points of agreement present a significant opportunity for linguists and NLP researchers to collaborate, as we hope our case study shows. The points of disagreement seem also to be opportunities for people to take new perspectives. We argue in particular that the facts surrounding *break* should lead linguists to reconsider their commitment to discreteness and embrace a more fluid, usage-based foundation for semantic theory.

2 English *Break*

English *break* is one of the best studied lexical items in lexical semantics, for a few reasons. First, it is a canonical instance of a change-of-state verb that undergoes the causative alternation:

- (1) The linguist broke the window
- (2) The window broke.

In fact, alternating change-of-state verbs are referred to as *break*-verbs (Acedo-Matellán and Mateu, 2014; Fillmore, 1970; Levin, 2017; Majid et al., 2008). The intransitive variant of the causative alternation (2) is analyzed in terms of the *unaccusativity hypothesis* (Perlmutter, 1978; Burzio, 1986; Levin and Rappaport Hovav, 1995), which says that the subjects in these cases are underlyingly internal arguments to the verb, bearing more theme-like semantic roles, and have been promoted to subject position to fulfill a subjecthood requirement. Research on *break* has also contributed to the study of the lexical properties of unaccusative verbs (Levin and Rappaport Hovav, 1995).

Second, *break* can take on a wide array of senses. Table 2 provides a partial list; we cannot hope to be comprehensive (there may not even be a fixed stock of senses; Section 5), but our examples convey the nature of the attested variation.

Third, the sense distinctions interact with the causative alternation. Whereas senses 1–4 all alternate, senses 5–11 are all strictly transitive. The non-alternating senses of *break* have informed the debate about which variant of the causative alternation (if any) is basic and which is derived (e.g. Levin and Rappaport Hovav, 1995; Alexiadou et al., 2006; Piñón, 2001). Though the debate is still unsettled, it has evinced that participation in the causative alternation is not a property of the verb itself, but of the verb in combination with its theme argument (Petersen, 2020; Spalek, 2012), just as

Frame	Sense	Frame	Sense
1. break the vase	shatter	14. break off the engagement	end
2. break the computer	render inoperable	15. break out	begin
3. break the news	reveal	16. break out of jail	escape
4. break the silence	interrupt	17. break out in hives	get
5. break the record	surpass	18. break into the building	intrude
6. break the code	decipher	19. break down the problem	analyze
7. break the law	violate	20. break down the proteins	decompose
8. break the habit	end	21. break in	enter
9. break the horse	tame	22. break in	interrupt
10. break a \$10 bill	make change	23. break free	escape
11. break the fall	lessen	24. break even	profit = loss
12. the weather broke	changed	25. break forth	emerge
13. the day broke	began	26. break to the right	turn

(a) Uses without particles/predicates.

(b) Uses with particles/predicates.

Table 2: Senses for *break*. A comprehensive account of senses may not be possible (Section 5.3).

with telicity and other aspectual properties (Dowty, 1976, 1979; Borer, 2005b).

Prior work has sought to capture the obligatorily transitive nature of some of these senses by appeal to a thematic role requirement: *break* in combination with its internal argument determines the range of semantic roles – agent, instrument, or natural force – that the subject of a transitive *break* frame may bear (Rappaport Hovav and Levin, 2012), and some frames require their subjects to be agentive (Levin and Rappaport Hovav, 1995; Piñón, 2001; Alexiadou et al., 2006; Schäfer, 2008). Since necessarily agentive subjects cannot be left unexpressed, these frames do not show intransitive variants. However, this cannot be the full story, as there are some obligatorily transitive *break* frames where the subject need not be an agent but which nonetheless do not alternate, like *the cushion broke her fall* vs. **the fall broke* (Petersen, 2020).

In addition, examples like *break the record* (sense 5) and *break the code* (sense 6) may be graded or uncertain in regard to their participation in the causative alternation. They are often assumed not to have intransitive uses (Levin and Rappaport Hovav, 1995; Piñón, 2001; Alexiadou et al., 2006; Schäfer, 2008; Rappaport Hovav and Levin, 2012), but there are attested cases like the following that suggest this is a point of variation.

- (3) Almost sixty years later, Frank Rowlett, a cryptologic pioneer and head of the “Purple” team, remembered that historic day when the code broke.

- (4) The Guinness World Record broke, our furniture didn’t.

There are also strictly intransitive uses, as in 12–13 of Table 2a. These are analyzed in the same way as the intransitive variant of alternating frames (2), i.e., as unaccusatives. Why these *break* frames do not allow a cause subject – e.g., **the Earth’s rotation broke the day* – is an open question.

As seen in Table 2b, *break* also combines with a wide range of predicates and particles to create new senses. Except for 14, 19, and 20, these uses are all intransitive, but they seem to differ from the particle-less uses in a key way: whereas intransitive particle-less *break* cases are all unaccusative, the particle cases vary in this regard. For example, *the war broke out* seems unaccusative, but *we broke into the building* has an agentive subject and so would not be analyzed as unaccusative.

A key question for lexical semantic theories is whether there is a single unifying semantic frame underlying this diverse array of senses – or, if not a single frame, then perhaps a few of them feeding into distinct sense clusters. This position is advanced, for example, by Kellerman (1978:65), for whom “[t]he various meanings of BREAK [...] can all be subsumed under a ‘deep’ meaning, ‘(cause) not to continue in existing state’, which links even the most disparate meanings of BREAK’ (see also Spalek 2012 for a similar position for Spanish *romper* ‘break’). Another approach would be to posit a few more primitive semantic dimensions that give rise to a combinatorial space of

	<i>Transitive</i>	<i>Unaccusative</i>	<i>Agent</i>	<i>Metaphorical</i>	<i>separate</i>	<i>violate</i>	<i>end</i>	<i>appear</i>	<i>out_escape</i>	<i>out_begin</i>
1. We broke the vase	1	0	1	0	1	0	0	0	0	0
2. The vase broke	0	1	0	0	1	0	0	0	0	0
3. We broke the law	1	0	1	1	0	1	0	0	0	0
4. The silence broke a procedural rule	1	0	0	1	0	1	0	0	0	0
5. We broke the silence	1	0	1	1	0	0	1	0	0	0
6. The day broke	0	1	0	1	0	0	0	1	0	0
7. The storm broke	0	1	0	1	0	0	0	1	0	0
8. Sweat broke on his forehead	0	1	0	1	1	0	0	1	0	0
9. We broke out (of jail)	0	0	1	0	0	0	0	0	1	0
10. Fighting broke out	0	1	0	1	0	0	0	0	0	1

Table 3: Partial feature-based analysis of *break* in different syntactic contexts.

predicted senses, which might in turn lead to predictions about argument structure realization and other structural and distributional properties.

3 Feature-based Theories

In this section, we take the somewhat unusual step of bringing together existing ideas from the linguistics literature into a feature space of the sort one is likely to encounter in NLP contexts. We do this for a few reasons. First, it reveals that, though theories in linguistics and NLP often take very different forms, there is actually a lot of common ground between them: on both sides, vector representations of data can serve as a common language. Second, the feature space reveals how deeply linguistic theories are committed to our contextual modulation tenet from Table 1: to honor the insights from the literature, we have to define the feature space in terms of (at least) full sentences.

Table 3 is our (highly partial) feature-based analysis. The Transitive feature captures whether a particular *break* frame has two nominal arguments or one. Causative alternation uses can then be reconstructed by looking at shared meaning dimensions that vary in their Transitive value, as in rows 1–2. We separately define an Unaccusative feature, since the uses in Table 2b show that these can come apart. This is evident especially in rows 9–10.

The Agent feature captures whether the subject of each example is agentive or not. We mentioned in Section 2 that the obligatory transitivity of some *break* frames has been traced, unsuccessfully in our view, to the agentivity of the subject of these

frames. The Agent feature in combination with the Transitive feature and the meaning dimensions reveals the incompleteness of this explanation: ‘violate’ examples, which are obligatorily transitive, may have subjects that are agentive (row 3) and non-agentive (row 4).

We have a column for Metaphorical, though coding this is sufficiently hard that it looks like a multidimensional category to us rather than a single feature. Due to the difficulty of classifying senses of *break* and other polysemous verbs as (non)metaphorical, previous literature that has engaged with this question (e.g. Kellerman, 1978; Piñón, 2001; McNally and Spalek, 2017, 2022) has used the heuristic of associating metaphorical senses with abstract participants, like *break the silence*, and non-metaphorical ones with concrete participants, such as *break the vase*. We follow this (admittedly simplifying) heuristic in our feature-based analysis. However, we agree with McNally and Spalek (2022, 6) that “the distinction between ‘literal’ and ‘figurative’ senses can become blurred over time, and sometimes can only be diachronically reconstructed”.

Following these features are a few meaning dimensions. The full class of meaning annotations we use in Section 5.3 has 72 classes, so this is just a sample. The sample was chosen to emphasize three aspects of the meanings of *break*. First, as already illustrated in Table 2, these meanings are highly diverse semantically. Second, it is difficult (and maybe even futile) to determine with confidence how many distinct (and non-overlapping) senses

1. break	11. up
2. breaks	12. trying
3. breaking	13. away
4. end	14. start
5. broke	15. get
6. down	16. again
7. take	17. 'll
8. let	18. back
9. going	19. out
10. leave	20. off

(a) GloVe, Common Crawl 840B tokens, 300d.

1. breaks	11. brief_respite
2. breaking	12. Nadal_netted_forehand
3. broke	13. loosen
4. broken	14. smash
5. Break	15. rip
6. Breaking	16. overhit_forehand
7. breather	17. miscued_forehand
8. shatter	18. cut
9. crack	19. slip
10. breaker	20. Breaks

(b) word2vec, GoogleNews, 300d.

1. break	11. breakin
2. breaks	12. breaked
3. breaking	13. broken
4. breake	14. legbreak
5. re-break	15. reak
6. break-	16. semi-break
7. unbreak	17. minibreak
8. breakes	18. breaker
9. break.	19. breaking-down
10. broke	20. tea-break

(c) fastText WikiNews, with sub-word modeling, 300d.

Table 4: Nearest neighbors of *break* in static embedding spaces. All the methods place morphological variants of *break* next to *break* itself and seem to sporadically find different senses and near synonyms of *break*. The lists given here are, in our judgment, the best from each of the three methods. For additional lists, see Appendix B.

break may express. For example, does *break* express the same meaning, ‘appear’, in row 6 as in row 7, as we suggest in Table 3? Or should these examples be seen as expressing distinct senses? Third, we believe there are some examples where *break* simultaneously expresses more than one meaning, as shown in row 8 of Table 3, where *break* shows both an ‘appear’ and a ‘separate’ meaning.

Break with particles/predicates can sometimes express meanings that particle-less *break* cannot convey: e.g. ‘escape’ in row 9. We assume that the particles/predicates contribute an irreducible meaning and reflect this in our feature-based analysis by preceding these meaning dimensions with the corresponding particles/predicates: ‘out_escape’. The particles/predicates do not determine a unique meaning, though, as we see with the two senses of *break out*: ‘out_escape’ and ‘out_begin’. And we could of course have extended this even further. There are additional clearly distinct senses like *His face broke out in hives* and *break out the champagne*, as well as cases like *break out in laughter* which might be subsumed under other senses (say, ‘out_begin’).

Potentially all of the columns are actually just informal stand-ins for much more complex concepts. The labels could be natural language predicates on par with *break*, in which case the column names are really just hooks into a larger lexical web, or they could be glosses for more intricate theoretical concepts that demand further decomposition before the theory can be regarded as complete.

In addition, we can seamlessly integrate this kind of analysis with more data-driven techniques. As an illustration, Appendix A reports on an experiment using the WordNet hypernym graph to identify extremely abstract latent meaning dimensions.

How many lexical items does this theory posit? The answer to this question is not clear. We could say that each attested combination of the features is a new sense, or we could select a few features and say that specific combinations of them correspond to distinct senses. Both decisions have a certain arbitrariness to them given the feature space itself, and we might infer from this that the theory does not posit distinct senses or distinct lexical items as first-class linguistic constructs. This may be a consequence of the contextual modulation tenet.

Relatedly, it is unclear to us what a *complete* analysis in these terms would look like. What would it mean to have determined all and only the correct features? Could it be that the investigation will always admit of further dimensions, or decomposition of existing dimensions?

In sum, it is easy to see how this analysis makes good on the central tenets in Table 1. The representations are high-dimensional vectors with discrete values. In addition, the representations themselves directly bring in context. The vector for *break* alone, if it exists in the theory at all, needs to be mostly unspecified values that only become values in specific syntactic or usage contexts.

4 Static Vector Modeling

The above feature-based analysis might be described as a *sparse vector representation* approach. We now contrast that with a dense vector representation approach that models individual lexical items as fixed (static) vectors. A variety of such methods have been developed. Here we look at the treatment of *break* by three prominent methods: word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and fastText (Mikolov et al., 2018). These methods have different learning objectives, but all

are closely related to Pointwise Mutual Information (PMI; Church and Hanks 1990; Turney 2001). In PMI, we assign weights to pairs of words w_i and w_j based on whether their observed joint probability of co-occurrence is larger or smaller than what we would expect given the null hypothesis that w_i and w_j have independent distributions. All three methods learn regularized, reduced dimensional representations according to roughly this same goal (Levy and Goldberg, 2014; Cotterell et al., 2017).

None of these methods use discrete features, and thus they are in conflict with our discreteness tenet from Table 1. The raw input to all of them is a matrix of co-occurrence counts, which could be viewed as a set of discrete distributional features. However, much of the power of these models derives from their ability to compress this information into a lower-dimensional space of continuous values in which the columns are unlikely to have direct interpretations as features.

The GloVe vocabulary is largely restricted to individual words from a fixed list. By contrast, the vocabulary used for our word2vec instance includes some phrase-like elements that were inferred by the authors using simple co-occurrence statistics, and our chosen fastText model includes sub-word components and so also ends up with a more expansive view of what counts as a lexical item.

All of these models have proven successful as representations of words and as components in larger systems. However, we find that these representations are disappointing for studying *break*. In Table 4, we show the top 20 nearest neighbors (according to cosine similarity) for some uses of these models. We chose what seemed to be the semantically richest instance of each model from a larger set of such results (see Appendix B). All of the models capture morphological variants very clearly. However, the other semantic associations generally only weakly indicate other specific senses (via associations with other words). We do see some positive benefits from the quasi-phrasal vocabulary used by word2vec and fastText, but overall these spaces look like only superficial pictures of the underlying semantic richness of *break*.

The cause for this semantic blandness likely traces to the basic design decision: every word-form has only a single representation. This means that a single vector must encode all the different senses that we see in Table 2 as well as others that we did not include there. The result is probably

something like a weighted average of these senses, which seems not to be in a particularly interesting part of the embedding space.

Adherents to our central tenets (Table 1) might have predicted this negative result. While static representations are high dimensional, they do not allow for contextual modulation. Each basic unit of the vocabulary is assigned exactly one representation. Contextual modulation may occur if the representations are embedded in a larger system, but it is not intrinsic to the vectors themselves.

5 LLM Investigations

We come now to our primary investigative tool: LLMs. We concentrate on models that have the core structure of the Transformer (Vaswani et al., 2017) and are trained at least in part using masked language modeling, which allows for bidirectional context. In the interest of space, we will mostly presuppose familiarity with these models. However, Appendix C provides an overview of their structure to try to bridge any gaps between the linguistics and NLP literature.

In our main text, we report results for RoBERTa-large (Liu et al., 2019), which has 24 layers. Our appendices cover BERT and DeBERTa. Our RoBERTa-large results are slightly better than all of these others, but the results are generally quite comparable, suggesting that all of these models can fruitfully be used for lexical semantics.

Before turning to our experiments, let's consider how LLMs relate to our core tenets from Table 1. First, the representations we obtain at each hidden layer are all high-dimensional and modulated by the context. Our experiments show that, for the case of *break*, this contextual modulation is rich and linguistically systematic. Thus, LLMs and traditional lexical semantic theories are aligned on these two tenets. However, the two theories part ways when it comes to the question of having discrete features. The column dimensions of LLM representations are continuous and highly abstract. Discrete linguistic features might be latently encoded in these representations, or extractable from them with some noise, but this does not detract from the fact that these representations are highly fluid and do not presuppose the existence of any particular features or dimensions. Rather, all the features are learned from data in a free-form way that is grounded entirely in distributions.

Layer	Probe	Control	Selectivity
1	0.33	0.03	0.30
6	0.81	0.03	0.79
12	0.83	0.03	0.80
18	0.80	0.03	0.76
24	0.86	0.03	0.83

(a) Meaning-class probing results.

Layer	Probe	Control	Selectivity
1	0.50	0.33	0.17
6	0.94	0.34	0.60
12	0.96	0.33	0.63
18	0.96	0.35	0.61
24	0.97	0.32	0.65

(b) Construction-type probing results.

Table 5: RoBERTa-large probing results. We report Macro F1 and Selectivity, which is the Macro F1 score for the task minus the Macro F1 for a control task (random assignment of tokens to classes). Results for other models are similar; see Appendix D.

5.1 Annotated Dataset

The basis for our investigation is an annotated dataset created by Petersen (2020) and subsequently updated by us to include more examples and senses. The examples are extracted from the Corpus of Contemporary American English (CoCA; Davies 2008). We focus on a subset of 1,042 sentences that have been annotated for, among other things, the core semantic class of the reading and the construction type (‘unergative’, ‘unaccusative’, ‘causative’). Petersen assigns a single semantic class to each example. However, as mentioned in Section 3, we believe that, in some cases, *break* can be said to simultaneously express more than one meaning. We use our experiment in Section 5.3 to identify examples with this property.

We rely primarily on the meaning class distinctions and make secondary use of the constructional annotations. Petersen’s annotation scheme uses 72 semantically rich meaning classes, which have a highly skewed distribution. The full distribution is given in Appendix E.

5.2 Probing Experiments

We want to explore the LLM representations in a fluid way that will lead us to identify new readings. Our tools for doing this are supervised probe models applied to the column of representations above the *break* token in each of our examples. We probe for meaning-class and construction-type (see also Papadimitriou et al. 2021). These probes serve as a quantitative evaluation of the extent to which these *break* representations encode these important properties, and they are also tools for heuristically finding new uses and readings.

For our construction-type probing work, we can use all 1,042 sentences, since there are only three classes and all have substantial representation in the

data (causative: 673 examples, unaccusative: 197, unergative: 172). For the meaning-type work, there are 72 classes, many with only a few instances. Thus, we limit attention to just the classes with at least 10 examples (Appendix E).

Our probe models are L2-regularized classifiers with a cross-entropy loss. Our core metric is the macro F1 score, which assigns equal weight to each class’s F1 score regardless of the class size. Following Hewitt and Liang (2019), we report *selectivity* scores, which are the probe scores minus the performance on a control task, which here is random assignment of *break* representations to meaning classes. We report selectivity scores averaged across 20 random 80%/20% train/test splits.

The probe results show a clear pattern: the lowest layers are not very robust when it comes to this probing work, but higher layers are very robust in this sense (see also Reif et al. 2019; Ethayarajh 2019). We see similar results for other LLMs in the class we are focused on, as reported in Appendix D. For this reason, we focus on layer 24 of RoBERTa-large from now on.

5.3 Discovering New Example Types

Our primary goal is to see whether it is possible to use LLMs to gain new insights about lexical semantics. Our probing results suggest that LLM representations are systematic enough to make this plausible, but they are very high-level. We need an investigative technique that is more free-form and that can bring to our attention new kinds of theory-relevant examples.

A natural choice is visualization. We provide t-SNE visualizations (van der Maaten and Hinton, 2008) in Appendix G, and we find that they are indeed useful: where an example of meaning class *a* is nestled among examples of class *b*, the *a*-class example is often an interesting blend of *a*-class

Sentence	Meaning		Construction	
	Gold	Predicted	Gold	Predicted
1. Patients will sometimes break out in a spontaneous recitation of the rosary	break_out_start	break_out_start	unacc.	unerg.
2. It was like you knew something, like you knew the story was getting ready to break again.	reveal	appear	unacc.	unacc.
3. @(Soundbite-of-music)@!Mr-GELB: (Singing) Tell me who’s going to pick up the pieces when you start to break down.	break_down_separate_into_parts	break_down_succumb	unacc.	unacc.
4. People have so many problems overcoming the disputes that occur when families break up	break_up_end_relationship	break_up_separate_into_parts	unacc.	unacc.
5. “So why tell the whole story now? Somebody, some male, has got to be willing to break this code of silence,” he says.	violate	end	unacc.	unacc.
6. So they forwarded the pictures to Madrid, where another officer noticed some printing on a towel that helped break the case.	decipher	end	causative	causative
7. Then too, stress can also work to break down the immune system, increasing the likelihood of respiratory and creating gastrointestinal and nervous disorders.	break_down_render_inoperable	break_down_destroy	causative	causative
8. Wind, naturally acidic rain, and physical processes such as freezethaw cycles also break down rock.	break_down_separate_into_parts	break_down_destroy	causative	causative
9. It didn’t take being an ICU exec to break the code: trade secret.	decipher	violate	causative	causative

Table 6: A curated sample of theoretically informative examples.

and *b*-class meanings, and such examples seem genuinely worthwhile to study further. However, these visualizations introduce known distortions resulting from compressing high-dimensional spaces into two dimensions (Wattenberg et al., 2016), and they can even vary in qualitatively substantive ways across models and runs.

For something more stable, we return to our probe models. The selectivity scores for both are conservative if we think of them as tools for finding new examples: the meaning-class probes achieve results above 80% macro F1, and the construction-type probes are nearly perfect in their performance (where chance is around 33%). Thus, we decided to extract and review the errors made by these models, with the expectation that many of these examples could inform lexical semantic theory itself.

Table 6 is a selection of examples that we extracted in this way for further analysis. This is a small curated set of (so-called) errors, though these examples do not look like errors to us, but rather like instances in which multiple senses and multiple construction types emerge in the same example.

Example 1 is predicted unergative, whereas the gold label is unaccusative. For us, this raises a new

question: what is the role of *spontaneous* in the overall agentivity of the subject, and should this play into how we characterize the syntactic frame?

Example 2 looks like a clear case where multiple senses can be activated and different utterance contexts might favor different readings. Is the breaking of a news story an agentive act of revealing information (the gold label), or can it be (or be described as) something more like a natural process of appearing (the predicted label)? Both readings seem available, and individual uses might blend them for a particular rhetorical effect.

We also find cases where both the gold meaning and the predicted meaning are in principle available, and the choice between them depends on whether we would like to focus on the metaphorical or the literal meaning of the expression. This can be seen in examples 3 and 4.

Example 4 also reveals a common pattern we found in our examples: in many cases where multiple senses are present, there is a contextual entailment relation between them. In example 4, should we focus on the direct and perhaps metaphorical reading (gold) or the more literal likely consequence (predicted)? In example 5, violating the

code of silence entails ending it. In example 6, deciphering a case entails solving or ending it. In example 7, the process of breaking down the immune system, which we paraphrased as rendering it inoperable (gold meaning), could culminate in its destruction (predicted meaning). And example 8 reveals that the event structure of *break* examples can be very complex. When a rock is broken down by natural forces, we can think of this as a process of breaking down into smaller parts (gold meaning) with the end state being total destruction (predicted meaning). Many examples in which the breaking event leads to the fragmentation of the theme participant can show similar blends.

We also see cases where there seems to be genuine uncertainty about which of the two senses is intended. Example 9 illustrates this. Depending on the meaning of the word *code* (‘encryption’ vs. ‘norm’), *break* can either mean ‘decipher’ (gold) or ‘violate’ (predicted). This example further evinces the importance of our contextual modulation tenet: sometimes we have to go even beyond full sentences to be able to determine the meaning of *break* in a particular case.

These are just a few examples of a much larger set of interesting cases that emerge from studying the interaction between our LLM-based probe models and our linguistic annotations. Appendix F provides a larger sample with brief annotations about potential theoretical relevance. We close this paper by reflecting on how best to incorporate these insights into the linguistic theory itself.

6 Discussion

The fact that linguistic theory and LLMs agree on the core tenets of high dimensionality and contextual modulation is a striking alignment of theoretical ideas with engineering success.

The fact that LLMs do not use discrete features, but rather derive dense, real-valued representations from data seems like an opportunity for linguists to reflect on the role of discreteness. As we noted in Section 3, it seems unlikely that purely analytic work and traditional corpus work will lead to an exhaustive hand-built representation for any lexical items. With LLMs, we can mine the existing representations while considering the LLM architecture and learned parameters to be a reflection of the core tenets of the theory.

The deep contextual modulation countenanced by linguistic theory and operationalized by LLM

embeddings invites a further question: do lexical items exist outside of their tokens of use? Even for the hand-built feature representations in Table 3, the rows could in principle vary based even on usage information, which would suggest a theory that is actually more about tokens (instances of use) than types. Similarly, for LLMs, though they do contain type-level representations (in the form of an embedding for the vocabulary), these play a minor role, and all the representations we have considered in this paper were in terms of representations that are more like token-level representations.

Overall, then, a theory of lexical semantics that draws heavily on LLMs as investigative tools, and even as ways to state theoretical ideas, is likely to become more usage-based than traditional theories would assume. This could lead them to focus less on pure representation and more on what is actually communicated between people when they use language. Traditional questions are likely to take on new forms in this setting, and exciting and relevant new questions – and new pieces of evidence – are likely to arise.

7 Limitations

Our general thesis is that LLMs are valuable tools both for conducting lexical semantic analyses and for providing valuable perspectives for lexical semantic theory design in general. Although we think this thesis is widely supported by prior literature, our own case study is limited to just a partial analysis of a single verb. This creates the risk that our general conclusions may be more specific to this verb, or to English, than we would like. The prior literature inherits many of our English-only biases as well (but see [Papadimitriou et al. 2021](#)).

Our main results use RoBERTa-large, and our appendices report on parallel analyses with different versions of BERT and DeBERTa. These models share core architectural features and were optimized in largely similar ways using very large – and largely uncontrolled – datasets. This means that these artifacts are certainly biased in ways that are relevant for lexical semantics. However, we are unlikely to be able to identify, isolate, and factor out these biases with the methods used in our paper. Our core methods are reasonably simple and lightweight, and we have released all our code. We hope that these steps allow easy reproduction of our core analyses whenever newer LLMs are released, so that we can begin to understand better how LLM

biases can affect linguistic theorizing in the mode we are advocating for.

Our current approach also has an analytic limitation: though we fit probe models and use them as devices for finding potentially relevant examples, the final step in our analysis involves inspection of those examples by linguists like ourselves. This means that the final step is not as reproducible as the others, and it means that any analytic biases that the linguists involved might have are likely to make their way into the analyses. We do not see a way to avoid these analytic steps entirely, since linguistic analysis favors this kind of low-level work, but we do think that we can mitigate the concerns about analyst bias by making all our data available for others to inspect, as a way of opening up many perspectives on the data and the associated theoretical questions.

Acknowledgements

Our thanks to Dan Lassiter, Beth Levin, Isabel Papadimitriou and participants at the DistCurate 2022 workshop for helpful discussion.

References

- Víctor Acedo-Matellán and Jaume Mateu. 2014. From syntax to roots: A syntactic approach to root interpretation. In Artemis Alexiadou, Hagit Borer, and Florian Schäfer, editors, *The Syntax of Roots and the Roots of Syntax*, pages 14–32. Oxford University Press, Oxford.
- Artemis Alexiadou, Elena Anagnostopoulou, and Florian Schäfer. 2006. The properties of anticausatives crosslinguistically. In Mara Frascarelli, editor, *Phases of Interpretation*, pages 187–211. Mouton de Gruyter, Berlin.
- Marianna Apidianaki. 2022. [From word types to tokens and back: A survey of approaches to word meaning representation and interpretation](#). *Computational Linguistics*, pages 1–60.
- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. [The Berkeley FrameNet project](#). In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.
- Collin F. Baker and Hiroaki Sato. 2003. [The FrameNet data and software](#). In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 161–164, Sapporo, Japan. Association for Computational Linguistics.
- Hagit Borer. 2005a. *In Name Only*, volume 1 of *Structuring Sense*. Oxford University Press.
- Hagit Borer. 2005b. *The Normal Course of Events*, volume 2 of *Structuring Sense*. Oxford University Press.
- Hagit Borer. 2013. *Taking Form*, volume 3 of *Structuring Sense*. Oxford University Press.
- António Branco, João Rodrigues, Małgorzata Salawa, Ruben Branco, and Chakaveh Saedi. 2020. [Comparative probing of lexical semantic theories for cognitive plausibility and technological usefulness](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4004–4019, Barcelona. International Committee on Computational Linguistics.
- Luigi Burzio. 1986. *Italian Syntax*. D. Reidel Publishing Company, Dordrecht.
- José Camacho-Collados and Mohammad Taher Pilehvar. 2018. [From word to sense embeddings: A survey on vector representations of meaning](#). *Journal of Artificial Intelligence Research*, 63(1):743–788.
- Kenneth Ward Church and Patrick Hanks. 1990. [Word association norms, mutual information, and lexicography](#). *Computational Linguistics*, 16(1):22–29.
- Eve V. Clark and Herbert H. Clark. 1979. When nouns surface as verbs. *Language*, 55(4):767–811.
- Herbert H. Clark. 1997. [Dogmas of understanding](#). *Discourse Processes*, 23(3):567–59.
- Ryan Cotterell, Adam Poliak, Benjamin Van Durme, and Jason Eisner. 2017. [Explaining and generalizing skip-gram through exponential family principal component analysis](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 175–181, Valencia, Spain. Association for Computational Linguistics.
- Mark Davies. 2008. The Corpus of Contemporary American English: 450 million words, 1990–present. Available online at <http://corpus.byu.edu/coca/>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- David Dowty. 1976. Montague grammar and the lexical decomposition of causative verbs. In Barbara H. Partee, editor, *Montague Grammar*, pages 201–245. Academic Press, New York.

- David Dowty. 1979. *Word Meaning and Montague Grammar*. D. Reidel, Dordrecht.
- Kawin Ethayarajh. 2019. [How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.
- Charles Fillmore. 1970. The grammar of hitting and breaking. In R.A. Jacobs and P.S. Rosenbaum, editors, *Readings in English Transformational Grammar*, pages 120–133. Ginn, Waltham, MA.
- Aina Garí Soler, Marianna Apidianaki, and Alexandre Allauzen. 2019. [Word usage similarity estimation with sentence representations and automatic substitutes](#). In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 9–21, Minneapolis, Minnesota. Association for Computational Linguistics.
- John Hewitt and Percy Liang. 2019. [Designing and interpreting probes with control tasks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.
- Eric Kellerman. 1978. Giving learners a break: Native language intuitions as a source of predictions about transferability. *Working Papers on Bilingualism*, 15:60–92.
- Angelika Kratzer. 1996. Severing the external argument from its verb. In Johan Rooryck and Laurie Zaring, editors, *Phrase Structure and the Lexicon*, pages 109–137. Kluwer, Dordrecht.
- Beth Levin. 2017. The elasticity of verb meaning revisited. In *Proceedings of SALT 27*, pages 571–599.
- Beth Levin and Malka Rappaport Hovav. 1995. *Unaccusativity: At the Syntax–Lexical Semantics Interface*. MIT Press, Cambridge, MA.
- Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*.
- Jiangtian Li and Marc F. Joannis. 2021. [Word senses as clusters of meaning modulations: A computational model of polysemy](#). *Cognitive Science: A Multidisciplinary Journal*, 45(4):1–30.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#). ArXiv:1907.11692.
- Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. [Analysis and evaluation of language models for word sense disambiguation](#). *Computational Linguistics*, 47(2):387–443.
- Asifa Majid, James S. Boster, and Melissa Bowerman. 2008. The cross-linguistic categorization of everyday events: A study of cutting and breaking. *Cognition*, 109:235–250.
- Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. [Learned in translation: Contextualized word vectors](#). In *Advances in Neural Information Processing Systems 30*, pages 6294–6305.
- John P. McCrae, Theodor Fransen, Sina Ahmadi, Paul Buitelaar, and Koustava Goswami. 2022. [Toward an integrative approach for making sense distinctions](#). *Frontiers in Artificial Intelligence*, 5:1–18.
- Louise McNally and Alexandra Anna Spalek. 2017. ‘Figurative’ uses of verbs and grammar. Unpublished manuscript.
- Louise McNally and Alexandra Anna Spalek. 2022. Grammatically relevant aspects of meaning and verbal polysemy. *Linguistics*, pages 1–45.
- Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhresch, and Armand Joulin. 2018. [Advances in pre-training distributed word representations](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. [Distributed representations of words and phrases and their compositionality](#). In Christopher J. C. Burges, Leon Bottou, Max Welling, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.
- Sathvik Nair, Mahesh Srinivasan, and Stephen Meylan. 2020. Contextualized word embeddings encode aspects of human-like word sense knowledge. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 129–141. Association for Computational Linguistics.
- Isabel Papadimitriou, Ethan A. Chi, Richard Futrell, and Kyle Mahowald. 2021. [Deep subjecthood: Higher-order grammatical features in multilingual BERT](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2522–2532, Online. Association for Computational Linguistics.
- Joe Pater. 2019. [Generative linguistics and neural networks at 60: Foundation, friction, and fusion](#). *Language*, 95(1):e41–e74.

- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- David M. Perlmutter. 1978. Impersonal passives and the Unaccusative Hypothesis. In *Proceedings of the Annual Meeting of the Berkeley Linguistics Society*, 38, pages 157–189. Berkeley Linguistics Society, Linguistic Society of America.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Erika Petersen. 2020. [Break + NP constraints and the causative alternation](#). Ms., Stanford University.
- Christopher Piñón. 2001. A finer look at the causative-inchoative alternation. In *Proceedings of SALT 11*, pages 346–364.
- James Pustejovsky. 1991. The generative lexicon. *Computational Linguistics*, 17(4):409–441.
- James Pustejovsky. 1995. *The Generative Lexicon*. The MIT Press, Cambridge, MA.
- James Pustejovsky. 2006. [Introduction to Generative Lexicon](#). Ms., Brandeis.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. [Improving language understanding by generative pre-training](#). Ms, OpenAI.
- Malka Rappaport Hovav and Beth Levin. 2012. Lexicon uniformity and the causative alternation. In Martin Everaert, Marijana Marelj, and Tal Siloni, editors, *The Theta System: Argument Structure at the Interface*, pages 150–176. Oxford University Press, Oxford.
- Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. [Visualizing and measuring the geometry of BERT](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. *FrameNet II: Extended Theory and Practice*. International Computer Science Institute, Berkeley, CA.
- Florian Schäfer. 2008. *The Syntax of (Anti-)Causatives: External Arguments in Change-of-State Contexts*. John Benjamins, Amsterdam.
- John R. Searle. 1980. The background of meaning. In John R. Searle, Ferenc Kiefer, and Manfred Bierwisch, editors, *Speech Act Theory and Pragmatics*, pages 221–232. D. Reidel, Dordrecht.
- Alexandra Anna Spalek. 2012. Putting order into literal and figurative uses of verbs: *romper* as a case study. *Borealis*, 1(2):140–167.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- Sean Trott and Benjamin Bergen. 2021. [RAW-C: relatedness of ambiguous words-in context \(A new lexical resource for English\)](#). *CoRR*, abs/2105.13266.
- Peter D Turney. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *European Conference on Machine Learning*, pages 491–502. Springer.
- Laurens van der Maaten and Geoffrey Hinton. 2008. [Visualizing data using t-SNE](#). *Journal of Machine Learning Research*, 9(86):2579–2605.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Martin Wattenberg, Fernanda Viégas, and Ian Johnson. 2016. [How to use t-SNE effectively](#). *Distill*, 1(10):e2.
- Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. *ArXiv*, abs/1909.10430.

Supplementary Materials

A WordNet-based Features

The feature-based analyses of Section 3 are easily extended with features obtained using more approximate, data-driven techniques. To illustrate this potential, we looked to WordNet (Fellbaum, 1998), which has a very rich picture of *break*. The lemma *break* participates in 59 SynSets in WordNet. We built a graph of these SynSets based on the hypernym relation. The resulting graph has 29 connected components (29 subgraphs). Figure 1 depicts the largest connected components as subgraphs. If we label these subgraphs with their most-specific shared hypernym, we get potentially new meaning dimensions like “Cause to change; make different; cause a transformation” and “undergo a change; become different in essence; losing one’s or its original nature”. These are similar, but only the first conveys agency. Both seem like plausible latent semantic dimensions that we could add to Table 3, either as primitive features or as sets of more basic meanings. And of course this is only a single example of many that WordNet would support, and additional features could be extracted from FrameNet (Baker et al., 1998; Baker and Sato, 2003; Ruppenhofer et al., 2006).

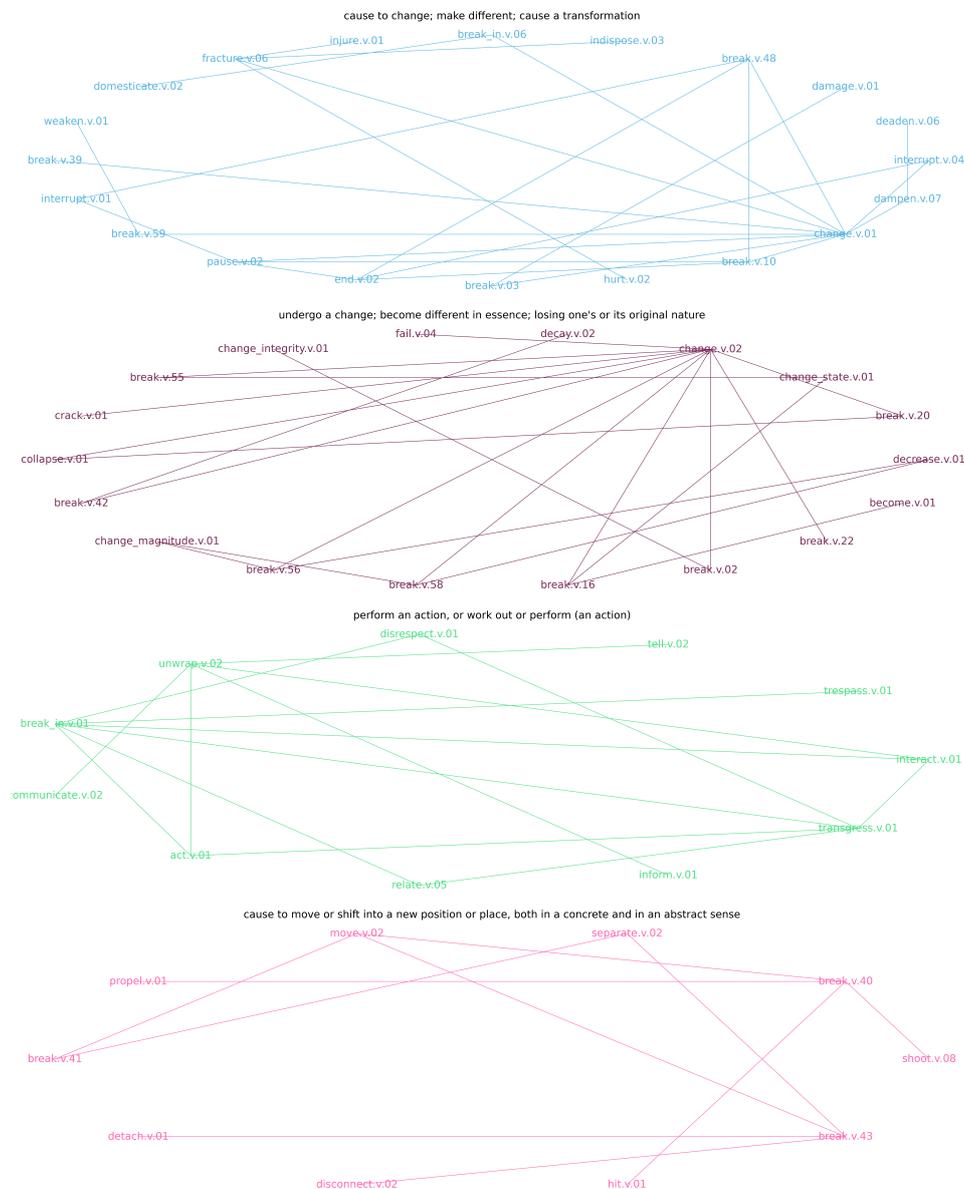


Figure 1: Largest WordNet connected components for *break* labeled with the name of their most specific shared hypernym. These hypernym labels suggest interesting abstract meaning dimensions for these senses.

B Additional Static Vector Analyses

Table 7 extends Table 4 from the main text with additional variants of word2vec, fastText, and GloVe. The overall picture seems consistent across these variants. For the main text, we simply chose the variant of each model that looked the best to us in terms of capturing meaning dimensions of *break*.

C LLM Structure

The input to the LLMs we consider is always a sequence of tokens $[x_1, \dots, x_n]$. Each token may correspond to a full word type or a word piece, depending on the tokenization method. For instance, whereas *the* is tokenized as a single unit, *breakage* is likely to be analyzed as two pieces, *break* and *##age*, where the *##* prefix indicates a word-internal piece. This is a detail we set aside in our analyses, since we consider only examples involving *break* and all the models we use analyze *break* as a single token.

The elements of the input sequence are looked up in a static embedding space. The result is a sequence of vectors $[\mathbf{x}_1, \dots, \mathbf{x}_n]$, where each \mathbf{x}_j has dimension d . These are akin to the static representations from models like those in Section 4: there is one vector per word piece and thus no contextual modulation.

The static embeddings are additively combined with one or more separate embeddings that record aspects of each token’s position in the sequence. In the simplest case, there is a single positional embedding that is used to create a sequence of vectors $[\mathbf{p}_1, \dots, \mathbf{p}_n]$, each of dimension d , and we obtain positionally enriched representations as $H_0 = [\mathbf{x}_1 + \mathbf{p}_1, \dots, \mathbf{x}_1 + \mathbf{p}_n]$. Thus, already at this point in the model, a single word will have different representations depending on where it appears in the input sequence.

The positionally-enriched embeddings are fed into the Transformer architecture itself. This creates numerous interactions between the representations. Each Transformer block $i > 0$ results in a sequence $H_i = [\mathbf{h}_{i,1}, \dots, \mathbf{h}_{i,n}]$ of hidden representations, each one of dimension d . The models we consider have between 12 and 24 of these layers.

We focus on models that are trained in the manner of the BERT model. The core of that training regime is *masked language modeling*, in which elements of the input sequence are randomly masked out or replaced with randomly chosen tokens from the vocabulary, and the task of the model is to learn to assign high likelihood to the actual token, using the entire surrounding sequence. This is a very advanced form of distributional learning, but the core intuition is very similar to that of the single vector models: we are learning linguistic properties entirely from co-occurrence patterns in corpus data.

In our main text, we report results for RoBERTa (Liu et al., 2019), which is ‘**Robustly optimized BERT approach**’. We focus on the ‘large’ variant, which has 24 hidden layers. In Appendix D, we report parallel experiments with the case-sensitive version of the original BERT model as well as two variants of the new DeBERTa model: version 1 and version 3 (which introduces some modifications to the pretraining regime compared to version 1). DeBERTa is potentially interesting from the perspective of lexical semantics, because it more fully separates the traditional static embeddings $[\mathbf{x}_1, \dots, \mathbf{x}_n]$ from the positional embeddings $[\mathbf{p}_1, \dots, \mathbf{p}_n]$. This might be taken to reify word types (as separate from token occurrences) more than the other models do. Like RoBERTa, BERT and the DeBERTa variants have ‘base’ (12-layer) and ‘large’ (24-layers) instances. For our main text, we chose to focus on RoBERTa-large because it seems slightly better overall than the rest, but our findings indicate that all these models perform about the same in our evaluations, suggesting that all of them can support lexical semantic investigation.

1. break	11. before
2. breaking	12. put
3. broke	13. start
4. breaks	14. take
5. set	15. trying
6. try	16. could
7. chance	17. to
8. time	18. broken
9. again	19. end
10. back	20. finally

(a) GloVe, Wikipedia+Gigaword, 300d.

1. break	11. weeks
2. time	12. start
3. breaks	13. last
4. before	14. end
5. then	15. broke
6. take	16. again
7. days	17. next
8. after	18. maybe
9. let	19. leave
10. up	20. down

(b) GloVe, Twitter, 2B tweets, 200d.

1. break	11. get
2. breaks	12. out
3. breaking	13. trying
4. broke	14. we
5. going	15. broken
6. let	16. again
7. away	17. come
8. take	18. down
9. up	19. make
10. 'll	20. before

(c) GloVe, Common Crawl 42B tokens, 300d.

1. break	11. up
2. breaks	12. trying
3. breaking	13. away
4. end	14. start
5. broke	15. get
6. down	16. again
7. take	17. 'll
8. let	18. back
9. going	19. out
10. leave	20. off

(d) GloVe, Common Crawl 840B tokens, 300d (from Table 4).

1. breaks	11. brief_respite
2. breaking	12. Nadal_netted_forehand
3. broke	13. loosen
4. broken	14. smash
5. Break	15. rip
6. Breaking	16. overhit_forehand
7. breather	17. miscued_forehand
8. shatter	18. cut
9. crack	19. slip
10. breaker	20. Breaks

(e) word2vec, GoogleNews, 300d (from Table 4).

1. break	11. follow
2. breaks	12. smash
3. breaking	13. BREAK
4. broke	14. knock
5. Break	15. water-main
6. broken	16. miss
7. crack	17. tie
8. take	18. go
9. shatter	19. relax
10. fix	20. start

(f) fastTest WikiNews, 300d.

1. break	11. breakin
2. breaks	12. breaked
3. breaking	13. broken
4. breake	14. legbreak
5. re-break	15. reack
6. break-	16. semi-break
7. unbreak	17. minibreak
8. breakes	18. breaker
9. break.	19. breaking-down
10. broke	20. tea-break

(g) fastTest WikiNews, subword modeling, 300d (from Table 4).

1. break	11. break.The
2. breaks	12. break.I
3. breaking	13. break.It
4. Break	14. break.This
5. broke	15. broken
6. break.And	16. break.So
7. Breaking	17. break.In
8. break.	18. break-
9. BREAK	19. breakck
10. Breaks	20. break.That

(h) fastTest Common Crawl 600B tokens, 300d.

1. break	11. take
2. breaks	12. broken
3. breaking	13. re-break
4. Break	14. breake
5. broke	15. abreak
6. break.	16. break.But
7. break.And	17. break-
8. rebreak	18. break.What
9. break.So	19. bend
10. breakck	20. break.That

(i) fastTest, Common Crawl 600B tokens, subword modeling, 300d.

Table 7: Static embedding spaces: closest neighbors of *break*.

D Additional Probing Results

Table 8a gives meaning-class probing results for all of the models described in Appendix C, and Table 8b provides a parallel set of results for the construction-type probes. The models are very consistent with each other in terms of layer-wise trends and overall performance. Only the DeBERTa variant stands out as showing differences that may be truly substantive.

		Probe	Control	Selectivity		Probe	Control	Selectivity	
bert-base-cased	1	0.64	0.04	0.60	bert-base-cased	1	0.75	0.34	0.40
	6	0.80	0.03	0.77		6	0.93	0.34	0.60
	12	0.81	0.03	0.78		12	0.95	0.33	0.63
bert-large-cased	1	0.65	0.04	0.61	bert-large-cased	1	0.72	0.33	0.39
	6	0.78	0.03	0.75		6	0.91	0.34	0.57
	12	0.83	0.03	0.80		12	0.94	0.33	0.62
	18	0.83	0.03	0.81		18	0.97	0.35	0.62
	24	0.84	0.03	0.81		24	0.97	0.33	0.63
deberta-base	1	0.72	0.03	0.68	deberta-base	1	0.88	0.34	0.54
	6	0.81	0.03	0.78		6	0.96	0.34	0.62
	12	0.85	0.03	0.82		12	0.97	0.32	0.64
deberta-large	1	0.72	0.04	0.68	deberta-large	1	0.86	0.33	0.53
	6	0.84	0.03	0.81		6	0.96	0.33	0.63
	12	0.81	0.04	0.78		12	0.96	0.33	0.64
	18	0.78	0.04	0.74		18	0.95	0.34	0.61
	24	0.83	0.03	0.81		24	0.96	0.34	0.63
deberta-v3-base	1	0.70	0.04	0.65	deberta-v3-base	1	0.87	0.32	0.54
	6	0.84	0.03	0.81		6	0.96	0.34	0.62
	12	0.75	0.03	0.72		12	0.94	0.32	0.61
deberta-v3-large	1	0.66	0.04	0.62	deberta-v3-large	1	0.80	0.34	0.45
	6	0.84	0.04	0.80		6	0.94	0.34	0.61
	12	0.83	0.03	0.79		12	0.96	0.33	0.64
	18	0.80	0.04	0.77		18	0.97	0.32	0.65
	24	0.79	0.04	0.75		24	0.95	0.36	0.60
roberta-base	1	0.66	0.03	0.63	roberta-base	1	0.82	0.33	0.49
	6	0.81	0.04	0.78		6	0.96	0.34	0.62
	12	0.83	0.03	0.80		12	0.96	0.32	0.64
roberta-large	1	0.33	0.03	0.30	roberta-large	1	0.50	0.33	0.17
	6	0.81	0.03	0.79		6	0.94	0.34	0.60
	12	0.83	0.03	0.80		12	0.96	0.33	0.63
	18	0.80	0.03	0.76		18	0.96	0.35	0.61
	24	0.86	0.03	0.83		24	0.97	0.32	0.65

(a) Meaning class.

(b) Construction type.

Table 8: Full probing results.

E Full Meaning Class Distribution

Table 9 gives the full set of meaning classes, with their counts, from the dataset of Petersen 2020. There are 72 classes in all. Our meaning-class probing experiments use only the 27 classes with at least 10 examples. Our construction-type probing experiments use the full dataset.

Meaning class		Meaning class	
separate_into_parts	150	break_open_open	5
end	126	break_in_interrupt	5
decipher	62	break_loose_detach	5
break_down_separate_into_parts	61	begin_construction	4
violate	59	eat_with_sb	4
break_up_separate_into_parts	35	change	4
surpass	34	break_from_detach	3
break_down_destroy	31	cost_too_much	3
break_into_intrude	28	break_loose_start	3
reveal	26	break_through_succeed	3
appear	25	break_up_destroy	3
break_through_pass_through	24	break_down_unclassified	3
render_inoperable	23	show_disagreement_with_group	3
unclassified	21	slow_down	3
break_down_render_inoperable	21	begin_to_sweat	2
break_free_escape	19	break_out_unclassified	2
break_down_succumb	18	break_down_pause	2
cause_to_fail	17	break_up_unclassified	2
break_up_end_relationship	17	happen	2
break_up_end	16	break_out_separate_into_parts	2
break_out_escape	15	break_out_prepare_for_consumption	2
break_even_profit=loss	14	break_in_mould_shoes	2
succumb	13	break_down_fail	2
break_out_start	12	dismantle_camp	2
experience_sorrow	11	break_loose_escape	1
break_away_detach	10	break_into_unclassified	1
break_off_end	10	break_off_stop	1
break_in_enter	9	go_bankrupt	1
break_apart_detach	9	break_away_pause	1
break_off_detach	7	break_in_train	1
break_for_pause	7	break_past_pass_through	1
destroy	6	tame	1
break_into_start	6	break_in_unclassified	1
pioneer	6	break_with_detach	1
lessen	6	break_out_have_skin_eruption	1
break_with_end_relationship	5	break_beef	1

Table 9: Full meaning-class distribution.

F Examples Selected as Theoretically Relevant

Here we provide the full set of examples extracted from our dataset using the procedure described in Section 5.3 and then selected by us as interesting for lexical semantic theory. The examples in bold are those that appear in Table 6.

Sentence	Meaning		Construction		Notes
	Gold	Predicted	Gold	Predicted	
Most of this information exchange takes place through what are known as newsgroups, which essentially just break all this international online babble up into different topics and areas of interest.	break_up_ separate_ into_parts	break_ down_ separate_ into_parts	causative	causative	Both senses seem active or possible.
What happens, when groups break up that means somebody got caught stealing the money or some guy does n't like it because another guy's a bigger star- KING: Or he married someone who- Mr. GATLIN: Right@!KING.	break_up_ separate_ into_parts	break_up_ end_ relationship	unacc.	unacc.	Both senses seem active.
Wind, naturally acidic rain, and physical processes such as freezethaw cycles also break down rock.	break_ down_ separate_ into_parts	break_ down_ destroy	causative	causative	Both senses seem active.
But her husband was determined not to break up the family.	break_up_ separate_ into_parts	break_up_ end_ relationship	causative	causative	Both senses seem active.
It was like you knew something, like you knew the story was getting ready to break again.	reveal	appear	unacc.	unacc.	Both senses seem active.
“So why tell the whole story now? Somebody, some male, has got to be willing to break this code of silence,” he says.	violate	end	causative	causative	Contextual entailment relation between the two labels.
Then too, stress can also work to break down the immune system, increasing the likelihood of respiratory and creating gastrointestinal and nervous disorders.	break_ down_ render_ inoperable	break_ down_ destroy	causative	causative	Contextual entailment relation between the two labels.
If you deprive yourself, you're going to break your diet and fall off it.	violate	end	causative	causative	Contextual entailment relation between the two labels.
Sen. BOB KERREY: I don't want to destroy Social Security or break a commitment.	violate	end	causative	causative	Contextual entailment relation between the two labels.
So they forwarded the pictures to Madrid, where another officer noticed some printing on a towel that helped break the case.	decipher	end	causative	causative	Contextual entailment relation between the two labels.
It's one example of how the standard model might break down.	break_ down_ render_ inoperable	break_ down_ succumb	unacc.	unacc.	Contextual entailment relation between the two labels.
Instead, crews will break down the structures over three years, releasing the water in the reservoirs at a rate that's more manageable for the animals and the people who live in the area.	break_ down_ separate_ into_parts	break_ down_ destroy	causative	causative	Contextual entailment relation between the two labels.

"The Comes would try to break the Saxon ranks with a mounted charge.	separate_ into_parts	end	causative	causative	Contextual entailment relation between the two labels.
Then the troops break formation and move out to a formation and stand guard, even from above, making sure the so-called detainees are safely behind the fence.	separate_ into_parts	end	causative	causative	Contextual entailment relation between the two labels.
The Soviet Union will break up into between six and twenty (or more) separate countries.	break_up_ separate_ into_parts	break_up_ end	unacc.	unacc.	Contextual entailment relation between the two labels.
It didn't take being an ICU exec to break the code: trade secret.	decipher	violate	causative	causative	Genuine uncertainty about which sense is intended.
@(Soundbite-of-music)@!Mr-GELB: (Singing) Tell me who's going to pick up the pieces when you start to break down.	break_ down_ separate_ into_parts	break_ down_ succumb	unacc.	unacc.	Gold meaning is literal; predicted meaning is metaphorical.
"People have so many problems overcoming the disputes that occur when families break up, and then to have to overcome the barriers that government puts up when they hold on to the money, literally sends children to bed hungry," says Jensen.	break_up_ end_ relationship	break_up_ separate_ into_parts	unacc.	unacc.	Gold meaning is metaphorical; predicted meaning is literal.
"I just don't want to break up such happy couples.	break_up_ end_ relationship	break_up_ separate_ into_parts	causative	causative	Gold meaning is metaphorical; predicted meaning is literal.
I had to break it up.	break_up_ end	break_up_ separate_ into_parts	causative	causative	Gold meaning is metaphorical; predicted meaning is literal.
Will the kibbutz movement "renew its days as of old" when it has recovered from the present crisis, as did the Hutterites at several points in their history? Will it continue to exist, but in a radically revised form, like Amana and other colonies? Or will the kibbutzim simply break up, to form part of the historical heritage of the Israeli nation, and no more— like so many of the well-preserved sites that aroused such powerful feelings in Yaakov Oved? The considerations I have advanced here seem to militate against the first of these possibilities and favor one of the others— perhaps a mixture of both.	break_up_ end	break_up_ separate_ into_parts	unacc.	unacc.	Gold meaning is metaphorical; predicted meaning is literal.
The past few days had consisted of a simple routine of drinking melted snow to stay hydrated and sleeping while waiting for the storm to break.	appear	end	unacc.	unacc.	Model prediction may be correct.
A small pair of scissors will easily break the seal, but bringing those scissors in your carry-on bag may no longer be permitted.	separate_ into_parts	decipher	causative	causative	The decipher prediction seems sensible given that a seal is like a lock or (easy) code that needs to be overcome.
Patients will sometimes break out in a spontaneous recitation of the rosary	break_out_ start	break_out_ start	unacc.	unerg.	The modifier "spontaneous" seems to affect agentivity and perhaps also argument structure.

Millennial darlings began to break down like virus-ridden websites, from the supercharged (Qualcomm, Oracle) to the superhyped (Amazon, Yahoo!) to the just plain super (Sun, Lucent, AOL).	<u>break_</u> <u>down_</u> succumb	<u>break_</u> <u>down_</u> render_	unacc.	unacc.	There is a comparison of "millennial darlings" with "virus-ridden websites". The gold meaning may apply to "millennial darlings" and the predicted meaning to "virus-ridden websites".
I felt disappointed, but I waited, hoping the clouds would break.	separate_ into_parts	appear	unacc.	unacc.	Weather events are persistently uncertain about whether they describe the start or end of something. Here, the clouds are leaving and other things are presumably appearing.

G Visualizations

Figure 2 uses t-SNE to visualize *break* embeddings from layer 1 of RoBERTa-large, and Figure 3 shows the embeddings from layer 24. We use color to distinguish the top 10 meaning classes (and the rest are gray). Underlined examples are unergative and boxed examples are unaccusative. The layer 24 visualization has much more structure than the layer 1 visualization. By layer 24, the model seems strikingly well-aligned with the meaning categories and construction types, as evidenced by how examples with the same color cluster together, and how the construction type annotations also cluster within those spaces. The other models we consider show effectively these same patterns.



Figure 2: t-SNE of break with RoBERTa-large, layer 1.

SWING : Balancing Coverage and Faithfulness for Dialogue Summarization

Kung-Hsiang Huang^{♦*} Siffi Singh[◇] Xiaofei Ma[◇] Wei Xiao[◇]
Feng Nan[◇] Nick Dingwall[◇] William Yang Wang[◇] Kathleen McKeown[◇]

[♦]University of Illinois Urbana-Champaign [◇]AWS AI Labs
khhuang3@illinois.edu

{siffis, xiaofeim, weixiaow, nanfen, nickding, wyw, mckeownk}@amazon.com

Abstract

Missing information is a common issue of dialogue summarization where some information in the reference summaries is not covered in the generated summaries. To address this issue, we propose to utilize natural language inference (NLI) models to improve coverage while avoiding introducing factual inconsistencies. Specifically, we use NLI to compute fine-grained training signals to encourage the model to generate content in the reference summaries that have not been covered, as well as to distinguish between factually consistent and inconsistent generated sentences. Experiments on the DIALOGSUM and SAMSUM datasets confirm the effectiveness of the proposed approach in balancing coverage and faithfulness, validated with automatic metrics and human evaluations. Additionally, we compute the correlation between commonly used automatic metrics with human judgments in terms of three different dimensions regarding coverage and factual consistency to provide insight into the most suitable metric for evaluating dialogue summaries.¹

1 Introduction

Dialogue summarization is a text generation task that aims to produce a compact summary given a piece of conversation. Conventional approaches to dialogue summarization rely on features of conversation data (Goo and Chen, 2018; Li et al., 2019; Oya et al., 2014). Recently, the rise of large pre-trained language models (LMs) has enabled coherent and fluent summaries to be generated without these features. However, low coverage and factual inconsistency remain two pressing issues as studies have shown that the summaries generated from these pre-trained LMs often do not fully cover the reference (Liu and Chen, 2021; Tang et al.,

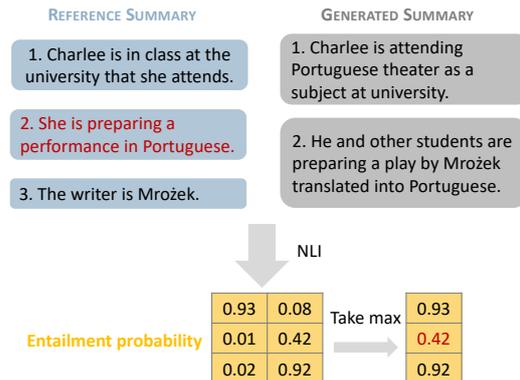


Figure 1: An illustration of how NLI can help determine whether a reference sentence is covered by the generated summary. We compute the entailment probability from each reference sentence (i.e. premise) to each generated sentence (i.e. hypothesis). By taking the max value along the row dimension, the resulting vector denotes the probability that each reference sentence entails a sentence in the generated summary. In this example, the entailment probability for the second reference sentence is low, indicating that this sentence is likely not covered by the generated summary.

2022) and that the generated summaries are often not factually consistent with the inputs (Zhang et al., 2020b; Maynez et al., 2020; Cao and Wang, 2021). If an unfaithful dialogue summarization model with low coverage is deployed for public use, it could spread misinformation and generate misleading content that only covers partial facts of a conversation. Hence, we are urgently in need of a solution to improve coverage without negatively impacting faithfulness for dialogue summarization.

Relatively little work addresses coverage and factual inconsistency for dialogue summarization. Some work addresses the issue of unfaithfulness with a controllable generation framework guided by person named entities (Liu and Chen, 2021) or summary sketches (Wu et al., 2021). Tang et al. (2022) categorize factual inconsistencies for dialogue summarization into different types of errors,

^{*}Work done while interning at Amazon.

¹We release our source code for research purposes: <https://github.com/amazon-science/AWS-SWING>.

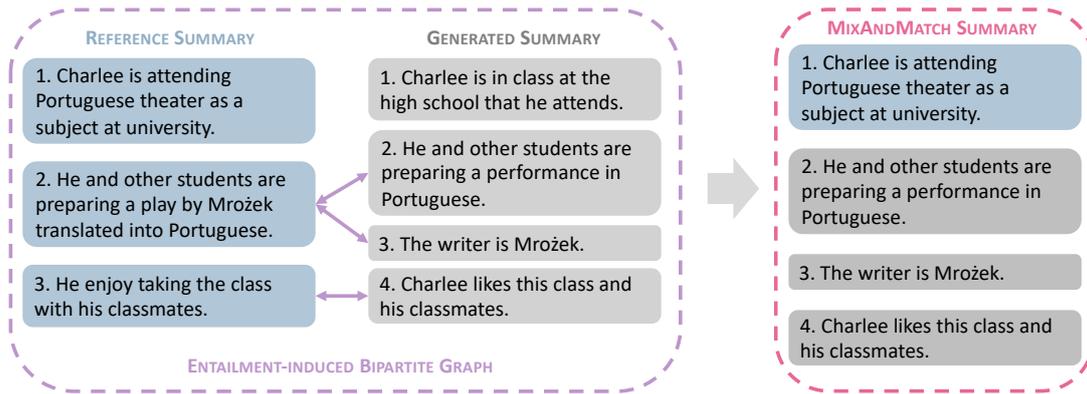


Figure 2: Illustration of how an entailment-induced bipartite graph is built and how a MIXANDMATCH summary is derived. With the NLI model, we determine which sentences from each summary contain equivalent information by computing the entailment probabilities between pairs of generated sentences and reference sentences, as indicated by the purple edges. Based on the graph, we determine that the generated summary does not cover the first reference sentence and that the first generated sentence is not faithful. Hence, the MIXANDMATCH summary is formed by combining the first reference sentence and the second to the fourth generated sentence.

such as missing information and wrong reference. Their framework integrates a contrastive loss and a self-supervised loss to reduce multiple types of errors. However, a great portion ($> 40\%$) of their outputs does not cover the full content of the reference summary. Thus, it is important to address coverage and factual consistency synergistically in dialogue summarization. The issue where the content in the reference does not occur in the generated summary is known as the missing information issue (Liu and Chen, 2021; Tang et al., 2022). In this work, we aim to mitigate missing information in the summary while being faithful to the dialogue.

We propose SWING , Summarizing Dialogue With NLI Guidance. Our approach samples a summary from the model and utilizes natural language inference (NLI) to determine (1) the faithfulness of each generated sentence and (2) whether each reference sentence has been covered by the generated summary. An example is shown in Figure 1. Based on the results computed by NLI, two losses are proposed to encourage the model to generate missing information and distinguish between factually consistent and inconsistent generated sentences.

Our contributions can be summarized as follows:

- We propose SWING, a dialogue summarization framework that effectively addresses missing information through two losses computed using NLI. The first loss encourages the model to recover content missing from the reference summaries. The second loss instructs the model to differentiate between factually consistent and inconsistent

generated sentences.

- Our approach achieves the best performance in mitigating missing information on two public dialogue summarization datasets, DIALOGSUM (Chen et al., 2021b) and SAMSUM (Gliwa et al., 2019), as validated by automatic metrics and human judges.
- We measure the correlation of human judgments with conventional and recently developed automatic metrics to provide intuition for future research on evaluating the faithfulness and coverage of dialogue summaries.

2 Method

Upon analyzing the dialogue summaries in SAMSUM, we observe that dialogues are often summarized linearly, consistent with the findings of Wu et al. (2021). Therefore, we segment the summaries into sentences and use a natural language inference (NLI) model to provide finer-grained training signals at the sentence level for two goals: (1) encourage generating sentences in the reference summaries that have not been covered by the generated sentences and (2) differentiate factually consistent generated sentences from inconsistent ones. To achieve these goals, we first determine the faithfulness of each sentence using an entailment-induced bipartite graph (§2.1). Then, we propose two new losses addressing each challenge in turn: an **Uncovered Loss** that encourages the model to recover missing information (§2.2) and a **Contrastive Loss** that brings closer the representations of the reference summary and the generated sentences that

Algorithm 1: Entailment-induced Bipartite Graph

Input: A reference summary $S^* = \{s_1^*, \dots, s_n^*\}$, a generated summary $S = \{s_1, \dots, s_m\}$;
Output: The bipartite mapping ϕ between sentences in S^* and S ;

- 1 Initialize ϕ as a zero matrix of size $n \times m$ where $n = |S^*|$ and $m = |S|$;
- 2 Let τ be the entailment threshold;
- 3 // Resolve 1-to-many mappings;
- 4 **for** $i \leftarrow 1$ **to** n **do**
- 5 $V \leftarrow \emptyset$;
- 6 **for** $j \leftarrow 1$ **to** m **do**
- 7 **if** $p_{\text{ent}}(s_i^*, s_j) > \tau$ **and** $\phi(i, j) = 0$ **then**
- 8 $V \leftarrow V \cup j$;
- 9 $s_V \leftarrow$ Concatenate sentences in $\{s_v, \forall v \in V\}$;
- 10 **if** V is consecutive **and** $p_{\text{ent}}(s_V, s_i^*) > \tau$ **then**
- 11 **for** $v \in V$ **do**
- 12 $\phi(i, v) \leftarrow 1$;
- 13 // Resolve many-to-1 mappings;
- 14 **for** $j \leftarrow 1$ **to** m **do**
- 15 $V \leftarrow \emptyset$;
- 16 **for** $i \leftarrow 1$ **to** n **do**
- 17 **if** $p_{\text{ent}}(s_j, s_i^*) > \tau$ **and** $\phi(i, j) = 0$ **then**
- 18 $V \leftarrow V \cup i$;
- 19 $s_V^* \leftarrow$ Concatenate sentences in $\{s_v^*, \forall v \in V\}$;
- 20 **if** V is consecutive **and** $p_{\text{ent}}(s_V^*, s_j) > \tau$ **then**
- 21 **for** $v \in V$ **do**
- 22 $\phi(v, j) \leftarrow 1$;
- 23 // Resolve 1-to-1 mappings;
- 24 **for** $i \leftarrow 1$ **to** n **do**
- 25 **for** $j \leftarrow 1$ **to** m **do**
- 26 **if** $\phi(i, j) = 0$ **and** $p_{\text{ent}}(s_j, s_i^*) > \tau$ **and**
- 27 $p_{\text{ent}}(s_i^*, s_j) > \tau$ **then**
- 28 $\phi(i, j) \leftarrow 1$;
- 28 Return ϕ ;

contain equivalent information to some sentences in the reference summary (§2.3). For the rest of this paper, we use *reference sentence* and *generated sentence* to refer to a sentence in the reference summary and the generated summary, respectively.

2.1 Entailment-induced Bipartite Graph

To determine which reference sentence has not been covered by the generated summary and which generated sentence is not faithful to the reference summary, we construct a bipartite graph that links sentences between a reference summary and a generated summary. An edge indicates the linked sentences contain equivalent information. If no edge connects to a reference sentence, we consider this sentence not covered by the generated summary. Similarly, if a generated sentence is not linked in the bipartite graph, this sentence is likely not

faithful to the reference summary. We use the entailment probabilities computed by an NLI model to determine whether a pair of sentences contain equivalent information. The procedure of constructing the bipartite graph is shown in Algorithm 1.

The NLI model takes in two sentences, a premise (P) and a hypothesis (H), and computes whether P entails, contradicts, or is neutral to H . Here, we only focus on the entailment probability from the i -th reference sentence to the j -th generated sentence $p_{\text{ent}}(s_i^*, s_j)$. We use the ROBERTA-LARGE model² trained on the MNLI dataset, achieving an accuracy of around 91%, which is on par with the performance of state-of-the-art models.

Let $\phi(i, j)$ denote the mapping between the i -th reference sentence and the j -th generated sentence. $\phi(i, j) = 1$ if a link exists between s_i^* and s_j ; otherwise, $\phi(i, j) = 0$. We first consider a simplified setting by assuming each reference sentence can be mapped to at most one generated sentence, and vice versa (i.e. $0 \leq \sum_j \phi(i, j) \leq 1$). In this setting, we can determine whether two sentences contain equivalent information by checking the entailment relation from both directions (lines 26-27).

$$\phi(i, j) = \begin{cases} 1, & p_{\text{ent}}(s_i^*, s_j) > \tau \wedge p_{\text{ent}}(s_j, s_i^*) > \tau \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Here, τ is a hyperparameter that indicates the entailment threshold.

However, one reference sentence may contain information equivalent to multiple generated sentences (one-to-many mappings) and vice versa (many-to-one mappings). In Figure 2, for example, the second reference sentence contains information equivalent to the second and the third generated sentences combined. This relation cannot be discovered if we only check the entailment relation between pairs of individual sentences.

Therefore, we must resolve one-to-many and many-to-one mappings before checking one-to-one mappings. To find one-to-many mappings, for every reference sentence s_i^* , we look for consecutive generated sentences $\{s_j, s_{j+1}, \dots, s_{j+k}\}$ s.t. $\max_i p_{\text{ent}}(s_i^*, s_m) > \tau \forall m \in \{j, \dots, j+k\}$ (lines 6-8). We only check for consecutive sentences based on our previous observation that dialogues are often summarized linearly. For every match, we concatenate the generated sentences $s_{j:j+k} = \{s_j, s_{j+1}, \dots, s_{j+k}\}$ and

²<https://huggingface.co/roberta-large-mnli>

check whether $s_{j:j+k}$ entails the reference sentence s_i^* (lines 8-9). If the entailment holds, we let $\phi(i, m) = 1 \ \forall m \in \{j, \dots, j+k\}$ (lines 11-12). The same approach is used to address many-to-one mappings (lines 14-22). Following Algorithm 1, a bipartite graph is built between the generated summary and the reference summary. Henceforth, we denote the reference sentences that have not been covered as $\underline{S}^* = \{s_i^* | \forall j \ \phi(i, j) = 0\}$ and generated sentences that can be mapped to some of the reference sentences as $\underline{S} = \{s_j | \exists i \ \phi(i, j) = 1\}$.

2.2 Uncovered Loss

The objective of the uncovered loss is to encourage the model to generate information from the reference summary that the generated summary has not covered. To this end, we train the model with MIXANDMATCH summaries, which are constructed by combining reference sentences that are not covered by the generated summary and generated sentences that contain information equivalent to some of the reference sentences. An example is shown in Figure 2.

The MIXANDMATCH summary \hat{S} is constructed by taking the union of \underline{S} and \underline{S}^* and sorting the sentences by their index,

$$\hat{S} = \text{SORT}(\underline{S} \cup \underline{S}^*). \quad (2)$$

The uncovered loss is effectively maximum likelihood estimation (MLE) with MIXANDMATCH summaries being the decoding targets:

$$\mathcal{L}_{\text{Uncovered}} = - \sum_t \log p(\hat{S}_t | \hat{S}_{<t}, \mathcal{D}), \quad (3)$$

where \mathcal{D} is the original dialogue and \hat{S}_t denotes the t -th token in the MIXANDMATCH summary.

The main advantages of constructing MIXANDMATCH summaries over other positive sample construction approaches, such as back translation and paraphrasing, are the two desired properties of this formulation. First, the model already has a high probability of generating sentences in \underline{S} . Therefore, the loss function (Equation (3)) does not penalize the model much for generating these sentences. Second, the penalty for generating sentences \underline{S}^* is larger since the model has a lower probability of generating those sentences.

2.3 Contrastive Loss

In the early stage of our experiment, the original goal was to discourage the model from generating

factually inconsistent sentences. We adopt unlikelihood training (Welleck et al., 2020) to decrease the probability of sampling these sentences from the model. However, we found that this objective causes the model to generate nonsense sequences. This phenomenon was also observed when we experimented with CONSEQ (Nan et al., 2021), which also incorporates such a loss function into its training process, as shown in §4.1. We hypothesize that it resulted from the fact that sentences in dialogue summaries share similar structures. Hence, using the unlikelihood training objective would confuse the model.

Instead, we pivoted our focus on differentiating factually consistent sentences from their inconsistent counterparts with the proposed contrastive loss. For each summary, we use the factually inconsistent sentences as negative samples (i.e. $s_j \notin \underline{S}$) and consistent sentences as positive samples (i.e. $s_j \in \underline{S}$). The contrastive learning objective takes a similar form as the InfoNCE loss (Oord et al., 2018):

$$\mathcal{L}_{\text{Contrastive}} = - \sum_{s_i \in \underline{S}} \frac{\exp(\cos(h_i, h_{S^*}))}{\sum_{s_j \in \underline{S}} \exp(\cos(h_j, h_{S^*}))} \quad (4)$$

, where h_i and h_j denote the representations of the generated sentences, h_{S^*} means the representations of the reference summary, and $\cos(\cdot, \cdot)$ denotes cosine similarity. The main difference between our contrastive objective and the other work (Cao and Wang, 2021; Tang et al., 2022) is granularity. Equation (4) operates at the sentence level rather than the summary level; therefore, it provides finer-grained training signals.

2.4 Training

The final loss function that our model is optimized with is a weighted sum of the two aforementioned loss functions and MLE,

$$\mathcal{L}_{\text{Final}} = \mathcal{L}_{\text{MLE}} + \alpha \mathcal{L}_{\text{Uncovered}} + \beta \mathcal{L}_{\text{Contrastive}}, \quad (5)$$

where \mathcal{L}_{MLE} is:

$$\mathcal{L}_{\text{MLE}} = - \sum_t \log p(S_t^* | S_{<t}^*, \mathcal{D}). \quad (6)$$

3 Experiments

3.1 Datasets

Experiments are conducted on two English-language dialogue summarization datasets: SAMSUM (Gliwa et al., 2019) and DIALOGSUM (Chen

Model	DIALOGSUM								SAMSUM							
	RL _F	RL _R	BS _F	BS _R	FC _F	FC _R	QS	QFE	RL _F	RL _R	BS _F	BS _R	FC _F	FC _R	QS	QFE
TextRank	27.74	29.16	-3.000	-3.039	60.55	59.54	-1.948	0.566	15.08	16.15	-4.374	-3.891	34.28	33.02	-2.172	0.237
BART-LARGE	50.82	56.78	-2.012	-1.960	82.90	85.86	-1.183	1.854	49.53	52.71	-2.248	-2.332	62.46	61.28	-0.912	2.335
CTRLDIASUMM	48.99	57.25	-2.145	-1.985	82.55	85.96	-1.214	1.817	47.79	51.17	-2.360	-2.414	61.50	61.76	-0.957	2.272
CODS	48.51	48.36	-2.379	-2.214	83.33	86.81	-1.246	1.860	48.39	47.68	-2.643	-2.593	61.21	62.01	-0.867	2.345
CONSEQ	22.82	19.50	-3.480	-3.588	84.24	73.14	-1.474	0.208	12.04	7.62	-5.908	-7.278	41.23	13.77	-2.058	0.035
CLIFF	51.87	56.22	-2.012	-1.973	85.38	86.30	-1.106	2.109	43.70	45.49	-2.485	-2.340	55.47	56.01	-1.063	1.891
CONFIT	50.44	55.65	-2.049	-2.016	83.34	86.37	-1.179	1.790	49.29	52.76	-2.188	-2.316	65.03	63.12	-0.819	2.343
SWING	51.96	59.04*	-1.999*	-1.904*	86.48	89.03	-1.082*	2.087	50.08	52.91	-2.228	-2.310*	64.19	63.52	-0.829	2.407*
- $\mathcal{L}_{\text{Uncovered}}$	50.94	60.06*	-2.044	-1.895*	83.26	87.45	-1.075*	2.339*	49.78	53.57	-2.231	-2.295*	63.81	63.11	-0.876	1.989
- $\mathcal{L}_{\text{Contrastive}}$	51.53	59.27*	-2.012	-1.901*	82.90	85.86	-1.130	2.399*	49.73	53.95	-2.185*	-2.143*	63.47	63.15	-0.886	2.027

Table 1: Performance comparison on DIALOGSUM and SAMSUM. - $\mathcal{L}_{\text{Uncovered}}$ and - $\mathcal{L}_{\text{Contrastive}}$ denote variants of SWING by ablating the corresponding loss. RL denotes ROUGE-L (%), BS denotes BARTSCORE, FC denotes FACTCC (%), QS denotes QUALS, and QFE denotes QAFACTEVAL. The subscripts F and R denote F1 score and recall, respectively. The proposed method outperforms previous systems on both DIALOGSUM and SAMSUM in most metrics, especially on the recall measures. Statistical significance over previous best systems computed with the permutation test (Fisher et al., 1937) is indicated with * ($p < .01$).

et al., 2021b). SAMSUM contains 16,369 online chitchat dialogues with an average of around 94 tokens per dialogue. DIALOGSUM is a spoken dialogue dataset that consists of 13,460 samples in total. With an average token count of about 131, the dialogues in DIALOGSUM are under real-life scenarios with clear communication patterns and intents. Details of the dataset statistics can be found in Appendix A.

3.2 Metrics

Our evaluation focuses on measuring the factual consistency, particularly the missing information challenge, of the summarization models. Therefore, we adopt recently developed metrics that have been shown to correlate well with human judgments in terms of faithfulness. BARTScore (Yuan et al., 2021) computes the semantic overlap between the generated summary and the reference summary by calculating the logarithmic probability of generating each summary conditioned on the other one. Since our goal is to assess how well the model reduce information missing from the reference summary, we consider the *Recall* (R) setting where we assess $p(S^* | S, \theta)$, the likelihood of generating the reference summary S given the generated summary S^* . FactCC (Kryscinski et al., 2020) is an entailment-based metric that predicts the faithfulness probability of a claim w.r.t. with the source texts. Similar to BARTScore, we use FactCC in the *Recall* setting where the claim is a reference sentence and the source text is the generated summary. We report the mean of the average CORRECT probability of each sentence within a generated summary.

In addition, we report the ROUGE-L metric (Lin, 2004), which has been also shown to better reflect

faithfulness compared to ROUGE-1 and ROUGE-2 (Pagnoni et al., 2021). For these metrics, we also consider the *F1* setting, where we compute each metric in the reverse direction ($S^* \rightarrow S$) and then take the average of both directions, to validate that the model is not generating too much redundant information. Finally, two recently introduced QA-based metrics that have demonstrated close approximation to human judgements in terms of factuality, QUALS (Nan et al., 2021) and QAFACTEVAL (Fabbri et al., 2022a), are also used for evaluation.

3.3 Implementation Details

We choose BART (Lewis et al., 2020) as the backbone seq2seq model as it has demonstrated better dialogue summarization performance than other pre-trained language models (Tang et al., 2022), such as PEGASUS (Zhang et al., 2020a) and T5 (Raffel et al., 2020). The proposed models are optimized using AdamW (Loshchilov and Hutter, 2019) with learning rate $3e-5$ and weight decay $1e-3$. The maximum input sequence length is set to 1024. For all baseline models, we use the best hyper-parameters reported in their papers. We fix τ to be 0.5 throughout all our experiments. α and β are both 1.0.

3.4 Baselines

We compare SWING with the following competitive baseline systems. **TextRank** (Mihalcea and Tarau, 2004) is a graph-based ranking algorithm that performs extractive summarization. **BART** (Lewis et al., 2020) is a seq2seq language model pre-trained on various denoising objectives. **CTRLDIASUMM** (Liu and Chen, 2021) and **CODS** (Wu et al., 2021) are controllable generation frameworks that generate summaries guided by named entity

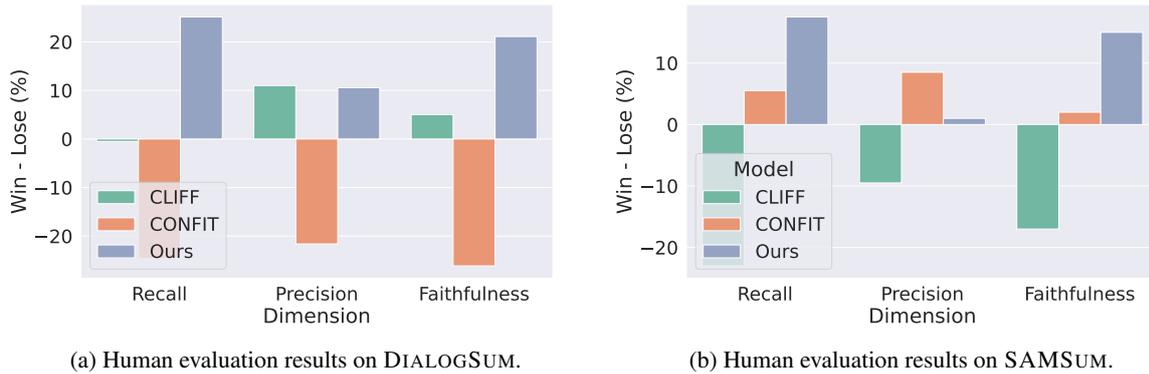


Figure 3: Human evaluation results. SWING achieves the highest RECALL and FAITHFULNESS scores on both datasets, suggesting the advantages of our approach in reducing missing information and improving the overall faithfulness of the generated dialogue summary.

planning and sketches, respectively. **CONSEQ** (Nan et al., 2021) learns a contrastive objective based on unlikelihood training, where positive and negative samples are selected by QUALS. **CLIFF** (Cao and Wang, 2021) and **CONFIT** (Tang et al., 2022) are trained with a similar contrastive learning loss that takes the form of the InfoNCE loss (Oord et al., 2018), except that CONFIT is optimized with an additional self-supervised loss that aims to reduce reference errors. BART-LARGE is used across all experiments that involve pre-trained language models for fair comparison.

4 Results

4.1 Main results

Table 1 summarizes the main results on DIALOGSUM and SAMSUM. SWING outperforms previous approaches in almost all metrics, especially recall measures. This result reflects that the proposed approach generates summaries that cover more content in the reference summaries lexically and semantically. One interesting observation was the deficient performance of CONSEQ on both datasets. We hypothesize that poor performance was the use of the unlikelihood training objective in their loss, as mentioned in §2.3. Since sentences of dialogue summaries often share similar structures, adopting such an objective could confuse the model. We verified this hypothesis by running a small experiment by training BART-LARGE with MLE and negative samples determined by QUALS, similar to CONSEQ. The resulting model also produces significantly lower performance than training with MLE alone. The finding confirms that the poor performance of CONSEQ is caused by the unlikelihood training and that such a loss function is unsuitable for dialogue summarization.

4.2 Human Evaluation

To further validate the effectiveness of SWING, we use Amazon’s Mechanical Turk (AMT) to recruit workers to conduct human evaluations on three methods: CLIFF, CONFIT and SWING. We sampled 100 dialogues from the test set of DIALOGSUM and SAMSUM, respectively. For each dialogue, human judges are presented with a pair of summaries produced by two different approaches and asked to select the better one with respect to three dimensions. **RECALL** assesses the portion of information in the reference summary covered by the generated summary. **PRECISION** considers whether all the content in the generated summary occurs in the reference summary. **FAITHFULNESS** examines whether the generated summary is factually consistent with the dialogue. "Tie" is selected if the judges consider the two summaries to be of equal quality. The final score of each system is calculated as the percentage of times the system is selected as the better one minus the percentage of times the system is not. To evaluate the annotation quality, we compute the inter-annotator agreement. The average Cohan’s Kappa (Cohen, 1960) is 54.35%, indicating a moderate agreement. Details of the human evaluation setup can be found in Appendix B.

The human evaluation results are demonstrated in Figure 3. We have the following observations. First, SWING achieves the highest RECALL scores on both datasets, indicating that our approach is the best in addressing the missing information issue for dialogue summarization. Second, while SWING does not score the highest on PRECISION, we achieve the highest scores on FAITHFULNESS. This implies that even though our approach often generates summaries with extra information,

Reference Summary	CONFIT	SWING
Mike took his car into garage today. Ernest is relieved as someone had just crashed into a red Honda which looks like Mike’s.	Mike took his car to the garage today. Someone crashed into his car.	Mike took his car into the garage today. Someone just crashed into a red Honda looking like Mike’s.
Hilary has the keys to the apartment. Benjamin wants to get them and go take a nap. Hilary is having lunch with some French people at La Cantina. Hilary is meeting them at the entrance to the conference hall at 2 pm. Benjamin and Elliot might join them. They’re meeting for the drinks in the evening.	Benjamin, Elliot, Daniel and Hilary will meet at La Cantina at 2 pm to have lunch with some French people who work on the history of food in colonial Mexico. They will try to avoid talking about their subject of research.	Hilary has the keys to Benjamin, Elliot and Daniel’s apartment. They will meet at the entrance to the conference hall at 2 pm and go to La Cantina for lunch with some French people who work on the history of food in colonial Mexico.

Table 2: Qualitative analysis on the outputs of SWING and CONFIT. The two rows demonstrate the *missing details* and the *missing sentences* issue of the summaries generated by CONFIT, respectively. The extra information in the outputs of CONFIT that also occurs in the reference summaries is highlighted in blue. In both cases, SWING is able to cover more content presented in the reference summaries.

the additional content is likely still faithful to the input. To measure the amount of additional information produced, we compute the average number of tokens per summary for each model. As seen in Table 3, the summaries generated by SWING is only slightly longer than those produced by CLIFF and CONFIT. This suggests that SWING achieves significantly higher faithfulness and coverage than CLIFF and CONFIT while maintaining conciseness.

Model	DIALOGSUM	SAMSUM
CONFIT	29.46	22.45
CLIFF	27.34	22.30
BART-LARGE	28.03	23.19
SWING	31.32	24.23

Table 3: Average token count per summary generated by different models.

4.3 Qualitative Analysis

To provide better insight into the effectiveness of the proposed method, we conduct a qualitative analysis using the 100 dialogues randomly sampled from the SAMSUM dataset. Specifically, we further categorize missing information errors into two sub-types: (1) *missing details* where partial information of a sentence in the reference summary is missing in the generated summary and (2) *missing sentences* where the model fails to generate an entire sentence in the reference summary. An example of each sub-type is shown in Table 2. By comparing the test sets outputs of CONFIT and SWING, we see that there are 10 improved cases with less *missing details* and 6 cases where *missing sentences* is mitigated by SWING. Meanwhile, our

proposed approach only introduces *missing details* error and *missing sentences* error in 1 and 2 examples, respectively. This implies that our approach is effective in alleviating both sub-types of missing information error while particularly advantageous in reducing *missing details* errors.

4.4 Correlation with Human Judgements

Although recently proposed metrics have been shown to be highly correlated with human judgements on news summarization in terms of factuality (Kryscinski et al., 2020; Yuan et al., 2021), no previous work has studied the transferability of these metrics to dialogue summarization. We seek to answer this question by computing the correlation of the automatic metrics in Table 1 with the human annotations discussed in §4.2. Using Kendall’s Tau (Kendall, 1938) as the correlation measure, the results are summarized in Table 4. We observe that: (1) $BARTSCORE_R$ is the most consistent and reliable metric across the three dimensions. It performs the best in RECALL on both datasets, indicating that $BARTSCORE_R$ is most suitable for measuring how well a model resolves the missing information issue in dialogue summarization. (2) Although a large number of invalid questions and answers are generated, QUALS is the best metric for assessing PRECISION overall. (3) $FACTCC_F$ and $FACTCC_R$ are two of the worst metrics in general. This could be explained by the fact that FACTCC constructs negative samples with some semantically variant transformations. However, these transformations may not be comprehensive enough to cover all cases. Hence, the poor transferability of FACTCC on these two datasets.

Metric	DIALOGSUM			SAMSUM		
	RECALL	PRECISION	FAITHFULNESS	RECALL	PRECISION	FAITHFULNESS
ROUGE-L _F	23.50	24.21	10.29	6.07	10.24	-0.75
ROUGE-L _R	23.46	2.51	4.24	29.52	9.61	17.88
BARTSCORE _F	18.35	25.94	3.17	15.50	8.00	10.69
BARTSCORE _R	26.48	14.87	9.25	32.10	9.68	24.11
FACTCC _F	6.15	6.93	1.19	-3.43	5.12	-2.28
FACTCC _R	4.79	6.86	10.56	4.13	10.32	-1.43
QUALS	14.23	23.61	-0.83	1.55	15.35	4.50
QAFACTEVAL	14.06	16.20	16.80	5.03	2.83	6.26

Table 4: Correlation (%) of automatic metrics with human judgements. We first convert human evaluation results and automatic metric scores into a scale of $\{-1, 0, 1\}$, which corresponds to $\{\text{LOSE, TIE, WIN}\}$. Then, Kendall’s Tau (Kendall, 1938) is used to compute the correlation between two sequences.

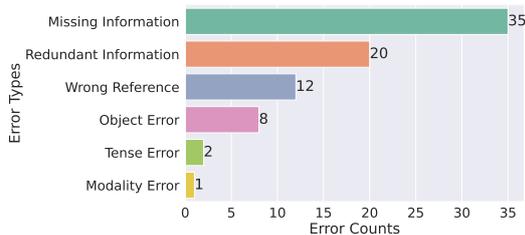


Figure 4: Remaining challenges.

4.5 Remaining Challenges

We analyzed the remaining errors by comparing 100 generated summaries with corresponding reference summaries on the SAMSUM datasets using the categories of factual errors defined in Tang et al. (2022). The results are shown in Figure 4. We observe that missing information still accounts for the largest portion of factual errors, even though our approach significantly exceeds prior methods in mitigating this issue. This reflects that this issue is challenging to tackle and that there is still a great opportunity to improve the reduction of missing information. As a comparison, we manually inspected outputs of BART-LARGE using the same 100 dialogues as input. We found 42 cases where information is missing from the dialogue summaries produced by BART-LARGE. This observation further confirms the effectiveness of SWING in addressing insufficient coverage. In addition, redundant information is another major source of errors. Although we have shown in §4.2 that the additional information generated by SWING is likely still faithful to the input dialogue, compactness is one of the important qualities of a summary. This can be improved by using NLI to guide the model to avoid generating extra information. Other common mistakes are wrong reference and object errors, both of which can be addressed with the self-supervised

loss discussed in Tang et al. (2022).³

5 Related Work

Dialogue Summarization Early work on dialogue summarization focus on the AMI meeting corpus (McCowan et al., 2005) due to the lack of dialogue summarization data. These studies enhance summarization performance by leveraging features of conversational data, such as dialogue act (Goo and Chen, 2018), visual features (Li et al., 2019), and the relationships between summary and dialogue (Oya et al., 2014). Later, Gliwa et al. (2019) released the SAMSUM dataset, the first large-scale dialogue summarization dataset, enabling abstractive summarization research on casual chat dialogue. With the rise of large language models (LMs), recent work focuses on improving the controllability of sequence-to-sequence models built upon large LMs. For instance, Wu et al. (2021) proposes to utilize a summary sketch to control the granularity of the summary generated. Liu and Chen (2021) conditions the generators with person name entities to control which people to include in the generating summary. Chan et al. (2021) improves controllability by formulating the summarization task as a constrained Markov Decision Process.

Factual Consistency Enhancement While factuality has been widely explored in the field of fact-checking and fake news detection (Thorne et al., 2018; Wadden et al., 2020; Huang et al., 2022b; Shu et al., 2018; Pan et al., 2021; Huang et al., 2022a), factual inconsistency remains a major challenge for abstractive summarization. One line of work attempts to improve the faithfulness of

³This analysis is not comparable to results reported in Tang et al. (2022) due to differences in the sampled examples.

the generated summary with a separate correction model that corrects the errors made by the summarization model (Dong et al., 2020; Cao et al., 2020; Fabbri et al., 2022b) or directly fix factual inconsistencies in the training data (Adams et al., 2022). Another line of work employs auxiliary loss functions to improve models’ representations or discourage the model from generating unfaithful outputs (Cao and Wang, 2021; Chen et al., 2021a; Nan et al., 2021; Tang et al., 2022). The main advantage of these approaches is the efficiency in inference time.

Some studies have attempted to use NLI to detect factual inconsistency in generated summaries. Early approaches rely on out-of-the-box NLI models, which did not yield satisfactory results (Falke et al., 2019). Barrantes et al. (2020) improved the detection accuracy by using an NLI model fine-tuned on the Adversarial NLI dataset (Nie et al., 2020). Laban et al. (2022) addresses the mismatch issue in input granularity between NLI datasets and inconsistency detection by passing sentence pairs as inputs instead of document-summary pairs. Kryscinski et al. (2020) and Yin et al. (2021) trains document-sentence entailment models to address the granularity mismatch issue. Utama et al. (2022) introduces a controllable generation framework that generates document-level NLI training data for identifying factual inconsistency. Our work leverages an NLI model to guide the dialogue summarization model to recover missing information.

6 Conclusion

We have proposed SWING, a dialogue summarization framework that generates summaries with mitigated missing information and improved faithfulness. To instruct the model to generate missing content from the reference summaries and to differentiate factually consistent generated sentences from their inconsistent counterparts, we propose two losses based on NLI. Experimental results on the DIALOGSUM and SAMSUM datasets showed that our approach achieves significantly higher faithfulness and coverage, while still maintaining conciseness, compared to prior methods. In addition, we measure the correlation between the reported automatic metrics and human judgments to provide insight into the most suitable metric for evaluating the coverage and factuality of dialogue summaries for future research.

7 Ethical Considerations

We acknowledge that the use of large language models pre-trained on the Web could lead to biased outputs. We did find out that our model may sometimes generate the incorrect pronouns for neutral names. For example, in Figure 1, Charlee is being referred to as a male in the generated summary, while Charlee is actually a female as shown in the reference summary. Such an issue is often caused by under-specified context (e.g. Charlee’s gender is not mentioned in the input dialogue). Fortunately, we found that such an error accounts for < 1% of the total outputs from our framework and the issue can be largely alleviated when enough context is provided.

8 Limitations

While our proposed approach is effective in mitigating missing information, this issue is still far from resolved, as shown in Figure 4. Significant effort is needed to ensure dialogue summarization models produce completely factual content. In addition, our method works as we found that most of the reference summaries in the two datasets we used are faithful to the corresponding dialogue. The proposed method may not work on other summarization datasets, such as XSum, which contains hallucinations in about 70% of the reference summaries (Maynez et al., 2020).

9 Acknowledgments

We would like to extend our gratitude to the reviewers for their valuable feedback and insights, which greatly contributed to the improvement of this paper. We would also like to thank the human evaluators for their time and effort in assessing the performance of our model. Their contributions have been essential in ensuring the quality of our research.

References

- Griffin Adams, Han-Chin Shing, Qing Sun, Christopher Winestock, Kathleen McKeown, and Noémie Elhadad. 2022. Learning to revise references for faithful summarization. *arXiv preprint arXiv:2204.10290*.
- Mario Barrantes, Benedikt Herudek, and Richard Wang. 2020. Adversarial nli for factual correctness in text summarisation models. *ArXiv*, abs/2005.11739.
- Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the*

- 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6251–6258, Online. Association for Computational Linguistics.
- Shuyang Cao and Lu Wang. 2021. **CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hou Pong Chan, Lu Wang, and Irwin King. 2021. **Controllable summarization with constrained Markov decision process**. *Transactions of the Association for Computational Linguistics*, 9:1213–1232.
- Wang Chen, Piji Li, Hou Pong Chan, and Irwin King. 2021a. **Dialogue summarization with supporting utterance flow modelling and fact regularization**. *Knowl. Based Syst.*, 229:107328.
- Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021b. **DialogSum: A real-life scenario dialogue summarization dataset**. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. **Multi-fact correction in abstractive text summarization**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331, Online. Association for Computational Linguistics.
- Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022a. **QAFactEval: Improved QA-based factual consistency evaluation for summarization**. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.
- Alexander R Fabbri, Prafulla Kumar Choubey, Jesse Vig, Chien-Sheng Wu, and Caiming Xiong. 2022b. **Improving factual consistency in summarization with compression-based post-editing**. *arXiv preprint arXiv:2211.06196*.
- Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevyich. 2019. **Ranking generated summaries by correctness: An interesting but challenging application for natural language inference**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.
- Ronald Aylmer Fisher et al. 1937. The design of experiments. *The design of experiments.*, (2nd Ed).
- Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. **SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization**. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.
- Chih-Wen Goo and Yun-Nung Chen. 2018. **Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts**. In *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece, December 18-21, 2018*, pages 735–742. IEEE.
- Kung-Hsiang Huang, Kathleen McKeown, Preslav Nakov, Yejin Choi, and Heng Ji. 2022a. **Faking fake news for real fake news detection: Propaganda-loaded training data generation**. *arXiv preprint arXiv:2203.05386*.
- Kung-Hsiang Huang, ChengXiang Zhai, and Heng Ji. 2022b. **CONCRETE: Improving cross-lingual fact-checking with cross-lingual retrieval**. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1024–1035, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.
- Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. **Evaluating the factual consistency of abstractive text summarization**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.
- Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. **SummaC: Re-visiting NLI-based models for inconsistency detection in summarization**. *Transactions of the Association for Computational Linguistics*, 10:163–177.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. **BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. **Keep meeting summaries on topic: Abstractive multi-modal meeting summarization**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196, Florence, Italy. Association for Computational Linguistics.

- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Zhengyuan Liu and Nancy Chen. 2021. [Controllable neural dialogue summarization with personal named entity planning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 92–106, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourbon, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, Dennis Reidsma, and P. Wellner. 2005. The ami meeting corpus. In *Proceedings of Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*, pages 137–140. Noldus Information Technology.
- Rada Mihalcea and Paul Tarau. 2004. [TextRank: Bringing order into text](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.
- Feng Nan, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Kathleen McKeown, Ramesh Nallapati, Dejian Zhang, Zhiguo Wang, Andrew O. Arnold, and Bing Xiang. 2021. [Improving factual consistency of abstractive summarization via question answering](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6881–6894, Online. Association for Computational Linguistics.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. [Adversarial NLI: A new benchmark for natural language understanding](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Tatsuro Oya, Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. [A template-based abstractive meeting summarization: Leveraging summary and source text relationships](#). In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 45–53, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.
- Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. [Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.
- Liangming Pan, Wenhui Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. [Zero-shot fact verification by claim generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 476–483, Online. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2018. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*.
- Xiangru Tang, Arjun Nair, Borui Wang, Bingyao Wang, Jai Desai, Aaron Wade, Haoran Li, Asli Celikyilmaz, Yashar Mehdad, and Dragomir Radev. 2022. [CONFIT: Toward faithful dialogue summarization with linguistically-informed contrastive fine-tuning](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5657–5668, Seattle, United States. Association for Computational Linguistics.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. [Multilingual translation with extensible multilingual pretraining and finetuning](#).
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. [FEVER: a large-scale dataset for fact extraction and VERification](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

- Prasetya Utama, Joshua Bambrick, Nafise Moosavi, and Iryna Gurevych. 2022. [Falsesum: Generating document-level NLI examples for recognizing factual inconsistency in summarization](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2763–2776, Seattle, United States. Association for Computational Linguistics.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. [Fact or fiction: Verifying scientific claims](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.
- Sean Welleck, Ilya Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. [Neural text generation with unlikelihood training](#). In *International Conference on Learning Representations*.
- Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong. 2021. [Controllable abstractive dialogue summarization with sketch supervision](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5108–5122, Online. Association for Computational Linguistics.
- Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. [DocNLI: A large-scale dataset for document-level natural language inference](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. [BARTScore: Evaluating generated text as text generation](#). In *Advances in Neural Information Processing Systems*.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. [Pegasus: Pre-training with extracted gap-sentences for abstractive summarization](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML'20*. JMLR.org.
- Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D. Manning, and Curtis Langlotz. 2020b. [Optimizing the factual correctness of a summary: A study of summarizing radiology reports](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5108–5120, Online. Association for Computational Linguistics.

A Dataset Statistics

We present the detailed statistics of DIALOGSUM and SAMSUM in Table 5.

Dataset	# Dialogues	Avg. Dialogue Words	Avg. Summ. Words
DIALOGSUM	13,460	187.5	31.0
SAMSUM	16,369	124.1	23.4

Table 5: Statistics of DIALOGSUM and SAMSUM. We use the NLTK tokenizer to compute word counts for both datasets.

B Human Evaluation Details

In this section, we describe the details of our human evaluation. We recruit AMT workers from the United States for ensuring language fluency. Qualification requirements are set such that only workers who have an acceptance rate greater than 99% and have more than 10,000 accepted HITs in the past are allowed to work on our annotation task. To further ensure annotation quality, we conducted two rounds of annotations. In the first round, we launched 100 HITs to select high-quality annotators in the first round. 8 qualified annotators are selected to enter the second round to conduct the remaining evaluation. We set the reward to \$0.8 per HIT to encourage experienced annotators to participate. Our annotation interface is displayed in Figure 5.

For each HIT, annotators are provided with a piece of dialogue and a corresponding reference summary as well as two summaries generated from different systems, demonstrated on the left segment of the interface. Based on the summaries and the dialogue, annotators are tasked to answer three questions shown on the right segment of the interface, each of which corresponds to RECALL, PRECISION, and FAITHFULNESS. They need to determine which summary is better with regard to each prompt.

C Comparison with Other Data Augmentation Methods

We compared our MIXANDMATCH summary construction technique with other data augmentation methods, including back translation (BACKTRANSLATE) and paraphrasing (PARAPHRASING). For back translation, we use mBART-50 (Tang et al., 2020) to translate a summary from English to German and then back to English. For paraphrase generation, we use this open

source package⁴. The experimental results are summarized in Table 6. Training with MIXANDMATCH summaries achieves the highest scores on most metrics, indicating that our proposed method is the most effective in improving the factuality of the generated summaries.

D Hardware and Software configurations

All experiments are conducted on a Linux machine with NVIDIA V100. We use PyTorch 1.11.0 with CUDA 10.1 as the Deep Learning framework and utilize Transformers 4.19.2 to load all pre-trained language models.

E Validation Set Performance

We report the validation set performance of our proposed model in Table 7.

F Number of Parameters

We do not introduce additional parameters to the backbone language model, BART-LARGE. During training time, the number of parameters equals to the sum of the number of parameters in BART-LARGE and ROBERTA-LARGE. In inference time, since we do not need the NLI component, the number of parameters is the same as that of BART-LARGE.

G Scientific Artifacts

The licenses for all the models and software used in this paper are listed below in parentheses: BART (MIT License), FACTCC (BSD-3-Clause License), QAFACTEVAL (BSD-3-Clause License), BARTSCORE (Apache License 2.0), QUALS (MIT License), py-ROUGE (Apache License 2.0), NLTK (Apache License 2.0).

⁴<https://github.com/Vamsi995/Paraphrase-Generator>

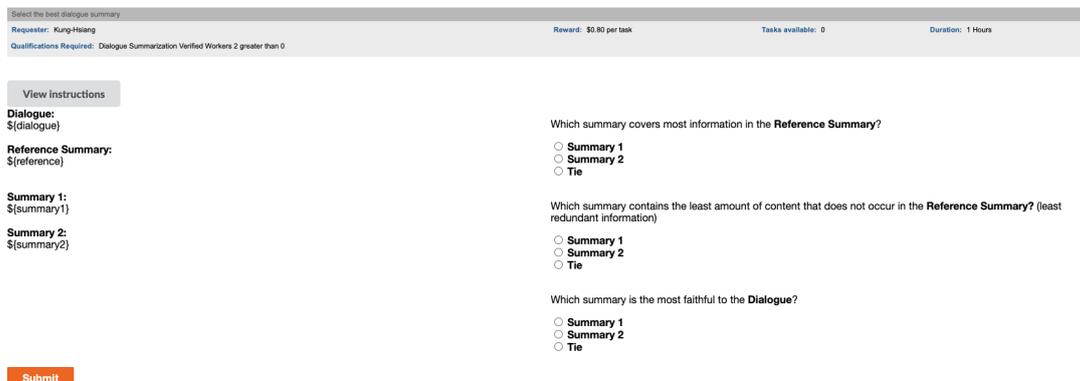


Figure 5: MTurk UI for our human evaluation.

Model	DIALOGSUM								SAMSUM							
	RL _F	RL _R	BS _F	BS _R	FC _F	FC _R	QS	QFE	RL _F	RL _R	BS _F	BS _R	FC _F	FC _R	QS	QFE
MIXANDMATCH	51.53	59.27	-2.012	-1.901	82.90	85.86	-1.130	2.399	49.73	53.95	-2.185	-2.143	63.47	63.15	-0.886	2.027
BACKTRANSLATE	50.41	58.22	-2.012	-2.032	83.20	84.23	-1.230	2.245	49.02	52.93	-2.234	-2.159	64.69	62.10	-1.230	1.984
PARAPHRASING	50.32	59.22	-2.133	-1.936	82.20	87.62	-1.198	2.333	49.23	53.94	-2.320	-2.178	64.78	63.98	-1.130	2.015

Table 6: Performance comparison on DIALOGSUM and SAMSUM with other positive data augmentation methods.

Model	DIALOGSUM						SAMSUM					
	RL _F	RL _R	BS _F	BS _R	FC _F	FC _R	RL _F	RL _R	BS _F	BS _R	FC _F	FC _R
SWING	48.45	51.27	-2.149	-2.169	71.36	70.65	50.61	53.74	-2.212	-2.134	64.27	64.56

Table 7: Validation set performance.

Language-Aware Multilingual Machine Translation with Self-Supervised Learning

Haoran Xu[♣], Jean Maillard[♡], Vedanuj Goswami[♡]

[♣]Johns Hopkins University, [♡]Meta AI

hxu64@jhu.edu
{jeanm, vedanuj}@meta.com

Abstract

Multilingual machine translation (MMT) benefits from cross-lingual transfer but is a challenging multitask optimization problem. This is partly because there is no clear framework to systematically learn language-specific parameters. Self-supervised learning (SSL) approaches that leverage large quantities of monolingual data (where parallel data is unavailable) have shown promise by improving translation performance as complementary tasks to the MMT task. However, jointly optimizing SSL and MMT tasks is even more challenging. In this work, we first investigate how to utilize **intra-distillation** to learn more *language-specific* parameters and then show the importance of these language-specific parameters. Next, we propose a novel but simple SSL task, **concurrent denoising**, that co-trains with the MMT task by concurrently denoising monolingual data on both the encoder and decoder. Finally, we apply **intra-distillation** to this co-training approach. Combining these two approaches significantly improves MMT performance, outperforming three state-of-the-art SSL methods by a large margin, e.g., 11.3% and 3.7% improvement on an 8-language and a 15-language benchmark compared with MASS, respectively¹.

1 Introduction

Multilingual machine translation (MMT) (Aharoni et al., 2019; Arivazhagan et al., 2019) comes with the problem of designing architectures where certain parameters are shared and certain parameters are more language-specific. In order to mitigate negative interference across languages, recent studies have investigated language-specific parameters, including searching for more language-specific parameters (Lin et al., 2021), or adding extra language-specific components to the original

¹Work done during an internship at Meta AI Research

¹Code is released at https://github.com/felixxu/CD_ID_MMT.

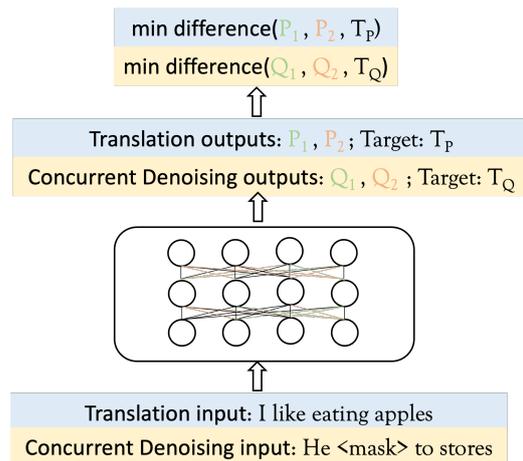


Figure 1: Concurrent denoising is a complementary task to the MMT task. Both tasks are applied with intra-distillation, where we forward pass model twice for the translation and masked inputs and each time we disable different subsets of parameters (illustrated by different colors). Then, for each task, we not only minimize the difference between the target and two outputs (e.g., minimize $\text{difference}(P_1, T_P)$ and $\text{difference}(P_2, T_P)$ in the MMT task), we also minimize the difference between two translated outputs as well as two denoised outputs (e.g., minimize $\text{difference}(P_1, P_2)$ for MMT).

model (Zhang et al., 2021; NLLB Team et al., 2022), or even utilizing language-specific pre-trained language models (Xu et al., 2021; Yarmohammadi et al., 2021). All these studies indicate the importance of language-specific parameters. *In this work, we first want to encourage parameters to have more language-specific attributes given a fixed model size.*

The difficulty of scaling MMT to low-resource and long-tail languages arises due to the scarcity of abundantly available parallel aligned data. Previous works (NLLB Team et al., 2022; Siddhant et al., 2022; Kim et al., 2021; Wang et al., 2020; Siddhant et al., 2020) try to tackle this by collecting massive amounts of monolingual data and using various types of self-supervised learning (SSL)

objectives, such as denoising AutoEncoder (DAE) (Liu et al., 2020) or Masked Sequence to Sequence (MASS) (Song et al., 2019) as auxiliary tasks to co-train with the MMT task, to compensate for the scarcity of parallel data for low-resource languages. *Following this line, we secondly aim to propose a more effective SSL objective.*

With the goal of learning language-aware MMT models and designing more effective SSL methods for MMT, we introduce two approaches. The first approach is **Intra-Distillation (ID)** (Xu et al., 2022), which performs a forward pass through the model K times², and in each pass disables a different set of parameters. This enforces consistent contributions between these disabled parameters by minimizing the difference between the K outputs. ID was originally proposed by Xu et al. (2022) to achieve a balanced parameter contribution in a model. Here, we study the effectiveness of ID in learning language-specific parameters for MMT models. Next, we introduce **Concurrent Denoising (CD)** which is an auxiliary self-supervised task jointly trained with the MMT task. CD predicts the same masked sentences both on the encoder and decoder side with a shared projection layer to facilitate the consistent understanding between encoder and decoder representations. We show that CD outperforms several state-of-the-art SSL methods for translation. Finally, we apply ID to our co-training scheme to further improve the MMT performance by learning more language-specific parameters. The overall framework is illustrated in Figure 1 and we summarize our main contributions below.

- We propose a method to quantify the degree of language-specificity of all parameters (Section 2) and perform a thorough analysis to demonstrate that intra-distillation helps the model learn more language-specific parameters. These parameters contribute more towards a specific language to improve the overall model generalization performance (Section 3).
- We propose the **concurrent denoising** SSL method and demonstrate its improvements over other existing SSL objectives for MMT. Moreover, we introduce a co-training method of MMT and CD with the help of intra-

distillation and shows the strong effectiveness of ID in improving MMT+SSL multi-task optimization (Section 4).

- We conduct extensive experiments on a 8-language dataset and a larger 15-language multilingual dataset, and demonstrate that MMT with concurrent denoising and intra-distillation outperforms multiple strong state-of-the-art methods (Section 5).

2 Preliminary

2.1 Quantify Language-Specific Parameters

Parameter sensitivity is a measure of the impact on the loss when a specific parameter of a model is zeroed-out. It is widely used in pruning as importance score (Ding et al., 2019; Molchanov et al., 2019; Lubana and Dick, 2021). A parameter can express different sensitivities depending on the language of the input data. Those parameters that have high sensitivity to a specific language but low sensitivity to others, are language-specific parameters. We define the i^{th} parameter in a model parameterized by Θ as $\theta_i \in \mathbb{R}$. We further define $\Theta_i = [0, \dots, 0, \theta_i, 0, \dots, 0] \in \mathbb{R}^{|\Theta|}$ and $\Theta_{-i} = [\theta_1, \dots, \theta_{i-1}, 0, \theta_{i+1}, \dots, \theta_{|\Theta|}] \in \mathbb{R}^{|\Theta|}$. The sensitivity of the i^{th} parameter given input batch b_l from language l is formulated as

$$\mathcal{S}(\theta_i, b_l) = |\mathcal{L}(\Theta, b_l) - \mathcal{L}(\Theta_{-i}, b_l)|, \quad (1)$$

where $\mathcal{L}(\cdot)$ is the loss function given the input batch and parameters. Then, we use a first-order Taylor decomposition to approximate the sensitivity of any arbitrary parameters. Equation 1 then becomes

$$\mathcal{S}(\theta_i, b_l) \approx |\Theta_i^T \nabla_{\Theta} \mathcal{L}(\Theta, b_l)|, \quad (2)$$

where $\nabla_{\Theta} \mathcal{L}(\Theta, b_l)$ is the gradient of the loss with respect to the model parameters. In our implementation, we randomly pick 500 batches and feed them to the model to retrieve the gradients and compute the average sensitivity. We then have

$$\mathcal{S}(\theta_i, \mathcal{B}_l) \approx \frac{1}{|\mathcal{B}_l|} \sum_{b_l \in \mathcal{B}_l} |\Theta_i^T \nabla_{\Theta} \mathcal{L}(\Theta, b_l)|, \quad (3)$$

where \mathcal{B}_l is a set containing 500 random b_l batches.

Now, we propose to quantify the degree of language-specificity of θ_i with respect to language l by measuring the relative sensitivity difference between language l and the other languages as

$$D(\theta_i, l) = \frac{\mathcal{S}(\theta_i, \mathcal{B}_l) - \mathcal{S}(\theta_i, \mathcal{B}_{-l})}{\mathcal{S}(\theta_i, \mathcal{B}_{-l}) + \sigma}, \quad (4)$$

²We use $K = 2$ in this work.

where \mathcal{B}_{-l} represents the set composed of mixed batches from all training languages except for the language l , and σ is a very small positive constant³. The larger $D(\theta_i, l)$ is, the more language-specific θ_i is to language l .

2.2 Intra-Distillation

A model with more balanced parameter sensitivity distribution shows better generalization (Liang et al., 2022). Xu et al. (2022) propose intra-distillation (ID) as an effective task-agnostic training method, aiming to encourage all parameters to contribute equally, which improves performance when model size is fixed. However we argue that, in the multilingual setting, ID actually helps the model learn more language-specific parameters resulting in improved performance. Given an input batch, ID needs to forward pass the model K times to obtain K outputs and each time a random subset of parameters is zeroed out. The core idea of ID is to minimize the difference of these K outputs to approximate minimizing the contribution gap of the parameters that are zeroed-out, because the K outputs are forced to be the same with different zeroed parameters. Let $\{p_1, \dots, p_i, \dots, p_K\}$ denote the K outputs. Note that the outputs are probability distributions in the translation and denoising task. The ID loss is then formulated by the X-divergence (Xu et al., 2022) to minimize the difference of K outputs as

$$\mathcal{L}_{id} = \frac{1}{K} \sum_{i=1}^K \text{KL}(p_i \parallel \bar{p}) + \text{KL}(\bar{p} \parallel p_i) \quad (5)$$

$$\text{where } \bar{p} = \frac{1}{K} \sum_{i=1}^K p_i$$

Let the original task loss be \mathcal{L}_i for the i^{th} pass. Then, the total loss is a combination of the original task losses and ID loss, given as

$$\min \frac{1}{K} \sum_{i=1}^K \mathcal{L}_i + \alpha \mathcal{L}_{id} \quad (6)$$

where α is a hyper-parameter to control the strength of ID. Similar to Xu et al. (2022), we use dropout to simulate *zeroed-out* parameters in all experiments.

Although the explanation for better performance after using ID is that the model parameters become more balanced, it is unclear how parameter contributions to different languages change after

³ σ is 1e-8 in our implementation.

applying ID in a multilingual (multitask) setting. For instance, do parameters become more language-agnostic and shareable across all languages, or do they become more language-specific? We investigate this in more details in Section 3.2.

3 Language-Aware MMT Models

In this section, we study how parameters can be prompted to be more language-specific by applying **intra-distillation**, which improves the model generalization performance. Specifically, certain parameters become more language-specific and tend to contribute more to their specific language and less to others. We demonstrate the importance of language-specific parameters by showing how much they can contribute in pruning experiments. We begin our analysis from a case study on MMT experiments with an 8-language dataset (M8), and then scale up our experiments to 15 languages (M15) with larger data size in Section 5. Here, we show results and analysis on $\text{xxx} \rightarrow \text{eng}$ directions. Similar discussions for $\text{eng} \rightarrow \text{xxx}$ directions are shown in Appendix A.

3.1 Experiments on Intra-Distillation

Dataset and Training We train MMT models with and without ID on the M8 dataset⁴. M8 is composed of Nigerian Fulfulde (fuv, 18K parallel sentences), Kimbundu (kmb, 82K), Ganda (lug, 278K), Chewa (nya, 693K), Swahili (swh, 2.1M), Umbundu (umb, 193K), Wolof (wol, 9K) and Zulu (zul, 1.2M). Datasets are extracted from the primary bitext used by the NLLB-200 model (NLLB Team et al., 2022). For ID, we pass the model twice ($K = 2$) considering the computational cost, and set α as 5 suggested by Xu et al. (2022). We use FLORES-200 as our dev and test sets (NLLB Team et al., 2022). Our model training is based on the Transformer_{big} architecture (Vaswani et al., 2017) with 32K vocabulary jointly trained by SentencePiece (Kudo

⁴The languages were selected in order to have a realistic dataset reflecting a specific use case. Multilingual training is crucial for languages that are low-resource, as is the case for many languages of Africa. We chose two different language groupings from the African continent: Benue-Congo languages (Kimbundu, Ganda, Chewa, Swahili, Umbundu, Zulu) and North-Central Atlantic languages (Nigerian Fulfulde, Wolof). While these languages may all belong to the Atlantic-Congo family, this is an extremely large, varied, and under-researched family, with Glottolog recording over 1,400 languoids in it – compare this to under 590 languoids recorded for the Indo-European family.

and Richardson, 2018). We report sacreBLEU scores (spm tokenizer) (Post, 2018).

Results Following NLLB Team et al. (2022), we categorized a language as *low-resource* if there are fewer than 1M parallel sentences, and as *very low-resource* if fewer than 100K (very low-resource is not the subset of low-resource). Otherwise, the language is considered as *high-resource*. We report the average BLEU scores for each of the three categories. In Table 1, we show that MMT with ID outperforms the regular MMT model by a large margin on all three categories by +1.21 BLEU averaged across all languages.

Method	High	Low	Very Low	All
Regular	31.70	12.57	6.92	15.94
Intra-Distillation	33.30	13.63	8.05	17.15

Table 1: M8 results on $\text{xxx} \rightarrow \text{eng}$ comparing regular MMT and MMT with ID. We observe that MMT with ID outperforms regular MMT by a significant margin.

3.2 Language-Specific or Language-Agnostic?

Next, we study whether parameter contributions are more language-specific or just shareable across all languages after ID. Given the i^{th} language l_i , we compute the sensitivities (Equation 3) of all parameters and flatten them into a list. Then, we calculate the Pearson correlation coefficients (PCC) p_{ij} between sensitivity lists of any arbitrary pair of languages l_i and l_j . A lower p_{ij} indicates that there are more contribution (sensitivity) disagreements between languages l_i and l_j . We plot a heat map to visualize p_{ij} for every language pair. Taking into account that the top 10% parameters usually dominate the contribution (Xiao et al., 2019; Sanh et al., 2020), we consider the performance of two groups of parameters, high-sensitive (top 10% most sensitive) and low-sensitive (the remaining 90%) parameters, respectively. Figure 2 shows that all p_{ij} in both groups become lower, indicating there is lower sensitivity similarity between different languages for the same parameters, which means the model becomes more language-specific after ID. For instance, sensitivity similarity between `zul` and `wol` drops from 0.67 to 0.57 in the low-sensitive group. However, the p_{ij} of low-sensitive parameters drops much more than high-sensitive ones, and high-sensitive parameters still hold high similarity (over 0.9). Thus, low-sensitive parameters mostly have language-specific properties while high-sensitive parameters

tend to play ‘language-agnostic’ roles. Overall, parameters are more language-specific after ID⁵. In fact, learning more language-specific parameters through ID in MMT leads to better performance as seen in Section 3.1. These findings align with the results of recent studies which investigate language-specific parameters (Lin et al., 2021; Zhang et al., 2021; NLLB Team et al., 2022), indicating the importance of language-specific parameters.

3.3 The Importance of Language-Specific Parameters

Here, we study the reason *why language-specific parameters are important and how much they contribute*. To investigate this, we first measure the degree of language-specificity of all parameters based on Equation 4. We explore the contribution of language-specific parameters with respect to the BLEU scores. Then, we conduct one-shot unstructured pruning with respect to BLEU scores in order of the degree of language-specificity for both models with and without ID, starting with the least language-specific parameters⁶. As more parameters are pruned, a slower performance drop means that a higher contribution comes from the remaining more language-specific parameters. Figure 3 shows the average BLEU drop across 8 languages versus the percentage of parameters pruned. After pruning the less language-specific parameters, the rest of the more language-specific parameters in the model with ID are able to preserve better performance, indicating the importance of more language-specific parameters.

4 Proposed Self-Supervision Method

We extend our study of language-awareness to MMT models co-trained with self-supervised objectives that have been shown to improve translation performance. We first propose a simple but effective self-supervised learning objective, **concurrent denoising** (CD), and then investigate the effectiveness of ID in helping improve multi-task optimization challenges of co-training CD and

⁵The overall parameter contribution is still more balanced as claimed in Xu et al. (2022). We leave further discussion on this to Appendix B.

⁶Note that, as shown in Figure 2b, the 10% most sensitive parameters are highly language-agnostic. They are easy to classify as less language-specific and can be pruned, but pruning them would lead to near-random performance ($\text{BLEU} \approx 0$), making it hard to evaluate the importance of more language-specific parameters. Thus, we keep the top 10% sensitive parameters and prune the rest of parameters that display a more language-specific behavior.

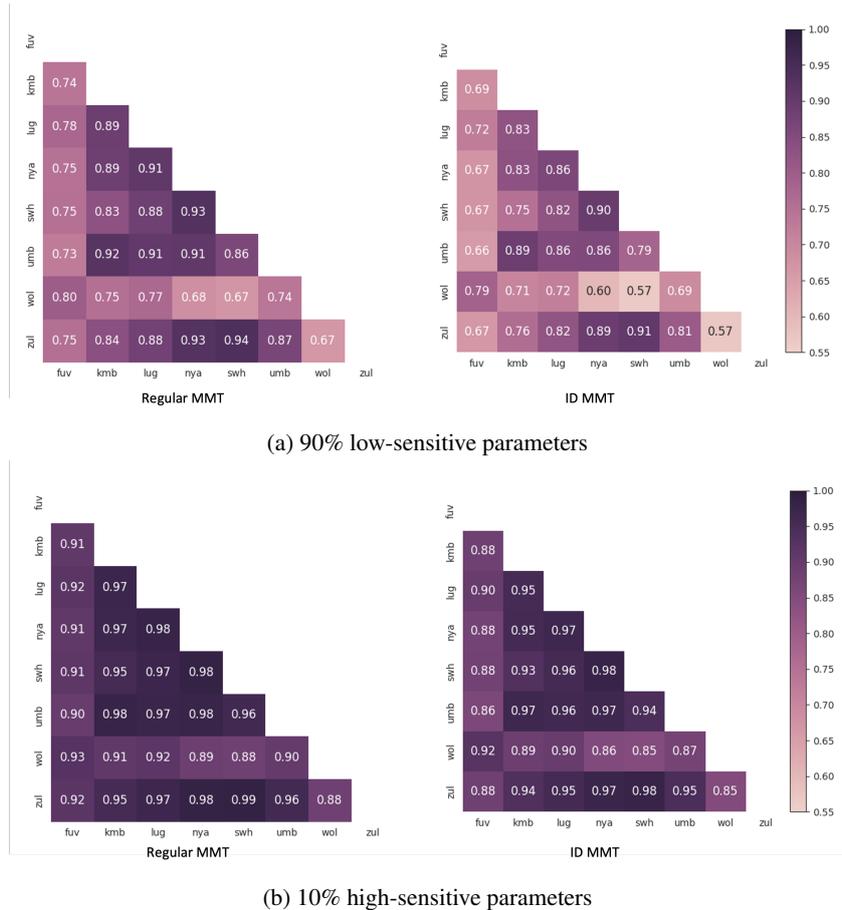


Figure 2: PCC between the lists of parameter sensitivity of every language (left for regular MMT and right for MMT with ID). We show contribution similarity of two groups of parameters, i.e., top 10% high-sensitive parameters and the remaining 90% parameters. The lower score between two languages represents the less similarity of parameter contributions for these two languages, which means more contribution disagreements and parameters are more language-specific.

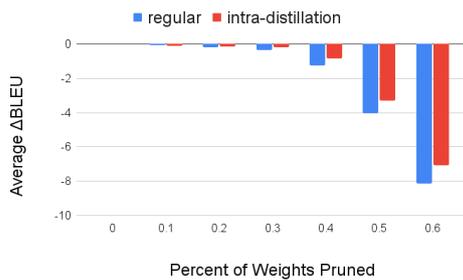


Figure 3: Change in average $xxx \rightarrow eng$ translation performance across 8 languages versus pruning ratio. Models are pruned starting with the least language-specific parameters.

MMT tasks together by learning more language-specific parameters.

4.1 Concurrent Denoising

Self-supervised learning objectives usually involve sentence denoising either on the encoder side, such

as MLM (Devlin et al., 2019), or on the decoder side, such as DAE (Liu et al., 2020). Jointly denoising sentences on both the encoder and the decoder sometimes is better than a single denoising objective (Wang et al., 2020; Kim et al., 2021) for MMT, but the training cost is doubled as we need to calculate the loss for the same monolingual sentence twice (masked in two different ways). We propose **concurrent denoising**, a self-supervised task that denoises a single masked sentence both on the encoder and decoder sides, which not only reduces the training time but also improves the language understanding of the model to result in better MMT performance.

We add noise to the monolingual data by whole-word masking (Devlin et al., 2019), where we randomly replace $r_m\%$ words with the special token `<mask>`. During the replacement process, each word has a 10% chance not to be masked, and another 10% chance to be replaced with other

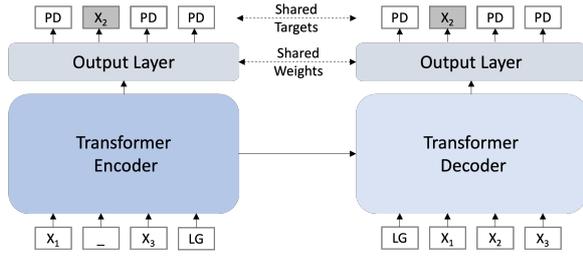


Figure 4: Concurrent denoising. In the example input sentence ‘ $X_1 X_2 X_3$ ’, the token X_2 is masked. The encoder and decoder share the same output projection layer and target tokens to predict the masked token. We only calculate the loss for the masked token prediction. PD represents the target token loss padding and LG is a special language token.

random tokens. The encoder and decoder use a shared output layer to reconstruct the original sentence. The loss for the encoder and decoder side are denoted as \mathcal{L}_e and \mathcal{L}_d respectively⁷. The total training loss combining translation loss \mathcal{L}_{MMT} and two self-supervised losses is

$$\mathcal{L} = \mathcal{L}_{MMT} + \mathcal{L}_e + \mathcal{L}_d. \quad (7)$$

Concurrent denoising is illustrated in Figure 4. Two key differences between our concurrent denoising method and regular MLM or DAE methods are worth highlighting.

Shared Output Projection Since the decoder has an output projection layer while the encoder does not, Wang et al. (2020) train the encoder with MLM by using an additional projection layer. However, we utilize the decoder projection layer as a shared layer for both encoder and decoder to reconstruct the sentence, which significantly reduces model parameters. This is because the projection layer is usually large when we have a large vocabulary size. We show the effect of using a shared projection layer in Appendix C.

Shared Target Tokens Since the output representations of the encoder and decoder are fed to the same projection layer, we want them to predict the same target token at the same position for the stability of the projection layer training. To achieve this, we carefully design our language token positions. Instead of only prepending a special language token at the beginning of the source sentence (Johnson et al., 2017), we append

⁷Unlike DAE training on the decoder side, we zero out the losses which predict non-masked tokens.

the special language token on the source side and also prepend it on the decoder side (As shown in Figure 4). This design also applies to MMT. In this way, we can avoid the encoder and decoder from predicting the same token at different positions.

4.2 Concurrent Denoising with Intra-Distillation

We investigate whether ID helps concurrent denoising to improve overall performance. We apply ID to the co-training of CD and MMT tasks. Following Equation 6 and 7, our final loss is

$$\mathcal{L} = \frac{1}{K} \left(\sum_{i=1}^K \mathcal{L}_{MMT_i} + \sum_{i=1}^K \mathcal{L}_{e_i} + \sum_{i=1}^K \mathcal{L}_{d_i} \right) + \alpha (\mathcal{L}_{id_MMT} + \mathcal{L}_{id_e} + \mathcal{L}_{id_d}), \quad (8)$$

where \mathcal{L}_{id_MMT} , \mathcal{L}_{id_e} and \mathcal{L}_{id_d} respectively represent the ID loss for translation, encoder denoising and decoder denoising (i.e., \mathcal{L}_{id_e} minimizes the difference of the K encoder outputs based on Equation 5, etc.). The i index in \mathcal{L}_{e_i} , \mathcal{L}_{MMT_i} and \mathcal{L}_{d_i} indicates that these losses are for the i^{th} forward pass.

5 MMT+SSL Experiments

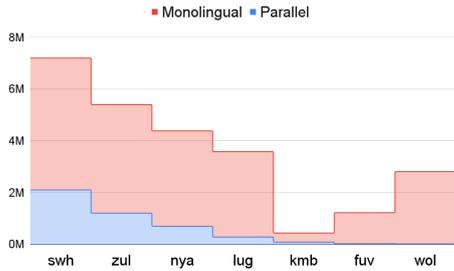
5.1 Baselines

We consider three strong baselines. All baselines are our own implementation following the settings from the original papers.

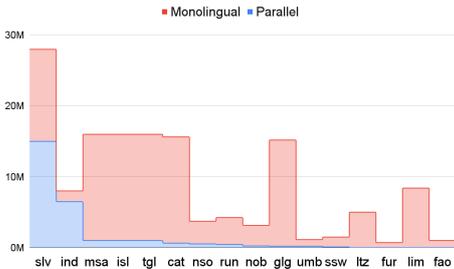
DAE NLLB Team et al. (2022) learn the effects of the causal language modeling (LM) and DAE objectives (Liu et al., 2020). Since they find that DAE performs better than LM or LM+DAE, we only compare our methods with the DAE objective.

DAE+MLM Wang et al. (2020) study a multi-task learning framework which jointly trains the MMT, MLM and DAE objectives, where MLM and DAE reconstruct sentences noised by different masking methods. Kim et al. (2021) also investigate the effectiveness of ELECTRA (Clark et al., 2020). They conclude that DAE+MLM is better than DAE+ELECTRA.

MASS Siddhant et al. (2020) and Siddhant et al. (2022) utilize MASS (masked sequence to sequence pre-training) (Song et al., 2019) to improve the MMT performance. Similar to MLM which predicts masked tokens on the encoder side, MASS masks a fragment of a sentence and predicts the masked fragment but on the decoder side.



(a) M8 dataset



(b) M15 dataset

Figure 5: The statistics of monolingual and parallel data for M8 and M15 are presented. The languages are arranged in descending order of parallel data size.

5.2 Datasets

In addition to the M8 dataset described in Section 3.1, we also build a larger dataset (M15), covering 15 languages. In composing this dataset, we take into account linguistic diversity and data size. The resulting dataset has languages from 6 linguistic families and a balanced number of high-resource, low-resource and very low-resource languages. Detailed information on this dataset is in Appendix D. We randomly sample at most 3M monolingual samples per language for M8, and 15M for M15. The distribution of monolingual data and parallel data for M8 and M15 is shown in Figure 5. Note that we also use parallel data for self-supervised learning, so the true monolingual data size includes bitext data. We use the FLORES-200 dataset for evaluation. All datasets come from the primary bitext and monolingual data used for the NLLB-200 model (NLLB Team et al., 2022).

5.3 Data Sampling

We use a data sampling temperature of $T = 1$ suggested by NLLB Team et al. (2022) to train on the MMT objective. For monolingual data, we use a temperature of $\frac{10}{7}$ to balance the SSL training, as suggested by Liu et al. (2020). During co-training, we mix the two sources in an equal ratio (50% monolingual data (including bitext used for SSL

Method	High	Low	Very Low	All
<i>M8 results</i>				
Regular MMT	31.70	12.57	6.92	15.94
+DAE (NLLB Team et al., 2022)	32.69	13.38	7.57	16.75
+DAE+MLM (Wang et al., 2020)	33.05	13.93	8.20	17.27
+MASS (Siddhant et al., 2020)	32.64	13.03	6.79	16.37
+CD (ours)	32.92	13.94	8.38	17.29
+CD+ID (ours)	35.16	15.18	9.18	18.69
<i>M15 results</i>				
Regular MMT	39.87	35.20	24.45	33.17
+DAE (NLLB Team et al., 2022)	38.46	34.05	26.23	32.91
+DAE+MLM (Wang et al., 2020)	38.60	34.00	25.49	32.70
+MASS (Siddhant et al., 2020)	38.53	33.93	22.79	31.75
+CD (ours)	39.23	34.88	28.21	34.11
+CD+ID (ours)	39.58	35.53	29.43	34.85

Table 2: Overall $xxx \rightarrow eng$ BLEU for M8 and M15.

training) with self-supervision and 50% parallel data).

5.4 Training and Evaluation Details

All experiments consider both the $eng \rightarrow xxx$ and $xxx \rightarrow eng$ directions and use the Transformer architecture (Vaswani et al., 2017). We use Transformer_{big} (242M parameters, 6 layers, 16 heads, 1,024 hidden dimension, 4,096 FFN dimension) for M8 experiments. For M15 experiments, we double the layers of Transformer_{big} (418M parameters). We use a vocabulary of size 32k for both M8 and M15 with SentencePiece (Kudo and Richardson, 2018). The batch size is 30K tokens. We warm-up for the first 8K steps. We set the total training steps to 100K and 300k for M8 and M15 respectively, with patience set to 10 for early stopping. We forward pass the model twice ($K=2$) to conduct ID. We set the ID weight $\alpha = 5$. During concurrent denoising, the masking ratio is set to $r_m = 30\%$. We also show the effect of masking ratio in Appendix E. During generation, we use beam search with a beam size of 5 and a length penalty of 1.0. All models are evaluated with sacreBLEU (spm tokenizer).

Method	High	Low	Very Low	All
<i>M8 results</i>				
Regular MMT	34.14	11.47	5.75	15.71
+DAE (NLLB Team et al., 2022)	34.35	11.41	5.79	15.74
+DAE+MLM (Wang et al., 2020)	34.48	11.45	5.20	15.64
+MASS (Siddhant et al., 2020)	34.02	11.53	4.75	15.46
+CD (ours)	34.87	11.50	5.90	15.94
+CD+ID (ours)	35.83	11.90	5.81	16.37
<i>M15 results</i>				
Regular MMT	38.44	31.62	16.46	28.84
+DAE (NLLB Team et al., 2022)	37.46	30.86	18.39	28.90
+DAE+MLM (Wang et al., 2020)	37.99	30.98	18.05	29.01
+MASS (Siddhant et al., 2020)	38.19	31.20	17.88	29.09
+CD (ours)	37.74	30.94	19.04	29.24
+CD+ID (ours)	38.29	31.71	19.43	29.81

Table 3: Overall $eng \rightarrow xxx$ BLEU for M8 and M15.

5.5 Results

The overall results for the $xxx \rightarrow eng$ and $eng \rightarrow xxx$ directions are shown in Tables 2 and 3. For both M8 and M15, and both translation directions, concurrent denoising is better than all aforementioned baselines, and combining it with ID further improves upon the baselines by an even larger margin. For instance, our method outperforms MASS by 11.3% and 3.7% on M8 and M15 respectively, averaged across all languages and directions. We also show the effectiveness of ID on other objectives like DAE in Section 6.1, but the results are subpar compared to CD+ID.

Aligned with the findings of Wang et al. (2020); Kim et al. (2021), we observe that DAE+MLM is better than DAE alone in M8 $xxx \rightarrow eng$, but the improvements become very minor when it comes to M8 $eng \rightarrow xxx$ or when scaling to 15 languages. MASS performs similarly or better than DAE in the $eng \rightarrow xxx$ but worse in the $xxx \rightarrow eng$.

In M15, high-resource languages perform slightly worse with SSL methods compared to the MMT only baseline, but improves other categories, similar to the observations of NLLB Team et al. (2022). It does not occur on M8, possibly due to the smaller dataset size allowing for sufficient model capacity to learn from additional monolingual data.

Note that the effectiveness of SSL such as DAE and MASS is not as pronounced as reported by Wang et al. (2020) and Siddhant et al. (2022). However, it is necessary to consider the for domain mismatch between the training and evaluation data. As demonstrated by Siddhant et al. (2022), a significant decline in performance can occur when either monolingual or bitexts diverge from the evaluation domain. In our study, the training data is sourced from NLLB-200 and FLORES-200, which encompasses a wide range of domains. we hypothesize that this contributes to the observed lessened effectiveness of SSL techniques in our experiments.

6 Analysis

6.1 Ablation Study

The final loss, described in Equation 8, has 6 loss terms. Except for the translation loss, we ablate the relative contribution of all the other 5 loss terms to the translation task performance. In Table 4, we show the results of this ablation study on M8 $xxx \rightarrow eng$ directions. Method ① is the regular MMT model and method ② is ID training only for

MMT (the same result as in Section 3.1). Method ③ is the same as the MMT+DAE method. With the help of ID for the decoder denoising (method ④) and an additional ID for translation (method ⑤), translation performance can respectively obtain +0.41 and +0.98 BLEU on average compared to ③. Note that method ⑤ is the MMT+DAE+ID method. Compared to our MMT+CD+ID method, it substantially underperforms our method (17.73 vs. 18.69), which shows that our method could better stimulate the potential of ID. The results for methods ⑥, ⑦ and ⑧ indicate the effectiveness of encoder denoising with CD and applying ID. Overall, the translation performance improves by including all the loss terms.

Method	Avg. BLEU
① \mathcal{L}_{MMT}	15.94
② $\mathcal{L}'_{MMT} + \mathcal{L}_{id_MMT}$	17.15
③ $\mathcal{L}_{MMT} + \mathcal{L}_d$	16.75
④ $\mathcal{L}'_{MMT} + \mathcal{L}'_d + \alpha \mathcal{L}_{id_d}$	17.16
⑤ $\mathcal{L}'_{MMT} + \mathcal{L}'_d + \alpha(\mathcal{L}_{id_d} + \mathcal{L}_{id_MMT})$	17.73
⑥ $\mathcal{L}_{MMT} + \mathcal{L}_e + \mathcal{L}_d$	17.29
⑦ $\mathcal{L}'_{MMT} + \mathcal{L}'_e + \mathcal{L}'_d + \alpha(\mathcal{L}_{id_d} + \mathcal{L}_{id_e})$	17.59
⑧ $\mathcal{L}'_{MMT} + \mathcal{L}'_e + \mathcal{L}'_d + \alpha(\mathcal{L}_{id_d} + \mathcal{L}_{id_e} + \mathcal{L}_{id_MMT})$	18.69

Table 4: Ablation study on loss terms. For simplicity, we use \mathcal{L}' to represent the mean loss of K forward pass, e.g., $\mathcal{L}'_e = \frac{1}{K} \sum_{i=1}^K \mathcal{L}_{e_i}$.

6.2 Language-Specific Parameters for SSL

In Section 3, we observed that ID helps MMT learn more language-specific parameters and improve model generalization. We are also interested in understanding 1) whether the model also learns more language-specific parameters for the SSL task (here we investigate CD), and 2) what is the relationship of parameter contribution between MMT and SSL tasks for the same language. We use the $xxx \rightarrow eng$ direction of the M8 dataset as an example to study these questions.

In Figure 6, we plot a heat map to illustrate the PCC of all parameter sensitivities between every language pair. As expected, parameter sensitivity similarity becomes lower for all languages, which means there are more language-specific parameters when we train SSL methods with ID. For the second question, in Figure 7 we show the parameter sensitivity similarity between the MMT and CD tasks for each language. The contribution similarity becomes higher between the two tasks for every language with ID. This is expected, since the losses of MMT and CD have the same objective on the decoder side, i.e., text generation conditioned on

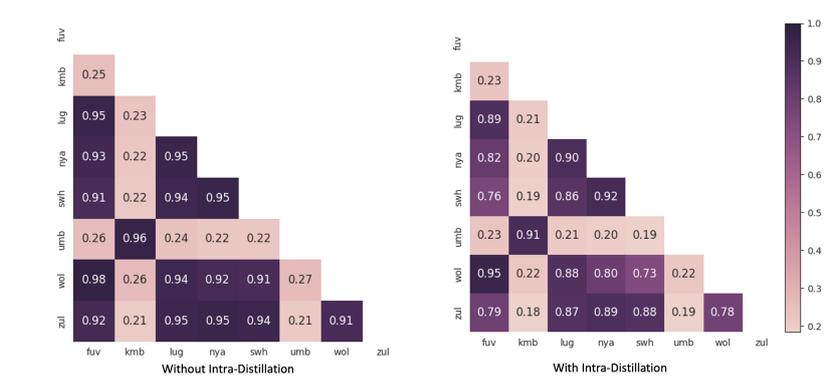


Figure 6: Parameter contribution similarity among all language pairs, evaluated by PCC for the CD task before (left) and after (right) ID.

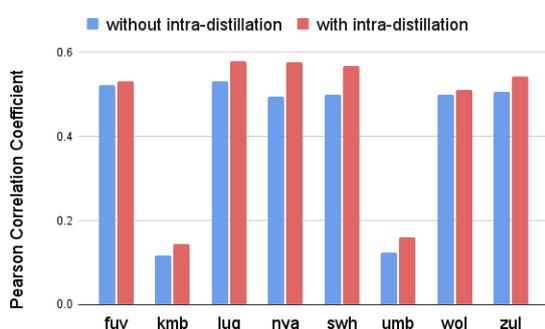


Figure 7: Parameter contribution similarity between MMT and CD for each language with and without ID.

another text. This is also another reason why SSL tasks can help multilingual translation.

7 Conclusions

We show extensive analysis that intra-distillation training helps multilingual translation by learning more language-specific parameters. We propose concurrent denoising, improving upon multiple state-of-the-art self-supervised learning methods. Moreover, we demonstrate that applying intra-distillation to the above co-training scheme offers further improvements to translation performance.

Limitations

Although we show improvements using our methods on multiple languages from diverse language families on multilingual machine translation, it should be noted that the generalizability of our findings to other multi-task learning settings, such as those involving the combination of tasks such as named entity recognition, part-of-speech tagging, and question answering, remains uncertain. This is due to

the fact that our study primarily focused on the utilization of intra-distillation to learn task-specific parameters on multilingual machine translation and did not investigate the aforementioned tasks. Furthermore, with intra-distillation we need to perform more than one forward pass, leading to a trade-off between higher performance and increased training time – which, for many use-cases, could be arguably acceptable.

Acknowledgements

We would like to thank anonymous reviewers for their valuable comments. We also thank Alex Guo, Simeng Sun, and Weiting Tan for their helpful suggestions.

References

- Roei Aharoni, Melvin Johnson, and Orhan Firat. 2019. [Massively multilingual neural machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.
- Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. [Massively multilingual neural machine translation in the wild: Findings and challenges](#). *CoRR*, abs/1907.05019.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. [Electra: Pre-training text encoders as discriminators rather than generators](#). In *International Conference on Learning Representations*.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Xiaohan Ding, Xiangxin Zhou, Yuchen Guo, Jungong Han, Ji Liu, et al. 2019. Global sparse momentum sgd for pruning very deep neural networks. *Advances in Neural Information Processing Systems*, 32.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. [Google’s multilingual neural machine translation system: Enabling zero-shot translation](#). *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andres Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. 2021. Scalable and efficient moe training for multitask multilingual models. *arXiv preprint arXiv:2109.10465*.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Chen Liang, Haoming Jiang, Simiao Zuo, Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Tuo Zhao. 2022. [No parameters left behind: Sensitivity guided adaptive learning rate for training large transformer models](#). In *International Conference on Learning Representations*.
- Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. [Learning language specific sub-network for multilingual machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 293–305, Online. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Ekdeep Singh Lubana and Robert Dick. 2021. A gradient flow framework for analyzing network pruning. In *International Conference on Learning Representations*.
- Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11264–11272.
- NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389.
- Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. [Leveraging monolingual data with self-supervision for multilingual neural machine translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2827–2835, Online. Association for Computational Linguistics.
- Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. 2022. Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *arXiv preprint arXiv:2201.03110*.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020. [Multi-task learning for multilingual neural machine translation](#). In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1022–1034, Online. Association for Computational Linguistics.

Xia Xiao, Zigeng Wang, and Sanguthevar Rajasekaran. 2019. Autoprune: Automatic network pruning by regularizing auxiliary parameters. *Advances in neural information processing systems*, 32.

Haoran Xu, Philipp Koehn, and Kenton Murray. 2022. The importance of being parameters: An intra-distillation method for serious gains. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 170–183, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. BERT, mBERT, or BiBERT? a study on contextualized embeddings for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6663–6675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mahsa Yarmohammadi, Shijie Wu, Marc Marone, Haoran Xu, Seth Ebner, Guanghui Qin, Yunmo Chen, Jialiang Guo, Craig Harman, Kenton Murray, Aaron Steven White, Mark Dredze, and Benjamin Van Durme. 2021. Everything is all it takes: A multipronged strategy for zero-shot cross-lingual information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1950–1967, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

A Analysis of Intra-Distillation for $\text{eng} \rightarrow \text{xxx}$

Method	High	Low	Very Low	All
Regular	34.14	11.47	5.75	15.71
Intra-Distillation	35.05	13.79	5.69	16.07

Table 5: M8 $\text{eng} \rightarrow \text{xxx}$ results of regular MMT and MMT with intra-distillation.

Similar to Section 3, the model with intra-distillation outperforms the regular MMT model by a large margin in the $\text{eng} \rightarrow \text{xxx}$ direction, as shown in Table 5. We still use a heat map to visualize the PCC of parameter sensitivity lists among every language pair in the $\text{eng} \rightarrow \text{xxx}$ direction. In Figure 8, we show that contribution similarity becomes lower as well, which means that the model also learns more language-specific parameters.

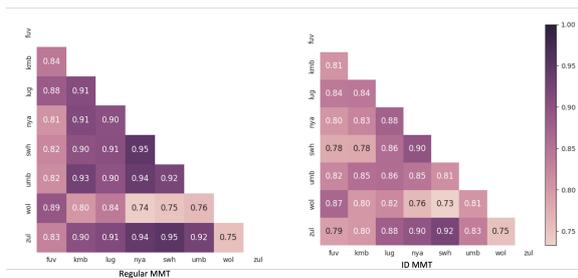


Figure 8: PCC between the list of all parameter sensitivities across every language in the M8 $\text{eng} \rightarrow \text{xxx}$ experiments. We compare the similarity between MMT with and without intra-distillation.

We also evaluate the importance of these language-specific parameters by following the same settings in Section 3.3. We conduct one-shot unstructured pruning, starting with the least language-specific parameters. We again see that the average BLEU scores of 8 languages from the model trained with intra-distillation drop slower after more parameters are pruned, indicating that these language-specific parameters learned by intra-distillation are able to preserve more performance.

B More Balanced Parameter Contribution

We compute the sensitivity of all parameters by feeding a set of batches \mathcal{B} that contains all language data in the M8 $\text{xxx} \rightarrow \text{eng}$ experiment. We illustrate parameter sensitivity distribution in Figure 10. Aligned with the findings in Xu et al.

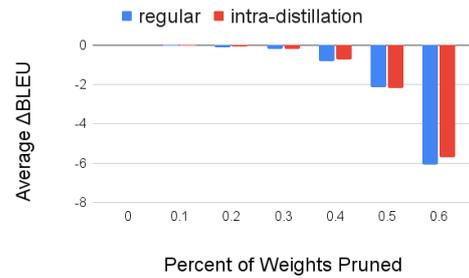


Figure 9: Change of model performance averaged across 8 languages against increasing pruning ratio for the $\text{eng} \rightarrow \text{xxx}$ translation task. Models are pruned starting with the least language-specific parameters.

(2022), the distribution of parameter sensitivity becomes more balanced after using ID.

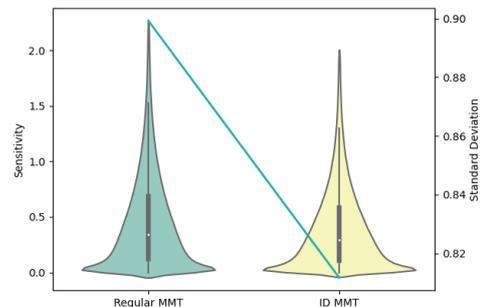


Figure 10: Sensitivity distribution (violin plots aligned with left y-axis) along with their standard deviation (green curve aligned with right y-axis, lower means more balanced parameter contribution). Note that we also remove the top 1% highest-sensitive parameters to ease the illustration.

C Ablation Study on Shared Projection Layer

Since we use a shared projection layer for both encoder and decoder denoising as well as for translation to reduce the model size and save memory, we investigate whether this sharing leads to a performance drop. We conduct experiments on M8 $\text{xxx} \rightarrow \text{eng}$ dataset. Table 6 shows that our method with shared layer slightly outperforms the one with separate output projection layers on average.

D M15 Language Information

We give a full account of the 15 languages in the M15 dataset in Table 7.

Method	High	Low	Very Low	All
CD+ID (shared layer)	35.16	15.18	9.23	18.69
CD+ID (NOT shared layer)	35.09	15.04	9.28	18.61

Table 6: Comparison of concurrent denoising + intra-distillation with and without using a shared projection layer.

E Effect of Masking Ratio

We take MMT+CD+ID as our study case to investigate the effect of masking ratio $r_m\%$ on the MMT performance. We conduct experiments on M8 $\text{xxx} \rightarrow \text{eng}$. Figure 11 shows that there is no big performance change when we set mask ratio between 0.3 and 0.6.

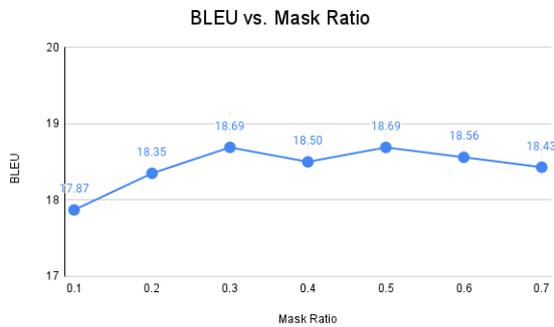


Figure 11: MMT performance change along with masking ratio on the MMT+CD+ID method.

Language	Language id	Parallel Data Size	Resource Level	Language family	Monolingual Data Size
Northern Sotho	nso	526K	Low	Central Narrow Bantu	3.2M
Rundi	run	454K	Low	Central Narrow Bantu	3.8M
Swati	ssw	94K	Very Low	Central Narrow Bantu	1.4M
Indonesian	ind	6.5M	High	Malayio-Polynesian	1.5M
Malay	msa	1M	High	Malayio-Polynesian	15M
Tagalog	tgl	1M	High	Malayo-Polynesian	15M
Bokmål (Norwegian)	nob	238K	Low	North Germanic	2.9M
Icelandic	isl	1M	High	North Germanic	15M
Faroese	fao	4K	Very Low	North Germanic	1.2M
Slovene	slv	15M	High	Southwestern Slavic	13M
Luxembourgish	ltz	8K	Very Low	Western Germanic	5M
Limburgish	lim	5K	Very Low	Western Germanic	8.4M
Catalan	cat	634K	Low	Western Romance	15M
Galician	glg	195K	Low	Western Romance	15M
Friulian	fur	6K	Very Low	Western Romance	730K

Table 7: The information of 15 languages in M15 dataset.

Cloze Quality Estimation for Language Assessment

Zizheng Zhang † and Masato Mita ‡† and Mamoru Komachi†

†Tokyo Metropolitan University

6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan

‡CyberAgent, Inc.

2-24-12 Shibuya Shibuya-ku, Tokyo 150-6121, Japan

zhang-zizheng@ed.tmu.ac.jp, mita_masato@cyberagent.co.jp, komachi@tmu.ac.jp

Abstract

Cloze tests play an essential role in language assessment and help language learners improve their skills. In this paper, we propose a novel task called Cloze Quality Estimation (CQE) — a zero-shot task of evaluating whether a cloze test is of sufficient “high-quality” for language assessment based on two important factors: reliability and validity. We have taken the first step by creating a new dataset named CELA for the CQE task, which includes English cloze tests and corresponding evaluations about their quality annotated by native English speakers, which includes 2,597 and 1,730 instances in aspects of reliability and validity, respectively. We have tested baseline evaluation methods on the dataset, showing that our method could contribute to the CQE task, but the task is still challenging.¹

1 Introduction

A cloze test (Taylor, 1953) is an efficient and comprehensive tool in language assessment, and thus it is widely used in language proficiency tests (Passage and Questions in Table 1), which measure multiple language abilities of examinees simultaneously, for example, grammatical knowledge (Rye, 1982; Alderson, 1979) and reading comprehension ability (Raymond, 1988; Klein-Braley, 1997). The most widespread format of the cloze test is the multiple-choice word-level cloze test, which consists of an incomplete passage with several blanks and a series of questions, where each question includes several options (four options in the usual setting), requiring examinees to fill the blanks by selecting words (or phrases) from options that make the passage coherent.

The significantly high price of creating cloze tests by experts has prompted automatic cloze generation methods (Goto et al., 2010; Sakaguchi

et al., 2013; Hill and Simha, 2016; Panda et al., 2022). However, automatically generated cloze tests do not always contribute well to language assessment and suffer from low quality. In language assessment, a reliable and valid cloze test shows a high ability to measure examinees’ language level (Bachman, 1985). Cloze question creation has two steps, word deletion and distractor generation, with the latter greatly affecting the ability to measure language level. As indicated by (Xie et al., 2018), some cloze tests, particularly automatically generated cloze tests, are created coarsely and cause two fatal issues in language assessment: (1) these tests do not guarantee that the answer is not ambiguous, which means there is a risk that multiple options fit almost equally well into the blank; (2) the test creation process considers less about which aspect of the language phenomenon is measured in the test; hence, such tests cannot measure the examinee’s language level. These two issues make the cloze test unsuitable for measuring examinees’ language level. The first issue makes the test unreliable (e.g., Question 4 in Table 1), which means even if an examinee has enough language knowledge to answer the test, the test might report a wrong score to indicate that the examinee lacks such knowledge. In other words, an unreliable cloze test cannot present the examinee’s language level. The second issue makes the test invalid (e.g., Question 3 in Table 1). An invalid test cannot identify the aspect in which an examinee lags in terms of language knowledge, which indicates that educators cannot identify the knowledge that the examinee has not acquired.

In this paper, we tackle the issues of evaluating the appropriateness of cloze tests for language assessment focusing on distractors. By following the test design principle (ALTE, 2011), we define a zero-shot task to evaluate cloze tests for language assessment considering two aspects: reliability and validity, which is called Cloze Quality Estimation

¹The CELA dataset and code of baselines are available at <https://github.com/zz-zhang/cloze-quality-estimation>.

Passage:

A policeman was walking along the street. In the doorway of a shop, a man was standing in the 1 light, with an unlighted cigar in his mouth. The policeman slowed down and then walked up to the man. "I'm just waiting for a friend here," the man said "It's an appointment 2 twenty years ago." The man struck a match and 3 his cigar. The light 4 a pale face with a little white scar near his right eye. "Twenty years ago tonight, when I said goodbye to Jimmy Wells, my best friend to start for the West to make my fortune ...

Questions:

- | | | | |
|------------------|-----------------------|---------------|------------------|
| 1. A. dark | B. bright | <u>C. dim</u> | D. colorful |
| 2. A. make | B. makes | C. making | <u>D. made</u> |
| 3. A. is stopped | <u>B. lighted</u> | C. burning | D. drop |
| 4. A. formed | <u>B. illuminated</u> | C. relieved | <u>D. showed</u> |
| ... | | | |

Evaluation:

(reliable, valid_grammar)
(reliable, valid_reading)
(reliable, not_valid)
(not_reliable, not_valid)
...

Table 1: Example of cloze test and qualities for each question in CELA. The input consists of a **passage** with multiple blanks and a series of **questions** (tuples of options). The expected output is **evaluation** tuples to indicate whether the questions are reliable and valid (and what language ability is tested if valid). Underlined options are the correct answers, which fit passage perfectly.

(CQE). In CQE, each question in a cloze test is asked to be estimated, whether it is reliable and valid. CQE provides a cloze as input (**Passage** and **Questions** in Table 1) and requires estimation of each question as output (**Evaluation** in Table 1). We introduce a new test set called the **Cloze Estimation dataset for Language Assessment (CELA)** which includes a variety of cloze tests and corresponding annotations. These cloze questions are specifically designed for junior high-school students in China. We prepared diverse English cloze tests including expert-designed and rule-generated tests and asked native English speakers to solve them and annotate the quality of each question in the two aspects of reliability and validity.

We also introduce baseline methods for the CQE task: we designed option-aware methods that evaluate cloze questions by analyzing their options. We tested the baseline methods with the CELA and compared them against option-agnostic baselines. We found that detection of unreliable questions is challenging and that all our baseline methods were wary to label a question as unreliable. The framework of our option-aware methods contributed to the validity evaluation, particularly when implemented by DNN-based approaches, which outperformed option-agnostic baselines significantly and showed potential for improvement.

The main contributions of this work are summarized as follows:

- We propose a new task of quality estimation

of cloze tests (CQE) for language assessment. We design two sub-tasks: reliability evaluation and validity evaluation.

- We create a new CQE dataset (CELA) for English learners, including annotations for both expert-designed and automatically generated cloze tests.
- We propose the first CQE methods considering the options of cloze questions. We report the experimental results using rule-based and DNN-based approaches.

2 Related Work

Language educators are capable of creating cloze tests rationally; they select words to be blanked and design distractors by their experience in language education to improve reliability and validity. CLOTH (Xie et al., 2018), SCDE (Kong et al., 2020), and CEPOC (Felice et al., 2022) are collections of human-created cloze tests, which are highly evaluated by experts in terms of measuring the English ability of examinees. However, designing cloze tests by experts is costly and difficult to generalize.

Automatic cloze generation methods could decrease the cost of creating cloze tests. To avoid generating useless tests in language assessment, these methods focus on distractor generation, which affects the quality of tests significantly. Previous works have conducted trials designing good rules or

machine learning models. Sakaguchi et al. (2013) described a method of generating distractors for assessing an English as second language (ESL) learner’s ability to distinguish semantic nuances between vocabulary words. They could generate valid questions, but these questions are domain-limited, which is not easy to generalize to cloze tests for human language assessment. The Children’s Book Test (CBT) (Hill et al., 2016) deletes named entities, common nouns, verbs, and prepositions and then designs distractors having the same part of speech (POS) as the deleted words to measure abilities in reading comprehension and vocabulary. Because of the naive distractor creation, the CBT method has the risk of generating unreliable questions. Coniam (1997); Goto et al. (2010); Correia et al. (2012); Hill and Simha (2016); Jiang et al. (2020) explore cloze test generation with various features, such as n -gram frequency and POS tag, and attempt to select deleted words and create distractors by using discriminative models including conditional random fields and support vector machine. Advanced distractor generation methods (Panda et al., 2022) that employ large pre-trained language models (LMs) can provide more valid distractors. These LMs produce better text representation that captures rich semantic information, and better text representation allows generation methods to produce more plausible distractors, which could measure language abilities better. However, designing good rules or models to improve the quality of cloze tests is not that easy. Furthermore, these works claimed they could generate better distractors, but it is difficult to perform comparisons to previous work. All the works performed human evaluations using their own metrics.

Crowdsourcing has been used to evaluate the quality of cloze tests and explore factors that affect quality (Skory and Eskenazi, 2010); workers were required to fill appropriate words in an open cloze-style sentence (a sentence with a blank, but without providing options). Answers from workers were used to calculate Cloze Easiness (Finn, 1977) to indicate whether the sentence is usable for the testing an examinee’s vocabulary. The Association of Language Testers in Europe provides a manual for developing a language test (ALTE, 2011), which specifies that the statistics of a test’s results reflect its reliability and validity. It asks various examinees to answer a test and analyzes the statistics of a question such as the accuracy and the answer

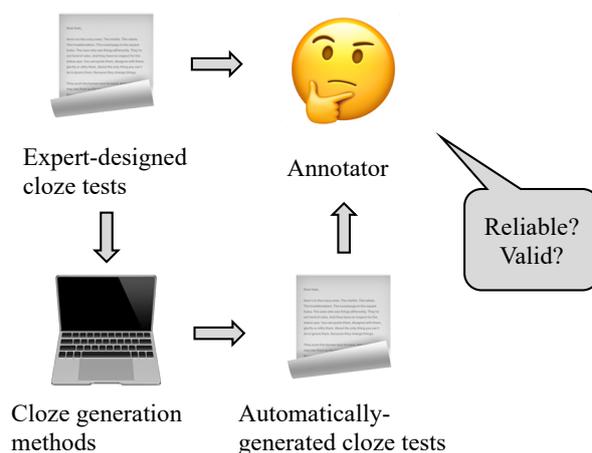


Figure 1: Flow of creating CELA. Cloze generation methods use passages and blanks from expert-designed tests to eliminate the effect of the word deletion strategy.

distribution. However, these evaluation methods require human resources or experts, which is time-consuming and sometimes difficult to obtain.

3 Cloze Quality Estimation

Motivated by related work, we propose the task of evaluating the quality of the cloze test. We introduce the definition of CQE task and CELA, our new dataset designed specifically for the CQE task. Figure 1 shows the flow of creating CELA. We collected expert-designed and automatically generated cloze tests and asked native English speakers to annotate whether these tests are reliable and valid.

3.1 Task Definition

First, we define a CQE task as follows. In a CQE task, given an incomplete passage with blanks and a series of questions with tuples of options, a quality estimation model should predict the quality of the questions and return tuples to indicate whether these questions are reliable and valid for language assessment. Here, we formalize a CQE task as two classification sub-tasks: *reliability evaluation* and *validity evaluation*.

Reliability. In the reliability evaluation, given a cloze passage with multiple questions (tuples of options), we perform a binary classification of whether the questions are reliable (*reliable*) or not (*not_reliable*). In terms of reliability, if a cloze question has more than one option that fits the context perfectly, there is no guarantee that the question can report a stable test score even when taken by the same examinee. Thus, in the reliability evaluation, we define that if a cloze question has more

than one correct answer, it is not reliable.

Validity. In the validity evaluation, we perform a three-class classification of either the question is valid for measuring grammatical knowledge (*valid_grammar*), valid for measuring reading comprehension ability (*valid_reading*), or invalid (*not_valid*). In terms of validity, a valid question should require examinees to use their language ability to distinguish the correct answer option from distractors. If a question measures more than one language ability, the question is considered to be too simple to measure the language ability of the examinee (Rankin, 1976). Thus, in the validity evaluation, we define that if a question requires the examinee’s single language ability (grammar or reading comprehension) to distinguish the answer option, the question is valid, otherwise it is invalid.

3.2 CELA data preparation

We collected English cloze tests from Chinese senior-high-school examinations (Xie et al., 2018) called CLOTH, which is expert-designed. To explore whether automatically generated cloze questions are sufficient for language assessment, we also employed four automatically generated cloze tests using previous generation methods: Randomized, Hill, Jiang, and Panda. All generated tests were based on the same cloze passages from expert-designed tests, that is, these five settings share the passages, blanks, and correct answers but have respective distractors in questions.

Randomized is generated using a random sampling method. In this method, we built vocabulary from CLOTH and randomly selected words from the vocabulary as distractor options.

Hill is generated using the same method of the CBT dataset (Hill et al., 2016), which selects words that have the same POS tag with the answer from the vocabulary as distractors.

Jiang employs the method of Jiang et al. (2020), which also selects words from the vocabulary but considers more factors including POS tag, word frequency, and spelling similarity. Their method is designed for the Chinese cloze test, but we adapted it to the English test.

Panda employs the method of Panda et al. (2022), which uses round trip translation to paraphrase a passage and align the paraphrased passages with the original one. They use aligned words to the answer as distractor candidates and select a distractor from candidates considering the syn-

onym and POS tag.

As a result, we collected and generated 150 cloze tests including 3,000 questions².

3.3 CELA annotation

We hired Amazon Mechanical Turkers to annotate the 3,000 questions. To ensure annotation quality, we required annotators to have approval rates over 98% and be native English speakers living in the United States. We also added attention checks to avoid bots and irresponsible annotators. Each question was annotated by three different annotators. Table 2 shows examples of our annotation task. As a reward, we paid each annotator \$1.5 for a test, which included 20 questions and took 5 to 7 minutes for completion.

We performed inter-annotator analysis on the annotations using Fleiss’ kappa score (Fleiss, 1971). Kappa scores were 0.67 and 0.45 for reliability (binary) and validity (3-class), respectively. Moderate kappa scores indicate that the annotation task was well-defined and the annotation result was trustable. Furthermore, to improve the annotation quality, we discarded all disagreed annotations. The majority of annotations that were rejected on the grounds of reliability pertained to long-term reasoning questions. These questions necessitated the integration of information from multiple sentences, and without taking into account this information, the distractors appeared to be equally plausible. This led to a divergence of opinions among some annotators and ultimately resulted in the determination that these questions were unreliable. The reasons for rejection in terms of validity were more varied. One pattern that emerged was the use of prepositions, where some annotators classified questions regarding preposition usage as *valid_reading* instead of *valid_grammar*, despite our explicit instructions on this matter. We posit that this may have been due to the fact that certain questions involving prepositions necessitate contextual information in order to deduce the correct answer (e.g., prepositions of location), causing some annotators to consider them as reading comprehension questions.

The processed data statistics are shown in Table 3. Because most blanks in CLOTH are content words and corresponding questions are designed to measure reading comprehension ability, there are few questions that measure grammatical knowledge.

²The cloze tests are collected/generated in five ways, each accounting for one-fifth of the total.

Passage	... He wished to find a good job. One day, he went to a company to ____ for a job.
Example 1	
Question	A. apply B. vote C. prepare D. wait
Explanation	In this question, only option A fits the passage perfectly, so please select “One” in Number of answer; options B, C, and D don’t fit the passage logically, and you will eliminate them by the knowledge (ability) of reasoning, so please select “Reading” option in Measured ability.
Example 2	
Question	A. apply B. applied C. look D. has applied
Explanation	In this question, both option A and C fit the passage perfectly, so please select “More than one” in Number of answer; except correct answers (option A and C), options B and D don’t fit the passage grammatically, and you will eliminate them by the knowledge (ability) of grammar, so please select “Grammar” option in Measured ability.
Example 3	
Question	A. apply B. vote C. applying D. waiting
Explanation	In this question, only option A fits the passage perfectly, so please select “One” in Number of answer; option B doesn’t fit the passage logically, option C doesn’t fit the passage grammatically, and you will eliminate them by the both of knowledge (abilities). So please select the “None” option in Measured ability. Also, since option D fits the passage neither logically nor grammatically, you will eliminate it by any of knowledge (abilities). So you can select the “None” option in Measured ability only considering option D.

Table 2: Example of annotation. We used following instructions: “Please select an option in the Number of answers list to indicate whether there is more than one option that fits the passage perfectly; please select what kind of language ability the question measures in the Measured ability list. You can refer to Table 2 for examples.”

Type	#
Reliability questions	2,597
reliable	2,324
not_reliable	273
Validity questions	1,730
valid_grammar	86
valid_reading	921
not_valid	723

Table 3: Statistics of the processed data. Because reliability is easier to annotate, it has higher agreement, and more annotations are retained than validity.

3.4 CELA analysis

We observed that the five types of cloze tests have various qualities. Figure 2 shows the quality statistic in CELA according to generation methods.

In reliability, Jiang is the most reliable and only includes 3.9% of unreliable questions, and Panda has 22.1%, which is the most unreliable. Surprisingly, CLOTH and Panda, which are expert-designed and generated by an advanced generation method, respectively, are not as reliable as the oth-

ers. We conjecture that these two types of tests tend to produce more plausible distractors that break only little coherence of the context. Plausible distractors are good at measuring learners’ language ability but have a higher risk of making the question unreliable. In particular, the Panda system utilizes round-trip translation and alignment to generate distractor candidates, which limits the scope of possible candidates and tends to produce more credible options compared to those generated by other systems. Furthermore, the Panda system does not impose strict limitations on eliminating distractors that are also suitable for the blank, which increases the likelihood of generating unreliable questions. On the other hand, the Randomized system selects distractors from the vocabulary without any constraints, which reduces the chance of selecting distractors that are also appropriate for the blank.

In validity, meeting our conjecture, there are fewer invalid questions in CLOTH and Panda, which means these two test types are better at measuring language ability than others. For automatic distractor generation methods, Panda has the strictest

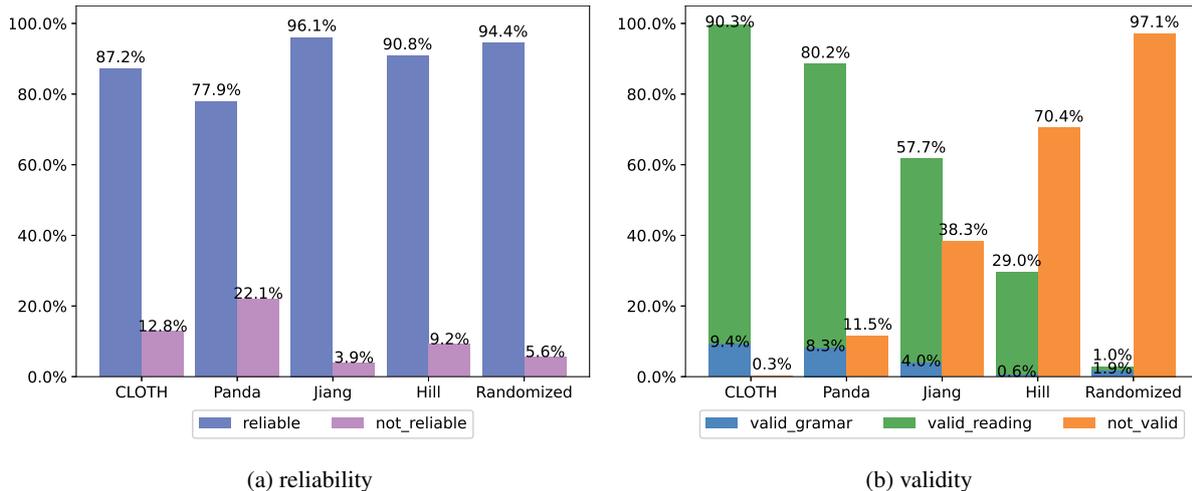


Figure 2: Quality statistics of cloze tests in CELA. The left and right buckets represent the ratio of high-quality and low-quality questions, respectively.

restrictions on distractor selection and produces the fewest invalid questions. Jiang has more filters for eliminating distractor candidates than Hill and could generate more valid questions. Randomized does not have any restrictions and is difficult to produce valid questions for language assessment.

We provide an example of the CELA dataset in Table 1. In the CELA dataset, each instance includes an incomplete passage with blanks and corresponding sets of options as input (questions). In a question, at least one option can be filled into the corresponding blank to make the passage coherent both grammatically and semantically. The label for each question is a tuple that denotes whether the question is reliable and valid, and if the question is valid the tuple also indicates which aspect of language ability the question measures.

4 Option-aware CQE Method

We propose two methods to tackle the CQE task, which analyze all options of cloze questions as baseline methods for the CQE task.

4.1 Intuition

We designed an option-aware CQE method considering how options in the question affect reliability and validity. We followed the definition in Subsection 3.1 and considered that the reliability and validity of a question is decided by its options. Thus, to tackle two sub-tasks in CQE, we need to inspect each option in terms of (1) whether it can be regarded as the sole answer to the question, and (2) what language ability it measures.

For the former (reliability), we consider that if an option breaks neither grammatical nor semantic coherence of the context, it fits the context perfectly and can be regarded as an answer option. For the latter (validity), if a distractor option only breaks grammatical (or semantic) coherence, examinees will use grammatical knowledge (or reading comprehension ability) to eliminate it, and in these cases, we say the distractor option is a grammatical (or reading) option; if a distractor option breaks both coherence, because it is too simple to measure one’s ability, we say it is a purposeless option.

For example, given a context: *I remember sitting in that dark hall listening to Mr. Zigler _____ everyone’s spirits up to the ceiling.* and options: *[raise, rise, educate, disappointed]*, the option *raise* does not break neither grammatical nor semantic coherence, so it is an answer option; the option *rise* breaks the grammatical coherence because the blank requires a transitive verb, so it is a grammatical option; the option *educate* obeys the grammatical rule but does not fit context semantically, so it is a reading option; the option *disappointed* is a purposeless option because it breaks both grammatical and semantic coherence of context.

Based on this intuition, we implement two functions, *BreakGrammar(·)* and *BreakSemantics(·)*, to judge whether an option breaks grammatical or semantic coherence. See Appendix A for a detailed description of the overall framework. To realize these two functions, we designed two different approaches: a rule-based approach and a DNN-based approach.

4.2 Rule-based approach

The rule-based approach is straightforward. It compares options with the answer to a question. The answer to a question can fit the context perfectly and does not break either grammatical or semantic coherence. Thus, we consider that if an option has the same grammatical/semantical feature as the answer, it does not break corresponding coherence either. In this case, functions *BreakGrammar(.)* and *BreakSemantics(.)* require one more parameter *answer*.

Given an answer option *answer* and an option *opt*, we fill *answer* and *opt* into context and obtain POS tags for them. If *opt* has the same POS tag as *answer*, we consider that it does not break grammatical coherence, otherwise it breaks grammatical coherence. For the implementation, we employed POS tagger in the Stanza library³.

Similarly, we use a synonym dictionary to judge if the option breaks grammatical coherence. If *opt* is a synonym of *answer*, *opt* does not break the semantic coherence, otherwise it breaks semantic coherence.

4.3 DNN-based approach

We also designed a DNN-based approach to implement these two functions. By using pretrained DNN models, we can plug in both grammatical and semantic knowledge into the CQE model. Unlike the rule-based approach, the DNN-based approach does not use *answer* but *opt* information for CQE.

We employ an English grammatical error corrector that can detect both grammatical and semantic errors and output the error types. We fill each option into context as input of the corrector and check the output. If the output indicates that there is no grammatical/semantic error, we regard that the option does not break grammatical/semantic coherence; otherwise, we think it breaks such coherence. We need to distinguish grammatical and semantic errors which affect the output of *BreakGrammar(.)* or *BreakSemantics(.)*. We design such a filter based on error types. To recognize the error type, we use the output tag of an error annotation toolkit.

5 Experiment

We conduct experiments to determine whether option-aware CQE methods (§4) can be a good baseline to estimate the quality of cloze tests,

³<https://github.com/stanfordnlp/stanza>

by comparing them with option-agnostic baseline methods (§5.2).

5.1 Configurations

To implement an option-aware baseline with a rule-based approach, we built an English synonym dictionary⁴. Considering that the word inflection or tense do not affect the meaning, we lemmatized both *answer* and *opt* into their basic form to judge if they were synonyms. The word lemmatization was implemented by employing the NLTK library⁵ and using the lemmatizer based on WordNet (Miller, 1998). We also employed POS tagger in the Stanza library to assign POS tags to *answer* and *opt*.

As for a DNN-based approach, we employed GECToR (Omelianchuk et al., 2020), a grammatical error corrector, that provided trained parameters and achieved a considerable performance on both CoNLL-2014 and BEA-2019 shared task (Ng et al., 2014; Bryant et al., 2019). We used GECToR which was implemented by RoBERTa (Liu et al., 2019). We fed original and corrected sentences into the Error ANnotation Toolkit (ERRANT)⁶ to obtain ERRANT tags. If the detected error’s ERRANT tag is one of **ADJ**, **ADV**, **NOUN**, and **VERB**, we considered the error to be a semantic one and not a grammatical one. Furthermore, we observed that the tag **OTHER** might contain both grammatical and semantic errors; therefore, we set two configurations for errors with tag **OTHER** as either grammatical or semantic errors.

5.2 Option-agnostic baselines

We employed the following *random* baseline and *majority prediction* baseline to show how well option-agnostic methods could perform on the CELA. Option-agnostic baselines can also be regarded as weak baselines.

Random baseline The random baseline predicts random class in reliability and validity classification. We chose the output class from the uniform distribution.

Majority prediction baseline The majority prediction baseline predicts the majority class in each classification sub-task. According to our CELA dataset, it always predicts *reliable* and

⁴collected from <https://www.thesaurus.com/>

⁵<https://www.nltk.org/>

⁶<https://github.com/chrisjbryant/errant>

Methods	Reliability			Validity				
	F_1	prec.	recall	micro F_1	macro F_1	$r.F_1$	$g.F_1$	$n.F_1$
Option-agnostic (weak)								
- Random	17.45	10.56	<u>49.82</u>	32.31	27.90	39.87	8.37	<u>35.45</u>
- Majority prediction	0.00	0.00	0.00	53.24	23.16	69.48	0.00	0.00
Option-aware (strong)								
- rule-based	2.87	1.47	66.67	42.35	41.31	30.28	37.05	56.61
- DNN-based (\bar{O})	19.50	98.53	10.82	<u>54.79</u>	<u>43.11</u>	72.33	<u>47.02</u>	9.96
- DNN-based (O)	<u>19.31</u>	<u>97.80</u>	10.71	58.54	48.25	<u>71.90</u>	53.79	19.05

Table 4: Performance of CQE baseline methods on the CELA dataset. $r.F_1$, $g.F_1$, and $n.F_1$ represent binary F_1 score for *valid_reading*, *valid_grammar*, and *not_valid* questions, respectively. **Bold** and underline indicate the best and second-best result, respectively. \bar{O} and O indicate we regard errors from GECToR with tag **OTHER** as grammatical and semantic errors, respectively.

valid_reading in the sub-task of reliability and validity classification, respectively.

5.3 Meta-evaluation metrics

To demonstrate the efficiency of CQE methods in estimating the quality of cloze tests, we provide baseline meta-evaluation metrics for the CQE task. Specifically, in the reliability evaluation, we used F_1 , precision, and recall score. Because unreliable cloze tests are harmful to language assessment, we must focus on how well CQE models can recognize unreliable tests; thus we set *not_reliable* as the positive label. For the validity evaluation, we used the micro-averaged and macro-averaged F_1 score. To indicate how well models perform in each class, we also split the overall F_1 score into three parts: F_1 for *valid_reading*, *valid_grammar*, and *not_valid*.

5.4 Result

The performance of the baselines on the CELA dataset is presented in Table 4. For option-agnostic baselines, because of imbalanced data distribution, the majority prediction baseline was not able to detect the *not_reliable* questions. Both baselines of random and majority prediction did not perform well on reliability compared with validity. Moreover, unreliable question detection is important to language assessment. In future work, improving the performance on reliability should be considered preferentially.

The option-aware baseline implemented by the rule-based approach performed worse than random baselines on some metrics. Although it achieved a moderate recall value, the precision was nearly zero, which denotes it tends to assign *reliable* to all questions. On the validity performance, it outperformed option-agnostic baselines on some metrics,

but it is still insufficient for evaluating the quality of cloze tests. One reason is that rules using the POS tag and synonym list are so naïve that they only consider partial cases of the option type. For example, given a context *This music made everyone want to _____. It was an early form of jazz.* and options [*dance, sing, laugh, ...*], though options *sing* and *laugh* are not the synonyms of the answer *dance*, they also fit the context semantically and should have not been classified into the reading option.

In most cases, the option-aware method using the DNN-based approach outperformed option-agnostic baselines. The DNN models utilized in this paper were straightforward and rudimentary, and there is potential for further improvement to make them more suitable for widespread use. Regarding reliability, errors with **OTHER** as grammatical or semantic errors have little effect on the performance. In terms of validity, when we regard **OTHER** errors as semantic errors, the micro F_1 value increased because the model could predict more *not_valid* questions correctly, which accounted for a significant proportion in CELA.

Except for hyperparameters, the mis-prediction caused by the underlying DNN models also leads to errors. GECToR did not perform well on long-term reasoning; thus, it was not able to detect some semantical errors. For example, given a context *I was ____ of flying, ... In order to get rid of my fear I decided to try a helicopter ride*, when filling word *proud* into the blank, we expect GECToR to correct the sentence with some words similar to *afraid*, but GECToR did not report any error.

6 Conclusion and Future Work

We proposed a novel task for evaluating cloze questions for human language assessment (CQE), which involved two important factors that affect the quality of cloze questions, reliability and validity, and also provided the CELA dataset. In addition, we explored automated CQE methods that can estimate the quality of cloze tests, by designing option-aware methods. Our experimental results on the CELA dataset showed that imbalanced data bring challenges.

In future work, we would like to investigate more factors that affect the quality of cloze questions and expand the CELA dataset. For example, given the context *The sun also ____.* and two different sets of options [*raises, rises, lifts, elevates*] as well as [*raises, lives, runs, lefts*], although these two sets of options both measure reading comprehension ability, answering the question with the former set of options is more difficult than the latter and requires a higher language level. Improving the performance of DNN models and optimizing implementation of *BreakGrammar(.)* and *BreakSemantics(.)* can improve the evaluation performance. For example, designing more detailed filters in the grammatical error corrector to filter out potential semantic errors or using fine-designed data to train different language models for grammatical and semantic error detection separately may boost the performance of DNN models.

We hope that the task and our resource will encourage further exploration from both computational linguistics and language education.

Limitations

The first limitation of this study is the coverage. In this study, the CQE task is defined to evaluate cloze tests that are generated by distractor generation methods. Cloze tests in this study are all based on expert-designed blanks. However, word deletion methods, which decide which word to be blanked, affect the quality of cloze tests, too. Investigation of how blanks influence the quality of cloze tests is necessary.

The second limitation of this study is the scalability of the annotation. In this study, the annotation of question quality is done by experts, which makes creating a large-scale dataset not that easy. This limitation could be mitigated by alternative choice of target data, i.e., there is room to replace native speakers with non-native speakers by selecting tar-

get data that do not require English knowledge at high-level proficiency (e.g., CEFR-A).

Finally, the CQE task and corresponding corpus are designed specifically for the English language. However, we are interested in exploring the possibility of adapting the task and dataset to other languages. The principles of test design, such as reliability and validity, apply to other languages as well, but the specific details may vary based on the language. For example, questions for a hieroglyph-based language may require learners to identify glyphs, which must be taken into consideration when defining reliability and validity. Adapting the CQE task to a new target language and creating a corresponding dataset requires a publicly available cloze question dataset or effective cloze question generation techniques in the target language, as well as experts in the language to evaluate question quality. In the future, we hope to develop an automatic adaptation method to transfer our task and dataset to multiple languages.

Acknowledgements

This work was supported by JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JP-MJFS2139.

References

- J Charles Alderson. 1979. The cloze procedure and proficiency in English as a foreign language. *TESOL quarterly*, pages 219–227.
- ALTE. 2011. *Manual for Language Test Development and Examining: For Use with the CEFR*. Language Policy division, Council of Europe.
- Lyle F Bachman. 1985. Performance on cloze tests with fixed-ratio and rational deletions. *TESOL Quarterly*, 19(3):535–556.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- David Coniam. 1997. A preliminary inquiry into using corpus word frequency data in the automatic generation of english language cloze tests. *Calico Journal*, pages 15–33.

- Rui Correia, Jorge Baptista, Maxine Eskenazi, and Nuno Mamede. 2012. Automatic generation of cloze question stems. In *International Conference on Computational Processing of the Portuguese Language*, pages 168–178. Springer.
- Mariano Felice, Shiva Taslimipoor, Øistein E. Andersen, and Paula Buttery. 2022. CEPOC: The Cambridge exams publishing open cloze dataset. In *Proceedings of the 2022 International Conference on Language Resources and Evaluation*. European Language Resources Association.
- Patrick J. Finn. 1977. Word frequency, information theory, and cloze performance: A transfer feature theory of processing in reading. *Reading Research Quarterly*, 13(4):508–537.
- Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.
- Takuya Goto, Tomoko Kojiri, Toyohide Watanabe, Tomoharu Iwata, and Takeshi Yamada. 2010. Automatic generation system of multiple-choice cloze questions and its evaluation. *Knowledge Management & E-Learning: An International Journal*, 2(3):210–224.
- Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks principle: Reading children’s books with explicit memory representations. In *International Conference on Learning Representations (ICLR)*.
- Jennifer Hill and Rahul Simha. 2016. Automatic generation of context-based fill-in-the-blank exercises using co-occurrence likelihoods and Google n-grams. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 23–30.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Christine Klein-Braley. 1997. C-tests in the context of reduced redundancy testing: An appraisal. *Language testing*, 14(1):47–84.
- Xiang Kong, Varun Gangal, and Eduard Hovy. 2020. SCDE: Sentence cloze dataset with high quality distractors from examinations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5668–5683, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach.
- George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzshanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Subhadarshi Panda, Frank Palma Gomez, Michael Flor, and Alla Rozovskaya. 2022. Automatic generation of distractors for fill-in-the-blank exercises with round-trip neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 391–401, Dublin, Ireland. Association for Computational Linguistics.
- Earl Rankin. 1976. Sequence strategies for teaching reading comprehension with the cloze procedure. In *Reading: Theory, Research and Practice, 26th Yearbook of the National Reading Conference*, pages 92–98, Atlanta, GA. National Reading Conference.
- Patricia M Raymond. 1988. Cloze procedure in the teaching of reading. *TESL Canada journal*, pages 91–97.
- James Rye. 1982. *Cloze procedure and the teaching of reading*. London.
- Keisuke Sakaguchi, Yuki Arase, and Mamoru Komachi. 2013. Discriminative approach to fill-in-the-blank quiz generation for language learners. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–242, Sofia, Bulgaria. Association for Computational Linguistics.
- Adam Skory and Maxine Eskenazi. 2010. Predicting cloze task quality for vocabulary training. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 49–56, Los Angeles, California. Association for Computational Linguistics.
- Wilson L Taylor. 1953. “Cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.
- Qizhe Xie, Guokun Lai, Zihang Dai, and Eduard Hovy. 2018. Large-scale cloze test dataset created by teachers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2344–2356, Brussels, Belgium. Association for Computational Linguistics.

A Algorithm for option-aware models

Algorithm 1 shows the framework of option-aware methods considering how to classify questions in aspects of reliability and validity by using types of options.

Please note that Algorithm 1 takes only content word as options. If the option is a functional word, we only assign *answer* or *grammar* as its type because questions including functional words as options only measure grammatical knowledge.

Algorithm 1: Framework of option-aware baseline

Input: context c ; a set of options in a question $O = \{opt_1, \dots, opt_n\}$;
function to judge if option breaks grammatical coherence
 $BreakGrammar(\cdot) \in \{true, false\}$;
function to judge if option breaks semantic coherence
 $BreakSemantics(\cdot) \in \{true, false\}$
Output: reliability and validity tuple of the input question (r, v) , where
 $r \in \{reliable, not_reliable\}$,
 $v \in \{valid_grammar, valid_reading, not_valid\}$
// Assign type to each option
1 $types = []$;
2 **for** $i \leftarrow 1$ to n **do**
3 **if** $BreakGrammar(c, opt_i) \wedge BreakSemantics(c, opt_i)$ **then**
 $types[i] \leftarrow purposeless$;
4 **if** $\neg BreakGrammar(c, opt_i) \wedge BreakSemantics(c, opt_i)$ **then**
 $types[i] \leftarrow reading$;
5 **if** $BreakGrammar(c, opt_i) \wedge \neg BreakSemantics(c, opt_i)$ **then**
 $types[i] \leftarrow grammar$;
6 **if** $\neg BreakGrammar(c, opt_i) \wedge \neg BreakSemantics(c, opt_i)$ **then**
 $types[i] \leftarrow answer$;
7 **end**
// Classify question in terms of reliability and validity by using option types
8 **if** $types.count(answer) = 1$ **then**
9 $r \leftarrow reliable$;
10 **if** $types.count(grammar) = n - 1$ **then** $v \leftarrow valid_grammar$;
11 **else if** $types.count(reading) = n - 1$ **then** $v \leftarrow valid_reading$;
12 **else** $v \leftarrow not_valid$;
13 **else**
14 $r \leftarrow not_reliable$;
15 $v \leftarrow not_valid$;
16 **end**
17 **return** (r, v) ;

Bag of Tricks for In-Distribution Calibration of Pretrained Transformers

Jaeyoung Kim
VUNO, Inc.
jaeyoung.kim@vuno.co

Dongbin Na
VUNO, Inc.
dongbin.na@vuno.co

Sungchul Choi
Pukyong National University
sc82.choi@pknu.ac.kr

Sungbin Lim*
Korea University
sungbin@korea.ac.kr

Abstract

While pre-trained language models (PLMs) have become a de-facto standard promoting the accuracy of text classification tasks, recent studies (Kong et al., 2020; Dan and Roth, 2021) find that PLMs often predict over-confidently. Although various calibration methods have been proposed, such as ensemble learning and data augmentation, most of the methods have been verified in computer vision benchmarks rather than in PLM-based text classification tasks. In this paper, we present an empirical study on confidence calibration for PLMs, addressing three categories, including confidence penalty losses, data augmentations, and ensemble methods. We find that the ensemble model overfitted to the training set shows sub-par calibration performance and also observe that PLMs trained with confidence penalty loss have a trade-off between calibration and accuracy. Building on these observations, we propose the **Calibrated PLM (CALL)**, a combination of calibration techniques. The CALL complements the drawbacks that may occur when utilizing a calibration method individually and boosts both classification and calibration accuracy. Design choices in CALL’s training procedures are extensively studied, and we provide a detailed analysis of how calibration techniques affect the calibration performance of PLMs.

1 Introduction

Trustworthy deployment of machine learning applications requires accurate and calibrated predictions to instill their reliability and help users be less confused about models’ decisions (Xiao and Wang, 2019; Liu et al., 2020).

However, modern deep neural networks (DNNs) produce miscalibrated predictions, i.e., a mismatch between a model’s confidence and its correctness. One of the reasons is that an over-parameterized

*Corresponding author. This work is partially done at UNIST.

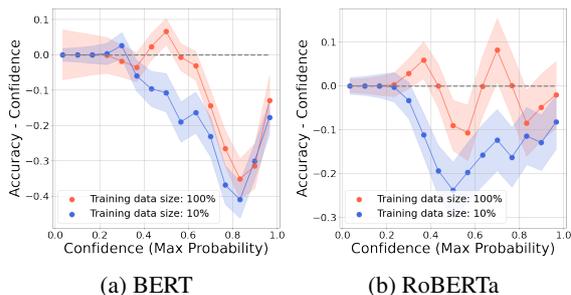


Figure 1: Reliability diagrams (DeGroot and Fienberg, 1983) on TREC (Li and Roth, 2002) with PLMs. A dashed line implies a perfect calibration while PLMs generally show over-confident predictions.

classifier typically produces over-confident predictions (Guo et al., 2017). Moreover, the miscalibration can be exacerbated when DNNs make predictions on test data different from the training distribution, i.e., distribution shift (Ovadia et al., 2019).

To obtain the well-calibrated predictions, many pioneering studies have shown the calibration effect of ensemble and regularization techniques focused on computer vision benchmarks. Ensemble learning has become one of the standard approaches to reduce calibration errors (Lakshminarayanan et al., 2017; Bonab and Can, 2019). Pereyra et al. (2017) propose the entropy regularized loss which penalizes confident output distributions in order to reduce overfitting. Hongyi Zhang (2018); Hendrycks et al. (2020) demonstrate that DNNs trained on diverse augmented data are less prone to produce over-confident predictions, leading to the calibration benefit under the distribution shift.

Intense research effort has focused on improving the calibration performance of vision models on image datasets. However, exploration of existing calibration methods with pre-trained Transformers (PLMs) has received less attention. Moreover, recent studies show that PLMs such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) pro-

duce miscalibrated predictions introduced by over-parameterization (Kong et al., 2020). Therefore, it is necessary to investigate how modern calibration techniques affect PLMs’ calibration.

In this paper, focused on PLMs in multi-class classification tasks, we explore widely used calibration families, including (1) confidence penalty loss functions that can be used instead of cross-entropy loss, (2) data augmentations, and (3) ensemble methods. We consider a low-resource regime since the small size of the training dataset amplifies the miscalibration of models (Rahaman et al., 2021). We also observe PLMs especially produce unreliable predictions in the data scarcity setting (see Figure 1).

Contributions. We conduct a comprehensive empirical study for the effectiveness of the above calibration methods. In this study, our findings are as follows:

- A PLM trained with imposing a strong penalty on the over-confident output shows significant improved calibration performance, but its accuracy can slightly deteriorate.
- For ensemble methods, Deep Ensemble (Lakshminarayanan et al., 2017) and MIMO (Havasi et al., 2021) increase the diversity of predictions, resulting in the well-calibrated predictions in the data scarcity setting. However, the ensemble methods show insufficient calibration when each ensemble member is overfitted to negative log-likelihood for the training dataset.
- Data augmentation methods that can expose diverse patterns such as MixUp (Hongyi Zhang, 2018) and EDA (Wei and Zou, 2019) are more effective for calibration in PLMs compared to weak text-augmentation methods (Kolomiyets et al., 2011; Karimi et al., 2021).

Building on our findings, we present Calibrated PLM (CALL), a blend of the discussed calibration methods. Numerical experiments demonstrate that the components of CALL complement each other’s weaknesses. For instance, data augmentation and ensemble methods offset the accuracy decline caused by the confidence penalty loss, while data augmentation and the confidence penalty loss counteract overfitting in the ensemble model. Through our extensive experiments, we show the CALL’s competitiveness on several text classification benchmarks.

2 Related Work

The calibration of machine learning models has been mainly studied for the trustworthy deployment of image recognition applications (Lakshminarayanan et al., 2017; Hongyi Zhang, 2018; Guo et al., 2017). Beyond the computer vision fields, research on the calibration ability of language models in the NLP domain has also recently been attracting attention (Desai and Durrett, 2020; Dan and Roth, 2021).

Desai and Durrett (2020) investigate the calibration ability of PLMs, and they demonstrate that RoBERTa produces more calibrated predictions than BERT. They also show that temperature scaling (Hinton et al., 2014) and label smoothing (Szegedy et al., 2016) improve the calibration performance of PLMs for language understanding tasks. Dan and Roth (2021) conduct an empirical study of the effects of model capacity on PLMs and show that smaller pre-trained transformers provide more reliable predictions. Moon et al. (2020) find that PLMs tend to produce over-confident outputs based on in-distribution (ID) keywords rather than contextual relations between words. They demonstrate that keyword-biased predictions can be over-confident even in out-of-distribution samples with ID keywords.

Kong et al. (2020) suggest two regularizers using generated pseudo-manifold samples to improve both ID and out-of-distribution calibration for PLMs. They use MixUp (Hongyi Zhang, 2018) as a regularization technique for BERT calibration and show that mixed training samples on the data manifold improve the calibration performance. Similarly, Park and Caragea (2022) propose a variant of MixUp utilizing saliency signals and also analyze the impact of combining additional calibration methods with MixUp. However, they only consider temperature scaling and label smoothing as additional calibration methods.

3 Why Re-assess Calibration Methods?

Guo et al. (2017) observe that a larger DNN tends to be more poorly calibrated than a smaller one. As the size of the parameters for modern DNNs continues to increase, the miscalibration issues need to be addressed more than ever.

At the same time, the unique character of PLMs raises concerns about whether previous findings on calibration obtained from standard convolutional neural networks (CNNs) can be successfully ex-

tended to PLM. For example, PLMs with ensemble learning may have different behavior compared to randomly initialized CNNs because naive PLMs have a massive amount of parameters and are initialized with pre-trained weights in the fine-tuning stage.

On the other hand, for the data augmentation, because image transformations (e.g., flipping, translation, and rotating) can not be directly applied to text-based samples, thus, it is also necessary to investigate the effect of text-specific augmentations on the calibration of PLMs.

4 Calibration Strategies

In this section, we review the existing literature used in our experiments and how we applied each method to PLMs. Calibration methods we explore are denoted by **bold**.

4.1 Preliminaries

Notation. Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ be a dataset consisting of N samples, where $x_i \in \mathcal{X}$ is an input and $y_i \in \mathcal{Y} = \{1, \dots, K\}$ is a ground truth label. We denote by $\bar{p}_i = f(y|x_i)$ the predicted distribution of a classifier f . Class prediction and associated confidence (maximum probability) of f are computed as $\hat{y}_i = \operatorname{argmax}_{k \in \mathcal{Y}} \bar{p}_i$ and $\hat{p}_i = \max_{k \in \mathcal{Y}} \bar{p}_i$, respectively.

In the BERT-style architecture, output of embedding layer, L attention blocks, and the output dense layer (with softmax function) are denoted by z_{embed} , $g = \{g_1, \dots, g_L\}$, and h , respectively.

Calibration Metrics. A calibrated model provides reliable predictive probability whose confidence aligns with its expected accuracy, i.e. $\mathbb{E}_{\hat{p}}[|\mathbb{P}(\hat{y} = y|\hat{p}) - \hat{p}|]$. Given a finite dataset, *Expected Calibration Error* (ECE; Naeini et al., 2015) is widely used as a calibration performance measure. ECE can be computed by binning predictions into T groups based on predictions of f and then taking a weighted average of each group’s accuracy/confidence difference:

$$\sum_{t=1}^T \frac{|B_t|}{N} |\operatorname{acc}(B_t) - \operatorname{conf}(B_t)|, \quad (1)$$

where B_t is the group of samples and their corresponding confidences belonging to the $(\frac{t-1}{T}, \frac{t}{T}]$. The $\operatorname{acc}(B_t)$ and $\operatorname{conf}(B_t)$ denote average accuracy and confidence of predictions for B_t , respectively.

Model calibration also can be measured using proper scoring rules (Gneiting and Raftery, 2007)

such as Brier score (Brier et al., 1950) and negative log likelihood (NLL).

4.2 Confidence Penalty Losses

We explore an alternative loss functions that can be used instead of cross-entropy (CE) loss.

Brier Loss (BL; Brier et al., 1950) is one of the proper scoring rules, defined as the squared error between the softmax output and the one-hot ground truth encoding. BL is related to ECE in that it is an upper bound of the calibration error by the calibration-refinement decomposition (Bröcker, 2009; Liu et al., 2020).

Entropy Regularized Loss (ERL; Pereyra et al., 2017) penalizes confident output distributions by adding the negative entropy:

$$\mathcal{L}_{\text{ERL}} = \mathcal{L}' + \beta \sum_{k=1}^K \bar{p}_k \log \bar{p}_k, \quad (2)$$

where \mathcal{L}' can be an arbitrary classification-based objective function (e.g., CE and BL), and β is the hyperparameter that controls the strength of the confidence penalty.

Label Smoothing (LS; Szegedy et al., 2016) is a commonly used *trick* for improving calibration that generates a soft label by weighted averaging the uniform distribution and the hard label.

4.3 Data Augmentations

Data augmentations have been widely used to improve the model’s calibration performance in computer vision fields (Hongyi Zhang, 2018; Hendrycks et al., 2020; Wang et al., 2021). However, text augmentations are often overlooked in the literature on the calibration in NLP tasks. To the best of our knowledge, we are the first to extensively study how text augmentation techniques such as Synonym Replacement (SR; Kolomiyets et al., 2011), Easy Data Augmentation (EDA; Wei and Zou, 2019), and An Easier Data Augmentation (AEDA; Karimi et al., 2021) affect calibration performance. We also investigate the recent variant of MixUp (Zhang and Vaidya, 2021).

SR randomly choose n words from the input sentence except for stop words and then replace each of these words with one of its synonyms chosen using WordNet (Miller, 1995).

EDA is a token-level augmentation method that consists of four random transformations: SR, Random Deletion, Random Swap, and Random Insertion.

AEDA only use Random Insertion operator that insert punctuation marks (i.e., “.”, “;”, “!”, “?”, “:”, “:”) into a input sentence.

MixUp (Hongyi Zhang, 2018) is a data augmentation strategy using convex interpolations of inputs and accompanying labels. Guo et al. (2019) investigate word- and sentence-level MixUp strategies to apply MixUp to recurrent neural networks. Zhang and Vaidya (2021) propose MixUp-CLS, that performs MixUp on the pooled [CLS] token embedding vector for a last attention layer of PLM. MixUp-CLS shows improved accuracy for natural language understanding (NLU) tasks compared to word-level MixUp. Unless otherwise specified, we use MixUp-CLS in our experiment.

4.4 Ensembles

Ensemble techniques utilize M models by combining them into an aggregate model and then average the predictions to produce calibrated outputs: $\frac{1}{M} \sum_{m=1}^M f_m(y|x)$. We compare the deterministic model with three ensemble approaches, and the computational cost of the ensemble methods used in the experiment is reported in Appendix A.

Deep-Ensemble (DE; Lakshminarayanan et al., 2017) consists of M randomly initialized models and provides a calibration effect leveraging the predictive diversity of ensemble members. When applying DE to PLMs, M independent models have different initialization weights only in a penultimate layer since PLMs are initialized with pre-trained weights.

Monte Carlo Dropout (MCDrop; Gal and Ghahramani, 2016) interprets Dropout as an ensemble model, leading to its application for uncertainty estimates by sampling M times dropout masks at test time.

Multi-Input and Multi-Output (MIMO). To alleviate the high computational cost and memory inefficiency of DE, Havasi et al. (2021) propose the multi-input and multi-output architecture by training M sub-networks inside a CNN.

In original MIMO, the M inputs (images) $\{x^m\}_{m=1}^M$ are sampled from $\mathcal{D}_{\text{train}}$. MIMO concatenates multiple inputs per channel before the first convolution layer and produces multiple outputs using M independent output dense layers. The feature extractor of CNN remains unchanged. For the training procedure, all ensemble members have the same mini-batch inputs with probability p , and the inputs are randomly sampled from the training

dataset with probability $1 - p$.

For applying MIMO to the PLMs, the following consideration arise; When multiple inputs are connected before the embedding layer, the length of tokens is M times longer. Thus, applying MIMO to PLMs in this manner is inefficient for a dataset that consists of long sentences.

Instead, we modify the original configuration of MIMO so that it can be applied to various NLP tasks. For PLM, the output of the first attention layer \bar{z} is calculated by averaging multiple outputs of M independent first attention blocks $\{g_1^m\}_{m=1}^M$:

$$\bar{z} = \frac{1}{M} \sum_{m=1}^M g_1^m(z_{\text{embed}}). \quad (3)$$

To produce multiple predictions, we use M modules that consist of the last attention blocks $\{g_L^m\}_{m=1}^M$ and dense layer h . The ensemble prediction is calculated by:

$$\bar{p} = \frac{1}{M} \sum_{m=1}^M h(g_L^m(g'(\bar{z}))), \quad (4)$$

where $g' = \{g_2, \dots, g_{L-1}\}$ is the shared attention blocks.

	# train	# dev	# test	l_{avg}	# classes
SST2	7.0k	0.7k	1.8k	19	2
20NG	9.1k	2.2k	7.5k	320	20
TREC	4.9k	0.5k	0.5k	10	6

Table 1: Summary of data statistics. l_{avg} : Sentence average length.

5 Experiments

This section presents the experimental results of the calibration methods. We describe experimental datasets and settings (Section 5.1 and 5.2), followed by empirical results for the low-resource regime (Section 5.3), overall calibration result (Section 5.4), and detailed analysis (Section 5.5). We then introduce the training procedure of CALL in Section 6. In our experiments, we set RoBERTa trained with CE as a baseline. Unless otherwise specified, ensemble and augmentation methods are applied to the baseline.

5.1 Datasets and Metrics

Dataset. Following Zhou et al. (2021), we use the following three text classification datasets. Data statistics are described in Table 1.

Acc \uparrow / ECE \downarrow / NLL \downarrow	TREC	SST2	20NG
RoBERTa (baseline)	94.04 / 4.08 / 24.86	91.23 / 7.42 / 43.08	76.58 / 11.37 / 90.40
CE+ERL	93.72 / 4.05 / 24.20	91.04 / 6.62 / 38.77	<u>76.79</u> / 11.21 / 90.32
CE+LS	93.84 / 3.37 / <u>23.71</u>	91.16 / 6.03 / 30.26	76.39 / 11.36 / 90.90
BL	93.24 / 2.69 / 26.55	89.48 / 7.15 / 36.02	75.74 / 7.21 / <u>86.02</u>
BL+ERL	93.84 / 2.48 / 24.78	90.32 / 5.68 / 29.61	76.13 / 6.62 / 86.11
BL+LS	93.52 / <u>2.32</u> / 25.16	91.15 / <u>5.56</u> / <u>29.37</u>	75.83 / <u>6.57</u> / 86.31
SR	94.24 / 3.37 / 22.24	90.54 / 7.22 / 38.03	76.45 / 10.54 / <u>87.64</u>
AEDA	93.76 / 4.68 / 28.36	91.45 / 6.69 / 37.67	76.41 / 11.49 / 91.21
EDA	93.40 / 2.83 / 23.46	91.56 / <u>5.01</u> / <u>29.86</u>	76.01 / <u>10.52</u> / 88.89
MixUp	<u>94.76</u> / 2.23 / <u>22.02</u>	90.86 / 6.46 / 31.89	<u>76.74</u> / 11.22 / 90.65
MCDrop	94.20 / 4.16 / 24.45	91.04 / 6.84 / 39.55	76.63 / 10.18 / 87.52
MIMO	94.88 / 3.13 / 20.38	91.26 / 6.21 / 32.78	76.25 / 5.61 / 81.43
DE	95.03 / <u>2.89</u> / 19.02	<u>91.44</u> / 4.88 / <u>29.51</u>	78.09 / 7.51 / 78.96

Table 2: Results for the low-resource regime. For each dataset, all methods are trained with 10% of training samples. The best results in each category are indicated in underline and the best results among all methods are indicated in **bold**. Accuracy is a percentile. We report ECE and NLL multiplied by 10^2 .

- Stanford Sentiment Treebank (SST2; Socher et al., 2013) is a sentiment analysis dataset that consists of sentences from movie reviews.
- 20 Newsgroups (20NG; Lang, 1995) is a topic categorization dataset which contains news articles with 20 categories.
- TREC (Voorhees and Tice, 2000) is a dataset for question classification, and we use its coarse version with six classes.

To evaluate the effectiveness for calibration methods in the data scarcity setting, we use 10% of the training set.

Metrics. We measure ECE and NLL for each calibration method. For ECE, we bin the predictions into $T = 15$ equidistant intervals. We report ECE and NLL multiplied by 10^2 in all experimental results for the convenience.

5.2 Training Configurations

We implement our framework upon Huggingface’s Transformers (Wolf et al., 2020) and build the text classifiers based on RoBERTa (roberta-base) in the main experiment. All models are optimized with Adam optimizer (Kingma and Ba, 2017) with a weight decay rate of 0.01, warmup proportion of 0.1, batch size of 16, a dropout rate of 0.1, and an initial learning rate of $1e-5$. We fine-tune the RoBERTa for 10 epochs. For each calibration method, hyper-parameters are tuned according to the classification performance, and the detailed hyper-parameter setting is described in Appendix B. We also provide empirical results for BERT

(bert-base-cased) in Appendix C. We report the averaged performance over 5 runs using different random seeds and implementation results are available at https://github.com/kimjeyoung/PLM_CALL.

5.3 Result for Low-resource Regime

Table 2 represents the classification accuracy and calibration performances for each dataset in the low-resource regimes. Most calibration strategies perform better than the baseline, even in cases where the baseline calibration results were already good, e.g., TREC. These results demonstrate that the existing methods can enhance PLM’s calibration ability when the annotation budget is small, as in many real-world settings.

Interestingly, augmentation methods except for AEDA also result in the calibration benefit. For example, MixUp and EDA show improved calibration performances for all datasets compared to the baseline.

Among confidence penalty losses, BL significantly reduces ECE for the three datasets. Moreover, the calibration performance is further improved when BL is combined with an additional regularization method (i.e., BL+ERL and BL+LS). However, BL+LS and BL+ERL underperform the baseline with respect to accuracy, and this performance drop is also observed when applied to BERT (Appendix C).

DE not only shows the most remarkable improvement of NLL but also improves accuracy for all datasets. MIMO also consistently outperforms the baseline for ECE. In summary, DE and MIMO are

Acc \uparrow / ECE \downarrow / NLL \downarrow	TREC	SST2	20NG
RoBERTa (baseline)	97.40 / 2.41 / 15.24	94.35 / 4.13 / 26.36	86.00 / 9.51 / 68.26
CE+ERL	97.24 / 2.44 / 14.64	94.05 / 4.05 / 26.94	86.13 / 9.41 / 70.18
CE+LS	97.28 / 2.06 / 13.11	94.21 / 3.75 / 20.17	86.14 / 9.81 / 70.16
BL	97.04 / 1.80 / 12.23	94.48 / 2.95 / 17.25	86.06 / 7.06 / 58.37
BL+ERL	97.28 / 1.35 / <u>12.09</u>	94.97 / 3.21 / 17.31	85.77 / 6.75 / 58.02
BL+LS	96.92 / 1.41 / 12.54	94.34 / <u>2.78</u> / 17.74	<u>86.15</u> / 6.76 / 58.17
SR	97.04 / 2.19 / 12.18	94.31 / 3.48 / 20.81	85.97 / 9.31 / 64.84
AEDA	<u>97.24</u> / 2.35 / 12.99	94.45 / 3.70 / 23.27	85.89 / 9.85 / 69.41
EDA	97.16 / 1.87 / 11.54	94.21 / <u>2.95</u> / 19.27	85.74 / <u>8.69</u> / <u>60.90</u>
MixUp	97.20 / <u>1.55</u> / 11.58	<u>94.57</u> / 3.61 / <u>19.04</u>	<u>86.21</u> / 8.72 / 64.48
MCDrop	97.56 / 2.37 / 13.84	94.01 / 3.64 / 24.02	85.97 / 8.61 / 64.49
MIMO	97.32 / 2.30 / 12.86	94.32 / 2.68 / <u>17.51</u>	85.80 / 8.68 / <u>60.92</u>
DE	97.32 / <u>2.09</u> / <u>12.83</u>	<u>94.64</u> / 3.10 / 19.15	86.81 / <u>7.90</u> / 62.31

Table 3: Overall calibration results for calibration techniques. For each dataset, all methods are trained with 100% of training samples.

more effective than the other calibration methods when considering both accuracy and calibration in the low-resource regime.

5.4 Overall Result

Overall performance result is reported in Table 3. Similar to the results in Table 2, most of calibration methods show better calibration performance compared to the baseline. In this setting, RoBERTa trained with BL+ERL works best. For example, BL+ERL shows NLL results of 17.31 and 58.02 in SST2 and 20NG, respectively, but DE obtain 19.15 and 62.31. In the data augmentation category, EDA and MixUp improve ECE and NLL compared to SR. AEDA underperforms the baseline for 20NG.

5.5 Analysis

Our empirical results raise the following questions: (1) Why do EDA and MixUp show better calibration performance than SR or AEDA? (2) How can we improve the accuracy of BL+ERL? (3) Why are ensemble methods more efficient than regularization methods in the low-resource setting, whereas BL+ERL is most effective for the full-data available setting? We further conduct a detailed analysis focusing on the above questions.

Role of Data Augmentation. Although the PLM trained on the proper scoring rule reduce calibration error for the training dataset, minimizing calibration errors for all unseen ID samples is challenging because we use finite training data (Liu et al., 2020). As an alternative, if models trained with augmented samples learn diverse representations, we expect to match the distribution of training data with the distribution of unseen ID data.

Distance	TREC	SST2	20NG
SR	11.56 / 17.44	7.12 / 12.01	15.54 / 23.02
AEDA	11.57 / 16.95	6.87 / 12.09	16.37 / 22.36
EDA	14.16 / 17.08	8.09 / 10.99	17.27 / 22.24
MixUp	14.52 / 15.44	7.69 / 11.18	16.65 / 21.62

Table 4: (Left) Distance between original and augmented sentences for the training samples. Higher is better. (Right) Distance between augmented training sentences and original test samples. Lower is better. The distance are computed at the last attention layer of RoBERTa.

Acc / ECE	TREC	SST2	20NG
BL+ERL	93.84 / 2.48	90.32 / 5.68	76.13 / 6.62
+SR	92.60 / 2.93	91.75 / 4.57	76.40 / 5.49
+AEDA	93.84 / 2.84	91.32 / 5.21	75.83 / 6.14
+EDA	93.40 / 2.83	90.76 / 4.97	76.45 / 5.18
+MixUp	94.76 / 2.23	90.89 / 4.52	76.25 / 6.39

Table 5: Comparison result for augmentation methods. Each method is trained with 10% of training data.

We analyze the distance between unseen and training data distribution, assuming that the augmentation scheme that pulls the distribution of training data towards the unseen data distribution will be effective for calibration.

To measure the distance between the two distributions, we use Hausdorff-Euclidean distance. In Table 4, RoBERTa trained with MixUp shows the closest distance between training data and test data, followed by EDA. In addition, the augmented data generated by MixUp and EDA are far away from the training data. It can be interpreted that EDA and MixUp generate more diverse patterns of

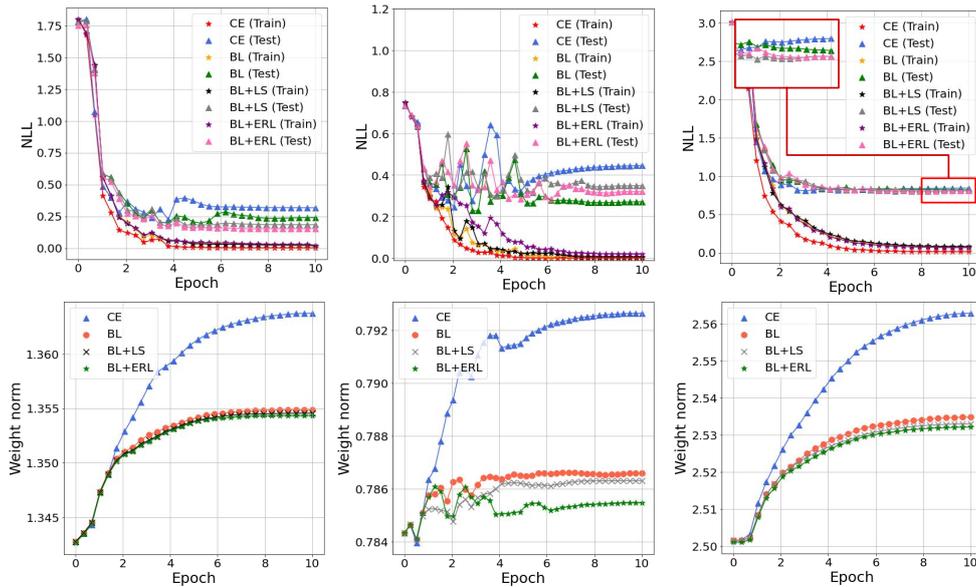


Figure 2: The plot of the NLL (Top) and the norm of weights (Bottom) while training RoBERTa on TREC (Left), SST2 (Middle), and 20NG (Right), respectively. The weights are extracted from the penultimate layer of RoBERTa and we use 10% of samples for training.

representations. Hence, matching the distribution of observed data with the distribution of unseen data by adopting a proper augmentation method that generates diverse patterns may help the model produces calibrated predictions.

On the other hand, since data augmentation generally helps to improve accuracy, we investigate whether augmentation methods improve the accuracy of BL+ERL. In Table 5, MixUp improves not only classification accuracy but also calibration performance on all datasets compared to the naive BL+ERL.

Role of Regularization. A crucial empirical observation by Guo et al. (2017) is that overfitting the NLL during training appears to be associated with the miscalibration of DNNs.

To better understand the role of strong regularization, we visualize the NLL during the training process of PLM. In Figure 2, training and test NLL are reduced at the beginning of training regardless of regularization methods. However, as training progresses, the test NLL of RoBERTa trained with CE increases¹. On the other hand, other regularization methods show an inhibitive effect on overfitting compared to CE.

A DNN can produce over-confident predictions if the network increases the norm of its weights, which results in the high magnitudes of the logits

¹Note that we use weight decay and dropout for training in order to alleviate overfitting.

(Mukhoti et al., 2020). Figure 2 (Bottom) shows that the RoBERTa trained with CE also has a larger norm than the regularized models.

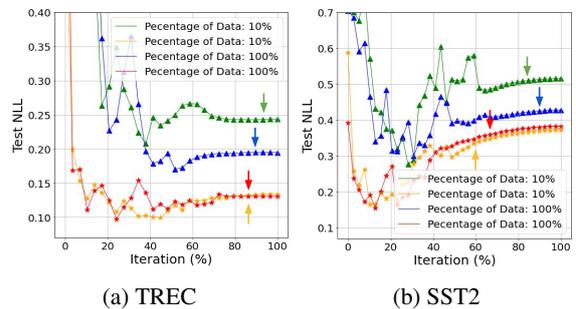


Figure 3: The test NLL for DE. Each arrow denotes the point at which the validation accuracy is the maximum.

Diversity Analysis in Ensembles. Diversity of predictions in ensemble is one of the key factor of determining calibration performances (Havasi et al., 2021). However, in the presence of overfitting, the diversity of predictions between ensemble members may decrease because the trained individual members would produce similar predictions that are overfitted to the same training data distribution (Shin et al., 2021).

We hypothesize ensemble members of DE applied to PLMs may also suffer from overfitting. Thus, we investigate whether the ensemble members are overfitted to NLL. In Figure 3, DE trained with 10% of the training data shows a different test NLL for each ensemble member, while DE trained

Acc \uparrow / ECE \downarrow / NLL \downarrow	TREC	SST2	20NG
Train samples	100 %		
RoBERTa (baseline)	97.40 / 2.41 / 15.24	94.35 / 4.13 / 26.36	86.00 / 9.51 / 68.26
DE (ensemble baseline)	97.32 / 2.09 / 12.83	94.64 / 3.10 / 19.15	86.81 / 7.90 / 62.31
BL + ERL	97.28 / 1.35 / 12.09	94.97 / 3.21 / 17.31	85.77 / 6.75 / 58.02
BL + ERL + MixUp	97.28 / <u>1.95</u> / 12.22	94.76 / <u>2.12</u> / 16.31	86.07 / 5.13 / 56.32
BL + ERL + MixUp + MCDrop	97.32 / 2.76 / 12.13	94.66 / 2.15 / <u>15.37</u>	86.12 / 4.73 / 55.61
BL + ERL + MixUp + MIMO	97.36 / 2.04 / <u>12.04</u>	<u>95.01</u> / <u>2.12</u> / 16.82	85.93 / <u>4.69</u> / <u>56.22</u>
BL + ERL + MixUp + DE	97.44 / 2.78 / 11.45	95.31 / 1.56 / 14.24	<u>86.67</u> / 3.67 / 53.21
Train samples	10 %		
RoBERTa (baseline)	94.04 / 4.08 / 24.86	91.23 / 7.42 / 43.08	76.58 / 11.37 / 90.40
DE (ensemble baseline)	95.03 / 2.89 / <u>19.02</u>	91.44 / 4.88 / 29.51	<u>78.09</u> / 7.51 / <u>78.96</u>
BL + ERL	93.84 / 2.48 / 24.78	90.32 / 5.68 / 29.61	76.13 / 6.62 / 86.11
BL + ERL + MixUp	94.76 / <u>2.23</u> / 22.02	90.89 / 4.52 / 26.59	76.25 / 6.39 / 84.20
BL + ERL + MixUp + MCDrop	94.68 / 2.41 / 21.92	90.93 / 4.26 / 26.16	76.16 / 4.69 / 82.54
BL + ERL + MixUp + MIMO	94.68 / 1.96 / 20.65	<u>91.75</u> / <u>3.13</u> / <u>23.96</u>	76.89 / <u>2.94</u> / 80.65
BL + ERL + MixUp + DE	<u>94.88</u> / 3.24 / 18.76	91.76 / 2.36 / 22.23	78.12 / 2.00 / 74.93

Table 6: CALL_{MIMO}: BL+ERL+MixUp+MIMO. CALL_{DE}: BL+ERL+MixUp+DE. The best and second best results are indicated in **bold** and underline, respectively.

with 100% of the training data results in a closer NLL for the ensemble members as the training progresses.

According to our experimental result, members within the ensemble often fail to produce different predictions due to the overfitting, indicating that additional effective regularization schemes can be adopted to prevent overfitting when applying the ensemble to the PLM. This finding also explains why ensemble techniques shows sub-par calibration performance compared to the regularization methods in the setting where full-data available.

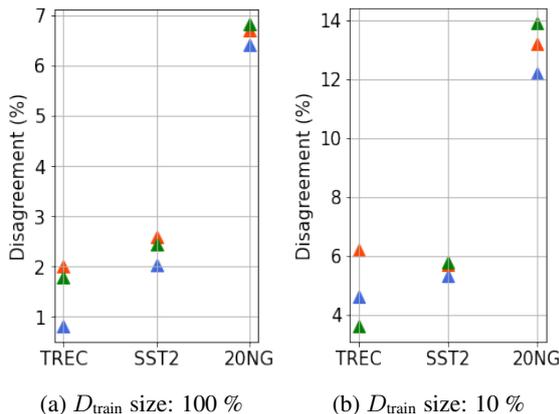


Figure 4: The diversity of predictions in ensemble with respect to the regularization methods. **Blue**: DE; **Orange**: DE+MixUp; **Green**: DE+BL+ERL. Results for MIMO and MCDrop are reported in Appendix D. A higher disagreement means that the models within the ensemble make different predictions.

We investigate whether BL+ERL and MixUp methods can compensate for the aforementioned

limitation of the ensemble method. We measure disagreement score (see Havasi et al., 2021) to analyze the degree of diversity for predictions. As shown in Figure 4, DE shows a high disagreement score in the low-resource regime. When full-data are available, the disagreement score of DE is consistently the lowest for all datasets. However, we observe that MixUp and BL+ERL significantly mitigate the reduction of predictive diversity for DE.

6 Calibrated PLMs

Through extensive analyses, we find that (1) MixUP that generate more diverse patterns helps improve the accuracy of BL+ERL, and (2) the reduced predictive diversity in the ensemble can be mitigated by BL+ERL and MixUp.

To this end, we report the calibration performance incrementally applying BL+ERL, MixUp, and ensemble techniques to the naive RoBERTa. Specifically, we denote BL+ERL+MixUP+DE, and BL+ERL+MixUP+MIMO by CALL_{DE}, and CALL_{MIMO}, respectively.

In Table 6, overall, CALL_{DE} achieves remarkable performance compared to DE on SST2 and 20NG datasets. CALL_{MIMO} shows competitive performance with DE with respect to ECE and NLL. This experiment shows that the calibration performance can be improved by the combinations using the ensemble, data augmentation, and confidence penalty losses in NLP tasks based on PLM, and each calibration method complements each other to further improve calibration performance without compromising accuracy.

7 Conclusion

In this work, we investigate the calibration effect of PLMs with various calibration methods applied. As a result of a comprehensive analysis of how calibration methods work in PLMs, we find that (1) the confidence penalty losses have a trade-off between accuracy and calibration, and (2) ensemble techniques lose predictive diversity as training progresses, resulting in reduced calibration effectiveness. To address these findings, we propose CALL, a combination of BL, ERL, MixUp, and ensemble learning. CALL reduces the risk of accuracy reduction through its data augmentation and ensemble techniques, and enhances the predictive diversity of ensemble methods by incorporating strong regularization and data augmentation. On multiple text classification datasets, CALL outperforms established baselines, making it a promising candidate as a strong baseline for calibration in text classification tasks.

Limitations

Although the proposed framework achieves significantly improved calibration performance compared to the baselines, CALL still has room for performance improvement and may require more diverse approaches (Zadrozny and Elkan, 2001; Hinton et al., 2014; Mukhoti et al., 2020; Liu et al., 2020). Another limitation is that we only address the ID calibration issue for PLMs. Therefore, whether CALL could work well for out-of-distribution detection and generalization tasks is unclear. We leave these questions for future research.

Ethics Statement

The reliability of deep-learning models is crucial to the stable deployment of real-world NLP applications. For example, the computer-aided resume recommendation system and neural conversational AI system should produce trustworthy predictions, because they are intimately related to the issue of trust in new technologies. In this paper, through extensive empirical analysis, we address diverse calibration techniques and provide a detailed experimental guideline. We hope our work will provide researchers with a new methodological perspective.

Acknowledgements

This work was also supported by Research Fund (1.200086.01) of UNIST, Institute of Information

& communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(No. 2022-0-00612, Geometric and Physical Commonsense Reasoning based Behavior Intelligence for Embodied AI), and National Research Foundation of Korea(NRF) funded by the Korea government(MSIT)(2021R1C1C1009256).

References

- Hamed Bonab and Fazli Can. 2019. Less is more: a comprehensive framework for the number of components of ensemble classifiers. *IEEE Transactions on neural networks and learning systems*, 30(9):2735–2745.
- Glenn W Brier et al. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.
- Jochen Bröcker. 2009. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 135(643):1512–1519.
- Soham Dan and Dan Roth. 2021. On the effects of transformer size on in-and out-of-domain calibration. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2096–2101.
- Morris H DeGroot and Stephen E Fienberg. 1983. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22.
- Shrey Desai and Greg Durrett. 2020. [Calibration of pre-trained transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.
- Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378.

- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.
- Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.
- Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M. Dai, and Dustin Tran. 2021. Training independent subnetworks for robust prediction. In *International Conference on Learning Representations*.
- Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2020. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. *Advances in Neural Information Processing System*.
- Yann N. Dauphin David Lopez-Paz Hongyi Zhang, Moustapha Cisse. 2018. [mixup: Beyond empirical risk minimization](#). *International Conference on Learning Representations*.
- Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. [Aeda: An easier data augmentation technique for text classification](#).
- Diederik P. Kingma and Jimmy Ba. 2017. [Adam: A method for stochastic optimization](#).
- Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2011. Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, volume 2, pages 271–276. ACL; East Stroudsburg, PA.
- Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. [Calibrated language model fine-tuning for in- and out-of-distribution data](#).
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.
- Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.
- Xin Li and Dan Roth. 2002. [Learning question classifiers](#). In *COLING 2002: The 19th International Conference on Computational Linguistics*.
- Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. 2020. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 33:7498–7512.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Seung Jun Moon, Sangwoo Mo, Kimin Lee, Jaeho Lee, and Jinwoo Shin. 2020. [Masker: Masked keyword regularization for reliable text classification](#).
- Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. 2020. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299.
- Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.
- Seo Yeon Park and Cornelia Caragea. 2022. [On the calibration of pre-trained language models using mixup guided by area under the margin and saliency](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5364–5374, Dublin, Ireland. Association for Computational Linguistics.
- Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. [Regularizing neural networks by penalizing confident output distributions](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.
- Rahul Rahaman et al. 2021. Uncertainty quantification and deep ensembles. *Advances in Neural Information Processing Systems*, 34:20063–20075.
- Minsuk Shin, Hyungjoo Cho, Hyun-seok Min, and Sungbin Lim. 2021. Neural bootstrapper. *Advances in Neural Information Processing Systems*, 34.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. [Recursive deep models](#)

- for semantic compositionality over a sentiment tree-bank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.
- Ellen M. Voorhees and Dawn M. Tice. 2000. [The TREC-8 question answering track](#). In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).
- Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. 2021. Augmax: Adversarial composition of random augmentations for robust training. *Advances in Neural Information Processing Systems*, 34.
- Jason Wei and Kai Zou. 2019. [Eda: Easy data augmentation techniques for boosting performance on text classification tasks](#).
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Huggingface’s transformers: State-of-the-art natural language processing](#).
- Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7322–7329.
- Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer.
- Wancong Zhang and Ieshan Vaidya. 2021. Mixup training leads to reduced overfitting and improved calibration for the transformer architecture. *arXiv preprint arXiv:2102.11402*.
- Wenxuan Zhou, Fangyu Liu, and Muhao Chen. 2021. [Contrastive out-of-distribution detection for pre-trained transformers](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Computational Cost for Ensemble Methods

Latency ↓ (s) (Train / Test)	TREC	SST2	20NG
RoBERTa	725.9 / 3.0	1031.9 / 8.2	1494.7 / 29.5
MCDrop (M=2)	725.9 / 5.8	1031.9 / 15.6	1494.7 / 58.8
MIMO (M=2)	840.7 / 3.5	1178.3 / 9.1	1720.0 / 34.0
DE (M=2)	1438.2 / 5.8	2060.7 / 15.6	3026.8 / 58.8
CALL _{MIMO}	841.9 / 3.5	1180.2 / 9.1	1721.5 / 34.0
CALL _{DE}	1440.3 / 5.8	2062.1 / 15.6	3028.4 / 58.8

Table 7: Comparison of training/test time for ensemble approaches. We measure the computational time on an NVIDIA-V100 single GPU.

Table 7 includes computational costs for ensemble methods on a single GPU. CALL_{DE} (RoBERTa+BL+ERL+MixUP+DE) is almost the same as DE since only the regularization term in the loss function and data augmentation process are added. Similarly, the computation cost of CALL_{MIMO} is almost the same as MIMO, and CALL_{MIMO} achieves a significant speedup in training/test time compared to DE.

B Hyperparameter Setting

Selected hyperparameters are highlighted in bold. **ERL**. Strength of the confidence penalty $\beta \in \{0.001, 0.005, 0.01, 0.1\}$. Empirically, PLMs trained with high beta (e.g., 0.1) showed sub-par classification accuracy. We set the low beta as 0.001 for all experiments.

LS. ϵ -smoothing parameter $\epsilon \in \{0.01, 0.05, 0.1\}$.

EDA. We follow the parameters recommended by the authors. Full-data setting: $\alpha = 0.1$. Data scarcity setting: $\alpha = 0.05$. α is a parameter that indicates the percent of the words in a sentence that are changed.

AEDA. For each input sentence, $p = \{5, 10, 15\}$ percentage of the words are changed for low-resource regime, otherwise $p = \{5, 10, 15\}$ words are changed.

SR. $p = \{5, 10, 15\}$ percentage of the words are changed for low-resource regime, otherwise $p = \{5, 10, 15\}$ words are changed.

MixUp. $\alpha \in \{0.1, 0.5, 1.0\}$ (strength of interpolation).

MCDrop. $p \in \{0.01, 0.02, 0.03, 0.04, 0.05, 0.1\}$ is the Dropout rate. $M \in \{2, 3, 4, 5\}$. We choose the hyperparameters when the validation accuracy is best in each experiment.

MIMO. $M \in \{2, 3, 4, 5\}$. Validation accuracy tends to decrease when M is increased.

We choose input repetition parameter $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ when the validation accuracy is best in each experiment. Overall, $p = 0.2$ is best.

DE. Full-data setting: $M \in \{2, 3, 4, 5\}$. Data scarcity setting: $M \in \{2, 3, 4, 5\}$.

C Empirical Result for BERT

We report empirical results for BERT in Table 8 and Table 9.

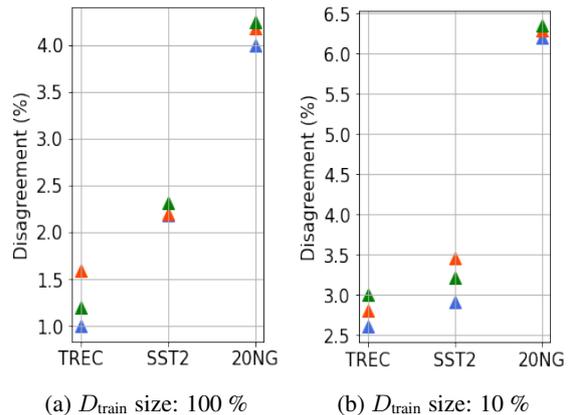


Figure 5: Effect of regularization with respect to diversity of predictions in ensemble. **Blue**: MCDrop; **Orange**: MCDrop+MixUp; **Green**: MCDrop+BL+ERL.

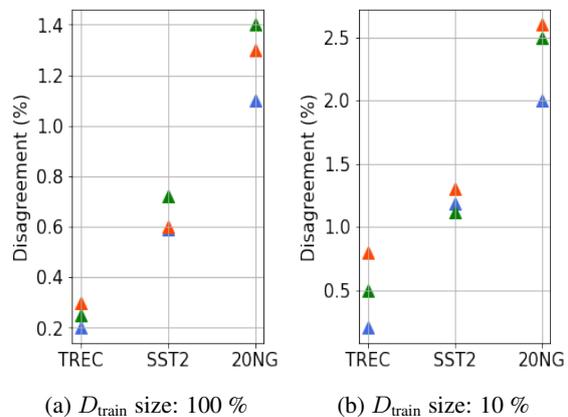


Figure 6: Effect of regularization with respect to diversity of predictions in ensemble. **Blue**: MIMO; **Orange**: MIMO+MixUp; **Green**: MIMO+BL+ERL.

D Analysis Diversity

We report diversity measure for MCDrop and MIMO in Figure 5 and Figure 6, respectively.

Acc↑ / ECE↓ / NLL↓	TREC	SST2	20NG
BERT (baseline)	97.24 / 2.44 / 13.20	91.26 / 5.19 / 33.77	85.45 / 9.98 / 70.33
CE+ERL	97.24 / 2.43 / 13.18	91.23 / 5.15 / 33.66	85.45 / 10.26 / 71.58
CE+LS	97.11 / 2.08 / 12.22	91.50 / 5.09 / 26.80	85.39 / 6.42 / 60.39
BL	97.64 / 1.29 / 10.38	91.33 / 5.18 / 28.01	85.28 / 7.25 / 60.46
BL+ERL	96.76 / 1.42 / 12.17	91.29 / 4.99 / 26.66	85.36 / 6.58 / 59.25
BL+LS	97.13 / 1.48 / 12.09	91.07 / 4.85 / 27.02	85.20 / 6.99 / 60.14
SR	97.48 / 1.96 / 10.37	91.83 / 5.11 / 29.53	85.50 / 9.60 / 68.14
AEDA	97.60 / 1.60 / 10.57	91.54 / 7.23 / 43.63	85.49 / 9.76 / 68.87
EDA	97.56 / 1.59 / 10.58	91.63 / 3.44 / 23.63	85.47 / 9.23 / 65.66
MixUp	97.40 / 1.30 / 11.12	91.66 / 5.89 / 28.78	85.63 / 8.92 / 66.20
MCDrop	97.32 / 2.08 / 12.97	91.52 / 5.89 / 31.28	85.35 / 9.90 / 68.56
MIMO	97.44 / 1.63 / 10.68	91.40 / 6.25 / 32.14	85.37 / 8.68 / 62.82
DE	97.32 / 1.98 / 11.26	91.92 / 4.14 / 27.27	85.86 / 7.99 / 62.81
BL+ERL+MixUp+MCDrop	97.34 / 2.01 / 12.37	91.59 / 3.61 / 28.54	85.37 / 5.62 / 60.18
BL+ERL+MixUp+MIMO (CALL _{MIMO})	97.56 / 1.52 / 10.40	91.37 / 5.03 / 25.96	85.33 / 4.87 / 58.06
BL+ERL+MixUp+DE (CALL _{DE})	97.79 / 2.82 / 10.18	91.82 / 2.58 / 22.19	86.05 / 3.62 / 54.03

Table 8: Result for BERT with diverse calibration techniques. The best results are indicated in **bold**.

Acc↑ / ECE↓ / NLL↓	TREC	SST2	20NG
BERT (baseline)	93.40 / 4.43 / 25.16	87.47 / 9.49 / 52.36	73.79 / 10.90 / 96.02
CE+ERL	93.40 / 4.40 / 25.13	87.48 / 9.50 / 51.66	73.77 / 10.84 / 95.96
CE+LS	93.28 / 3.87 / 24.13	87.44 / 7.99 / 37.05	73.57 / 8.07 / 94.78
BL	93.60 / 2.33 / 21.54	87.26 / 7.25 / 38.74	73.96 / 6.63 / 91.02
BL+ERL	93.25 / 2.38 / 21.95	87.56 / 6.83 / 36.96	74.21 / 5.63 / 90.94
BL+LS	93.14 / 2.41 / 22.03	87.78 / 6.01 / 36.76	73.91 / 5.89 / 92.37
SR	92.52 / 4.67 / 28.37	87.74 / 8.62 / 46.59	74.00 / 10.93 / 95.34
AEDA	93.44 / 4.36 / 24.48	87.71 / 9.03 / 48.55	73.65 / 11.52 / 97.43
EDA	91.88 / 4.30 / 28.30	87.44 / 8.93 / 44.94	74.04 / 10.33 / 94.26
MixUp	93.88 / 2.76 / 20.47	87.65 / 7.20 / 37.47	74.01 / 9.04 / 95.31
MCDrop	93.56 / 3.53 / 24.89	87.43 / 8.87 / 50.13	73.81 / 10.24 / 94.77
MIMO	93.88 / 2.62 / 21.53	87.55 / 6.09 / 34.82	73.80 / 7.25 / 88.65
DE	93.68 / 2.91 / 21.13	87.92 / 6.76 / 38.44	75.19 / 7.52 / 85.81
BL+ERL+MixUp+MCDrop	93.45 / 3.51 / 23.77	87.58 / 5.42 / 34.31	73.80 / 7.35 / 90.69
BL+ERL+MixUp+MIMO (CALL _{MIMO})	93.56 / 2.91 / 21.20	87.70 / 5.85 / 34.35	74.11 / 5.21 / 89.93
BL+ERL+MixUp+DE (CALL _{DE})	94.24 / 3.41 / 19.79	88.25 / 2.48 / 28.65	75.68 / 2.20 / 82.90

Table 9: Result for BERT with diverse calibration techniques on the low-resource regime. The best results are indicated in **bold**.

Fine-Tuning Deteriorates General Textual Out-of-Distribution Detection by Distorting Task-Agnostic Features

Sishuo Chen¹, Wenkai Yang¹, Xiaohan Bi¹, Xu Sun²

¹Center for Data Science, Peking University

²MOE Key Laboratory of Computational Linguistics, School of Computer Science,

Peking University

{chensishuo, xusun}@pku.edu.cn

{wkyang, bxh}@stu.pku.edu.cn

Abstract

Detecting out-of-distribution (OOD) inputs is crucial for the safe deployment of natural language processing (NLP) models. Though existing methods, especially those based on the statistics in the feature space of fine-tuned pre-trained language models (PLMs), are claimed to be effective, their effectiveness on different types of distribution shifts remains underexplored. In this work, we take the first step to comprehensively evaluate the mainstream textual OOD detection methods for detecting semantic and non-semantic shifts. We find that: (1) no existing method behaves well in both settings; (2) fine-tuning PLMs on in-distribution data benefits detecting semantic shifts but severely deteriorates detecting non-semantic shifts, which can be attributed to the distortion of task-agnostic features. To alleviate the issue, we present a simple yet effective general OOD score named GNOME that integrates the confidence scores derived from the task-agnostic and task-specific representations. Experiments show that GNOME works well in both semantic and non-semantic shift scenarios, and further brings significant improvement on two cross-task benchmarks where both kinds of shifts simultaneously take place. Our code is available at <https://github.com/lancopku/GNOME>.

1 Introduction

The pre-training and fine-tuning paradigm based on Transformers (Vaswani et al., 2017) has achieved tremendous success in various natural language understanding (NLU) tasks (Devlin et al., 2019; Liu et al., 2019; Qiu et al., 2020). However, fine-tuned pre-trained language models (PLMs) notoriously suffer from over-confident predictions on out-of-distribution (OOD) inputs (Hendrycks et al., 2020). As this issue threatens the reliability of NLP models deployed in the open world, textual OOD detection has attracted great attention recently (Podolskiy et al., 2021; Zhou et al., 2021, 2022; Duan et al.,

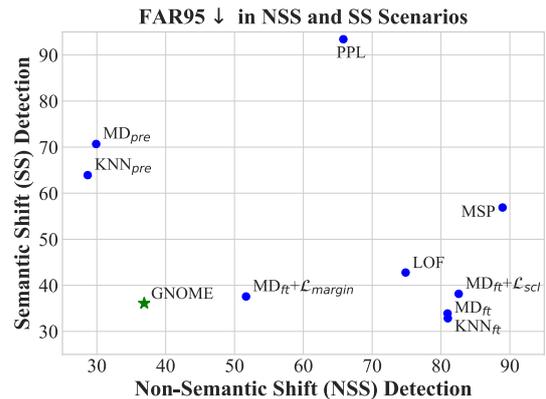


Figure 1: OOD detection performance (FAR95↓, lower is better) in non-semantic and semantic shift scenarios. No single existing method works well in both scenarios, but our proposed GNOME mitigates the trade-off.

2022, etc.), which aims to enable the model to abstain from making unreasonable predictions on OOD data and resort to human intervention.

Nonetheless, almost all of the current approaches are assessed under certain assumptions about the type of OOD texts. One line of works creates in-distribution (ID) and OOD pairs from arbitrary datasets for different tasks (Hendrycks et al., 2020), while another line assumes that OOD data belong to classes in the ID task but unseen during training, e.g., in intent recognition (Podolskiy et al., 2021). Arora et al. (2021) reveal the inconsistency among the evaluation protocols and category the distribution shifts to non-semantic shifts (NSS) and semantic shifts (SS), but a thorough comparison of existing methods in different settings is missing as later works either focus on either detecting NSS (Duan et al., 2022) or SS (Zhou et al., 2022).

In this work, we systematically evaluate the mainstream textual OOD detection methods on a comprehensive suite of benchmarks covering both NSS and SS scenarios. As shown in Figure 1, no single method wins across the board. Notably, the detec-

tors based on the pre-trained features, e.g., the Mahalanobis distance detector MD_{pre} (Xu et al., 2021), excel at detecting non-semantic shifts but fail in detecting semantic shifts. In contrast, when the PLM is fine-tuned on annotated ID data, the detectors based on fine-tuned features, e.g., MD_{ft} (Podolskiy et al., 2021), perform well in the SS scenario but disastrously fail in the NSS setting. These observations uncover an intriguing trade-off: *fine-tuning contributes to the detection of semantic shifts but impairs the detection of non-semantic shifts*. This trade-off raises two critical research questions:

RQ1: *Why does fine-tuning undermine the detection of non-semantic shifts?* It is relatively easy to attribute the positive effect of fine-tuning in the SS setting to the learned class-discriminative features (Fort et al., 2021), but it remains unknown why fine-tuning plays a negative role in the NSS setting. We empirically find that the adverse effect comes from the fact that fine-tuning gradually destructs the pre-trained task-agnostic knowledge about general linguistic properties, which are useful cues for the detection of non-semantic shifts.

RQ2: *How to develop a general textual OOD detection method?* Since the type of distribution shifts is unknown in practice, our findings suggest that a practical method able to detect different kinds of OOD texts is yet to be developed. To this end, we aggregate the distance scores estimated in the feature space of both pre-trained and fine-tuned models to derive a **GeNeral textual OOD Measurement scorE** (GNOME) capable of detecting both NSS and SS. On the suite of benchmarks covering both NSS and SS settings, GNOME (the green star in Figure 1) surpasses the previous SOTA by 8.13 FAR95 points on average; on two cross-task benchmarks where both kinds of shifts happen simultaneously, GNOME reduces the average FAR95 by 4.88 points. Note that GNOME is not meant to be a SOTA method in all settings but rather a simple, principled way to get reasonable detection performance under various kinds of distribution shifts—we hope our analysis inspires better approaches for general textual OOD detection.

2 Related Work

OOD detection aims to detect abnormalities coming from a different distribution from the training data so that the model can refuse to make predictions on them (Amodei et al., 2016; Yang et al.,

2021). Since it is essential for the security of machine learning models deployed in the open-world environment, OOD detection has gained great attention, first in computer vision (CV). We categorize the mainstream OOD detection methods into three groups by way to derive confidence scores: (1) *confidence-based methods* using the output probabilities of classifiers trained on in-distribution (ID) data (Hendrycks and Gimpel, 2017; Lakshminarayanan et al., 2017; Liang et al., 2018; Liu et al., 2020); (2) *density-based methods* using density scores derived from generation models (Zong et al., 2018; Ren et al., 2019; Xiao et al., 2020); (3) *distance-based methods* using the distance statistics in the feature space of neural networks (Lee et al., 2018; Huang et al., 2021; Sun et al., 2022).

Following the progress in CV, textual OOD detection based on PLMs has also attracted increasing attention. Hendrycks et al. (2020) show that the maximum softmax probability (MSP) score (Hendrycks and Gimpel, 2017) is a strong baseline for PLMs, followed by a group of works on confidence-based textual OOD detection (Li et al., 2021; Shen et al., 2021; Yilmaz and Toraman, 2022). As for the density-based branch, Gangal et al. (2020) and Arora et al. (2021) apply the idea to textual OOD detection by leveraging language models such as LSTM (Hochreiter and Schmidhuber, 1997) and GPT-2 (Radford et al., 2019). Regarding the distance-based methods, Podolskiy et al. (2021) revisit the Mahalanobis distance-based detector (Lee et al., 2018) for textual OOD detection based on fine-tuned PLMs and achieve performance gains over confidence-based methods, which is then further improved by introducing contrastive regularization (Zhou et al., 2021), utilizing nearest-neighbor distance (Zhou et al., 2022), and leveraging intermediate features (Chen et al., 2022).

Nonetheless, the NLP community lacks uniform evaluation criteria for OOD detection. Generally, ID/OOD pairs for evaluation are constructed in three ways: (1) *the non-semantic shift (NSS) setting* (a.k.a., the background shift setting) (Li et al., 2021; Arora et al., 2021; Duan et al., 2022), where ID and OOD data consist of the same semantic classes but differ in background information,¹ e.g.,

¹Although the model can also make predictions on the samples with only non-semantic shifts, the accuracy tends to significantly drop. As the cost of wrong predictions is great in safety-critical scenarios, a conservative method for handling these samples by rejecting them is practical.

tweets as ID and Wikipedia comments as OOD in toxicity detection; (2) *the semantic shift (SS) setting* (Podolskiy et al., 2021; Zhou et al., 2022), where OOD data are composed of unseen classes belonging to the ID task, e.g., new classes in intent classification; (3) *the cross-task setting* (Hendrycks et al., 2020; Zhou et al., 2021), where the ID and OOD data are from datasets for different tasks and both semantic and non-semantic shifts happen, e.g., sentiment analysis data as ID and news classification data as OOD. Arora et al. (2021) first notice the inconsistency and compare confidence-based and density-based methods in NSS and SS settings, but they neglect the crucial branch of distance-based methods and the cross-task setting. In this work, we fill in this gap by presenting a comprehensive evaluation and developing a general textual OOD score motivated by our observations.

3 Observations and Explanations

In this section, we first give preliminaries in § 3.1. Then we introduce our benchmark for evaluating textual OOD detection (§ 3.2) and the evaluated methods (§ 3.3). Finally, we present the evaluation results (§ 3.4) and our interpretation of the observed trade-off between NSS and SS scenarios (§ 3.5).

3.1 Preliminaries

Problem Formulation The OOD detection problem can be formulated as a binary classification problem to decide whether an input example \mathbf{x} belongs to the training data distribution \mathcal{P}_{in} (ID) or not (OOD). An OOD detector D makes decisions for the input \mathbf{x} based on the following formula:

$$D(\mathbf{x}) = \begin{cases} \text{ID} & \text{if } S(\mathbf{x}) \geq \gamma \\ \text{OOD} & \text{if } S(\mathbf{x}) < \gamma \end{cases}, \quad (1)$$

where $S(\mathbf{x})$ is the confidence score output by the detector and γ is the threshold chosen by the user.

Metrics We adopt two widely-used metrics AUROC and FAR95 following prior works (Podolskiy et al., 2021; Zhou et al., 2021). AUROC can be interpreted as the probability that the model ranks a random ID sample higher than a random OOD sample, and FAR95 is the proportion of negative samples (OOD) wrongly judged as positive (ID) when the true positive rate is 95%. Higher AUROCs and lower FAR95s indicate better performance.

Notations Assume M_θ is a PLM where θ denotes its parameters and $\mathbf{z} = M(\mathbf{x})$ denotes the feature

Setting	Task	ID	OOD
Non-Semantic Shift	Sentiment Analysis	SST-2 IMDB	IMDB SST-2
	Toxicity Detection	Twitter Jigsaw	Jigsaw Twitter
Semantic Shift	News Categorization	AGNews NC	AGNews _{OOD} NC _{OOD}
	Dialogue Intent Classification	ROSTD CLINC	ROSTD _{OOD} CLINC _{OOD}

Table 1: The architecture of the constructed suite of benchmarks categorized as either non-semantic shift (NSS) or semantic shift (SS). NC is short for the News Category dataset.

vector for the input sample \mathbf{x} derived from M (e.g., the last-layer CLS embedding in Transformers). For a classification task with C classes, the user fine-tunes M together with a classification head h and get the fine-tuned model $F_{\theta^*,h} = h \circ M_{\theta^*}$ where θ^* denotes the fine-tuned parameters. The output of F is $F_{\theta^*,h}(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_C(\mathbf{x}))^T$, which denotes the predicted probabilities.

3.2 Benchmark Construction

We aim to build ID/OOD pairs where either the non-semantic shift (NSS) or the semantic shift (SS) dominates so that we can fairly compare existing methods on the ability to detect these two kinds of shifts separately. (1) For NSS, we choose SST-2 (Socher et al., 2013) and IMDB (Maas et al., 2011) for sentiment analysis, and Twitter (Founta et al., 2018) and Jigsaw² for toxicity detection. Among the four, any two datasets from the same task can be regarded as an ID/OOD pair. (2) For SS, we use four datasets: New Category (NC) (Misra and Grover, 2021; Misra, 2022), AGNews (Corso et al., 2005), ROSTD (Gangal et al., 2020), and CLINC (Larson et al., 2019). For each dataset, we use some classes as ID and the remaining classes as OOD. We show the architecture of the constructed suite of benchmarks categorized as either non-semantic shift (NSS) or semantic shift (SS) in Table 1 and more details can be found in Appendix A. Compared with Arora et al. (2021), we additionally include toxicity detection data for NSS and intent recognition data for SS, which make the suite of benchmarks more representative of real-world scenarios.

²Available at this [link](#).

3.3 Evaluated Baselines

OOD Detection Methods We evaluate the mainstream methods as follows. (1) For *confidence-based methods*, we test the MSP baseline ($S(\mathbf{x}) = \max_{y \in \{1, 2, \dots, C\}} p_y(\mathbf{x})$) (Hendrycks and Gimpel, 2017) and its three variants: Scaling (Liang et al., 2018), Energy Score (Liu et al., 2020), and D2U (Yilmaz and Toraman, 2022); (2) For *density-based methods*, we evaluate the PPL method (Arora et al., 2021) using the GPT-2 model for language modeling ($S(\mathbf{x}) = 1/\text{PPL}(\mathbf{x})$); (3) For *distance-based methods*, we test the LOF method (Lin and Xu, 2019) that trains a local outlier detector on fine-tuned features of ID data, and the basic variants of the Mahalanobis detector (MD): MD_{pre} (Xu et al., 2021) built on pre-trained features ($\mathbf{z} = M_\theta(\mathbf{x})$) and MD_{ft} (Podolskiy et al., 2021) built on fine-tuned features ($\mathbf{z} = M_{\theta^*}(\mathbf{x})$). Also, we evaluate two variants of MD built on features derived from PLMs fine-tuned with supervised contrastive and margin-based auxiliary targets (Zhou et al., 2021), namely $\text{MD}_{\text{ft}} + \mathcal{L}_{\text{scl}}$ and $\text{MD}_{\text{ft}} + \mathcal{L}_{\text{margin}}$. Generally, the confidence score in MD is formulated as:

$$\text{MD}(\mathbf{x}) = \min_{c \in \{1, 2, \dots, C\}} (\mathbf{z} - \mu_c)^T \Sigma^{-1} (\mathbf{z} - \mu_c), \quad (2)$$

$$S(\mathbf{x}) = -\text{MD}(\mathbf{x}),$$

where μ_c is the class centroid for class c and Σ is the global covariance matrix (μ and Σ can be estimated on ID training data). Besides, we evaluate the nearest-neighbor detectors (Sun et al., 2022) based on pre-trained (KNN_{pre}) and fine-tuned features (KNN_{ft}). We refer readers to Appendix C for more details about the baselines.

Model Configuration For the methods based on fine-tuned PLMs, we build text classifiers by fine-tuning the RoBERTa_{base} (Liu et al., 2019) model (110M parameters) on annotated ID data. For MD_{pre} , we use the pre-trained RoBERTa_{base} model. For the PPL method, we fine-tune the GPT-2_{small} model (117M parameters) for language modeling on ID data. More details can be found in Appendix B.

3.4 Evaluation Results and Findings

We display the main evaluation results in Table 2. As shown, the confidence-based methods underperform the density-based method PPL in the NSS setting, while they outperform PPL in the SS setting, in line with the observations in Arora et al. (2021). Notably, we notice that the distance-based methods achieve the best results in both NSS and SS

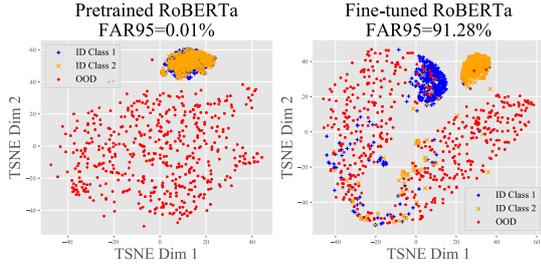
Category	Method	Avg.	NSS	SS
Confidence	MSP	71.63/72.92	65.47/88.94	77.78/56.89
	Scaling	71.96/71.62	65.45/88.94	78.47/54.30
	Energy	71.75/71.63	64.90/89.05	78.61/54.20
	D2U	71.99/71.49	65.47/88.94	78.52/54.04
Density	PPL	67.65/79.61	74.28/65.81	61.03/93.42
Distance	MD_{ft}	80.39/57.42	72.25/80.95	88.54/33.89
	$\text{MD}_{\text{ft}} + \mathcal{L}_{\text{scl}}$	82.22/60.36	76.71/82.57	87.73/38.15
	$\text{MD}_{\text{ft}} + \mathcal{L}_{\text{margin}}$	86.50/44.63	85.45/51.68	87.54/37.57
	MD_{pre}	83.76/50.29	93.29/29.90	74.22/70.68
	KNN_{ft}	81.02/56.91	72.58/80.99	89.47/32.84
	KNN_{pre}	85.69/46.29	92.66/ 28.67	78.72/63.92

Table 2: The performance (AUROC \uparrow /FAR95 \downarrow values in percentage) of the evaluated approaches. All results are averaged over five random seeds, and best results are highlighted in **bold**. We report results averaged on the ID/OOD pairs in NSS and SS setting in the last two columns, respectively, and report the results averaged on all eight benchmarks in the third column. See full results on each benchmark in Table 4 and Appendix E.

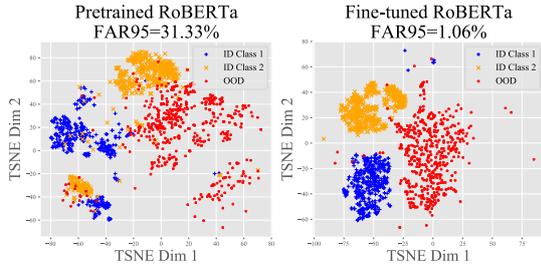
settings. Concretely, MD_{pre} and KNN_{pre} built on pre-trained features are the best in the NSS setting, while MD_{ft} and KNN_{ft} built on fine-tuned features are the best in the SS setting. However, no single method wins across the board. Thus, we draw an intriguing trade-off: *In textual OOD detection, fine-tuning PLMs on ID data boosts semantic shift detection but impairs non-semantic shift detection.*

To intuitively understand the effect of fine-tuning, we visualize the features using t-SNE (Van der Maaten and Hinton, 2008) in both settings. As plotted in Figure 2, before fine-tuning, the ID and OOD samples are sharply separated in the NSS setting, but show a significant overlap in the SS setting; after fine-tuning, the ID samples are well clustered on class in both settings and unseen classes (OOD) in the SS setting are also pulled away from the ID data, but OOD samples in the NSS setting become almost indistinguishable from the ID data.

The observed benefits of fine-tuning in the SS setting match the observation in the near-OOD image detection (Fort et al., 2021), suggesting that the fine-tuned task-specific representations are more suitable for detecting unseen classes belonging to the ID task. Regarding the negative effect of fine-tuning in the NSS setting, we speculate that it can be explained in this way: task-agnostic features important for detecting non-semantic shifts are learned during pre-training but discarded in the fine-tuning stage, for which we will present empirical evidence in § 3.5. To our knowledge, we are the first to study the impact of fine-tuning on the de-



(a) Non-Semantic Shift: SST-2 (ID) vs. IMDB (OOD)



(b) Semantic Shift: ROSTD (ID) vs. ROSTD_{OOD} (OOD)

Figure 2: T-SNE visualizations for the features derived from pre-trained and fine-tuned RoBERTa models and the corresponding FAR95 of the Mahalanobis detector.

tection of different kinds of OOD texts and reveal the trade-off between the NSS setting (fine-tuning harms) and the SS setting (fine-tuning helps).

In addition, we notice that when the model is fine-tuned with margin-based contrastive auxiliary targets ($\mathcal{L}_{\text{margin}}$) (Zhou et al., 2021), the MD detector ($\text{MD}_{\text{ft}} + \mathcal{L}_{\text{margin}}$) substantially surpasses MD_{ft} in the NSS setting with marginal sacrifice in the SS setting, thus it achieves the best performance on average. However, it still falls far behind MD_{pre} in the NSS setting. *As no single existing method behaves well in both settings, a general textual OOD detection method capable of detecting different kinds of OOD texts is yet to be developed, given the broad range of distribution shifts in realistic scenarios.*

3.5 Empirical Explanations

Probing Analysis. From the view of the oracle, the OOD data in the NSS setting can be easily distinguished from the ID data by certain task-agnostic linguistic features. For example, the IMDB data are long movie reviews with an average length of 230 tokens, which can be well distinguished by length from the SST-2 reviews with an average length of 19 tokens. Therefore, we speculate that the negative effect of fine-tuning arises from the deletion of general linguistic features during fine-tuning. To test the conjecture, we evaluate the sentence em-

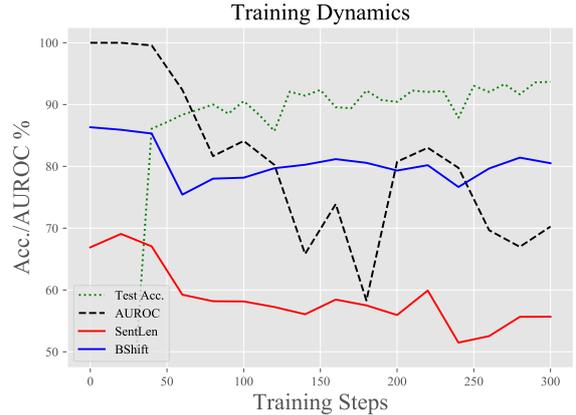


Figure 3: The dynamics of test accuracy, probing accuracies of *SentLen* and *BShift*, and the performance (AUROC \uparrow) to detect IMDB as OOD in the fine-tuning process of the RoBERTa model on SST-2. We have seen similar trends on other datasets in the NSS setting.

Methods	Avg.	NSS	SS
MD_{pre}	50.29	29.90	70.68
MD_{ft}	57.42	80.95	33.89
$\text{MD}_{\text{ft}} + \text{RecAdam}$	55.76	78.39	33.13
MD_{ft} (head lr $\times 10$)	53.90	76.18	31.63
$\text{MD}_{\text{ft}} + \text{LP-FT}$	50.92	60.26	41.59

Table 3: The performance of MD_{ft} coupled with regularization techniques on textual OOD detection. We report the FAR95 \downarrow values on average.

beddings produced by the model checkpoints in the fine-tuning process on SST-2 on two classic probing tasks designed by Conneau and Kiela (2018): *SentLen* (sentence length) and *BShift* (bigram shift). They are general linguistic features irrelevant to the class labels in downstream classification tasks, so the probing accuracies can be regarded as indicators of the preservation of task-agnostic features (see details in Appendix A.3). We show the tendency of the probing accuracies along with corresponding OOD detection performance (IMDB as OOD) and test accuracy in Figure 3. We find that as fine-tuning goes on, although the classification performance on ID test data shows an upward trend, the OOD detection performance (AUROC) gradually declines along with the probing accuracies. The observed correlations between the OOD detection performance and the probing accuracies on *SentLen* and *BShift* empirically indicate that fine-tuning impairs NSS detection by distorting the task-agnostic features in pre-trained models.

Does Regularization Help? As stated, fine-tuning may destruct the pre-trained features and

thus harm NSS detection. If the cause-effect holds, the negative effect can be alleviated by regularization techniques for preserving pre-trained features. To verify this deduction, we investigate three regularization approaches: (1) the RecAdam optimizer (Chen et al., 2020); (2) a 10× larger learning rate for the head (Prabhu et al., 2021); (3) the linear-probing then fine-tuning approach (LP-FT) (Kumar et al., 2022). As shown in Table 3, the regularization techniques applied to MD_{ft} bring moderate improvements in the NSS setting, but they still fall far behind MD_{pre} that exploits the original pre-trained features. The results provide further empirical support for our reasoning about the demerits of fine-tuning. Moreover, they suggest that effectively preserving pre-trained task-agnostic features suitable for NSS detection in fine-tuned PLMs is challenging. A plausible solution is to decouple task-agnostic and task-specific features in a single fine-tuned model, which we leave for future study. Another possible solution is to directly leverage the pre-trained model, which we will introduce next.

4 GNOME for Textual OOD Detection

In view of the observed trade-off, we are motivated to combine the strengths of task-agnostic and task-specific representations to obtain a confidence score capable of modeling both non-semantic shifts and semantic shifts. A straightforward way is to take the mean of MD_{pre} and MD_{ft} scores.³ However, given that the norm of features can fluctuate and thus the distance scores are not comparable across different spaces, simple averaging may cause the integrated score to be skewed towards the side with the larger norm. To alleviate the issue, we normalize MD_{pre} and MD_{ft} before aggregation:

$$\begin{aligned} \text{Norm}(\text{MD}_{\text{pre}}(\mathbf{x})) &= \frac{\text{MD}_{\text{pre}}(\mathbf{x}) - \mu_{\text{pre}}}{\sigma_{\text{pre}}}, \\ \text{Norm}(\text{MD}_{\text{ft}}(\mathbf{x})) &= \frac{\text{MD}_{\text{ft}}(\mathbf{x}) - \mu_{\text{ft}}}{\sigma_{\text{ft}}}, \end{aligned} \quad (3)$$

where μ and σ are the mean and standard deviation of Mahalanobis distance scores, respectively, which can be estimated on ID validation samples. Then we obtain the integrated score GNOME (GeNeral textual OOD Measurement scoreE):

$$S_{\text{GNOME}}(\mathbf{x}) = -\text{Agg}(\text{Norm}(\text{MD}_{\text{pre}}(\mathbf{x})), \text{Norm}(\text{MD}_{\text{ft}}(\mathbf{x}))), \quad (4)$$

³Our core idea is orthogonal to the distance-based scoring function, so we can also combine KNN_{pre} and KNN_{ft}. We have tested in this way and got similar results to those obtained by combining MD_{pre} and MD_{ft}.

where Agg is the aggregation operator (such as the mean or max). We use the mean operator for aggregation in our main experiments. Note that We do not use a weighted average because it is not possible to tune the weights when OOD data is unknown, which follows the mainstream setting in OOD detection. If the user has prior knowledge about the type of OOD data, he/she can train the weights to aggregate the scores.

5 Experiments

5.1 Experimental Setup

Benchmarks Besides the eight benchmarks introduced in § 3.2 that are categorized as either NSS or SS, we also evaluate GNOME and baselines in the cross-task setting (Hendrycks et al., 2020; Zhou et al., 2021) where both kinds of shifts happen simultaneously. Following Zhou et al. (2021), we choose SST-2 (Socher et al., 2013) and 20 News-groups (Lang, 1995) as ID data, and regard a series of datasets from different tasks as OOD data: TREC-10 (Li and Roth, 2002), WMT-16 (Bojar et al., 2016), Multi30k (Elliott et al., 2016), RTE (Dagan et al., 2005), and SNLI (Bowman et al., 2015). Refer to Appendix A for more details.

Models and Metrics We follow the same model configuration as that in § 3.3 in main experiments and report FAR95 values in the main text (the trend of AUROC results in Appendix E is similar).

5.2 Results and Analysis

GNOME works well in both SS and NSS settings and significantly surpasses baselines in terms of average performance. As shown in Table 4, GNOME is competent in both settings (close to MD_{pre} for NSS and MD_{ft} for SS) and achieves the best performance on average. The average FAR95 is 36.50%, 8.13% lower than the previous SOTA MD_{ft}+ $\mathcal{L}_{\text{margin}}$ requiring extra margin-based targets.

GNOME also achieves superior performance in the cross-task setting. As results in Table 5, GNOME outperforms all baseline methods (4.88% FAR95 reduction on average) in the cross-task setting, demonstrating the power of integrating task-agnostic and task-specific representations when non-semantic shifts and semantic shifts happen simultaneously. Note that among existing methods, MD_{pre} and KNN_{pre} are the best on the 20 News-groups benchmark, suggesting that non-semantic shifts dominate there; MD_{ft}+ $\mathcal{L}_{\text{margin}}$ is the best on

Methods	Avg.	Non-Semantic Shift (NSS)				Semantic Shift (SS)			
		SST-2	IMDB	Twitter	Jigsaw	NC	AGNews	ROSTD	CLINC
MSP (Hendrycks and Gimpel, 2017)	72.92	90.50	79.20	89.76	96.29	72.16	85.06	51.24	19.08
Scaling (Liang et al., 2018)	71.62	90.50	79.20	89.76	96.29	68.94	80.19	52.15	15.90
Energy (Liu et al., 2020)	71.63	90.58	79.65	89.70	96.27	69.35	79.17	52.53	15.76
D2U (Yilmaz and Toraman, 2022)	71.49	90.50	79.20	89.76	96.29	68.86	79.49	52.14	15.66
PPL (Arora et al., 2021)	79.61	64.65	7.25	94.53	96.79	93.50	95.78	86.99	97.40
LOF (Lin and Xu, 2019)	58.82	93.39	24.28	88.86	92.90	70.83	81.00	4.69	14.58
MD _{fit} (Podolskiy et al., 2021)	57.42	91.28	47.78	88.49	96.25	58.16	64.06	1.06	12.26
MD _{fit} + \mathcal{L}_{scl} (Zhou et al., 2021)	60.36	88.05	63.11	84.53	94.57	67.00	69.26	3.42	12.90
MD _{fit} + \mathcal{L}_{margin} (Zhou et al., 2021)	44.63	32.61	4.70	72.51	96.90	59.31	67.48	11.07	12.40
MD _{pre} (Xu et al., 2021)	50.29	0.01	1.54	44.40	73.65	90.14	87.64	31.33	73.30
KNN _{fit} (Sun et al., 2022)	56.91	87.87	56.81	83.67	95.60	56.51	60.64	0.71	13.50
KNN _{pre} (Sun et al., 2022)	46.29	0.00	1.48	33.11	80.03	84.56	81.89	18.51	70.70
GNOME (Ours)	36.50	0.04	8.24	53.36	85.88	64.81	63.25	1.47	14.94

Table 4: OOD detection performance (FAR95↓, lower is better) on the constructed suite of benchmarks. All values are percentages averaged over five different random seeds, and the best results are highlighted in **bold**. The second column gives the average performance on eight benchmarks.

Methods	Avg.	ID Datasets	
		SST-2	20 NG
MSP	59.98	70.00	49.95
Scaling	50.68	70.00	31.36
Energy	52.31	72.43	32.31
D2U	51.15	70.00	32.29
LOF	51.55	66.29	36.81
MD _{fit}	32.29	48.82	15.75
MD _{fit} + \mathcal{L}_{scl}	35.30	49.04	21.56
MD _{fit} + \mathcal{L}_{margin}	23.97	29.43	18.51
MD _{pre}	17.90	35.79	0.01
KNN _{fit}	43.58	63.73	23.42
KNN _{pre}	20.79	41.57	0.01
GNOME (ours)	13.02	26.02	0.01

Table 5: OOD detection performance in the cross-task setting. For each ID dataset, we report the macro average of FAR95↓ on all corresponding OOD datasets, averaged over five random seeds.

the SST-2 benchmark, indicating that both kinds of shifts matter there. Without any prior knowledge about the type of distribution shifts, GNOME yields the best performance on both benchmarks.

5.3 Ablation Study

We examine the rationality of the key components of GNOME here. As shown in Table 6, when the normalization operation is absent, the performance in the SS setting is slightly enhanced ($\sim 3\%$ FAR95 reduction), but the performance in the NSS and cross-task settings drops by around 7% FAR95 points, which suggests that the normalization operation helps strike a balance between the two scenarios and thus achieve better performance on average. These results also empirically verify that the mean operator is more suitable than the max

Norm.	Agg.	Avg.	NSS	SS	CT
✓	mean	28.67	36.88	36.12	13.02
	max	30.08	38.33	37.90	14.01
✗	mean	32.61	44.00	33.32	20.52
	max	34.67	45.31	33.88	24.83

Table 6: The performance (FAR95↓) corresponding to different normalization choices and score aggregators in GNOME. CT denotes the cross-task setting.

operator for the score aggregation step in GNOME. We have also tested other common normalization methods such as min-max and found that they underperform the standardization normalization employed in GNOME.

Besides the score-level fusion in GNOME, we have also tested feature-level fusion (concatenating or averaging pre-trained and fine-tuning features), but they lead to a significant drop in the SS setting ($+20\%$ FAR95) while only a slight improvement in the SS setting. Thus we argue that the score-level fusion by the mean operator is better.

6 Further Discussion

6.1 Comparison with Ensemble Methods

On the top of MD_{fit} based on the fine-tuned PLM, GNOME is free of modification to the model architecture or training, and only requires an extra inference of the off-shelf PLM to obtain pre-trained features, thus being practical for real-world deployment. For a strictly fair comparison under the same inference overhead constraint, we compare GNOME with previous ensemble meth-

Methods	#Passes	Avg.	NSS	SS
<i>Single Pass</i>				
MSP	1	72.92	88.94	56.89
MD _{fit}	1	57.42	80.95	33.89
MD _{pre}	1	50.29	29.90	70.68
<i>Model Ensemble</i>				
MSP	2	70.25	88.45	52.05
MD _{fit}	2	56.09	81.35	30.82
MSP	5	68.59	88.37	48.80
MD _{fit}	5	55.35	80.64	30.11
<i>Dropout Ensemble</i>				
MC Dropout	2	72.68	88.41	56.95
MC Dropout	5	70.75	85.88	55.62
GNOME	2	36.50	36.88	36.10

Table 7: Comparison with ensemble methods on the developed benchmark. We report FAR95 values averaged on the ID/OOD pairs in both SS and NSS settings.

ods, which can be divided into two groups: (1) *Model ensemble* (Lakshminarayanan et al., 2017): summing confidence scores derived from models trained over different random seeds (we apply it to MSP and MD_{fit}); (2) *MC Dropout* (Gal and Ghahramani, 2016): summing the probabilities output by multiple inferences with dropout on.

As the results shown in Table 7, previous ensemble methods that require 2× or 5× forward passes only slightly raise the performance compared with their single-pass counterparts, and fall far behind GNOME in terms of the average detection performance. These results also substantiate the power of integrating pre-trained and fine-tuned features. We do not compare with the *k*-Folden method (Li et al., 2021) that needs $(C - 1)$ sub-models (C is the number of ID classes) because it does not apply to binary classification problems and is expensive for large-scale problems where C is large.

6.2 The Choice of Pre-Trained Features

In the main experiments, we adopt the last-layer CLS embeddings as the pre-trained features for simplicity and fair comparison between MD_{pre} and MD_{fit}. As works on unsupervised textual OOD detection (Xu et al., 2021) and unsupervised sentence embedding (Su et al., 2021) show, pooling operations such as token-level and layer-level averaging produce better pre-trained features. We then alternatively use *last-avg* (the average of token embeddings in the last layer) and *first-last-avg* (the average of token embeddings in the first and last layers) embeddings as pre-trained features in MD_{pre} and GNOME. As shown in Table 8, when

Pre-trained Features	Methods	Avg.	NSS	SS
-	MD _{fit}	57.42	80.95	33.89
<i>last-cls</i>	MD _{pre}	50.29	29.90	70.68
	GNOME	36.50	36.88	36.10
<i>last-avg</i>	MD _{pre}	46.28	36.14	56.41
	GNOME	35.88	38.33	33.43
<i>first-last-avg</i>	MD _{pre}	41.77	36.00	47.54
	GNOME	35.89	37.93	33.85

Table 8: The OOD detection performance (FAR95↓ in percentage) of different pre-trained features.

Backbone	Methods	Avg.	NSS	SS
BERT _{base-uncased}	MD _{pre}	67.81	60.61	75.01
	MD _{fit}	59.06	85.05	33.08
	GNOME	50.98	65.42	36.55
RoBERTa _{base}	MD _{pre}	50.29	29.90	70.68
	MD _{fit}	57.42	80.95	33.89
	GNOME	36.50	36.88	36.10
RoBERTa _{large}	MD _{pre}	71.17	58.72	83.60
	MD _{fit}	58.98	83.33	34.62
	GNOME	44.92	54.08	35.76

Table 9: Textual OOD detection performance (FAR95↓ values on average) with different pre-trained backbones.

the *last-cls* embeddings are replaced with the *last-avg* or *first-last-avg* embeddings, MD_{pre} is moderately degraded in the NSS setting (~7% FAR95 increase), but it is drastically improved in the SS setting (~14% or ~23% FAR95 reduction). Notably, the trade-off before and after fine-tuning still holds when the *last-avg* or *first-last-avg* is used to get pre-trained features. However the pre-trained features are derived, GNOME consistently brings improvements to the average detection performance.

6.3 Generalization on Other PLMs

To demonstrate the generality of GNOME, we also test on another two PLMs: BERT_{base-uncased} (Devlin et al., 2019) (110M parameters) and RoBERTa_{large} (Liu et al., 2019) (355M parameters). As shown in Table 9, we observe that: (1) The NSS-SS trade-off is prevalent on different PLMs and GNOME brings consistent gains over baselines in terms of average performance. (2) RoBERTa_{base}, which uses more diverse pre-training data, beats BERT_{base-uncased}, suggesting that pre-training on diverse data boosts textual OOD detection; RoBERTa_{large} underperforms RoBERTa_{base}, indicating that larger models are not necessarily better at OOD detection.

7 Conclusion

Aware of the lack of a fair and comprehensive evaluation of current textual OOD detection methods, we take the first step to systematically assess them under different distribution shifts. Interestingly, we find that no single method works well in both the non-semantic shift setting and the semantic shift setting, and there exists a trade-off: fine-tuning pre-trained language models on in-distribution data benefits detecting semantic shifts but undermines detecting non-semantic shifts. After presenting empirical explanations for the trade-off from the perspective of feature distortion, we are then motivated to fully utilize both the pre-trained and fine-tuned features to obtain an efficient measurement score GNOME for better detecting diverse distribution shifts. Extensive experimental results demonstrate the efficacy and generality of GNOME. Overall, GNOME is a first step in leveraging the intuition from our observations and analysis, and we hope that this work sheds light on the behavior of pre-trained language models upon detecting different kinds of distribution shifts and inspires new methods for general textual OOD detection.

Limitations

Although our approach GNOME yields the best overall performance on the suite of benchmarks where either NSS or SS dominates and also performs best in the cross-task setting where both kinds of shifts take place, it slightly underperforms MD_{pre} and KNN_{pre} in the NSS setting and marginally lags behind MD_{ft} and KNN_{ft} in the SS setting. This is comprehensible because it is challenging for a single method to function perfectly for arbitrary OOD data without priors on the type of distribution shifts as analyzed in visual OOD detection works (Ahmed and Courville, 2020). Note again that we do not intend to present a perfect textual OOD detector capable of tackling all kinds of distribution shifts; instead, our core contributions are that we discover the trade-off between NSS and SS settings, present an empirical analysis to explain the phenomenon and provide insights to mitigate the trade-off for general textual OOD detection.

Ethical Considerations

We believe that our work leads to a better understanding of the behavior of pre-trained language models on OOD texts. We also believe that the

proposed method will facilitate the reliable deployment of NLU models since a model may face various types of OOD inputs in the wild and our method contributes to the detection performance on unknown OOD data in the average sense. All experiments in this work are conducted on open datasets and all pre-trained models that we investigate are publicly available. We do not anticipate any negative social consequences to our work and we hope to continue to build on our method and develop more effective textual OOD detectors in the future.

Acknowledgement

We sincerely thank all the anonymous reviewers for their valuable comments and advice. This work is supported by Natural Science Foundation of China (NSFC) No. 62176002. Xu Sun is the corresponding author of this paper.

References

- Faruk Ahmed and Aaron C. Courville. 2020. [Detecting semantic anomalies](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3154–3162. AAAI Press.
- Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.
- Udit Arora, William Huang, and He He. 2021. [Types of out-of-distribution texts and how to detect them](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10687–10701. Association for Computational Linguistics.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. Lof: identifying density-

- based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM.
- Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. [Recall and learn: Fine-tuning deep pretrained language models with less forgetting](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Sishuo Chen, Wenkai Yang, Xiaohan Bi, and Xu Sun. 2022. [Holistic sentence embeddings for better out-of-distribution detection](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6676–6686, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Alexis Conneau and Douwe Kiela. 2018. [SentEval: An evaluation toolkit for universal sentence representations](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. [What you can cram into a single \$\mathbb{R}^d\$ vector: Probing sentence embeddings for linguistic properties](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2126–2136. Association for Computational Linguistics.
- Gianna M. Del Corso, Antonio Gulli, and Francesco Romani. 2005. [Ranking a stream of news](#). In *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, pages 97–106. ACM.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Hanyu Duan, Yi Yang, Ahmed Abbasi, and Kar Yan Tam. 2022. [Barle: Background-aware representation learning for background shift out-of-distribution detection](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 750–764, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Desmond Elliott, Stella Frank, Khalil Sima’an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. *arXiv preprint arXiv:1605.00459*.
- Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. 2021. [Exploring the limits of out-of-distribution detection](#). In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 7068–7081.
- Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. [Large scale crowdsourcing and characterization of twitter abusive behavior](#). In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018*, pages 491–500. AAAI Press.
- Yarin Gal and Zoubin Ghahramani. 2016. [Dropout as a bayesian approximation: Representing model uncertainty in deep learning](#). In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org.
- Varun Gangal, Abhinav Arora, Arash Einolghozati, and Sonal Gupta. 2020. [Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7764–7771. AAAI Press.
- Dan Hendrycks and Kevin Gimpel. 2017. [A baseline for detecting misclassified and out-of-distribution examples in neural networks](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. [Pretrained transformers improve out-of-distribution robustness](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. 2020. [Generalized ODIN: detecting out-of-distribution image without learning from out-of-distribution data](#). In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10948–10957. IEEE.
- Haiwen Huang, Zhihan Li, Lulu Wang, Sishuo Chen, Xinyu Zhou, and Bin Dong. 2021. [Feature space singularity for out-of-distribution detection](#). In *Proceedings*

of the Workshop on Artificial Intelligence Safety 2021 (SafeAI 2021) co-located with the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021), Virtual, February 8, 2021, volume 2808 of CEUR Workshop Proceedings. CEUR-WS.org.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. [Supervised contrastive learning](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. [Fine-tuning can distort pretrained features and underperform out-of-distribution](#). In *International Conference on Learning Representations*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. [Simple and scalable predictive uncertainty estimation using deep ensembles](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413.

Ken Lang. 1995. [Newsweeder: Learning to filter net-news](#). In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 331–339. Morgan Kaufmann.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. [A simple unified framework for detecting out-of-distribution samples and adversarial attacks](#). In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177.

Xiaoya Li, Jiwei Li, Xiaofei Sun, Chun Fan, Tianwei Zhang, Fei Wu, Yuxian Meng, and Jun Zhang. 2021. [kfolden: k-fold ensemble for out-of-distribution detection](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3102–3115. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. [Learning question classifiers](#). In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Shiyu Liang, Yixuan Li, and R. Srikant. 2018. [Enhancing the reliability of out-of-distribution image detection in neural networks](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Ting-En Lin and Hua Xu. 2019. [Deep unknown intent detection with margin loss](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5496, Florence, Italy. Association for Computational Linguistics.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. [Energy-based out-of-distribution detection](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475. Curran Associates, Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. [Learning word vectors for sentiment analysis](#). In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

Rishabh Misra. 2022. News category dataset. *arXiv preprint arXiv:2209.11429*.

Rishabh Misra and Jigyasa Grover. 2021. *Sculpting Data for ML: The first act of Machine Learning*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. [Revisiting mahalanobis distance for transformer-based out-of-domain detection](#). In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13675–13682. AAAI Press.

Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. 2021. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision, pages 8558–8567.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. 2019. [Likelihood ratios for out-of-distribution detection](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14680–14691.

Yilin Shen, Yen-Chang Hsu, Avik Ray, and Hongxia Jin. 2021. [Enhancing the generalization for intent classification and out-of-domain detection in SLU](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2443–2453. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Yiyong Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. 2022. [Out-of-distribution detection with deep nearest neighbors](#). In *International Conference on Machine Learning*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al.

2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Zhisheng Xiao, Qing Yan, and Yali Amit. 2020. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *Advances in Neural Information Processing Systems*, 33.

Keyang Xu, Tongzheng Ren, Shikun Zhang, Yihao Feng, and Caiming Xiong. 2021. [Unsupervised out-of-domain detection via pre-trained transformers](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1052–1061. Association for Computational Linguistics.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. 2021. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*.

Eyup Yilmaz and Cagri Toraman. 2022. [D2U: Distance-to-uniform learning for out-of-scope detection](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2093–2108, Seattle, United States. Association for Computational Linguistics.

Wenxuan Zhou, Fangyu Liu, and Muhao Chen. 2021. [Contrastive out-of-distribution detection for pretrained transformers](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1100–1111. Association for Computational Linguistics.

Yunhua Zhou, Peiju Liu, and Xipeng Qiu. 2022. [KNN-contrastive learning for out-of-domain intent classification](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5129–5141, Dublin, Ireland. Association for Computational Linguistics.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae-ki Cho, and Haifeng Chen. 2018. [Deep autoencoding gaussian mixture model for unsupervised anomaly detection](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

A Dataset Introduction and Statistics

A.1 The Constructed Suite of Benchmarks

We show the included datasets in Table 10 and give an introduction to them as follows.

For the NSS setting, we consider two tasks: sentiment analysis and toxic detection. For sentiment analysis, we choose SST-2 (Socher et al., 2013) and IMDB (Maas et al., 2011). SST-2 contains short

Dataset	# Classes	# Train	# Dev	# Test	L
SST-2	2	6,920	872	1,821	19
IMDB	2	23,000	2,000	25,000	230
Twitter	2	69,632	7,737	8,597	17
Jigsaw	2	143,614	15,957	63,978	68
AGNews	4	115,778	3,994	3,993	23
NC	5	68,859	8,617	8,684	30
ROSTD	12	30,521	4,181	8,621	7
CLINC	150	15,000	3,000	4,500	8
AGNews _{OOD}	-	-	-	3,600	21
NC _{OOD}	-	-	-	11,402	29
ROSTD _{OOD}	-	-	-	3,090	7
CLINC _{OOD}	-	-	-	1,000	9

Table 10: Statistics of the datasets used for the constructed suite of benchmarks. **L** denotes the average length of each sample.

movie reviews by the audience, while IMDB contains longer and more professional movie reviews. Therefore, the two datasets can regard each other as OOD data representing a non-semantic shift. For toxic detection, we choose Twitter (Founta et al., 2018) and the Jigsaw dataset from a Kaggle challenge.⁴ The Twitter dataset consists of short comments on Tweet, while the Jigsaw dataset consists of longer Wikipedia comments, so they can regard each other as OOD data of the NSS type.

For the SS setting, we consider two tasks where newly emerging classes are common: news topic categorization and dialogue intent classification. For news topic categorization, we choose the AGNews (Corso et al., 2005) and News Category datasets (Misra and Grover, 2021; Misra, 2022) to construct ID/OOD pairs. Specifically, we choose four classes from AGNews and five classes from News Category as ID data and regard the remaining classes from the original datasets as OOD data. For dialogue intent classification, we use the ROSTD (Gangal et al., 2020) and CLINC (Larson et al., 2019) datasets as ID data and regard the annotated unknown intents from the original datasets as OOD data.

A.2 Cross-Task Benchmarks

For the cross-task setting, we follow Zhou et al. (2021) to use SST-2 and 20 Newsgroups (20 NG) (Lang, 1995) as ID data. 20 NG is a news categorization dataset containing 10,182 training samples, 1,132 validation samples, and 7,532 test samples. The average sample length in 20 NG is 289. Naturally, SST-2 and 20 NG can regard each other as

⁴Available at this [link](#).

Dataset	# Test	L
TREC-10	500	10
Multi30k	1,014	13
WMT16	2,000	22
RTE	3,000	48
SNLI	2,000	21

Table 11: Statistics of OOD datasets in the cross-task setting. **L** denotes the average length of each sample.

Dataset / Loss	\mathcal{L}_{ce}	$\mathcal{L}_{ce} + \mathcal{L}_{scl}$	$\mathcal{L}_{ce} + \mathcal{L}_{margin}$
SST-2	93.96	94.23	93.69
IMDB	94.56	94.53	94.21
Twitter	93.67	93.64	93.81
Jigsaw	81.82	82.08	82.43
NC	95.39	95.21	95.43
AGNews	91.28	91.03	91.18
ROSTD	99.23	99.21	99.26
CLINC	96.21	96.16	96.08
20 NG	84.52	84.65	84.53

Table 12: Accuracies / F1 scores on the test set of in-distribution data (averaged over five random seeds). We report F1 scores for Twitter and Jigsaw toxic detection and accuracies for other tasks.

OOD data. Besides, we use five additional datasets from different datasets as OOD test data for each ID dataset: TREC-10 (Li and Roth, 2002), WMT-16 (Bojar RTE (Dagan et al., 2005), and SNLI (Bowman et al., 2015)). TREC-10 is a question classification dataset; Multi30k (Elliott et al., 2016) and WMT16 (Bojar et al., 2016) are parts of the English side data of English-German machine translation datasets; RTE (Dagan et al., 2005) and SNLI (Bowman et al., 2015) are the concatenations of the precise and respective hypotheses from NLI datasets. The statistics of the OOD datasets are listed in Table 11.

A.3 Probing Benchmarks

To probe the linguistic information contained in pre-trained and fine-tuned features, we use two probing tasks designed by Conneau et al. (2018). Each probing dataset contains 100k training samples, 10k validation samples, and 10k test samples. We use the SentEval toolkit (Conneau and Kiela, 2018) along with the recommended hyperparameter space to search for the best probing classifier according to the validation accuracy and report test accuracies.

B Performance on In-Distribution Data

We fine-tune the RoBERTa_{base} model on the ID training data to build text classifiers in our main experiments. The model is optimized with the

Adam (Kingma and Ba, 2015) optimizer using a learning rate of $2e-5$. We use a batch size of 16 and fine-tune the model for 5 epochs. We evaluate the model on the ID validation set after every epoch and choose the best checkpoint as the final model. The setting is the same for other pre-trained Transformers studied in the paper (BERT_{base-uncased} and RoBERTa_{large}). The performance of fine-tuned RoBERTa_{base} models is given in Table 12, where \mathcal{L}_{ce} denotes the vanilla cross-entropy loss, \mathcal{L}_{scl} denotes the supervised contrastive loss (Khosla et al., 2020), and \mathcal{L}_{margin} denotes the margin-based contrastive loss (Zhou et al., 2021). We report the F1 scores on the test set for toxic detection on Twitter and Jigsaw and test accuracies for other tasks.

C Details of OOD Detection Baselines

C.1 Confidence-Based Baselines

Notations In a classification problem with C classes, assume the input is \mathbf{x} , we denote $f_i(\mathbf{x})$ is the output logit of class i , and the predicted softmax probability of class i is defined as:

$$p_i(\mathbf{x}) = \max_i \frac{\exp(f_i(\mathbf{x}))}{\sum_{j=1}^C \exp(f_j(\mathbf{x}))}. \quad (5)$$

Confidence-based methods obtain the OOD score based the output logits and softmax probabilities.

MSP Hendrycks and Gimpel (2017) propose the maximum softmax probability (MSP) baseline, in which the confidence score is defined the predicted maximum softmax probability among C classes:

$$S(\mathbf{x}) = \max_i p_i(\mathbf{x}). \quad (6)$$

Scaling In the ODIN paper (Liang et al., 2018), temperature scaling is applied to the scoring function:

$$S(\mathbf{x}) = \max_i \frac{\exp(f_i(\mathbf{x})/T)}{\sum_{j=1}^C \exp(f_j(\mathbf{x})/T)}, \quad (7)$$

where T is the temperature term. Following Hsu et al. (2020), we fix $T = 1000$ in our experiments. Note that ODIN (Liang et al., 2018) also propose an input pre-processing step adding adversarial perturbation to the input image, while we do not use it because it is not directly applicable for discrete inputs in NLP.

Energy Score (Liu et al., 2020) propose to use the free energy function for OOD detection, which

is formulated as follows:

$$E(\mathbf{x}) = \sum_{i=1}^C e^{f_i(\mathbf{x})}, \quad (8)$$

$$S(\mathbf{x}) = -E(\mathbf{x}).$$

D2U Yilmaz and Toraman (2022) propose to improve out-of-scope detection by exploiting the shape of the entire output distribution. Specifically, the distance of the output distribution $P(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_c(\mathbf{x}))$ to the uniform distribution U as the OOD score:

$$S(\mathbf{x}) = \text{dst}(P(\mathbf{x}), U), \quad (9)$$

where dst is the distance function. We use the KL divergence as the distance function as recommended in Yilmaz and Toraman (2022) in our experiments. Note that Yilmaz and Toraman (2022) also propose to use D2U for loss calculation when out-of-scope training data is available, while we do not use it in the training because we follow the mainstream setting in OOD detection works where OOD data is not available for training.

C.2 Density-Based Baselines

PPL Arora et al. (2021) propose to use the token perplexity (PPL) score derived from the GPT-2 language model (Radford et al., 2019) as the OOD score. Following the implementation of Arora et al. (2021),⁵ we fine-tune the GPT-2_{small} model (117M parameters, similar to RoBERTa_{base} in size) for language modeling on the ID training data and use the inverse of the PPL score as the OOD score. Formally, for an input text sequence $\mathbf{x} = \{x_1, \dots, x_t\}$,

$$\text{PPL}(\mathbf{x}) = \exp \left\{ -\frac{1}{t} \sum_i \log p_\theta(x_i | x_{<i}) \right\}, \quad (10)$$

$$S(\mathbf{x}) = 1/\text{PPL}(\mathbf{x}),$$

where t is the number of tokens in \mathbf{x} .

C.3 Distance-Based Baselines

Local Outlier Factor (LOF) Lin and Xu (2019) propose to identify unknown user intents by feeding feature vectors derived from LSTM models to the density-based novelty detection algorithm, local outlier factor (LOF) (Breunig et al., 2000). In our implementation, we use the last-layer CLS vector embeddings by the fine-tuned RoBERTa models as the input and train a LOF model following the

⁵Available at this Github repository.

implementation details of Lin and Xu (2019) on the ID training set. Finally, we use the local density output as $S(\mathbf{x})$.

Mahalanobis Distance Detector The Mahalanobis distance detector (MD) (Lee et al., 2018) is a classical distance-based OOD detection method that exploits the sample distance to the nearest ID class in the embedding space to obtain the OOD score. Formally, for a given feature extractor ψ , the Mahalanobis distance score is defined as:

$$S(\mathbf{x}) = -\min_{c \in \Upsilon} (\psi(\mathbf{x}) - \mu_c)^T \Sigma^{-1} (\psi(\mathbf{x}) - \mu_c), \quad (11)$$

where $\Upsilon = \{1, 2, \dots, C\}$ is the label space containing C classes in the ID task, $\psi(\mathbf{x})$ is the embedding vector of the input \mathbf{x} , μ_c is the class centroid for a class c , and Σ is the covariance matrix. The estimations of μ_c and Σ are defined as:

$$\begin{aligned} \mu_c &= \frac{1}{N_c} \sum_{\mathbf{x} \in \mathcal{D}_{in}^c} \psi(\mathbf{x}), \\ \Sigma &= \frac{1}{N} \sum_{c \in \Upsilon} \sum_{\mathbf{x} \in \mathcal{D}_{in}^c} (\psi(\mathbf{x}) - \mu_c) (\psi(\mathbf{x}) - \mu_c)^T, \end{aligned} \quad (12)$$

where $\mathcal{D}_{in}^c = \{\mathbf{x} \mid (\mathbf{x}, y) \in \mathcal{D}_{in}, y = c\}$ denotes the training samples belonging to the class c , N is the size of the training set, and N_c is the number of training instances belonging to the class c . As for textual OOD detection based on pre-trained language models, when the feature extractor ψ is the off-the-shelf pre-trained model, i.e. detecting anomalies in the pre-trained feature space (Xu et al., 2021), it is called MD_{pre} in our paper; when ψ is the fine-tuned model, i.e. detecting anomalies in the fine-tuned feature space (Podolskiy et al., 2021), it is called MD_{fit} in our paper.

Contrastive Fine-Tuning Targets Coupled with the MD Detector Zhou et al. (2021) propose to use two forms of contrastive losses to boost textual OOD detection, i.e., the supervised contrastive loss (\mathcal{L}_{scl}) and the margin-based contrastive loss (\mathcal{L}_{margin}). For a classification task containing C classes, given a batch of training examples $\{x_i, y_i\}_{i=1}^M$, where x_i is the input and y_i is the label, the supervised contrastive loss term \mathcal{L}_{scl} and the final optimization target \mathcal{L} can be formulated as:

$$\begin{aligned} \mathcal{L}_{scl} &= \sum_{i=1}^M \frac{-1}{M|P(i)|} \sum_{p \in P(i)} \log \frac{e^{\mathbf{z}_i^T \mathbf{z}_p / \tau}}{\sum_{a \in A(i)} e^{\mathbf{z}_i^T \mathbf{z}_a / \tau}}, \quad (13) \\ \mathcal{L} &= \mathcal{L}_{ce} + \mathcal{L}_{scl}, \end{aligned}$$

where $A(i) = \{1, \dots, M\} \setminus \{i\}$ is the set of all anchor samples, $P(i) = \{p \in A(i) : y_i = y_p\}$ is the

set of anchor samples from the same class as i , τ is a temperature hyper-parameter, \mathbf{z} is the L2-normalized CLS embedding before the softmax layer, \mathcal{L}_{ce} is the cross-entropy loss, and λ is a positive coefficient. Following the implementation of Zhou et al. (2021),⁶ we use $\tau = 0.3$ and $\lambda = 2$.

The margin-based loss term \mathcal{L}_{margin} and the final optimization target \mathcal{L} is formulated as:

$$\begin{aligned} \mathcal{L}_{pos} &= \sum_{i=1}^M \frac{1}{|P(i)|} \sum_{p \in P(i)} \|\mathbf{h}_i - \mathbf{h}_p\|^2, \\ \mathcal{L}_{neg} &= \sum_{i=1}^M \frac{1}{|N(i)|} \sum_{n \in N(i)} (\xi - \|\mathbf{h}_i - \mathbf{h}_n\|)^2_+, \\ \mathcal{L}_{margin} &= \frac{1}{dM} (\mathcal{L}_{pos} + \mathcal{L}_{neg}), \\ \xi &= \max_{i=1}^M \max_{p \in P(i)} \|\mathbf{h}_i - \mathbf{h}_p\|^2, \\ \mathcal{L} &= \mathcal{L}_{ce} + \lambda \mathcal{L}_{margin}, \end{aligned} \quad (14)$$

where $N(i) = \{n \in A(i) : y_i \neq y_n\}$ is the set of anchor samples from other classes than y_i , $\mathbf{h} \in \mathbb{R}^d$ is the unnormalized CLS embedding before the classification head, ξ is the margin, d is the number of dimensions of \mathbf{h} , and λ is a positive coefficient. We use $\lambda = 2$ following Zhou et al. (2021).

Except for the optimization target, we use the same hyper-parameters for the two tuning methods as vanilla tuning.

Nearest-Neighbor-Based Detector Sun et al. (2022) explore the efficacy of non-parametric nearest-neighbor distance for OOD detection and show its advantages over the Mahalanobis distance detector on visual OOD detection benchmarks. Specifically, it takes the minus of the average distance from the test sample to the k -nearest training samples in the normalized feature space. We reproduce two variants, i.e., KNN_{pre} using the pre-trained features and KNN_{fit} using the fine-tuned features. We set the neighborhood size $k = 10$ in our experiments.

D Software and Hardware Requirements

We implement our code based on the PyTorch (Paszke et al., 2019) and HuggingFace Transformers (Wolf et al., 2020) Python libraries. All experiments (training and inference) in this paper can be conducted on a single NVIDIA TITAN RTX GPU (24 GB memory), except that the fine-tuning of the RoBERTa_{large} model needs 4 TITAN RTX GPUs.

⁶Available at this Github repository

Methods	Avg.	Non-Semantic Shift (NSS)				Semantic Shift (SS)			
		SST-2	IMDB	Twitter	Jigsaw	NC	AGNews	ROSTD	CLINC
MSP (Hendrycks and Gimpel, 2017)	71.62	67.92	74.09	48.75	71.13	75.12	64.84	75.42	95.72
Scaling (Liang et al., 2018)	71.96	67.92	74.09	48.76	71.03	74.60	67.35	75.71	96.20
Energy (Liu et al., 2020)	71.63	69.73	72.84	47.56	69.49	74.19	67.55	76.52	96.18
D2U (Yilmaz and Toraman, 2022)	71.99	67.92	74.09	48.75	71.13	74.62	67.46	75.72	96.26
PPL (Arora et al., 2021)	67.65	79.65	98.51	34.61	84.36	57.91	50.67	85.05	50.47
LOF (Lin and Xu, 2019)	76.42	53.39	94.87	62.14	57.88	78.39	70.07	97.49	97.17
MD _{fit} (Podolskiy et al., 2021)	80.39	69.86	90.87	65.83	62.42	3.41	73.51	99.66	97.57
MD _{fit} + \mathcal{L}_{scf} (Zhou et al., 2021)	82.22	83.12	88.28	71.13	64.32	81.68	72.74	99.09	97.39
MD _{fit} + $\mathcal{L}_{\text{margin}}$ (Zhou et al., 2021)	86.50	93.84	98.99	83.04	65.94	82.91	72.00	97.68	97.56
MD _{pre} (Xu et al., 2021)	83.76	99.99	98.75	90.59	83.84	57.45	61.53	95.22	82.69
KNN _{fit} (Sun et al., 2022)	81.02	72.00	86.07	74.90	57.33	84.82	75.85	99.67	97.53
KNN _{pre} (Sun et al., 2022)	85.69	99.99	98.31	92.80	79.53	65.90	69.28	96.89	82.81
GNOPE (Ours)	89.34	99.98	98.25	89.64	81.10	75.50	73.77	99.63	96.84

Table 13: OOD detection performance (AUROC \uparrow , higher is better) on the developed suites of benchmarks. All values are percentages averaged over five different random seeds, and the best results are highlighted in **bold**. The last column gives the average performance on eight datasets.

E Additional Experimental Results

We display the AUROC results of GNOPE and the baselines on the constructed suite of benchmarks in Table 13. The overall trend is consistent with that of the FAR95 results reported in Table 4 in the main text.

A Question of Style: A Dataset for Analyzing Formality on Different Levels

Elisabeth Eder and Ulrike Krieg-Holz and Michael Wiegand

Universität Klagenfurt, Klagenfurt, Austria

{elisabeth.eder | ulrike.krieg-holz | michael.wiegand}@aau.at

Abstract

Accounting for different degrees of formality is crucial for producing contextually appropriate language. To assist NLP applications concerned with this problem and formality analysis in general, we present the first dataset of sentences from a wide range of genres assessed on a continuous informal-formal scale via comparative judgments. It is the first corpus with a comprehensive perspective on German sentence-level formality overall. We compare machine learning models for formality scoring, a task we treat as a regression problem, on our dataset. Finally, we investigate the relation between sentence- and document-level formality and evaluate leveraging sentence-based annotations for assessing formality on documents.

1 Introduction

Textual style can be approached from various points of view. We focus on its inherent formality dimension stretching from informal to formal language use. See these two sentences, for example:

- (1) We gave thorough thought to an adequate example.
Wir haben gründlich über ein adäquates Beispiel nachgedacht.
- (2) racked our brains about a niice example... :D
haben uns den kopf über ein schöönes beispiel zermartert... :D

While both sentences transport the same content, they differ in their degree of formality. (2) is less formal than (1). It may be suitable only for more informal discourse contexts and inappropriate in formal settings. Understanding these different nuances of formality is crucial for effective communication. Consequently, striking the right tone is relevant not only for humans but also for various NLP applications. May it be machine translation in need to transfer expressions of formality between different languages adequately (Niu and Carpuat, 2020; Anastasopoulos et al., 2022), chatbots aiming to produce contextually appropriate language to increase user satisfaction (Chaves et al., 2019;

Elsholz et al., 2019), or writing assistance systems altering content to be more formal (Saber et al., 2020). Hence, intra-lingual formality style transfer, which deals with generating a formal phrase given its informal version and vice versa, has recently also received increased attention (e.g., Shang et al., 2019 or Zhang et al., 2020).

Our paper addresses a prerequisite for this task: **assessing linguistic formality**. Rating the transferred style strength is necessary for evaluating formality style transfer models. Further, parallel corpora with formal and informal language pairs, often the basis for style transfer, are commonly built by automatically grading and extracting informal sentences first (Rao and Tetreault, 2018; Briakou et al., 2021b). For facilitating such formality assessments and analyzing linguistic formality in general, we make the following **contributions**:

1. We present the **first dataset of sentences from a wide range of genres with human formality assessments on a continuous informal-formal scale**. We ensure a comprehensive perspective on formality by collecting sentences from diverse domains. Formality annotations are obtained via a comparative annotation variant (annotators compare items to each other), which is not only more reliable than the rating scale method (Kiritchenko and Mohammad, 2017) but also satisfies the principle that a “continuum of formality” (Heylighen and Dewaele, 1999) exists rather than categorical distinctions. The dataset is the **first** to target **German** sentence-level formality unrestrictedly overall.

2. We **evaluate several machine learning models** for formality scoring on our dataset, which we treat as a **regression task**. Regression models have been found to be more suitable than classifiers for evaluating formality style transfer models since they grasp the broad spectrum of linguistic formality (Briakou et al., 2021a). Besides fine-tuning transformers on our dataset, we examine utilizing formality-informed corpora from different lan-

guages with coarser or narrower representations of formality. Further, we employ feature-based approaches for formality scoring and analyze linguistic properties that constitute formality. For such analyses, we provide a tool with a variety of features for profiling characteristics of registers, genres, and author styles for various languages.

3. We investigate the applicability of sentence-level formality annotations for the formality assessment of documents. Lately, [Jin et al. \(2022\)](#) proposed extending formality style transfer, which so far exclusively focuses on the sentence level, to stylistically more complex documents. However, datasets targeting formality on this scope are rare and limited in size, probably because obtaining annotations is more expensive. Therefore, we analyze how sentence formality contributes to the formality of documents.

2 Related Work

With their continuous formality score based on frequencies of parts of speech, [Heylighen and Dewaele \(1999\)](#) established a milestone for the definition of formality. [Lahiri et al. \(2011\)](#) adapted this measure from the document to the sentence level. Most approaches targeting the lexical dimension of formality also regarded formality as a continuum ([Brooke et al., 2010](#); [Brooke and Hirst, 2014](#); [Pavlick and Nenkova, 2015](#); [Eder et al., 2021](#)).

To the best of our knowledge, datasets comprising sentences with human formality assessments on a continuous informal-formal scale have not been constructed before. [Pavlick and Tetreault \(2016\)](#) built an English dataset collecting formality annotations on a 7-point Likert scale for sentences from only four sources (compared to the twelve in our dataset). They introduced formality detection as a regression task using features based on analyzing human perceptions of formality for a ridge regression model. Other datasets targeting sentence-level formality have binary labels since they primarily serve as parallel data for formality style transfer and contain formal and informal language pairs. They cover English ([Rao and Tetreault, 2018](#); [Cheng et al., 2020](#)), Brazilian Portuguese, French and Italian ([Briakou et al., 2021b](#)), and Hindi, Bengali, Kannada and Telugu ([Krishna et al., 2022](#)).

Work on formality style transfer mainly used classification for measuring style strength and a handful of different classifiers (e.g., [Lai et al. \(2021\)](#) employed a CNN, [Wang et al. \(2019\)](#) an

LSTM, and [Krishna et al. \(2020\)](#) transformers). Evaluating the style strength as a regression task, [Rao and Tetreault \(2018\)](#) borrowed the approach from [Pavlick and Tetreault \(2016\)](#), and [Briakou et al. \(2021b\)](#) relied on fine-tuning transformers.

For the German language, not yet considered for intra-lingual formality style transfer, two sentence collections with binary formality annotations based on formal and informal direct address exist ([Faruqui and Padó, 2012](#); [Nadejde et al., 2022](#)). (Since these formality levels do not exist in English, they pose a problem for machine translation ([Nadejde et al., 2022](#).) Hence, these datasets target a very constrained view of formality only.

Focusing on the document level, several works used traditional machine learning models for binary formality classification based on linguistic features. As training data, [Abu Sheikha and Inkpen \(2010\)](#) assumed binary labels for formality from the text genre, and [Peterson et al. \(2011\)](#) manually annotated emails from the English *ENRON* corpus ([Klimt and Yang, 2004](#)) with four formality classes. Treating formality assessment on documents as a regression task, [Chhaya et al. \(2018\)](#) employed linguistic features for formality scoring on *ENRON* emails, which have been rated on a 5-point Likert scale, whereas [Eder et al. \(2021\)](#) evaluated word formality scoring on emails from the German corpus *CodE Alltag* ([Eder et al., 2020](#)) based on continuous formality annotations. All these manually labeled document collections are small in size ($\sim 1k$) and built from a single domain only, i.e., emails. None of these works leverages formality-annotated sentences nor fine-tunes transformer models to assess the formality of documents.

3 Data

To build our dataset, we collected 3,000 German (DE) sentences from different domains and let crowdworkers assess their formality on a continuous formality scale via comparative annotations.

3.1 Collecting Sentences

We chose twelve different text sources, which we assumed to be related to diverse levels of formality, to cover the broad spectrum of linguistic formality best possible. From each source, we took 250 sentences. We picked these sentences randomly, but they had to consist of at least one word. Additionally, we attempted to enhance language variety by selecting a minimum number of sentences per au-

thor. We also tried spreading the data over different topics whenever such information was available.¹

We utilized the following sources:

Tweets. We rehydrated tweets from a German *Twitter* snapshot (Scheffler, 2014).

Reddit. We extracted posts from the *GeRedE* corpus, which contains German communication on *Reddit* (Blombach et al., 2020).

Subtitles. To account for spoken language, we included German sentences from the *OpenSubtitles* collection of parallel corpora with movie and TV subtitles (Lison and Tiedemann, 2016).

Comments. 250 sentences were collected from the *One Million Posts Corpus*, which comprises comments on news articles (Schabus et al., 2017).

Emails. We took sentences from *Code Alltag*, a corpus with German emails (Eder et al., 2020).

Blogs. Using the *DWDS* platform (Geyken et al., 2017), we obtained sentences from a blog corpus (Barbaresi and Würzner, 2014).

Fiction. Due to the lack of accessible corpora covering contemporary fictional texts, we reverted to an archive that, besides fan fiction, contains original work from nonprofessional writers.² We extracted 250 sentences from their short stories.

News. We gathered sentences from the German news corpus from 2020 provided in the *Leipzig Corpora Collection* (Goldhahn et al., 2012).

Wikipedia. From the *Leipzig Corpora Collection*, we also used sentences from the German *Wikipedia* corpus from 2021.

Political. For potentially more formal spoken language examples, we extracted sentences from German political speeches that are included in the parallel corpus *EuroParl* (Koehn, 2005).

Legal. We gained sentences from the legal domain by utilizing a dataset with German court decisions (Leitner et al., 2019).

Science. We used *Springer Link*³ to manually collect sentences from scientific journals, proceedings, and books published between 2000 and 2022 under open access.

3.2 Human Assessment

We gathered human formality assessments for the resulting 3,000 sentences using Best-worst scaling (BWS) (Louviere et al., 2015), a form of comparative annotation. BWS delivers more reliable an-

notations than the rating scale method mitigating issues such as a scale region bias or inconsistent annotations (Kiritchenko and Mohammad, 2017). Further, it complies with the notion of formality as a continuum (Heylighen and Dewaele, 1999).

For BWS, annotators are presented with n items at a time (typically $n = 4$). They have to decide which item from the n -tuple is the *best* and which is the *worst* (i.e., the highest and the lowest regarding the property of interest). To get real-valued scores from these BWS annotations, the percentage of times the term is chosen as worst is subtracted from the percentage of times the term is chosen as best (Counts Analysis (Orme, 2009)). Thus, each item receives a score between +1 (most formal) and -1 (most informal).

We randomly generated $2N$ 4-tuples (where N denotes the number of sentences) under the premise that each term occurs only once in eight different tuples and each tuple is unique.⁴ For the annotation process proper, we chose crowdsourcing to ensure the heterogeneity of annotators. Using the crowdsourcing platform *Clickworker*⁵, German native speakers assessed each of the 6,000 tuples five times. Thus, we collected 30,000 annotations from 1,084 different annotators, with an average of 27.7 annotations per annotator.

All five annotators agreed in 19% of the annotations. In two-thirds, three or four annotators chose the same item, while only in 15% just two of the answers matched. The higher the difference between the real-valued formality scores of two sentences, the higher the agreement of the crowdworkers. For a score difference of just 0.1, the agreement is 64%. It rises to over 70% for higher score differences, with over 80% for differences higher than 0.4 and at least 90% for differences over 0.7.

We computed the split-half reliability⁴ for our formality-assessed dataset by randomly splitting the annotations of a tuple into two halves, calculating scores independently for these halves, and measuring the correlation between the resulting two sets of scores. We got an average Spearman's ρ of 0.919 (± 0.002) over 100 trials, which indicates a high reliability of the annotations.

3.3 The Final Dataset

Figure 1 displays the distribution of human-assessed formality scores for each of the twelve

¹For some corpora, we subsumed subreddits, blogs, genres, or articles, to which comments refer, in place of topic.

²<https://www.fanfiktion.de/>

³<https://link.springer.com/>

⁴We employed scripts developed for emotion scaling by Kiritchenko and Mohammad (2016, 2017).

⁵<https://www.clickworker.de>

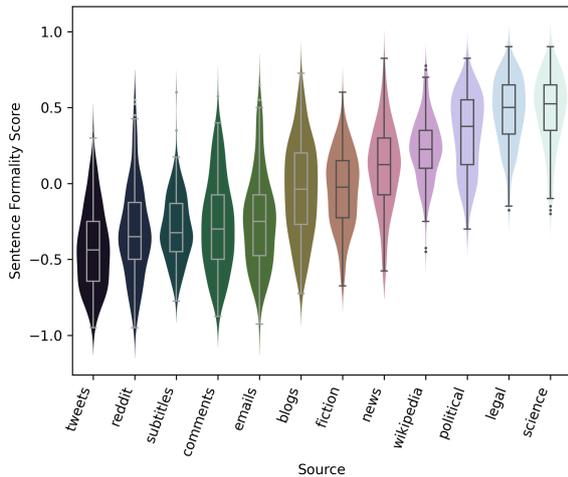


Figure 1: Distribution of formality scores for our 3,000 sentences per each of the twelve sources ordered by the average formality score of the source.

sources of the 3,000 sentences in our dataset. As expected, sentences from online communication or sources with more spontaneous language use, e.g., *tweets* or *comments*, tend to be linked to lower scores, while sentences with more elaborated language use, e.g., *legal* or *scientific* texts, have higher scores. However, sources scatter broadly, and assuming the same degree of formality per genre seems inappropriate.

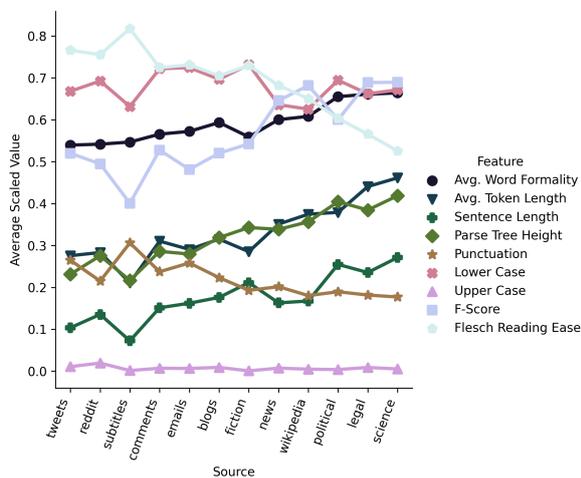


Figure 2: Averages of simple linguistic characteristics (scaled to a range between 0 and 1) of sentences for each source; sources ordered by their mean formality.

In Figure 2, we plot some simple linguistic features, which have been studied in relation to formality (Heylighen and Dewaele, 1999; Pavlick and Tetreault, 2016, i.a.) for each source. The mean word formality, token length, sentence length and parse tree height rise for sources with higher av-

erage sentence formality scores. The proportion of punctuation characters tends to sink, whereas the ratios of upper- or lower-case tokens are more stable. Heylighen and Dewaele’s (1999) F-score indicates a higher formality and the readability score Flesch Reading Ease (Flesch, 1948) signals a lower readability for sources with higher mean formality.

In the following, we explore such properties for scoring the formality of the individual sentences.

4 Formality Scoring on Sentences

We compared different models for predicting formality scores for sentences on our dataset.

4.1 Within-Dataset Experiments

Transformers. We experimented with fine-tuning transformer models on our dataset. For that, we employed GBERT-base (Chan et al., 2020), a German BERT language model.⁶ For all transformer-based experiments, we used the NLP library *FLAIR* (Akbi et al., 2019) as a framework.

Feature-based Models. We evaluated two feature-based models, which allowed us to examine the influence of linguistic characteristics more directly. The first ridge regression model employs eleven different feature groups and was developed for scoring the formality of English sentences (Pavlick and Tetreault, 2016). The second was created for English documents, more precisely emails (Chhaya et al., 2018). It borrows features from the first model and extends them with affect-based features. We adapted these feature sets to German and adjusted them to work on sentences and documents. We also employed a ridge regression model. Table 5 in the Appendix contains a detailed breakdown of the features we implemented.

4.2 Cross-Dataset Experiments

Learning from Other Languages. We examined using English sentences with formality scores determined via averaging over individual annotations on a 7-point Likert scale (Pavlick and Tetreault, 2016). This dataset (*PT16* in the following) contains about 11k sentences from four sources: news and blogs from Lahiri (2015) extended by emails and Q&A sites. We evaluated three different settings. We fine-tuned GBERT-base transformers on *PT16* translated to German and tested them on

⁶Other German transformers (Chan et al., 2020; Minixhofer et al., 2022) either yielded no significant difference or performed worse (see Table 3 in the Appendix).

Training		Testing		Model	Spearman’s ρ
sentences	ours (de)	sentences	ours (de)	GBERT	0.919 (± 0.009)
	ours (de)		ours (de)	feature-based (\sim Pavlick and Tetreault, 2016)	0.857* (± 0.007)
	ours (de)		ours (de)	feature-based (\sim Chhaya et al., 2018)	0.830* (± 0.018)
	<u>PT16 (de)</u>		ours (de)	GBERT	0.877* (± 0.018)
	PT16 (en)		ours (de)	XLM-RoBERTa	0.847* (± 0.017)
	PT16 (en)		ours (<u>en</u>)	BERT	0.844* (± 0.022)
	XFORMAL (br-pt+fr+it)		ours (de)	XLM-RoBERTa	0.768* (± 0.020)
	GYAFC (en)		ours (de)	XLM-RoBERTa	0.716* (± 0.023)
	FP12 (de)		ours (de)	GBERT	0.595* (± 0.042)

Table 1: Evaluation of different models for formality scoring on our sentences; ‘*’ stands for a statistically significant difference of $p < 0.005$ with respect to **best** model (using two-sided Wilcoxon signed-rank test on Spearman’s ρ); language(s) of datasets in brackets, translated data underlined.

our dataset. Consequently, we utilized BERT-base (Devlin et al., 2019) for fine-tuning on the original English *PT16* and testing on the English translation of our dataset. Further, we fine-tuned multilingual XLM-RoBERTa-base transformers (Conneau et al., 2020) on the English *PT16* and tested them on our German sentences. For the translations in both directions, we employed the models from Edunov et al. (2018) via the *fairseq* toolkit (Ott et al., 2019).

Formality Classifiers. Since there are huge datasets with binary formality annotations, we evaluated binary formality classifiers leveraging these data. We used the probability of the class determined by the classifiers as a prediction of a formality score. For the informal class, we took the probability as a negative number, thus ending up with scores from -1 to $+1$. Lacking more comprehensive German data, we experimented with a dataset from Faruqui and Padó (2012) that comprises 60k German sentences with binary formality annotations based only on formal and informal direct address (unmarked in English yet explicitly marked in German). We fine-tuned GBERT on this dataset, *FP12* in the following, for binary classification. As *FP12* is limited to this particular case of formality, we further utilized parallel datasets with formal and informal language pairs from languages other than German. These parallel datasets, containing informal sentences from a Q&A forum and their formal rewrites, are *GYAFC* with 110k English sentences (Rao and Tetreault, 2018) and *XFORMAL* with 23k Brazilian Portuguese, French and Italian sentences (Briakou et al., 2021b). We employed binary XLM-RoBERTa-based classifiers fine-tuned on *GYAFC*⁷ and *XFORMAL*⁸.

⁷<https://github.com/martiansideofthemoon/style-transfer-paraphrase> (Krishna et al., 2020)

⁸https://huggingface.co/SkolkovoInstitute/xlmr_formality_classifier

4.3 Evaluation

Table 1 reports the average Spearman’s ρ for the different setups. Evaluated in a 10-fold cross-validation manner, the two feature-based models yielded high results. To explore their relation to formality, Figure 3 shows several linguistic features used by these models per the formality score of the sentences. While sentiment seems to be a relatively constant feature across the formality scale, other factors correlate better with formality. The punctuation ratio and the Flesch readability score tend to sink, whereas word formality, token length, constituency tree height, and the number of tokens rise with increasing sentence formality. According to *SHAP* (Lundberg and Lee, 2017)⁹, among the most important features of the approach by Chhaya et al. (2018) are indeed the sentence length, the average word formality, the Flesch score and the average token length (already achieving 0.8 Spearman’s ρ on their own). This shows that such simple linguistic properties are good indicators of formality, at least at the sentence level.

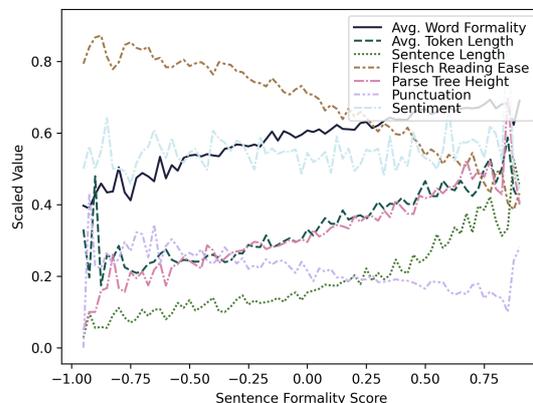


Figure 3: Relation between several linguistic features (scaled values) and the formality scores of the sentences.

⁹*SHAP* is a game theoretic approach that facilitates interpreting predictions of machine learning models.

However, fine-tuning transformers significantly outperformed the feature-based approaches (Table 1). In Figure 4, we plot the predictions of GBERT transformers fine-tuned on our dataset versus the human-assessed formality scores. The errors are lower on both ends of the scale. Sentences nearer to the scale’s middle are more difficult to predict for the model since they carry fewer linguistic markers than sentences with extreme (in)formality scores. But in general, predictions are relatively accurate.

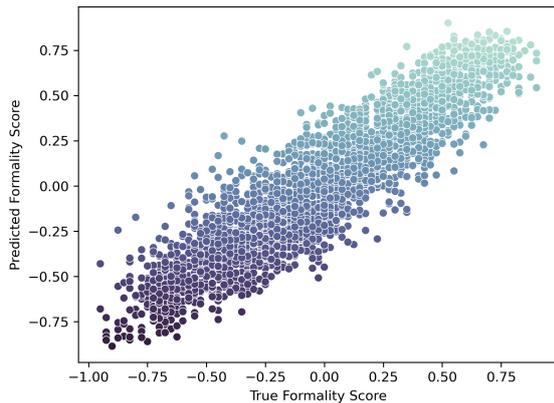


Figure 4: Predictions of the best model versus gold formality scores (brighter colors mean higher predictions).

Table 1 also shows that from the settings utilizing the *PT16* dataset, the model fine-tuned on *PT16* translated to German performed best. The formalization effect of machine translation (informal sentences get more formal through translation (Briakou et al., 2021b)) seems to influence the models using translated data since they tended to predict higher formality scores, especially for more neutral sentences. However, the results indicate that this is less critical when compared to the cross-lingual regression model fine-tuned on English and tested on German data. Contrasted to fine-tuning and testing on our dataset, the *PT16* models were still significantly worse, although *PT16* comprises over three times more sentences than our dataset. This may also be ascribed to its narrower scale of formality. *PT16* models tended to yield lower results on more formal domains of our dataset (*science*, *legal* and *Wikipedia*). Scoring these genres seems more challenging for those models since *news*, the most formal source in *PT16* (Pavlick and Tetreault, 2016), has only the fifth-highest average formality score in our dataset (see Figure 1).

The probabilities for being either formal or informal from the binary formality classifiers fine-tuned on *GYAFC* and *XFORMAL* in a cross-lingual set-

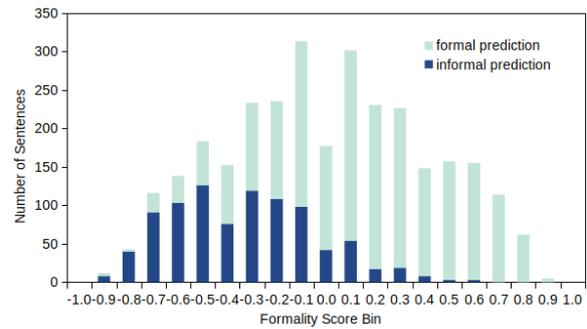


Figure 5: Formal and informal predictions of the *GYAFC* model per formality score bin of our sentences.

ting also showed a correlation to the human assessments (Table 1). However, these models performed worse than regression models. Figure 5 exemplifies the class predictions of the binary formality classifier fine-tuned on *GYAFC* per formality score bin (formality scores rounded to one decimal place) on our dataset. It shows that sentences with lower formality scores tended to be classified as informal and sentences with higher scores as formal. However, formal and informal sentences were predicted in nearly every formality score bin. From that, we infer that a binary separation of formality into formal and informal sentences is not reasonable.

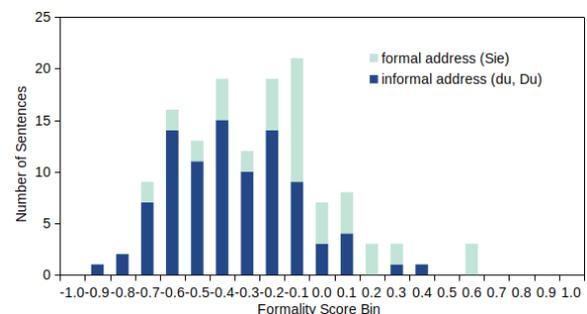


Figure 6: Distribution of formality scores of sentences with formal and informal address.

The monolingual binary classifier fine-tuned on *FP12*, which includes only formal and informal address sentences, performed significantly worse than all other setups. Figure 6 shows the number of sentences with formal and informal address in our dataset (only 137 in total) per formality score bin. Although they lean towards the lower end, even these sentences scatter broadly over the formality scale (average formality scores are -0.10 (± 0.30) for formal and -0.36 (± 0.25) for informal address). Formality is not only expressed via these different forms of address. (3) shows a sentence from our dataset with formal address but a

Training		Testing	Model	Spearman’s ρ
<i>sent.</i>	<i>d.</i>	E21 (de)	GBERT	0.891 (± 0.059)
	<u>ours</u> (de)	E21 (de)	GBERT	0.847 (± 0.028)
	<u>ours</u> (de)	E21 (de)	feature-based (\sim Pavlick and Tetreault, 2016)	0.686* (± 0.039)
	<u>ours</u> (de)	E21 (de)	feature-based (\sim Chhaya et al., 2018)	0.603* (± 0.095)
<i>sentences</i>	<i>d.</i>	C18 (en)	BERT	0.827 (± 0.041)
	<u>ours</u> (en)	C18 (en)	BERT	0.729* (± 0.059)
	<u>ours</u> (de)	C18 (en)	XLNet	0.703* (± 0.054)
	<u>ours</u> (de)	C18 (de)	GBERT	0.674* (± 0.054)
	<u>PT16</u> (en)	C18 (en)	BERT	0.603* (± 0.063)

Table 2: Results for formality scoring on documents; statistically significant differences (calculated with the two-sided Wilcoxon signed-rank test) are marked with ‘*’ for $p < 0.005$ with respect to the **best** models; language(s) of datasets in brackets, translated data underlined.

low formality score because of other indicators. Consequently, formality is a much broader concept, and restricting it to this use case is insufficient for comprehensive formality analysis.

- (3) Wollen *Sie* *formal address* nicht *reingucken* *informal*?
 Don’t *you* want to *have a look*?

5 Formality Scoring on Documents

Documents may assemble an even more diverse range of clues for degrees of formality than sentences. Only recently, Jin et al. (2022) proposed extending style transfer to the more complex document level, but manual formality annotations of documents are more expensive to obtain than sentence-level assessments. Therefore, this section investigates how single sentences and linguistic properties contribute to the overall document formality. We examine if sentence-level formality annotations are useful for assessing formality on documents.

5.1 Evaluation on German Documents

We conducted experiments and analyses on German documents. For that, we utilized 800 emails with continuous formality scores (Eder et al., 2021). Sentences from emails show the highest standard deviation of formality of all domains in our dataset and the corpus from Pavlick and Tetreault (2016). Thus they possess a high stylistic variability. We denote the dataset *E21* in the following.

We compared transformers and feature-based approaches trained on our formality-informed sentences with transformer models fine-tuned on *E21* for predicting formality on this document collection. The upper half of Table 2 presents the average Spearman’s ρ for these models. Fine-tuning GBERT on *E21* itself (10-fold cross-validation) performed best, but there is no statistically significant difference between utilizing the documents

or our formality-assessed sentences as training data. The transformer models grasped the concept of formality more comprehensively since the feature-based ridge regression models yielded significantly worse results. It seems that linguistic features do not generalize well. Figure 7 shows some of the most predictive linguistic features for formality scoring on the sentence level for the documents. The average word formality and the Flesch Reading Ease correlate with document formality in a similar way than with sentence formality (Figure 3). However, the average sentence length and average token length are comparably more static across the formality scale of documents and thus less suitable features.

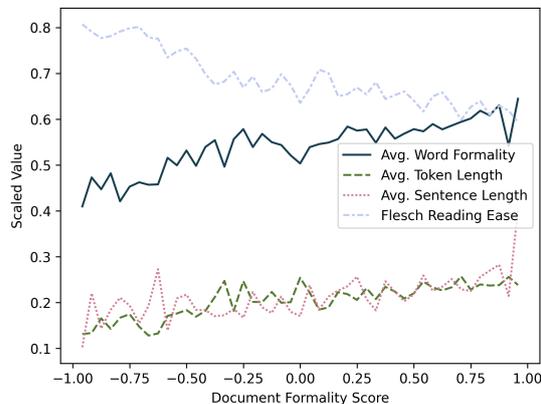


Figure 7: Linguistic characteristics (scaled values) of the documents per their formality scores.

To further understand how the formality of a document is affected by its sentences, we split the documents of *E21* into separate sentences. Then, we ran the GBERT model fine-tuned on our dataset on these sentences to determine their formality. Taking the average of the calculated scores as document score still returned a Spearman’s ρ of 0.801. Although this result is significantly worse

($p < 0.01$) than running the model on the documents directly, it still shows a strong correlation between the scores of the sentences and the document formality score. In Figure 8, we plot the number of sentences per calculated formality score bin for each formality score bin of the corresponding documents. The sentence and document formality scores show some overlap. Nevertheless, the sentences in the documents have quite a range of formality scores.

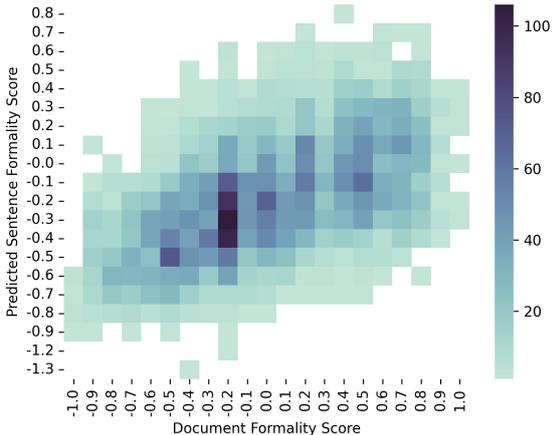


Figure 8: Frequency of calculated formality scores of sentences per formality score bin of documents.

How the formality of sentences changes throughout a document is shown in Figure 9, which depicts the mean sentence formality by position in the documents. The formality tends to decrease with increasing position in the text. This observation is in line with the assumption of Heylighen and Dewaele (1999) that formality is higher at the beginning of a text because of the lack of previous discourse to relate to. For threaded online discussions, Pavlick and Tetreault (2016) reported congruent findings.

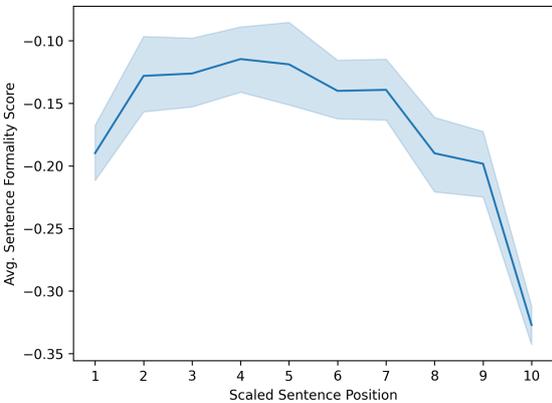


Figure 9: Average sentence formality score by position in documents; sentence positions are scaled to ten bins.

Concluding, fine-tuning transformers on sentences is applicable for assessing the formality of documents, as our results show. However, due to the variety of sentence formality scores, it may not be helpful to map formality assessments of documents to their sentences to save annotation efforts or assume mono-style documents regarding the formality dimension.

5.2 Evaluation on English Documents

To investigate the applicability of transformer models fine-tuned on our sentences for other languages, we evaluated them on English documents. We used 960 emails (C18 in the following) with formality annotations obtained via averaging over individual assessments on a 5-point Likert scale (Chhaya et al., 2018). The lower half of Table 2 shows the results. Fine-tuning on the documents (10-fold cross-validation) significantly outperformed sentence-based models. We ascribe this performance decline also to the manual annotations of C18 since we only calculated an average split-half reliability of 0.573 (± 0.015) Spearman’s ρ over 100 trials. Given these conditions, a BERT model fine-tuned on our translated dataset still achieved a high correlation also compared to the English PT16 model. Hence, we assume our dataset is beneficial for formality assessment of English-language documents too.

6 Conclusion

We presented the first dataset of sentences with highly reliable human formality assessments on a continuous informal-formal dimension obtained via Best-worst scaling. Our dataset comprises 3,000 sentences evenly distributed over twelve different domains to cover the broad spectrum of formality best possible. It is the first for the German language with a comprehensive perspective on sentence-level formality altogether.

We evaluated various machine learning models for the regression task of assessing formality on our dataset. We found that a transformer model fine-tuned on an existing German dataset including only sentences of formal and informal address (Sie vs. Du/du) yielded the worst results. Hence, this restricted view on formality is insufficient to capture a more comprehensive concept of formality. Cross-lingual settings utilizing transformer-based classifiers pre-trained on huge datasets with formal and informal language pairs not restricted

to a particular form of formality performed better. However, a binary categorization of formality strikes as inappropriate since ridge regression models employing simple linguistic features outperformed them. Fine-tuning transformers for regression on an English dataset produced similar (for the cross-lingual setting or the English translation of our dataset) or higher (for the German translation of the training data) results. In comparison, a transformer model fine-tuned on our dataset with its broader formality scale outperformed all other settings significantly.

Expanding the scope to longer texts, a requested future research direction of style transfer (Jin et al., 2022), we investigated the influence of the formality of sentences on a document’s formality. We observed that the sentences included in the documents cover a wide spectrum of formality with higher formality scores at the beginning. Our results indicate that a transformer model fine-tuned for formality scoring on our sentences generalizes better across text levels than linguistic features and can be used to predict the degree of formality of German and English documents. We anticipate our dataset to facilitate future work on German formality style transfer and formality analysis in general on both the document and the sentence level. It may also be valuable for other languages.

Our **dataset** and a **tool** for analyzing styles with a wide range of linguistic features are **available** under https://github.com/ee-2/in_formal_sentences and <https://github.com/ee-2/register>.

Limitations

This work assesses the formality of texts in isolation, excluding any conventional and situational contexts. However, for different genres and situations different expectations have to be met. For example, an expression regarded as formal in one genre may be perceived as too informal in another. We also do not take forms of formality beyond the pure text level into account. Properties that contribute to formality besides the text itself may include the structure of a text (e.g., blank lines in emails (Chhaya et al., 2018)) or the volume, the pitch, the speech rate, or the rhythm of speech (Labov, 1972). For future research and downstream applications, it might be helpful to consider the contextual circumstances and non-textual varieties of formality too.

Our experiments on the document level include only emails due to the lack of other corpora with formality annotations on this text level. With their composition, often including greeting, signoff, and signature, emails present a particular genre. Potentially, the greeting provides already a good indication of the formality of the text that follows (e.g., ‘Dear Mrs. Doe’ vs. ‘Hi Jane’). Although we anticipate congruent findings, future work should experiment with other types of documents, possibly more challenging to assess. Further, extending the cross-lingual experiments on the document level to languages other than English (e.g., languages with multiple forms of honorifics, such as Japanese) will be required.

Ethical Considerations

We ensured that our dataset can be made publicly available (sentences from *comments* are restricted to non-commercial use only). Since our data originates from several different domains, we gave careful consideration to finding a balance between copyright and data privacy regulations. Finally, we pseudonymized text spans containing personal information in user-generated content where necessary (*tweets*, *Reddit* posts, *comments* and *blogs*). This means we replaced sensitive text with automatically generated substitutes, e.g., female names with other female names or locations with other locations. We only release the IDs for *tweets*, *Reddit* posts and *comments*. For *blogs*, we follow the license requirements and publish the respective reference. The corpora with *emails* and *legal* texts had been pseudonymized already, no information on authors is available. For less-privacy-sensitive text sources, such as *subtitles*, *political* speeches, *news* and *Wikipedia*, we report all information shared in the original corpus, e.g., URLs. The sentences from *fiction* and *science*, which we collected ourselves, are cited appropriately in order to acknowledge intellectual property rights. People involved in creating our dataset were compensated at least following minimum wage requirements.

Acknowledgments

This work was partially funded by the Faculty of Humanities of the University of Klagenfurt. Further, we especially thank Udo Hahn for valuable input and discussions.

References

- Fadi Abu Sheikha and Diana Z. Inkpen. 2010. [Automatic classification of documents by formality](#). In *International Conference on Natural Language Processing and Knowledge Engineering*, pages 1–5. IEEE.
- Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. [FLAIR: An easy-to-use framework for state-of-the-art NLP](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.
- Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marceley Zanon Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanouel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gabbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alexander Waibel, Changhan Wang, and Shinji Watanabe. 2022. [Findings of the IWSLT 2022 evaluation campaign](#). In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98–157, Dublin, Ireland (in-person and online). Association for Computational Linguistics.
- Adrien Barbaresi and Kay-Michael Würzner. 2014. [For a fistful of blogs: Discovery and comparative benchmarking of republishable German content](#). In *Proceedings of NLP4CMC workshop (KONVENS 2014)*, pages 2–10. Hildesheim University Press.
- Andreas Blombach, Natalie Dykes, Philipp Heinrich, Besim Kabashi, and Thomas Proisl. 2020. [A corpus of German Reddit exchanges \(GeRedE\)](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6310–6316, Marseille, France. European Language Resources Association.
- Eleftheria Briakou, Sweta Agrawal, Joel Tetreault, and Marine Carpuat. 2021a. [Evaluating the evaluation metrics for style transfer: A case study in multilingual formality transfer](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1321–1336, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Eleftheria Briakou, Di Lu, Ke Zhang, and Joel Tetreault. 2021b. [Olá, bonjour, salve! XFORMAL: A benchmark for multilingual formality style transfer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3199–3216, Online. Association for Computational Linguistics.
- Julian Brooke and Graeme Hirst. 2014. [Supervised ranking of co-occurrence profiles for acquisition of continuous lexical attributes](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2172–2183, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Julian Brooke, Tong Wang, and Graeme Hirst. 2010. [Automatic acquisition of lexical formality](#). In *Coling 2010: Posters*, pages 90–98, Beijing, China. Coling 2010 Organizing Committee.
- Sven Buechel, Susanna Rücker, and Udo Hahn. 2020. [Learning and evaluating emotion lexicons for 91 languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1202–1217, Online. Association for Computational Linguistics.
- Branden Chan, Stefan Schweter, and Timo Möller. 2020. [German’s next language model](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Ana Paula Chaves, Eck Doerry, Jesse Egbert, and Marco Gerosa. 2019. [It’s how you say it: Identifying appropriate register for chatbot language design](#). In *Proceedings of the 7th International Conference on Human-Agent Interaction*, pages 102–109, Kyoto, Japan. Association for Computing Machinery.
- Yu Cheng, Zhe Gan, Yizhe Zhang, Oussama Elachqar, Dianqi Li, and Jingjing Liu. 2020. [Contextual text style transfer](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2915–2924, Online. Association for Computational Linguistics.
- Niyati Chhaya, Kushal Chawla, Tanya Goyal, Projjal Chanda, and Jaya Singh. 2018. [Frustrated, polite, or formal: Quantifying feelings and tone in email](#). In *Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media*, pages 76–86, New Orleans, Louisiana, USA. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of](#)

- deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. 2020. **Code Alltag 2.0 — a pseudonymized German-language email corpus**. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4466–4477, Marseille, France. European Language Resources Association.
- Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. 2021. **Acquiring a formality-informed lexical resource for style analysis**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2028–2041, Online. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. **Understanding back-translation at scale**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Ela Elsholz, Jon Chamberlain, and Udo Kruschwitz. 2019. **Exploring language style in chatbots to increase perceived product value and user engagement**. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, pages 301–305, Glasgow, United Kingdom. Association for Computational Machinery.
- Manaal Faruqui and Sebastian Padó. 2012. **Towards a model of formal and informal address in English**. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 623–633, Avignon, France. Association for Computational Linguistics.
- Rudolf Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.
- Alexander Geyken, Adrien Barabresi, Jörg Didakowski, Bryan Jurish, Frank Wiegand, and Lothar Lemnitzer. 2017. **Die Korpusplattform des “Digitalen Wörterbuchs der deutschen Sprache” (DWDS)**. *Zeitschrift für germanistische Linguistik*, 45(2):327–344.
- Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. **Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages**. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).
- Francis Heylighen and Jean-Marc Dewaele. 1999. **Formality of language: Definition, measurement and behavioral determinants**. Technical report, Center “Leo Apostel”, Free University of Brussels, Brussels.
- Matthew Honnibal, Ines Montani, Sofie Van Lan-deghem, and Adriane Boyd. 2020. **spaCy: Industrial-strength natural language processing in python**.
- Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. **Deep learning for text style transfer: A survey**. *Computational Linguistics*, 48(1):155–205.
- Svetlana Kiritchenko and Saif Mohammad. 2017. **Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation**. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.
- Svetlana Kiritchenko and Saif M. Mohammad. 2016. **Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling**. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 811–817, San Diego, California. Association for Computational Linguistics.
- Nikita Kitaev, Steven Cao, and Dan Klein. 2019. **Multilingual constituency parsing with self-attention and pre-training**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.
- Nikita Kitaev and Dan Klein. 2018. **Constituency parsing with a self-attentive encoder**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.
- Bryan Klimt and Yiming Yang. 2004. **The Enron corpus: A new dataset for email classification research**. In *Machine Learning: Proceedings of the 15th European Conference on Machine Learning (ECML 2004)*, 3201, pages 217–226, Pisa, Italy. Springer.
- Philipp Koehn. 2005. **Europarl: A parallel corpus for statistical machine translation**. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.
- Kalpesh Krishna, Deepak Nathani, Xavier Garcia, Bidisha Samanta, and Partha Talukdar. 2022. **Few-shot controllable style transfer for low-resource multilingual settings**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7439–7468, Dublin, Ireland. Association for Computational Linguistics.
- Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. **Reformulating unsupervised style transfer as paraphrase generation**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.

- William Labov. 1972. *Sociolinguistic Patterns*. University of Philadelphia Press, Philadelphia, Pennsylvania, USA.
- Shibamouli Lahiri. 2015. [SQINKY! A corpus of sentence-level formality, informativeness, and implicature](#). ArXiv, abs/1506.02306.
- Shibamouli Lahiri, Prasenjit Mitra, and Xiaofei Lu. 2011. [Informality judgment at sentence level and experiments with formality score](#). In *Computational Linguistics and Intelligent Text Processing. Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 446–457, Tokyo, Japan. Springer.
- Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021. [Thank you BART! rewarding pre-trained models improves formality style transfer](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 484–494, Online. Association for Computational Linguistics.
- Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. [Fine-grained named entity recognition in legal documents](#). In *Semantic Systems. The Power of AI and Knowledge Graphs (SEMANTiCS 2019)*, pages 272–287, Karlsruhe, Germany. Springer.
- Pierre Lison and Jörg Tiedemann. 2016. [OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).
- Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. 2015. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press, Cambridge, United Kingdom.
- Scott M Lundberg and Su-In Lee. 2017. [A unified approach to interpreting model predictions](#). In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS 2017)*, volume 30, pages 4765–4774. Curran Associates, Inc.
- Benjamin Minixhofer, Fabian Paischer, and Navid Rekasaz. 2022. [WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.
- Maria Nadejde, Anna Currey, Benjamin Hsu, Xing Niu, Marcello Federico, and Georgiana Dinu. 2022. [CoCoA-MT: A dataset and benchmark for contrastive controlled MT with application to formality](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 616–632, Seattle, United States. Association for Computational Linguistics.
- Xing Niu and Marine Carpuat. 2020. [Controlling neural machine translation formality with synthetic supervision](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8568–8575.
- Bryan Orme. 2009. Maxdiff analysis: Simple counting, individual-level logit, and HB. *Sawtooth Software, Inc.*
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [fairseq: A fast, extensible toolkit for sequence modeling](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ellie Pavlick and Ani Nenkova. 2015. [Inducing lexical style properties for paraphrase and genre differentiation](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 218–224, Denver, Colorado. Association for Computational Linguistics.
- Ellie Pavlick and Joel Tetreault. 2016. [An empirical analysis of formality in online communication](#). *Transactions of the Association for Computational Linguistics*, 4:61–74.
- Kelly Peterson, Matt Hohensee, and Fei Xia. 2011. [Email formality in the workplace: A case study on the Enron corpus](#). In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 86–95, Portland, Oregon. Association for Computational Linguistics.
- Sudha Rao and Joel Tetreault. 2018. [Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.
- Dariush Saberi, John Lee, and Jonathan James Webster. 2020. [Automatic assistance for academic word usage](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2163–2168, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. [One million posts: A data set of German online discussions](#). In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017*, page

1241–1244, Shinjuku, Tokyo, Japan. Association for Computing Machinery.

Tatjana Scheffler. 2014. [A German Twitter snapshot](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 2284–2289, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mingyue Shang, Piji Li, Zhenxin Fu, Lidong Bing, Dongyan Zhao, Shuming Shi, and Rui Yan. 2019. [Semi-supervised text style transfer: Cross projection in latent space](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4937–4946, Hong Kong, China. Association for Computational Linguistics.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. [Harnessing pre-trained neural networks with rules for formality style transfer](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3573–3578, Hong Kong, China. Association for Computational Linguistics.

Yi Zhang, Tao Ge, and Xu Sun. 2020. [Parallel data augmentation for formality style transfer](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3221–3228, Online. Association for Computational Linguistics.

A Appendix

A.1 Models for Formality Scoring

Fine-tuned Transformer Models. For fine-tuning transformers, we used the recommended and default parameter settings of the *FLAIR* framework (Akbik et al., 2019) (version 0.10):

- learning rate = 5.0e-5
- maximal epochs = 10
- optimizer = AdamW
- scheduler = linear scheduler with warmup
- warmup fraction = 0.1
- mini batch size = 4

Table 3 shows the results for fine-tuning different transformers on our dataset in a 10-fold cross-validation setting. We experimented with the German transformer models GBERT-base, GBERT-large, GELECTRA-base, GELECTRA-large (all from Chan et al. (2020)), and WECHSEL-RoBERTa-base-german (Minixhofer et al., 2022). The large models possess a high fluctuation in performance. Therefore, we chose the best-performing

Model	Spearman’s ρ
GBERT-base	0.919 (± 0.009)
GELECTRA-base	0.918 (± 0.011)
GBERT-large	0.109* (± 0.274)
GELECTRA-large	0.322* (± 0.426)
WECHSEL-RoBERTa-base	0.912* (± 0.009)

Table 3: Results for different transformer models on our dataset (10-fold cross-validation); significant differences (at least $p < 0.05$) are marked with ‘*’.

(and less expensive) GBERT-base model for our experiments on German data.

Table 4 displays the performances of transformer models used in a cross-dataset setting on the original data. We report results for fine-tuning regression models on our dataset and *PT16* in a 10-fold cross-validation setting. For the formality classifier we fine-tuned ourselves, the GBERT model fine-tuned on *FP12*, we achieved perfect accuracy on the original test split of this dataset.

Dataset	Model	Spearman’s ρ
ours (de)	XLM-RoBERTa	0.893 (± 0.010)
ours (en)	BERT	0.891 (± 0.010)
PT16 (de)	GBERT	0.762 (± 0.011)
PT16 (en)	XLM-RoBERTa	0.776 (± 0.016)
PT16 (en)	BERT	0.820 (± 0.010)

Table 4: Results for transformer-based regression models used in a cross-dataset setting on the original dataset (10-fold cross-validation).

Feature-based Models. For the feature-based models, we used *spaCy* (3.3) (Honnibal et al., 2020) and its language model *de_core_news_sm* for basic NLP processing routines. We utilized the *benepar* library (Kitaev and Klein, 2018; Kitaev et al., 2019) (version 0.2) for constituency parsing and scored the formality of a word given its word embedding as proposed by Eder et al. (2021). Emotional features are based on the lexicon by Buechel et al. (2020), whereas sentiment was determined with the German *TextBlob* module (0.4.3).¹⁰ We used *scikit-learn.org* (1.0.2) for the ridge regression implementation with the default parameters. We compared two sets of features adapted from Pavlick and Tetreault (2016) and Chhaya et al. (2018). In Table 5, we list the concrete features we employed per setting.

¹⁰<https://textblob-de.readthedocs.io/>

~ Pavlick and Tetreault (2016)	~ Chhaya et al. (2018)
<ul style="list-style-type: none"> • average token length • average sentence length in tokens • Flesch Reading Ease • proportion of hedge phrases • proportion of first person pronouns • proportion of third person pronouns • proportion of upper case words • proportion of lower case words • proportion of title case words • proportion of punctuation • proportion of emoticons and emojis • proportion of contractions • one-hot features for named entity types (e.g., <i>person</i>, <i>location</i>) • average word formality score • sentiment 	
<ul style="list-style-type: none"> • average sentence length in characters • one-hot features for token uni-, bi- and trigrams <ul style="list-style-type: none"> • relative frequencies of POS tags • average height of constituency trees • relative frequencies of constituency productions <ul style="list-style-type: none"> • one-hot features for combinations of dependency relation, POS tag of governor and POS tag of subordinate • GBERT embeddings 	
	<ul style="list-style-type: none"> • average word values for the emotions: valence, arousal, dominance, joy, anger, sadness, fear and disgust

Table 5: Linguistic features used for formality scoring.

Number of Parameters. Table 6 shows the number of parameters for the feature-based architectures and the transformer models.

Model	Parameters
~ Chhaya et al. (2018)	26
~ Pavlick and Tetreault (2016)	106K
GBERT-base	110M
BERT-base	110M
XLM-RoBERTa-base	125M
GELECTRA-base	110M
GBERT-large	335M
GELECTRA-large	335M
WECHSEL-RoBERTa-base	125M

Table 6: Number of parameters per model.

A.2 Annotation

We restricted the pool of crowdworkers to German native speakers from Germany, Austria, and Switzerland who were older than 18 years. No further information on the demographics of the annotators is accessible. The crowdworkers were compensated following the minimum wage defined by the German government (€ 9.60 per hour at the time of annotation). *Clickworker*, the crowdsourcing platform we used, does not provide separate qualification tests. Rather it ensures the qualification

of the crowdworkers by their own filtering methods (e.g., project-independent online tests/training or evaluation of the work results). The German annotation guidelines can be found in the project repository alongside the dataset.

A.3 Computing Details

We carried out our experiments on a NVIDIA RTX A40 GPU with 48GB RAM. We estimate a total computational budget of 72 GPU hours. Fine-tuning GBERT-base, BERT-base, or XLM-RoBERTa-base on our dataset took under 15 minutes per model. Fine-tuning these models on *PT16* required about 45 minutes per model. Fine-tuning GBERT on *FP12* took about two hours, and fine-tuning models on German or English documents needed under five minutes. Training ten ridge regression models for 10-fold cross-validation was completed in under two minutes for the feature set based on Chhaya et al. (2018) and in under 15 minutes for the feature set based on Pavlick and Tetreault (2016).

Task-specific Compression for Multi-task Language Models using Attribution-based Pruning

Nakyeong Yang¹, Yunah Jang¹, Hwanhee Lee², Seohyeong Jung³ and Kyomin Jung¹

¹Seoul National University, ²Chung-Ang University, ³Hyundai Motor Group and 42dot Inc
{yny0506, vn2209, kjung}@snu.ac.kr
hwanheelee@cau.ac.kr, seohyeong.jeong@42dot.ai

Abstract

Multi-task language models show outstanding performance for various natural language understanding tasks with only a single model. However, these language models utilize an unnecessarily large number of model parameters, even when used only for a specific task. This paper proposes a novel training-free compression method for multi-task language models using a pruning method. Specifically, we use an attribution method to determine which neurons are essential for performing a specific task. We task-specifically prune unimportant neurons and leave only task-specific parameters. Furthermore, we extend our method to be applicable in low-resource and unsupervised settings. Since our compression method is training-free, it uses few computing resources and does not destroy the pre-trained knowledge of language models. Experimental results on the six widely-used datasets show that our proposed pruning method significantly outperforms baseline pruning methods. In addition, we demonstrate that our method preserves performance even in an unseen domain setting.

1 Introduction

Various pre-trained language models with large-scale data and parameters have emerged (Devlin et al., 2018; Lewis et al., 2019; Raffel et al., 2019; Brown et al., 2020). Specifically, pre-trained language models like T5 (Raffel et al., 2019) and GPT-3 (Brown et al., 2020) have shown outstanding performance on many natural language understanding tasks. These language models can perform various tasks with a single model by treating every text processing problem as a text generation problem. However, these language models may utilize unnecessary large-scale model parameters even when performing only a specific task. Previous works have introduced various compression methods for language models such as pruning (Chen et al., 2020; Goyal et al., 2020; He et al., 2021),

knowledge distillation (Sanh et al., 2019; Hou et al., 2020; Mao et al., 2020; Sun et al., 2020), quantization (Shen et al., 2020), and low-rank factorization (Liu et al., 2021). However, these studies have (1) not compressed the language models task-specifically or (2) demanded an additional training process like the case of knowledge distillation. This additional training process requires excessive computing resources and a massive training dataset. Furthermore, this training process can destroy inherent pre-trained knowledge in language models since it updates the model’s pre-trained parameters (Toneva et al., 2018). Due to the catastrophic forgetting (McCloskey and Cohen, 1989) caused by pre-trained knowledge destruction, models which are compressed and trained for a specific task, tend to show degraded performance on solving other pre-trained tasks (Kirkpatrick et al., 2017; Ritter et al., 2018). Also, additional memory space is required to store the trained parameters separately.

In this paper, we propose a novel training-free attribution-based task-specific pruning method that enables more efficient compression and inference by extracting only task-specific parameters from multi-task language models. We can determine which neurons are essential to derive a specific output for each neural network layer by using attribution so that we can extract only task-specific parameters from the entire model, as shown in Figure 1. We can efficiently process input data while preserving the model’s task performance by selecting only the important neurons determined by the attribution method. Furthermore, we extend our method to be applicable in two challenging scenarios: low-resource and unsupervised scenarios. The former alleviates insufficient labeled data situations, and the latter handles settings when labels are unavailable. Both methods can relieve the cost of obtaining labeled datasets, which requires excessive human resources and is time-consuming. Especially under the low-resource setting, our attribution-based task-

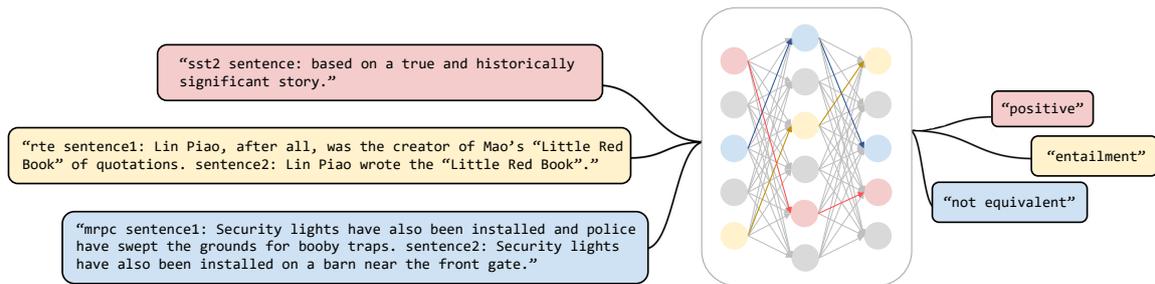


Figure 1: Task-specific Knowledge of Multi-task Language Models. Not all parameters in a language model behave as important parameters when performing a single task. For example, in this figure, when a language model receives SST-2 data, sentiment analysis data, only the parameters expressed in red color behave as essential parameters.

specific pruning requires only a single forward and backward propagation computation for few-shot data samples (e.g., only ten samples) to derive attribution of each neuron. Since this pruning process does not update the model’s parameters, it does not destroy the pre-trained knowledge of the language models. Therefore, it is irrelevant to the various disadvantages that arise during an additional training process. Since our method is model-agnostic, it can be applied to any neural network model broadly and generally. Even we can use it to extract only task-specific knowledge after other compression methods are applied.

Experimental results on the six widely-used natural language understanding tasks show that our proposed method significantly outperforms baseline training-free pruning methods. Furthermore, we demonstrate that our method shows robust performance in both low-resource and unsupervised settings. Also, we reveal that our proposed method shows outstanding knowledge preservation even for an unseen related domain, which suggests that our method can preserve task-specific knowledge effectively. We additionally investigate to offer a guideline for our task-specific compression method by analyzing which types of layers are significant for processing task-specific knowledge.

2 Related Works

2.1 Efficient Language Models

As transformer-based (Vaswani et al., 2017) language models (Devlin et al., 2018; Radford et al., 2018; Raffel et al., 2019; Liu et al., 2019; Yang et al., 2019) have become state-of-the-arts on many NLP tasks in the last few years, deep neural network model compression methods have been vastly applied to large-scale language models. Fan et al. (2019) randomly drops layers at training time, which enables structured pruning on transformer

layers at inference time. Michel et al. (2019) prunes less important attention heads at inference time. Other works (Goyal et al., 2020; Kim et al., 2021) focus on pruning less important tokens and progressively remove them during inference. However, many of the pruning methods (Goyal et al., 2020; Kim et al., 2021; Chen et al., 2020) require a following fine-tuning step of the model parameters after fixing the configuration of a pruned network, which makes such methods undesirable for efficient task-specific compression.

On the knowledge distillation side, Sun et al. (2019); Jiao et al. (2019); Sanh et al. (2019) employ teacher-student framework (Hinton et al., 2015) to transfer knowledge from an original large model (teacher), to a lightweight shallow model (student). They differ in how the student network is initialized and to which components knowledge distillation is applied. On the other hand, Shen et al. (2020) uses the mixed precision group-wise quantization based on Hessian information to compress BERT.

There are other streams of works that explore efficient language models by solving the bottleneck of the Transformer-based model computation. Beltagy et al. (2020) and Zaheer et al. (2020) sparsify the attention matrix to make transformer-based language models more efficient and Wang et al. (2020) applies low-rank approximation to increase inference speed. However, such works sparsify the full self-attention matrix according to attention score, which does not directly reduce the dimension of the matrices in the model such as query, key, value, and feed-forward matrices.

2.2 Network Pruning

One of the ways to categorize network pruning is to compare structured pruning to unstructured pruning. For structured pruning (Li et al., 2016; Hu et al., 2016; Wen et al., 2016), groups of weight con-

nections are removed from a network together, such as entire channels or filters in CNN-based networks and layers or attention heads in transformer-based networks. For unstructured pruning (Han et al., 2015a,b), weight connections are removed from a network individually. However, unstructured pruning methods produce large sparse weight matrices which are computationally inefficient unless equipped with a specifically designed hardware. In this paper, we utilize the structured pruning method to propose a compression method that enables efficient weight matrix multiplication computation.

2.3 Attribution Method

We utilize an attribution method (Shrikumar et al., 2016) to extract the importance of neurons from the pre-trained language models. Attribution methods are mostly used to derive important features (*i.g.*, *pixel*, *token*) to extract interpretability from deep neural networks (Baehrens et al., 2010; Springenberg et al., 2014; Shrikumar et al., 2016). Specifically, attribution methods are used to compute the importance of each feature for performing a specific task. Formally, suppose we have a function $\mathcal{P} : \mathbb{R}^d \rightarrow [0, 1]^m$ that represents deep neural networks for multi-class classification. The contribution of the i -th feature in x to the prediction of c -th class using \mathcal{P} is defined as follows:

$$A_i^{(x,c)}(x) = x_i \times \frac{\partial \mathcal{P}(c|x)}{\partial x_i} \quad (1)$$

where $\partial \mathcal{P}(c|x)/\partial x_i$ is the gradient of $\mathcal{P}(c|x)$ with respect to the i -th feature.

3 Methodologies

In this section, we describe our attribution-based pruning method for extracting only the task-specific knowledge from a multi-task language model T5 (Raffel et al., 2019), where attribution is obtained using gradient information. Furthermore, we extend our method to low-resource and unsupervised settings to alleviate insufficient labeled data situations. We select T5 because it is a multi-task solving model and can be used in any natural language understanding setting by treating every text processing problem as a text generation problem. For our problem setting, suppose we have input text $x = \{x_1, \dots, x_n\}$ and output text $y = \{y_1, \dots, y_m\}$ mapped as $(x, y) \in \mathcal{D}$, where each text corresponds to a sequence of tokens, and an input text contains a prefix task description. We

can represent a standard conditional language modeling objective to maximize the following likelihood:

$$\mathcal{L}(x, y) = \sum_i \log \mathcal{P}(y_i|x, y_1, \dots, y_{i-1}; \Theta) \quad (2)$$

where the conditional probability \mathcal{P} is modeled using a neural network with parameters Θ .

3.1 Task-specific Knowledge Extraction

Applying Pruning for Transformer variants

Deep neural networks can be compressed by pruning unimportant i -th neurons of the layer representation h (Han et al., 2015a,b). The architecture of Transformer-based models mainly consists of multi-head attentions and fully connected feed-forward networks as follows.

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, \dots, head_h)W^O \\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \\ FFN(x) &= \sigma(xW_1 + b_1)W_2 + b_2 \end{aligned} \quad (3)$$

where $W_i^{Q,K,V} \in \mathbb{R}^{d_{model} \times d_{q,k,v}}$ and $W_i^O \in \mathbb{R}^{d_v \times d_{model}}$ are the projection matrix parameters for multi-head attentions. For the fully connected feed-forward network (FFN), two linear transformations, denoted with the projection matrix parameters W_1 and W_2 and biases b_1 and b_2 , with an activation function are used. Transformer (Vaswani et al., 2017) variants can be compressed by pruning $W^{Q,K,V,O}$, $W_{1,2}$, and $b_{1,2}$ for each transformer block.

Deriving Attribution for Language Models

Language models generate text outputs by iteratively selecting a word-piece from the vocabulary dictionary. Therefore, the text generation process can be seen as a classification task dealt with in the attribution methods, and we can apply the attribution methods to compute the importance of features for language models. However, the purpose of this study is to derive the importance of each neuron h_i in the layer representation $h \in \mathbb{R}^d$, rather than deriving the importance for the input feature x_i . Hence, the attribution formula is adapted to compute a neuron attribution $A_i^{(x,y_j)} \in \mathbb{R}$ as follows:

$$A_i^{(x,y_j)}(h) = h_i \times \frac{\partial \mathcal{P}(y_j|x, y_{1:j-1})}{\partial h_i} \quad (4)$$

If the target output text consists of multiple word-pieces rather than a single word-piece, language

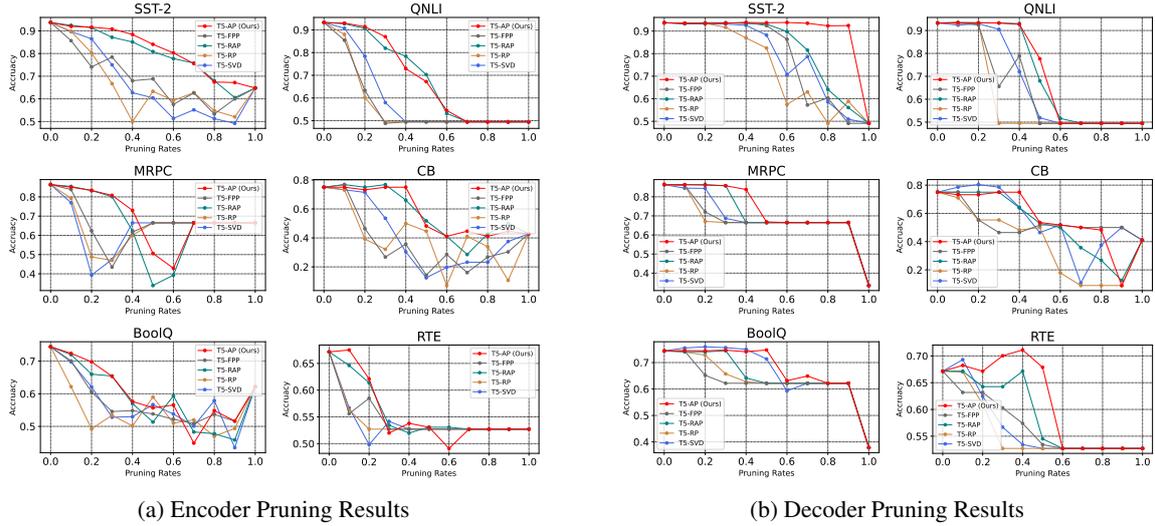


Figure 2: Module-specific Pruning Results. Our proposed attribution-based pruning significantly outperforms the other pruning methods in most cases. Especially, our task-specific pruning is more effective on decoder compression; these results suggest that most task-specific knowledge exists in the decoder of language models. The standard deviations of T5-RP and T5-RAP are shown in appendix B.

models must derive the multiple word-piece output distributions. Therefore, we change the attribution formula to handle multiple word-piece outputs as follows:

$$A_i^{(x,y)}(h) = h_i \times \sum_{j=1}^{|y|} \frac{\partial \mathcal{P}(y_j|x, y_{1:j-1})}{\partial h_i} \quad (5)$$

Since $A_i^{(x,y)}$ is attribution for one sample data x , we obtain the final neuron attribution by summing attributions for multiple sample data as shown in the following formula:

$$A_i^{(\mathcal{D})}(h) = \sum_{(x,y) \in \mathcal{D}} A_i^{(x,y)}(h) \quad (6)$$

where \mathcal{D} means the entire task-specific dataset. In low-resource environments, few-shot samples can be used for \mathcal{D} (e.g., only ten samples), which are sufficient to derive a precise importance score for each neuron. Experimental results for low-resource setting are described in section 4.3.

Attribution-based Layer Pruning We focus on applying attribution-based pruning on the Transformer encoder and decoder, more specifically on multi-head attention and fully connected feed-forward networks. We use neuron attribution $A_i^{(\mathcal{D})}$ as the importance for each neuron of a specific layer. We sort the importance of each neuron in order of magnitude at each layer, and we can compress the model by pruning neurons with lower importance.

$$\text{argsort}_i(A) = |\{j | (A_i < A_j) \cup (A_i = A_j, j < i)\}| \quad (7)$$

where $i, j \in \{1, \dots, k\}$

Once neurons are sorted according to the importance score, we prune neurons from each layer with the pruning rate p by constructing a set \mathcal{M} of neuron indices to be secured.

$$\mathcal{M} = \{i | \text{argsort}_i(A) < \lfloor k \times p \rfloor\} \quad (8)$$

where $i \in \{1, \dots, k\}$

The algorithm for deriving a set \mathcal{M} is shown in appendix A. Suppose $W \in \mathbb{R}^{d \times k}$ is a linear matrix multiplication parameter we want to prune, the matrix after pruning is denoted as $\tilde{W} = (W_{ij})_{\substack{1 \leq i \leq d \\ j \in \mathcal{M}}}$.

If the bias term $b \in \mathbb{R}^k$ is added to the operation for an affine transformation, the bias term can also be compressed by performing the $\tilde{b} = (b_i)_{i \in \mathcal{M}}$ operation similarly. The compressed parameters are used to compute the new representation by performing the transformation operation $h\tilde{W}$ or $h\tilde{W} + \tilde{b}$.

More specifically, for W_i^Q , W_i^K , and W_i^V from eq. (3), second dimension (the number of columns) of the matrix is pruned and for W_i^O , W_1 , and W_2 , the first dimension (the number of rows) is pruned to preserve the original architecture by matching shape with input processed from the previous layer. After pruning, multi-head attention and fully connected feed-forward network computations are precisely the same as before but with the pruned weight matrices:

$$\begin{aligned}
MultiHead(Q, K, V) &= Concat(head_1, \dots, head_h) \tilde{W}^O \\
head_i &= Attention(Q \tilde{W}_i^Q, K \tilde{W}_i^K, V \tilde{W}_i^V) \\
FFN(x) &= \sigma(x \tilde{W}_1 + b_1) \tilde{W}_2 + \tilde{b}_2
\end{aligned} \tag{9}$$

Note that attribution scores are sorted locally within each layer, and the pruning rate p is applied to each prunable layer uniformly.

Our proposed compression process utilizes a structured pruning without any training process. Therefore, our method can conduct on-demand real-time task-specific compression and inference for each task while preserving pre-trained parameters. The detailed algorithm for on-demand real-time task-specific compression and inference is shown in appendix A.

3.2 Unsupervised Pruning

Obtaining labeled data usually requires excessive human resources and is time-consuming. Therefore, we propose an additional method to derive attributions in an unsupervised setting to mitigate this problem. If the label for the dataset is given, we can simply compute attribution by summing the gradients values for the word-piece set composing the label. However, when the label is not given, the target word-piece set is ambiguous. To resolve this problem, we compute task-specific importance by summing the absolute values of attributions for all candidate labels as follows:

$$A_i^{(x, \mathcal{Y})}(h) = \sum_{y \in \mathcal{Y}} |h_i| \times \sum_{j=1}^{|y|} \left| \frac{\partial \mathcal{P}(y_j | x, y_{1:j-1})}{\partial h_i} \right| \tag{10}$$

where \mathcal{Y} is the candidate label set. The above importance computation formula does not require supervision for any data. Hence, we may not reflect definite label information when computing each neuron’s importance under our unsupervised compression setting. However, this setting is helpful for a resource-constrained environment, where obtaining labeled data is challenging.

4 Experiments

4.1 Experimental Setup

Datasets We conduct experiments on six downstream tasks (Wang et al., 2018, 2019). Specifically, we utilize SST-2 (sentiment analysis); MRPC

(semantic textual similarity); BoolQ (question answering); and QNLI, CB, RTE (natural language inference).

Implementation Details We select pre-trained *T5-base*¹ as a backbone for the following experiments. *T5-base* consists of 12 encoder and 12 decoder layers. Each encoder layer contains 6 prunable matrices: 4 for the multi-head self-attention networks and 2 for the feed-forward networks. Each decoder layer contains 10 prunable matrices: 4 for the multi-head self-attention networks and 2 for the feed-forward networks, and 4 for the cross-attention networks. *T5-base* used in our experiments has been fine-tuned by multi-task learning using the six datasets above. We experiment with pruning rates ranging from 0.1 to 1.0, and a pruning rate is applied to each prunable layer uniformly.

4.2 Task-specific Pruning Efficiency

In this section, we validate the effectiveness of our task-specific attribution-based pruning by comparing the performance with other pruning methods. We collect compressed models using various pruning methods and evaluate the model’s performance on testset for all six datasets.

Baselines We select four other training-free pruning methods to compare with our task-specific **T5 Attribution Pruning (T5-AP)**.

- **T5 Forward Propagation Pruning (T5-FPP)** derives the importance of each neuron with the absolute value of the forward propagation value of each neuron. This method is widely used to compress model in various studies (Han et al., 2015b; Hu et al., 2016; Li et al., 2016). Previous studies using FPP generally fine-tune the compressed model to increase the model’s performance. However, we eliminate the fine-tuning process to maintain a fair evaluation scenario since we focus on studying training-free compression.
- **T5 Low Rank Factorization (T5-SVD)** prunes weight matrices of neural networks using Singular Value Decomposition (SVD). SVD is commonly used as a main matrix compression idea in various researches (Wang et al., 2020; Noach and Goldberg, 2020). Specifically, SVD is used to compress a ma-

¹<https://huggingface.co/t5-base>

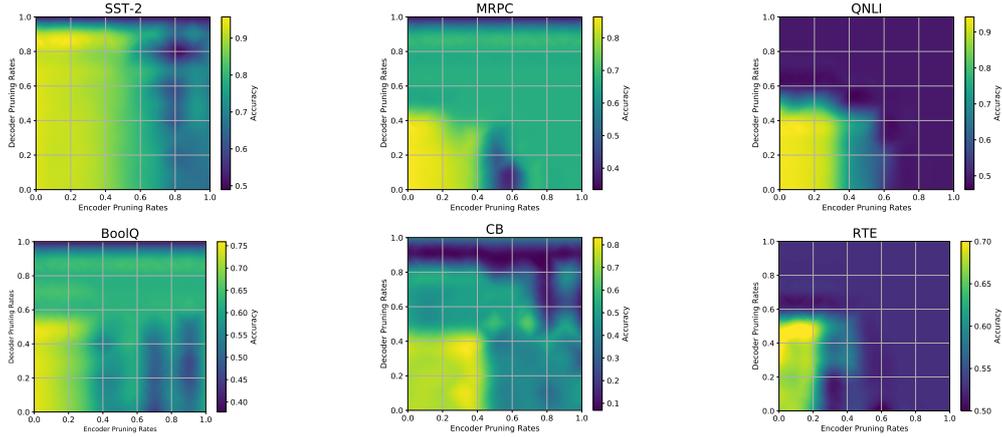


Figure 3: Module-integrated Pruning Results. These results reveal that compressing the whole architecture of the model does not additionally degrade the model’s performance compared to module-specific pruning. We experiment with the combinations of ten pruning rates for the encoder and decoder, and plot the interpolated results.

trix based on low rank factorization formula as follows:

$$W = U\Sigma V \approx \sum_{j=1}^r \sigma_j \times (U_j \times V_j) \quad (11)$$

where $W \in \mathbb{R}^{d \times k}$ is a matrix to compress, and $U \in \mathbb{R}^{d \times r}$ and $V \in \mathbb{R}^{r \times k}$ are the decomposed matrices. $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ is a diagonal matrix consisting of the singular values σ_i , where $r \leq \min(d, k)$ is the matrix rank. U_j is the j -th column of U and V_j is the j -th row of V . We can compress the matrices of T5 by determining the rank $r = \lfloor \frac{d \times k \times p}{d+k+1} \rfloor$ to have the same number of parameters as T5-AP, where p is the pruning rate defined in formula 8.

- **T5 Random Attribution Pruning (T5-RAP)** randomly selects word-pieces that are not label, and uses them to compute attribution. RAP does not derive appropriate task-specific importance for each neuron since this method randomly selects word-pieces output. We calculate the final performance of T5-RAP by averaging the accuracy derived from five trials of random word-pieces selection.
- **T5 Random Pruning (T5-RP)** randomly selects which neuron to prune. This method can achieve the lower-bound performance of overall training-free pruning methods since it randomly selects which neuron to prune without any knowledge. We calculate the final performance of T5-RP by averaging the accuracy derived from five trials of random pruning.

Module-specific Pruning For each dataset, we separately compressed the encoder and decoder at varying pruning rates to reveal the effect of our method on the encoder and decoder, respectively. Figure 2 shows the experimental results for five compression methods, including our proposed method. Experimental results show that our method outperforms other compression methods in most cases. Specifically, there is almost no performance difference between the T5-RP and T5-FPP. These results suggest that the T5-FPP does not extract task-specific knowledge. In addition, T5-SVD performs not badly in some cases, but generally performs similarly to T5-RP. It is because the low-rank approximation of T5-SVD does not work task-specifically. Surprisingly, T5-RAP sometimes performs similarly to T5-AP, probably due to the use of partial gradients information calculated from model parameters. Our experiments show that the decoder part of T5 has the robustness for task-specific compression than the encoder part of T5. These results demonstrate that T5 decoder processes more task-specific information than T5 encoder.

Module-integrated Pruning To maximize the compression efficiency of a language model, we should compress the whole model instead of compressing the encoder or decoder, respectively. Therefore, we also validate our method by compressing the whole architecture of T5. Figure 3 shows the experimental results of simultaneously compressing both the encoder and decoder using our method. These experimental results reveal that compressing the whole architecture of the model, not compressing each encoder or decoder sepa-

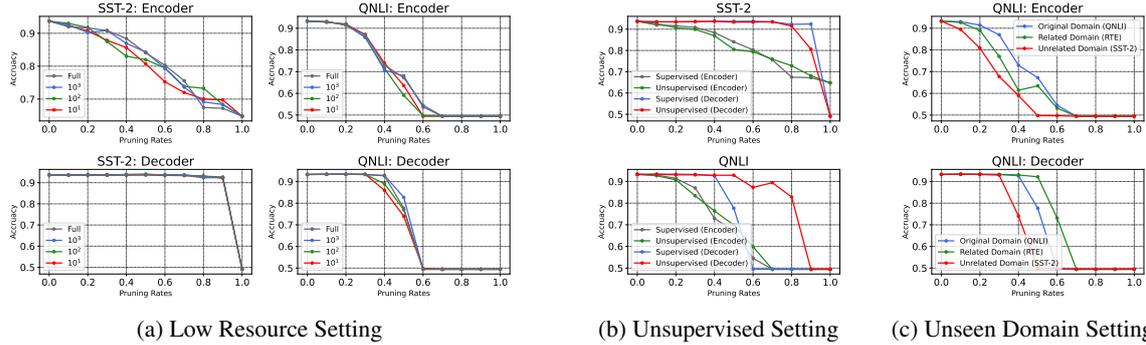


Figure 4: Experimental results extending our pruning method to challenging settings: (a) Low-resource setting experiment results. (b) Unsupervised setting experiment results. (c) Unseen domain setting experiment results. These extensions make our method more practical for use in a real-world setting.

rately, does not degrade the model’s performance additionally.

Our method focuses on compressing a multi-task language model without any additional training process in a model-agnostic way. Therefore, it is difficult to compare our method with previous compression research due to the inconsistent experimental setting since previous studies have treated training-based and model-specific compression methods. Since our method is model-agnostic, it can be utilized broadly and generally to prune multi-task language models containing only task-specific knowledge after applying other compression methods.

4.3 Low-resource Setting

In this section, we demonstrate the results for compressing language models based on the attribution computed from only few-shot. Specifically, we compute neuron importance using only 10^3 and 10^2 , and 10^1 samples of SST-2 and QNLI datasets and prune the T5 model with the computed importance, where we balance the number of samples for each class when sampling a subset of the whole dataset. All results are reported by averaging five trials of random sampling. Figure 4-(a) represents the pruning results in low-resource setting. For SST-2 dataset, we find that compression using only 10^1 data samples yields comparable performance to the results of using the entire training dataset. The total number of data samples of SST-2 is 67k, and 10^1 of data samples corresponds to about 10^{-4} of the whole dataset. For the QNLI dataset, we demonstrate that compression using only 10^3 data samples of the labeled training dataset yields comparable performance to the results of using the entire training dataset. Furthermore, the performance degradation is also insignificant when using only

10^1 samples of the labeled QNLI training dataset. The total number of data samples of QNLI is 105k, and 10^3 and 10^1 data samples correspond to about only 10^{-2} , 10^{-4} of the whole dataset, respectively. These results suggest that most of the task-specific knowledge is derived from computing gradients for only the candidate outputs. We can effectively reduce the time consumption in this low-resource setting by using a few labeled instances to compute the attribution, and it is the most significant advantage over other training-based compression methods.

4.4 Unsupervised Setting

We suggest an additional method to compute attributions using an unlabeled text dataset in section 3.2. We present the pruning results by computing attributions for an unsupervised setting in Figure 4-(b). Results of encoder compression with the unsupervised setting for both SST-2 and QNLI datasets show competitive scores to that of labeled data. For the decoder, the performance of SST-2 decreases slightly, but the performance of QNLI rather increases. The experimental result on SST-2 reveals that the compression in an unsupervised setting shows robust performance maintenance. In the QNLI result, we observe that computing attributions using information from all output candidates enhances the model’s performance.

4.5 Unseen Domain Setting

In this section, we validate the effect of our task-specific compression on unseen domains. We compress the T5 using related and unrelated datasets, and then compare the performance preservation for the original dataset. Specifically, we compress the T5 using attribution computed with SST-2 and RTE, respectively. And then, we evaluate the compressed models with the QNLI dataset. QNLI and RTE are

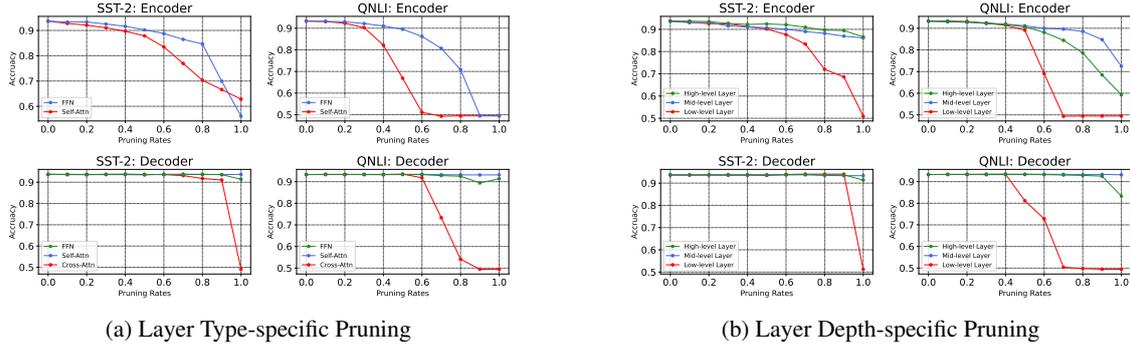


Figure 5: Layer types analysis: (a) Layer architecture experiment results. (b) Layer depth experiment results. The higher the degradation, the more essential layers are.

related domains since both are natural language inference datasets, and SST-2 is an unrelated domain built for sentiment analysis. Figure 4-(c) shows the evaluation results of the compressed model for related and unrelated domains. Experimental results reveal our method’s robust performance maintenance for the related domain. Surprisingly the case of decoder compression shows even better performance maintenance in the related domain than in the original domain.

4.6 Layer-specific Pruning Analysis

This section further investigates the pruning effect per layer type. We select two pruning settings: (1) Layer type-specific and (2) Layer depth-specific.

Layer Type-specific Pruning Analysis Layer type-specific pruning analysis focuses on understanding how the performance of the model varies depending on the type of compressed layers. The encoder investigates pruning results for feed-forward neural networks and self-attention networks, and the decoder focuses on feed-forward neural networks, self-attention networks, and cross-attention networks.

Layer Depth-specific Pruning Analysis Layer depth-specific pruning analysis investigates how the performance of the model changes depending on the depth of the compressed layers. We select SST-2 for experiments and separate each encoder and decoder into three parts: (1) Low-level layer, (2) Mid-level layer, and (3) High-level layer. Since *T5-base* consists of 12 layers for each encoder and decoder, each depth consists of 4 layers.

Layer-specific pruning results are shown in Figure 5. For the encoder, self-attention networks are more critical for preserving the performance than feed-forward neural networks. For the decoder, cross-attention networks are more important than

feed-forward neural networks and self-attention networks. For each layer-depth, we can conclude that the low-level features are more crucial to preserving the model’s performance. Especially, experimental results reveal that the model’s performance is preserved even if the pruning rate of a specific layer is 1.0. These results demonstrate that there is redundant information processing between layers for performing a specific task. Note that although the pruning rate is 1.0 for a layer, the representation propagated through the pruned layer does not lose every knowledge completely. It is because transformer variants have residual connections to preserve the knowledge of previous layers.

5 Conclusion

This paper proposes a novel training-free attribution-based task-specific knowledge extraction method for multi-task language models. Specifically, we use attribution to determine which neurons are important to derive a specific output for each task. Then, we prune task-specific unimportant neurons to extract only task-specific knowledge from the entire model. We further propose a method for computing attributions in low-resource and unsupervised settings. We demonstrate that our method outperforms the other pruning methods on the widely used text datasets. In addition, we examine that our task-specific language model pruning method shows outstanding performance in the unseen domain, especially when the unseen domain is related to the dataset used to configure the compressed version. Our compression method does not update the pre-trained parameters of the language models, which enables efficient on-demand compression and inference. Also, our proposed method is valuable because it can be universally applied to any neural network-based model architecture.

Limitations

To the best of our knowledge, this is the first work to compress a multi-task language model without extra training on the target task. Due to insufficient prior work on these training-free compression methods, we couldn't include a thorough comparison with other baseline algorithms. Also, our work focused on analyzing the results of six widely-used natural language understanding datasets among GLUE benchmark. We believe that extra experiments on various challenging natural language understanding tasks will show our work's generalization performance. We have conducted experiments on various settings; varying layer types, layer depth, low resource, unsupervised, and unseen domain. However, there are still extra room for improving this work, such as exploring and applying layer-specific pruning rates, which we leave for future work.

Acknowledgements

We thank anonymous reviewers for their constructive and insightful comments. K. Jung is with ASRI, Seoul National University, Korea. This work was supported by AIRS Company in Hyundai Motor Company & Kia Motors Corporation through HMC/KIA-SNU AI Consortium Fund. This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics & NO.2021-0-02068, Artificial Intelligence Innovation Hub (Artificial Intelligence Institute, Seoul National University) & NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)]

References

- David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang, and Jingjing Liu. 2020. Earlybert: Efficient bert training via early-bird lottery tickets. *arXiv preprint arXiv:2101.00063*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*.
- Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pages 3690–3699. PMLR.
- Song Han, Huizi Mao, and William J Dally. 2015a. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*.
- Song Han, Jeff Pool, John Tran, and William Dally. 2015b. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.
- Xuanli He, Iman Keivanloo, Yi Xu, Xiang He, Belinda Zeng, Santosh Rajagopalan, and Trishul Chilimbi. 2021. Magic pyramid: Accelerating inference with early exiting and token pruning. *arXiv preprint arXiv:2111.00230*.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).
- Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793.
- Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. 2016. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.
- Sehoon Kim, Sheng Shen, David Thorsley, Amir Ghلامي, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. 2021. Learned token pruning for transformers. *arXiv preprint arXiv:2107.00910*.

- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yuanxin Liu, Zheng Lin, and Fengcheng Yuan. 2021. **ROSITA: refined BERT compression with integrated techniques**. *CoRR*, abs/2103.11367.
- Yihuan Mao, Yujing Wang, Chufan Wu, Chen Zhang, Yang Wang, Yaming Yang, Quanlu Zhang, Yunhai Tong, and Jing Bai. 2020. **Ladabert: Lightweight adaptation of BERT through hybrid model compression**. *CoRR*, abs/2004.04124.
- Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.
- Matan Ben Noach and Yoav Goldberg. 2020. Compressing pre-trained language models by matrix decomposition. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 884–889.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8815–8821.
- Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2016. Not just a black box: Interpretable deep learning by propagating activation differences. *arXiv preprint arXiv:1605.01713*, 4.
- Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.
- Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.
- Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2018. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.
- Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.

A Algorithms

Our pruning method consists of two stages: (1) Derivation of task-specific neuron indices per layer for a specific task (2) Real-time task-specific inference with previously pruned layers.

Algorithm 1 Deriving task-specific neuron indices per layer for a task t

Input: task-specific dataset \mathcal{D}^t ; model \mathcal{P} ; pruning rate p
Output: list \mathcal{M}^t with task-specific neuron indices per layer

- 1: Initialize all \mathcal{M}_i^t as an empty set and all $A_i^{(\mathcal{D}^t)}$ to zero
- 2: $\mathcal{B} \leftarrow$ split \mathcal{D}^t into mini-batches of size β
- 3: **for** each batch $b \in \mathcal{B}$ **do**
- 4: **for** each layer $l \in \mathcal{P}$ **do**
- 5: **for** $i = 1$ to k^l **do**
- 6: compute neuron importance $A_i^{(b)}(h^l)$
- 7: $A_i^{(\mathcal{D}^t)}(h^l) \leftarrow A_i^{(\mathcal{D}^t)}(h^l) + A_i^{(b)}(h^l)$
- 8: **for** each layer $l \in \mathcal{P}$ **do**
- 9: **for** $i = 1$ to k^l **do**
- 10: **if** $\text{argsort}_i(A^{(\mathcal{D}^t)}(h^l)) < \lfloor k^l \times p \rfloor$ **then**
- 11: $\mathcal{M}_i^t \leftarrow \mathcal{M}_i^t \cup \{i\}$

return \mathcal{M}^t

In the first stage, we sort neuron indices in descending order by computed attribution scores, leaving high-importance neurons by $(1 - p)$ ratio.

Algorithm 2 Real-time task-specific inference with pruned layers

Input: task t ; text inputs x ; indices container \mathcal{M} ; model \mathcal{P}
Output: text outputs y

- 1: For task t , load corresponding \mathcal{M}^t
- 2: **for** each layer $l \in \mathcal{P}$ **do**
- 3: $W^l \leftarrow (W_{ij}^l)_{\substack{i \in \mathcal{M}_i^t \\ j \in \mathcal{M}_i^t}}$ ▷ match rows with a previous layer l'
- 4: **if** bias b^l exists in layer l **then**
- 5: $b^l \leftarrow (b_i^l)_{i \in \mathcal{M}_i^t}$
- 6: compute outputs y with x using the pruned model $\tilde{\mathcal{P}}$

return y

In the second stage, we prune task-specifically unimportant neurons when given a user request for a specific task. We task-specifically compress a model in real-time and conduct an inference with the pruned model.

B Statistic of Pruning Results

We compute the pruning results of the baselines of T5-RP and T5-RAP through five random trials. The standard deviations of the accuracy for the two baselines are shown in Table 1.

		SST-2	MRPC	QNLI	RTE	CB	BoolQ
T5-RP	Encoder	0.0202	0.0046	0.0143	0.0426	0.0168	0.0020
	Decoder	0.0241	0.0108	0.0003	0.0580	0.0010	0.0060
T5-RAP	Encoder	0.0082	0.0080	0.0119	0.0165	0.0099	0.0061
	Decoder	0.0159	0.0089	0.0029	0.0123	0.0027	0.0063

Table 1: Standard deviations of Pruning results.

We calculate the standard deviations by averaging the values derived by all pruning rates. These results reveal that the variances of T5-RP and T5-RAP are not significant.

Zero-shot Transfer of Article-aware Legal Outcome Classification for European Court of Human Rights Cases

Santosh T.Y.S.S¹, Oana Ichim², Matthias Grabmair¹

¹School of Computation, Information, and Technology;
Technical University of Munich, Germany

²Graduate Institute of International and Development Studies, Geneva, Switzerland
{santosh.tokala, matthias.grabmair}@tum.de
oana.ichim@graduateinstitute.ch

Abstract

In this paper, we cast Legal Judgment Prediction on European Court of Human Rights cases into an article-aware classification task, where the case outcome is classified from a combined input of case facts and convention articles. This configuration facilitates the model learning some legal reasoning ability in mapping article text to specific case fact text. It also provides an opportunity to evaluate the model's ability to generalize to zero-shot settings when asked to classify the case outcome with respect to articles not seen during training. We devise zero-shot experiments and apply domain adaptation methods based on domain discrimination and Wasserstein distance. Our results demonstrate that the article-aware architecture outperforms straightforward fact classification. We also find that domain adaptation methods improve zero-shot transfer performance, with article relatedness and encoder pre-training influencing the effect.

1 Introduction

Legal Judgment Prediction (LJP) has recently gained considerable attention in the mainstream NLP community (e.g., Aletras et al. 2016; Chalkidis et al. 2019, 2021, 2022b; Santosh et al. 2022, 2023). In LJP, the outcome of a case should be classified/predicted based on a textual description of case facts. In actual legal reasoning, legal practitioners (e.g., advocates, judges) determine relevant rules from the sources of law (e.g., statutes, regulations, precedent) that are relevant to the case at hand. They then carry out an analysis to determine which rules apply to the case at hand, and deduce the outcome of the case by applying them. Subsuming case facts under elements of rules given in legal sources plays a critical role in this process. Many current LJP approaches (e.g., Aletras et al. 2016; Chalkidis et al. 2019, 2022b; Santosh et al. 2023) tackle this as a straightforward classification problem with the textual descriptions of case fact as

the sole input. This reliance on the model learning statistical correspondences from case fact descriptions directly to outcomes neglects the role of legal sources in this relationship. As a consequence, the model may learn sub-optimal fact-outcome patterns that are informed by the case distribution in the data rather than learning to align facts with the legal source text containing applicable rules. The models may also attend to outcome-correlating distractors present in the dataset rather than engage in the legal fact-vs-law reasoning that is required of legal practitioners for a proper justification of the outcome (Santosh et al., 2022).

This work seeks to remedy this incomplete inference and enable the model to learn more authentic reasoning between rules and case facts by casting LJP into an article-aware classification setting and subjecting it to a zero-shot transfer challenge. Article-aware classification has been explored on Chinese criminal case corpora (Wang et al., 2018, 2019b; Yue et al., 2021; Chen et al., 2022). Similarly, Holzenberger et al. 2020 has modeled statutory reasoning by classifying US tax law provisions concatenated with textual case descriptions. We build on this prior work in two ways. First, we develop and evaluate our model on a public dataset (Chalkidis et al., 2022b) of cases by the European Court of Human Rights (ECtHR), which hears complaints by individuals about possible infringements of their rights enshrined in the European Convention on Human Rights (ECHR) by states. To the best of our knowledge, this is the first work applying article-aware case outcome prediction setting to human rights adjudication. Our approach pairs case fact descriptions with candidate ECHR articles and assigns a binary target label depending on whether the article has been alleged/deemed to have been violated, or not. Our results show that the article-aware classification model outperforms the traditional classification setup by a small but consistent margin.

Second, we subject the model to a zero-shot transfer task. Models trained on case facts alone cannot produce inferences about convention articles they did not observe during training. By contrast, human judges can conduct outcome analysis with new/amended legal provisions because they are trained to understand the rules they contain and apply them to case facts in an expertise-informed way, even in the absence of secondary sources (e.g., commentaries to the rule, etc.). Article-aware classification allows an emulation of this process by means of a zero-shot benchmarking task on articles unseen at training time. We compare two conditions where (1) the model either has no access to the target articles, or (2) it is allowed to ‘read’ the target articles but is not given any prediction outcome labels for case-target article pairs.

We experiment with domain adaptation by means of a domain discriminator (Ganin et al., 2016) and Wasserstein distance (Shen et al., 2018). Our results show that this improves performance on unseen articles compared to a vanilla model. We study the impact of law-specific pre-trained encoders on this zero-shot transferability compared to the standard language pre-trained one. Intuitively, we observe that our models perform better in zero-shot transfer if the target/unseen articles are semantically related to articles seen at training time.

It should be noted that, despite these tasks being typically referred to as instances of ‘legal judgment prediction’, ECtHR fact statements are typically not finalized until the decision outcome is known, making the task effectively one of retrospective classification rather than prediction (Medvedeva et al., 2021). While this does lead to distracting and confounding phenomena (see our prior work in Santosh et al. 2022), the dataset remains a useful resource for the development of NLP models that analyze these fact statements for text patterns that correspond to specific convention articles as drafted by the court. Consequently, in this paper we hence speak of our models as engaging in *case outcome classification (COC)*.

Our main contributions in this paper are¹:

- We cast LJP/COC on ECtHR cases as an article-aware classification task by pairing case fact descriptions with candidate articles. Assuming a frozen pre-trained encoder network, our article-aware prediction model out-

performs straightforward fact classification.

- We conduct zero-shot transfer benchmarking of article-aware COC models. We find this to be a difficult testing task for the generalization of COC models. We show that domain adaptation using a domain discriminator and a Wasserstein distance method improves generalization.
- We conduct auxiliary experiments validating that article relatedness positively affects transfer performance and show an interaction between domain adaptation and domain specific encoder pre-training.

2 Related Work

Legal Judgement Prediction: LJP/COC as an NLP task has been studied using corpora from different jurisdictions, such as the ECtHR (Chalkidis et al., 2019, 2021, 2022b; Aletras et al., 2016; Liu and Chen, 2017; Medvedeva et al., 2020; SAYS, 2020; Medvedeva et al., 2021; Santosh et al., 2023) Chinese Criminal Courts (Luo et al., 2017; Zhong et al., 2018; Yang et al., 2019; Yue et al., 2021; Zhong et al., 2020), US Supreme Court (Katz et al., 2017; Kaufman et al., 2019), Indian Supreme Court (Malik et al., 2021; Shaikh et al., 2020) the French court of Cassation (Şulea et al., 2017b,a), Brazilian courts (Lage-Freitas et al., 2022), the Federal Supreme Court of Switzerland (Niklaus et al., 2021), UK courts (Strickson and De La Iglesia, 2020) and German courts (Walzl et al., 2017)

Early works (Aletras et al., 2016; Şulea et al., 2017a,b; Virtucio et al., 2018; Shaikh et al., 2020; Medvedeva et al., 2020) used bag-of-words features. More recent approaches use deep learning (Zhong et al., 2018, 2020; Yang et al., 2019). Large pre-trained transformer models have since become the dominant model family in COC/LJP (Chalkidis et al., 2019; Niklaus et al., 2021), including legal-domain specific pre-trained variants (Chalkidis et al., 2020; Zheng et al., 2021) that have been employed for the benchmark ECtHR corpus we use in this paper (Chalkidis et al. 2021, 2022b).

Prior work on Chinese criminal case corpora case extends fact-based classification by providing the text of legal source articles as additional input. Luo et al. 2017 used an attention-based neural network which jointly models charge prediction and relevant article extraction in a unified framework whose input includes the text of legal articles. Sim-

¹Our code is available at <https://github.com/TUMLegalTech/zeroshotLJP>

ilarly, Wang et al. 2018, 2019b; Chen et al. 2022; Yue et al. 2021 employ matching mechanism between case facts and article texts. To the best of our knowledge, ours is the first work to adapt article-aware prediction to the ECtHR corpus, which is situated in the in human rights litigation domain. Going beyond previous works, we further benchmark the zero-shot transfer performance of such models, providing a test bed to evaluate their capability to process article texts they have not seen during training time and applying them to case facts towards classifying allegations/outcomes.

Domain Adaptation (DA): In transfer learning, the field of domain adaptation (DA) addresses the covariate shift between source and target data distributions (Ruder, 2019). It is tackled under three different settings: (1) Semi-supervised DA (Bollegala et al., 2011; Daume III and Marcu, 2006) where labels for the source and a small set of labels for the target domain are available, (2) unsupervised DA (Ganin et al., 2016; Blitzer et al., 2006) where only labels for the source domain and unlabelled target data are given, and (3) Any Domain Adaptation / Out of Distribution generalization (Ben-David et al., 2022; Volk et al., 2022) where only labeled source data is given. In this work, we distill the existing public LexGLUE ECtHR dataset into a new benchmark on more challenging unsupervised and any domain adaptation settings for COC to emulate legal reasoning involving previously unseen convention articles.

DA variants have been benchmarked for various NLP tasks, such as Question answering (Yu et al., 2018), duplicate question detection (Shah et al., 2018), sentiment analysis (Li et al., 2017; Ganin et al., 2016), dependency parsing (Sato et al., 2017), relation extraction (Wu et al., 2017), POS tagging (Yasunaga et al., 2018), named entity recognition (Jia et al., 2019), event trigger identification (Naik and Rose, 2020), machine reading comprehension (Wang et al., 2019a), and machine translation (Yang et al., 2018). To the best of our knowledge, this work is the first to benchmark domain adaptation in COC/LJP. While previous works typically involve short text, COC on ECtHR data involves case facts and articles, both of which typically are long documents.

Methods proposed for domain adaptation can be categorized into four types: (a) Instance-based data selection methods (Jiang and Zhai, 2007; Remus, 2012) which employ similarity metrics to

sample source data points to match the distribution of the target domain and train models based on obtained subsamples from the source domain, (b) Pseudo-labeling approaches (Ruder and Plank, 2018; Rotman and Reichart, 2019) which train a classifier based on source data initially and use it to predict labels on unlabeled target data towards further adapting the model, (c) Pivot-based methods (Blitzer et al., 2006; Ziser and Reichart, 2017) which aim to map different domains to a common latent space (where the feature distributions are close) by employing auto encoders and structural correspondence learning, and (d) Loss-based methods (Ganin and Lempitsky, 2015; Shen et al., 2018) which employ domain adversaries aiming to minimize the discrepancies between source and target data distributions. In this work, we employ loss-based approaches using a domain discriminator (Ganin et al., 2016) and Wasserstein distance (Shen et al., 2018) to enable domain adaptation for our COC models.

3 Dataset, Tasks & Settings

We use the LexGLUE ECtHR dataset provided by (Chalkidis et al., 2022b), which consists of 11k case fact descriptions along with target label information about which convention articles have been alleged to be violated (task B), and which the court has eventually found to have been violated (task A). The dataset is chronologically split into training (2001–2016), validation (2016–2017), and test set (2017–2019) with 9k, 1k, and 1k cases, respectively. The label set includes 10 prominent ECHR articles, which forms a subset of all the rights contained in the convention and its protocols. In both the ECtHR A and B benchmarks, it is assumed that the model classified the target from the fact description alone, which we refer to as the **fact classification variant**.

For our article-aware classification settings, we augment the dataset with the texts of the 10 articles copied from the publicly available ECHR convention document². We formulate the **article-aware prediction variant** for both tasks: Given both the case fact statements and a particular article information, the model should classify the binary outcome of whether an article has been alleged to be violated by the claimant (task B) or found to have been violated by the court (task A).

²https://www.echr.coe.int/documents/convention_eng.pdf

Our zero-shot transfer task then involves determining violation/allegation from case facts with respect to articles which are not seen during training time. We consider a ‘domain’ to be a particular convention article (i.e., 10 convention articles form 10 domains). The objective is to train a model on a source domain (seen articles) with the goal of performing well at test-time on a target domain (unseen articles). Following (Yin et al., 2019; Ramponi and Plank, 2020), we propose two settings under zero-shot COC:

Zero-Shot Restrictive / Unsupervised Domain Adaptation (UDA): In this setting, the model is given a pair of case facts and the text of training set articles (i.e., the source domain) along with their corresponding violation/allegation outcome label. In the target domain, it is provided with case facts and article text pairs as well, but the outcome label is withheld. The goal of UDA is to learn an outcome classifier from the outcome labelled source domain which should generalize well on the target domain by leveraging outcome-unlabelled target data. This setting is legally realistic, as the text of new or modified written legal sources is typically known for a given task and available for domain adaptation (e.g., a public administration decision support tool receives an update after relevant legislation has changed).

Zero-shot Wild / Any Domain Adaptation (ADA) / Out of Distribution Generalization: In this setting, the model never sees any article data from the target domain during training, yet should be able to generalize to it. In the legal setting, this corresponds to a model which is required to work with texts of sources only available at query time (e.g., complex retrieval settings where multiple legal sources potentially apply).

We reorganize the dataset to evaluate our zero-shot transfer/adaptation models by splitting the 10 ECHR articles into two non-overlapping groups, such that both contain articles of various frequencies (common, moderate, rare).

- *split_0*: 6, 8, P1-1, 2, 9
- *split_1*: 3, 5, 10, 14, 11

We evaluate UDA and ADA on *split_0* as source and *split_1* as target, and vice-versa.

4 Method

We employ a hierarchical neural model which takes the case fact description x along with the article

a as input and outputs a binary outcome (allegation in Task B and violation in Task A) for case x with respect to article a . Our architecture is a modified version of the Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017) incorporating conditional encoding (Augenstein et al., 2016; Rocktäschel et al., 2016) that has been adapted to deal with long input sequences following hierarchical attention networks (Yang et al., 2016). We experiment with two domain adaptation components based on adversarial training: (1) a classification-based domain discriminator and (2) a Wasserstein-distance based method which aims to reduce the difference between the source and the target domain distributions.

4.1 Article-aware prediction Model

Given the facts of the case $x = \{x_1, x_2, \dots, x_m\}$ where $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ and the article $a = \{a_1, a_2, \dots, a_k\}$ where $a_j = \{a_{j1}, a_{j2}, \dots, a_{jl}\}$, the model outputs a binary label. x_i / a_i and x_{jp} / a_{jp} denote the i^{th} sentence and p^{th} token of the j^{th} sentence of the case facts / article, respectively. m/k and n/l denote the number of sentences and tokens in the i^{th} sentence of case facts / article, respectively. Our model contains an encoding layer, followed by an interaction layer, a post-interaction encoding layer, and a classification header. See Fig. 1 for an overview of our architecture.

4.1.1 Pre-interaction Encoding Layer

Our model encodes the facts of the case x sentence-wise with LegalBERT (Chalkidis et al., 2020) to obtain token level representations $\{z_{i1}, z_{i2}, \dots, z_{in}\}$. These are aggregated into sentence level representations using token attention:

$$u_{it} = \tanh(W_w z_{it} + b_w) \quad (1)$$

$$\alpha_{it} = \frac{\exp(u_{it} u_w)}{\sum_t \exp(u_{it} u_w)} \quad \& \quad f_i = \sum_{t=1}^n \alpha_{it} z_{it} \quad (2)$$

where W_w, b_w and u_w are trainable parameters. The sentence level representations $\{f_1, \dots, f_n\}$ are passed through a GRU encoder to obtain context-aware sentence representations of the facts $h = \{h_1, h_2, \dots, h_m\}$. The analogous article encoder takes a as input and outputs $s = \{s_1, s_2, \dots, s_k\}$.

4.1.2 Interaction Layer

Interaction between the sentences of the case facts and articles is done via dot product attention between the two sequences of sentences as follows:

$$e_{ij} = h_i^T s_j \ \& \ h'_i = \sum_{j=1}^k \frac{\exp(e_{ij})}{\sum_{l=1}^k \exp(e_{il})} s_j \quad (3)$$

$$s'_j = \sum_{i=1}^m \frac{\exp(e_{ij})}{\sum_{l=1}^m \exp(e_{lj})} h_i \quad (4)$$

where e_{ij} represents the dot product interaction score between the context-aware representations of the i^{th} sentence of the case facts and the j^{th} sentence of the article. h'_i and s'_j represent article-aware representations corresponding to the i^{th} sentence of the case facts and the fact-aware representation corresponding to the j^{th} sentence of the article, respectively. Finally, we obtain interaction-aware sentence representations of the facts $h' = \{h'_1, h'_2, \dots, h'_m\}$. Similarly for the article, we obtain $s' = \{s'_1, s'_2, \dots, s'_k\}$

4.1.3 Post-Interaction Encoding Layer

The article-dependent final representation of the case facts is obtained in two steps: (i) we compute the final representation of the article text and (ii) use it as a conditional encoding (Augenstein et al., 2016; Rocktäschel et al., 2016) to obtain the final article-dependent fact representation.

Final representation of article: We first combine the pre-interaction sentence encodings and fact-aware sentence representations of the article:

$$p_i = [s_i, s'_i, s_i - s'_i, s_i \odot s'_i] \quad (5)$$

where \odot denotes element-wise product. This representation aims to capture high-order interaction between the pre- and post- interaction elements (Chen et al., 2017). The sentence representations p_i are passed over a non-linear projection and a GRU (as in the pre-interaction encoder) to perform context-level modelling among sentence sequences. The final article representation A is obtained via sentence attention analogous to eq. 2.

Final Representation of Case Facts: Similarly, we pass the combined representation of case facts using pre- and post- interaction similar to Eq. 5 over a non linear projection, a GRU layer, and sentence level attention to the obtain article-dependent final representation of case facts. To ensure conditioning, we initialize the GRU hidden state with the final representation of the articles A . This facilitates capturing the salient case fact information with respect to the specified article.

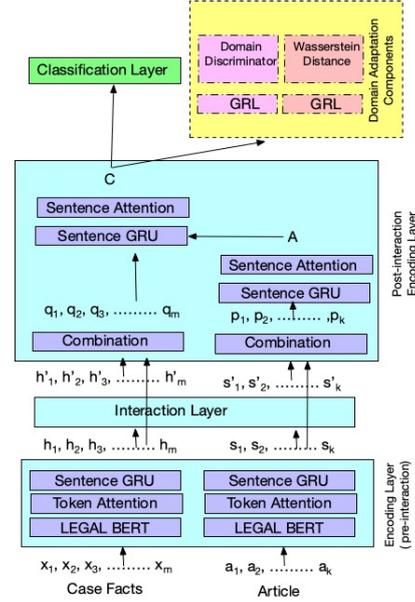


Figure 1: Our article-aware prediction model architecture

4.1.4 Classification Layer

We pass the article-dependent final representation of the case facts through a nonlinear projection to classify the outcome.

4.2 Domain Adaptation Components

Domain Adaptation aims to make models generalize well from a source to a target domain. Both domains are mapped to a common latent space, reducing differences between their distributions and facilitating domain invariant feature representations. In our case of article-aware COC, we regard reasoning with respect to every ECHR article as a domain and seek to learn article-invariant case facts representations. Put differently, we want our model to learn how to read two texts and interrelate them towards an outcome determination (as lawyers do) with minimal encoding of the information contained in the texts into the model itself. This way, the models can achieve generalization capability to adapt and perform reasoning with regard to articles not seen during training time.

4.2.1 Domain Discriminator

We employ a two layer feed forward network as a discriminator which takes the article-dependent case fact representation as input to predict the article (i.e., the domain). We train the discriminator in an adversarial fashion to maximize the model’s ability to capture information required for the outcome task while minimizing its ability to

predict the article. This guides the model to generate article-invariant feature representations and improves transferability. Following (Ganin and Lempitsky, 2015; Ganin et al., 2016), we perform a min-max game adversary objective optimization using a gradient reversal layer (GRL) between the feature extractor and discriminator. It acts as the identity during the forward pass but, during the backward pass, scales the gradients flowing through by $-\lambda$, making the feature extractor receive the opposite gradients from the discriminator. The overall objective function reduces to:

$$\arg \min_{\theta_F, \theta_C, \theta_D} [L_c(C(F(x, a)), y_e) + \lambda L_d(D(\text{GRL}(F(x, a))), y_a)] \quad (6)$$

where L_c, L_d represents the loss function corresponding to classifier and domain discriminator, respectively, λ is the GRL hyperparameter, x is the input, y_e is the outcome label, y_a is the class-id of the article, F, C and D represents feature extractor, classifier, and discriminator with parameters θ_F, θ_C and θ_D , respectively. In case of UDA (where the model has access to the text of target domain articles), we discriminate among all the source and target articles. While in case of ADA, we discriminate among source articles only.

4.2.2 Wasserstein Method (Distance based)

Our second method aims to reduce the Wasserstein distance (Shen et al., 2018) between different domain feature distributions. In a given batch, the final feature representations will be fed into the domain critic (Arjovsky et al., 2017), which is a feed-forward network whose output is a single scalar for each batch element. These scalars are then averaged per domain in the batch, resulting in two numbers representing source and target domains, respectively. Their difference can be considered an approximation of the Wasserstein distance between the two feature distributions and becomes the Wasserstein loss component of the network. If the domain critic neural network satisfies the constraint of the Lipschitz-1 continuous function, we calculate the approximate empirical Wasserstein distance by maximizing the following domain critic loss:

$$L(X_p, X_q) = \frac{1}{n_p} \sum_{x_p \in X_p} f_w(F(x_p)) - \frac{1}{n_q} \sum_{x_q \in X_q} f_w(F(x_q)) \quad (7)$$

where f_w, F denote the Wasserstein domain critic and feature extractor, respectively, X_p and X_q de-

Table 1: Fact Classification vs Article-aware prediction Performance on Task A and Task B. mic. and mac. indicates micro-F1 and macro-F1 scores, respectively.

Model	Task B		Task A	
	mac.	mic.	mac.	mic.
Fact Classification	71.96	77.40	61.21	72.21
Article-aware pred.	74.14	78.49	67.09	74.77

note datasets from two domains p and q with n_p and n_q samples, respectively.

During optimization, a gradient reversal layer (Ganin et al., 2016) between the feature extractor and domain critic ensures that (a) the domain critic weights are updated such that the Wasserstein loss becomes maximal, while the encoder weights are updated towards minimizing it. Through this procedure, we encourage the model to learn feature representations that are invariant to the covariate shift between the source and the target domain. Since the Wasserstein distance is continuous and differentiable everywhere, we can train the domain critic end-to-end. In case of UDA, we minimize the distance between the source and the target domains, while in case of ADA, we minimize among the different source domains. To enforce the Lipschitz constraints, we clip the weights of the domain critic within a compact space $[-c, c]$ after each gradient update following (Arjovsky et al., 2017).

5 Experiments & Discussion

5.1 Baseline

For the **fact classification variant**, we employ an architecture similar to the article-aware prediction model but reduced to the case fact based encoding without the interaction mechanism. The output layer is modified to 10 classes and trained against a multi-hot target vector using a binary cross entropy loss. Notably, we freeze the weights in the LegalBERT sentence encoder, both to save computational resources and to reduce the model’s susceptibility to shallow surface signals and ensure the comparability of our domain adaptation methods. We describe the detailed hyperparameters for the article-aware prediction model in Appendix Sec. A

5.2 Does Article-aware Classification Perform Better than Fact-only Classification?

Micro-F1 and macro-F1 scores for both tasks A and B with regard to the 10 target articles are given in Table 1. The article-aware model performs better than fact-only classification across the board. In

Table 2: Task B F1 performance of baseline and domain adaptation models

Setting	Model	Transfer 0 → 1				Transfer 0 ← 1			
		source : <i>split_0</i>		target : <i>split_1</i>		source : <i>split_1</i>		target : <i>split_0</i>	
		mac.	mic.	mac.	mic.	mac.	mic.	mac.	mic.
Baseline	Source only	73.45	75.63	7.32	7.37	70.26	77.10	8.49	9.08
UDA	Domain Disc.	73.81	76.95	13.92	14.94	70.63	77.43	22.50	26.27
	Wasserstein	69.63	74.86	13.17	18.16	66.89	75.21	20.78	30.30
ADA	Domain Disc.	73.76	76.13	9.62	10.77	69.71	76.85	9.30	10.45
	Wasserstein	70.17	74.80	9.14	9.89	67.46	75.25	9.26	10.38

particular, we notice a greater improvement in the macro-F1 score, indicating the article-aware classification approach helps the model to improve performance for sparser articles which are not prominently represented in the case distribution. We conjecture that this performance difference can be explained with article-aware classification being subjected to a different training regime. In fact-only classification, a given case’s fact text will always be associated with the same multi-hot outcome vector. By contrast, in the fact-aware setting, it will occur multiple times alongside different article texts and the model is forced to predict a single binary outcome variable. This seems to lead the model away from shallow signals towards capturing fact-article correspondence, resulting in a better model. Additionally, the beneficial effect is greater for the harder task of violation classification.

5.3 Does Domain Adaptation Help to Improve Zero Shot Transferability ?

We evaluate UDA and ADA on both Task A and Task B with the two article splits. A baseline *source only* model is trained without domain adaptation using the labelled source data only and tested on the target test data directly. Tables 2 and 3 show the performance of different models with our two splits on task B and A, respectively.

Baseline vs Domain Adaptation: From both tables, we observe that the performance of the *source only* model on target data is lower compared to their domain adaptation counterparts with a significant margin. This indicates that, intuitively, models trained on source data without any adaptation do not generalize to unseen articles. This also highlights the need to have domain adaptation components to achieve a generalizable model.

UDA: Under unsupervised domain adaptation, we observe that the Wasserstein distance method performs better on target data than the Domain Discriminator in micro-F1 by a significant margin. It also improved macro-F1 marginally in Task A tar-

get data, but is inferior in Task B. Most strikingly, however, Wasserstein performance on source data is lower than the source only baseline across the board, especially with respect to macro-F1. These observations also indicate that the Wasserstein distance method is able to transfer well to certain articles more than others. This can be attributed to the method influencing feature representations towards a reduction of the mean difference across articles. The distribution of target articles which are closer to the source articles distributions might have gained well. We further validate this hypothesis using an experiment illustrated in sec 5.5. On source data, the Domain Discriminator performed better than the source only model, albeit by very small margins but consistent across the tables.

ADA: On target data, both the Domain Discriminator and Wasserstein distance are comparable across the tables in both metrics. With respect to source data, in task B, the Domain Discriminator performed better than the Wasserstein distance method in both micro and macro F1. Strikingly, in Task A, Wasserstein performance on source data picks up in micro-F1 (slightly even better than source only baseline) but stays behind in macro-F1.

ADA vs UDA: Unsurprisingly, the performance on target data under ADA tends to be lower compared to UDA due to no access to target article information in this setting compared to UDA.

The absolute performance levels on the target data immediately suggest that the zero-shot transfer task we propose is very difficult and the discrepancy of performance between source and target data is still large, even in the case of domain adaptation components. This indicates ample opportunity for further research on neural models capable of reasoning with legal text in a way that transfers well to unseen legal domains. Some of the source-target performance divergence can likely be attributed to the model falling prey to spurious correlations that exist in the data, which is especially prominent in the ECtHR datasets that suffer from fact statements

not being finalized until the case outcome is known (see our prior work in Santosh et al. 2022). Given this limitation, our zero-shot framework serves as a challenging benchmark in the development of legal NLP models that learn to interrelate case facts and legal source text towards supporting domain experts.

5.4 How does Encoder Pre-Training influence Zero-Shot Transferability?

We conduct an additional experiment on Task A with *split_1* as source and *split_0* as target, where we replace LegalBERT embeddings used in the encoding layer with BERT base embeddings (Kenton and Toutanova, 2019), and report its performance in Table 4. Comparing it to Transfer 0 \leftarrow 1 in Table 3, we observe that the BERT base model performs worse on target data than the LegalBERT encoder. In particular, the best performing Wasserstein domain adaptation model drops from 26.2 to 16.36, much more than the Domain Discriminator. We leave an exploration of this asymmetric effect of the pre-training regime across different domain adaptation strategies to future work.

Base BERT performs similarly on the source domain. This indicates that even a non-legally pre-trained encoder can be harnessed to reach comparable in-domain performance. However, to generalize to unseen target articles, domain specific pre-training is beneficial. It should be noted that LegalBERT (Chalkidis et al., 2020) has been pre-trained on a collection of ECtHR decisions that may include cases from LexGLUE’s test partition, thereby possibly injecting domain-specific information about the target articles into the encoding.

5.5 How does Article Relatedness Affect Zero-Shot Transferability?

To test whether article relatedness between source and target domains affect performance, we experiment with Article P1-1 (Article 1 of Additional Protocol 1 - The Protection of Property) as the target domain. This simulates the realistic scenario of our zero shot setting where the convention is amended with an additional protocol. We then constructed one related and one unrelated source domain based on the suggestion provided by a legal expert (the second author) while ensuring training sets of similar size. The related domain consists of articles 6 (right to a fair trial) and 8 (right to respect for private and family life). The unrelated domain articles comprise articles 2 (right to life), 3

(prohibition of torture, and 5 (right to liberty and security).

We report the performance on Task A for target P1-1 in Table 5. We observe that the related source domain is able to perform better across the board, confirming the intuition that relatedness between source and target is an important factor to be considered when training a model for transferability. As before, we observe that UDA achieves higher performance overall as it has the chance to see article P1-1 during training. Interestingly, we observe the Wasserstein method outperforming the Domain Discriminator for the related source, but vice versa for the unrelated source. We believe this is owed to related articles forming similar feature distributions and thereby making it easy for the Wasserstein distance to facilitate adaptation. This case study suggests the design of domain adaptation components which derive information more from related articles than unrelated ones when transferring to a target article. This raises a related question of how article relatedness could be determined by the model itself rather than a priori by an expert.

6 Conclusion

We cast case outcome classification on ECtHR data into an article-aware architecture. This configuration is inspired by realistic legal reasoning involving both the case facts and convention articles to determine possible allegations/violations. Assuming non-finetuned pre-trained encoders, we observe a performance improvement over a simple fact-only classification model. It also enables us to conduct experiments in zero shot transfer COC with and without access to unlabeled target data during domain adaptation. While we show that domain adaptation techniques are in principle suitable to facilitate generalization, the divergence between source and target domain performance is large and this task variant is very difficult. We further observe that the effectiveness of domain adaptation interacts with law-specific pre-training of transformer-based encoders and with the relatedness of the source and target domains. Overall, this zero-shot COC task formulation opens up new research opportunities towards legal NLP models that are more aligned with expert reasoning.

Limitations

We cast the legal judgment prediction task into an article-aware classification setting and create a

Table 3: Task A F1 performance of baseline and domain adaptation models

Setting	Model	Transfer 0 → 1				Transfer 0 ← 1			
		source : split_0		target : split_1		source : split_1		target : split_0	
		mac.	mic.	mac.	mic.	mac.	mic.	mac.	mic.
Baseline	Source only	63.62	71.98	3.14	3.78	67.79	74.57	5.80	8.02
UDA	Domain Disc.	64.65	72.52	9.52	9.87	68.19	75.32	14.47	16.51
	Wasserstein	60.26	71.46	11.04	18.20	63.56	74.89	15.23	26.20
ADA	Domain Disc.	64.89	72.08	7.18	7.78	67.12	74.43	6.45	9.34
	Wasserstein	61.78	72.36	7.27	7.61	65.71	74.88	6.71	9.71

Table 4: Task A F1 Performance in one split using BERT base embeddings (as opposed to Legal Bert)

Setting	Model	source : split_1		target : split_0	
		mac.	mic.	mac.	mic.
UDA	Dom. Disc.	68.01	75.26	13.68	15.21
	Wasserstein	62.15	74.32	14.12	16.36
ADA	Dom. Disc.	67.92	75.32	4.77	7.44
	Wasserstein	66.71	74.95	4.73	7.65

Table 5: Task A F1 target performance on article P1-1 with related and unrelated source domains

Setting	Source Model	Related		Unrelated	
		mac.	mic.	mac.	mic.
UDA	Dom. Disc.	54.13	73.71	43.52	65.72
	Wasserstein	62.35	74.64	34.01	49.62
ADA	Dom. Disc.	42.79	68.46	37.12	56.16
	Wasserstein	43.25	69.91	26.87	38.28

zero-shot benchmark on a corpus of ECtHR cases. Matching between the text of legal sources and case fact descriptions varies greatly between different legal systems and subdomains, and is highly dependent on the textual nature of the case fact and legal sources. Specific to our context, for example, we have discussed the ECtHR fact statements as being influenced by the eventual case outcome and not suitable for prospective prediction in sec 1. COC as article-aware classification in other jurisdictions will likely lead to different levels of task difficulty, absolute performance, and zero shot transferability. In particular, many legal areas require multiple sources to be applied in conjunction to a set of case facts.

Technically, a major hurdle dealing with corpora related to the legal domain is their lengthy nature. We resort to hierarchical models, which are inherently limited in that tokens across long distances cannot directly attend to one another. This restriction of hierarchical models is still underexplored (but see preliminary work in, e.g. Dai et al. 2022; Chalkidis et al. 2022a). Additionally, we freeze the weights in the LegalBERT sentence encoder, both to save computational resources and to reduce the model’s susceptibility to shallow surface signals

and ensure the comparability of our domain adaptation methods, in particular with respect to the impact of domain-specific pre-training. We leave an exploration of COC as article-aware classification with fine-tuned encoders for future work.

Ethics Statement

We experiment with a publicly available datasets of ECtHR decisions, which has been derived from the public court database HUDOC³. These decisions contain real names of the parties involved without any anonymization. We hence do not consider our experiments to produce any additional harmful effects relating to personal information.

The task of legal judgment prediction raises ethical, civil rights, and legal policy concerns, both general and specific to the European Court of Human Rights (e.g., (Fikfak, 2021) on system bias and court caseload). The main premise of this work is to make incremental technical progress towards enabling systems to work with case outcome information in a way that is aligned with how human experts analyze case facts through an interplay with complex legal sources. We do not advocate for the practical application of COC/LJP systems by courts, but rather explore how their core functionality of processing legal text can be made as expert-aligned as possible. Our research group is strongly committed to research on such models as a means to derive insight from legal data for purposes of increasing transparency, accountability, and explainability of data-driven systems in the legal domain.

We are conscious that, by adapting pre-trained encoders, our models inherit any biases they contain. Similarly, the ECtHR case collection as historical data may contain a data distribution in which sensitive attributes of individuals (e.g., applicant gender) may have some predictive signal for the allegation/violation variable (see, e.g., (Chalkidis et al., 2022c)). We believe the results we observe

³<https://hudoc.echr.coe.int>

in our COC experiments to not be substantially related to such encoded bias. However, legal NLP systems leveraging case outcome information and intended for practical deployment should naturally be scrutinized against applicable equal treatment imperatives regarding their performance, behavior, and intended use.

All models of this project were developed and trained on Google Colab. We did not track computation hours.

References

- Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93.
- Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.
- Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885.
- Eyal Ben-David, Nadav Oved, and Roi Reichart. 2022. Pada: Example-based prompt learning for on-the-fly adaptation to unseen domains. *Transactions of the Association for Computational Linguistics*, 10:414–433.
- John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128.
- Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2011. Relation adaptation: learning to extract novel relations with minimum supervision. In *Twenty-Second International Joint Conference on Artificial Intelligence*.
- Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323.
- Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022a. An exploration of hierarchical attention transformers for efficient long document classification. *arXiv preprint arXiv:2210.05529*.
- Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904.
- Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. Paragraph-level rationale extraction through regularization: A case study on european court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 226–241.
- Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022b. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330.
- Ilias Chalkidis, Tommaso Pasini, Sheng Zhang, Letizia Tomada, Sebastian Schwemer, and Anders Søgaard. 2022c. Fairlex: A multilingual benchmark for evaluating fairness in legal text processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4389–4406.
- Junyi Chen, Lan Du, Ming Liu, and Xiabing Zhou. 2022. Mulan: A multiple residual article-wise attention network for legal judgment prediction. *Transactions on Asian and Low-Resource Language Information Processing*, 21(4):1–15.
- Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.
- Xiang Dai, Ilias Chalkidis, Sune Darkner, and Desmond Elliott. 2022. [Revisiting transformer-based models for long document classification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7212–7230, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Hal Daume III and Daniel Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126.
- Veronika Fikfak. 2021. What future for human rights? decision-making by algorithm. *Decision-making by algorithm (September 3, 2021)*. *Strasbourg Observers*, 19.
- Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.
- Nils Holzenberger, Andrew Blair-stanek, and Benjamin Van Durme. 2020. A dataset for statutory reasoning in tax law entailment and question answering. In *NLLP@KDD*.
- Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain ner using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2464–2474.
- Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271.
- Daniel Martin Katz, Michael J Bommarito, and Josh Blackman. 2017. A general approach for predicting the behavior of the supreme court of the united states. *PLoS one*, 12(4):e0174698.
- Aaron Russell Kaufman, Peter Kraft, and Maya Sen. 2019. Improving supreme court forecasting using boosted decision trees. *Political Analysis*, 27(3):381–387.
- Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- Diederik P Kingma and Jimmy Ba. 2015. Adam: a method for stochastic optimization 3rd int. In *International Conference on Learning Representations*.
- André Lage-Freitas, Héctor Allende-Cid, Orivaldo Santana, and Lívia Oliveira-Lage. 2022. Predicting brazilian court decisions. *PeerJ Computer Science*, 8:e904.
- Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2237–2243.
- Zhenyu Liu and Huanhuan Chen. 2017. A predictive performance comparison of machine learning models for judicial cases. In *2017 IEEE Symposium series on computational intelligence (SSCI)*, pages 1–6. IEEE.
- Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to predict charges for criminal cases with legal basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2727–2736.
- Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021. Ildc for cjpe: Indian legal documents corpus for court judgment prediction and explanation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4046–4062.
- Masha Medvedeva, Ahmet Üstün, Xiao Xu, Michel Vols, and Martijn Wieling. 2021. Automatic judgment forecasting for pending applications of the european court of human rights. In *ASAIL/LegalAIIA@ICAIL*.
- Masha Medvedeva, Michel Vols, and Martijn Wieling. 2020. Using machine learning to predict decisions of the european court of human rights. *Artificial Intelligence and Law*, 28(2):237–266.
- Aakanksha Naik and Carolyn Rose. 2020. Towards open domain event trigger identification using adversarial domain adaptation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7618–7624.
- Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. 2021. Swiss-judgment-prediction: A multilingual legal judgment prediction benchmark. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 19–35.
- Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855.
- Robert Remus. 2012. Domain adaptation using domain similarity-and domain complexity-based instance selection for cross-domain sentiment analysis. In *2012 IEEE 12th international conference on data mining workshops*, pages 717–723. IEEE.
- Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention.
- Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.
- Sebastian Ruder. 2019. *Neural transfer learning for natural language processing*. Ph.D. thesis, NUI Galway.
- Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054.
- T. Y. S. S Santosh, Marcel Perez San Blas, Phillip Kemper, and Matthias Grabmair. 2023. Leveraging task dependency and contrastive learning for case

- outcome classification on european court of human rights cases. *arXiv preprint arXiv:2302.00768*.
- T.y.s.s Santosh, Shanshan Xu, Oana Ichim, and Matthias Grabmair. 2022. [Deconfounding legal judgment prediction for European court of human rights cases towards better alignment with experts](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1120–1138, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. 2017. Adversarial training for cross-domain universal dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 71–79.
- JURI SAYS. 2020. Prediction system for the european court of human rights. In *Legal Knowledge and Information Systems: JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020*, volume 334, page 277. IOS Press.
- Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1056–1063.
- Rafe Athar Shaikh, Tirath Prasad Sahu, and Veena Anand. 2020. Predicting outcomes of legal cases based on legal factors using classifiers. *Procedia Computer Science*, 167:2393–2402.
- Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- Benjamin Strickson and Beatriz De La Iglesia. 2020. Legal judgement prediction for uk courts. In *Proceedings of the 2020 the 3rd international conference on information science and system*, pages 204–209.
- Octavia-Maria Şulea, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P Dinu, and Josef van Genabith. 2017a. Exploring the use of text classification in the legal domain.
- Octavia-Maria Şulea, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. 2017b. Predicting the law area and decisions of french supreme court cases. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 716–722.
- Michael Benedict L Virtucio, Jeffrey A Aborot, John Kevin C Abonita, Roxanne S Avinante, Rother Jay B Copino, Michelle P Neverida, Vanesa O Osiana, Elmer C Peramo, Joanna G Syjuco, and Glenn Brian A Tan. 2018. Predicting decisions of the philippine supreme court using natural language processing and machine learning. In *2018 IEEE 42nd annual computer software and applications conference (COMPSAC)*, volume 2, pages 130–135. IEEE.
- Tomer Volk, Eyal Ben-David, Ohad Amosy, Gal Chechik, and Roi Reichart. 2022. Example-based hypernetworks for out-of-distribution generalization. *arXiv preprint arXiv:2203.14276*.
- Bernhard Waltl, Georg Bonczek, Elena Scepankova, Jörg Landthaler, and Florian Matthes. 2017. Predicting the outcome of appeal decisions in germany’s tax law. In *International conference on electronic participation*, pages 89–99. Springer.
- Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. 2019a. Adversarial domain adaptation for machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2510–2520.
- Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. 2019b. Hierarchical matching network for crime classification. In *proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 325–334.
- Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, and ShaoZhang Niu. 2018. Modeling dynamic pairwise attention for crime classification over legal articles. In *the 41st international ACM SIGIR conference on research & development in information retrieval*, pages 485–494.
- Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783.
- Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. 2019. Legal judgment prediction via multi-perspective bi-feedback network. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4085–4091.
- Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Unsupervised domain adaptation for neural machine translation. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 338–343. IEEE.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489.

- Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. 2018. Robust multilingual part-of-speech tagging via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 976–986.
- Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923.
- Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. 2018. Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 682–690.
- Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021. Neurjudge: a circumstance-aware neural framework for legal judgment prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 973–982.
- Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. 2021. When does pre-training help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 159–168.
- Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549.
- Haoxi Zhong, Yuzhong Wang, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Iteratively questioning and answering for interpretable legal judgment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1250–1257.
- Yftah Ziser and Roi Reichart. 2017. Neural structural correspondence learning for domain adaptation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 400–410.
- The sentence level GRU encoder dimension is 200 (i.e. 400 bidirectional), and the sentence level attention vector dimension is 200. The entailment classifier hidden layer also has size 200. Domain discriminator and critic have two layered networks with hidden layers of size 200 and 100. The entailment classifier is trained with a binary cross entropy loss while the domain discriminator is trained with cross entropy loss over a one-hot domain vector. The model is optimized end-to-end using Adam (Kingma and Ba, 2015). The dropout rate (Srivastava et al., 2014) in all layers is 0.1. To handle data skewness in the entailment setup, we employ a custom batch sampler which ensures every batch contains 4 different articles as well as 2 positive and 2 negative instances per article. Our batch size is 16. We employ a learning rate scheduler based on loss plateau decay. For adversarial training using GRL, following (Ganin and Lempitsky, 2015), we set the λ in gradient reversal to be $\lambda = \frac{2}{1+\exp(-\gamma p)} - 1$ where $p = \frac{t}{T}$, where t and T denote current training step and total training steps. γ is determined using a grid search over [0.05, 0.1, 0.15, 0.2]. We employ a 10 class domain discriminator (5 from source and 5 from target) in the case of UDA and a 5 class discriminator in the case of ADA. We reduce the mean between instances of a particular article of source and target in the case of UDA. In the case of ADA, we reduce the mean between instances of different articles in the source domain.

A Implementation Details

We employ a maximum sentence length of 256 and document length (number of sentences) of 50. Our word level attention context vector size is 300.

Abstractive Document Summarization with Summary-length Prediction

Jingun Kwon¹, Hidetaka Kamigaito^{1,2}, and Manabu Okumura¹

¹Tokyo Institute of Technology

²Nara Institute of Science and Technology (NAIST)

kwon.j.ad@m.titech.ac.jp

kamigaito.h@is.naist.jp

oku@pi.titech.ac.jp

Abstract

Recently, we can obtain a practical abstractive document summarization model by fine-tuning a pre-trained language model (PLM). Since the pre-training for PLMs does not consider summarization-specific information such as the target summary length, there is a gap between the pre-training and fine-tuning for PLMs in summarization tasks. To fill the gap, we propose a method for enabling the model to understand the summarization-specific information by predicting the summary length in the encoder and generating a summary of the predicted length in the decoder in fine-tuning. Experimental results on the WikiHow, NYT, and CNN/DM datasets showed that our methods improve ROUGE scores from BART by generating summaries of appropriate lengths. Further, we observed about 3.0, 1.5, and 3.1 point improvements for ROUGE-1, -2, and -L, respectively, from GSum on the WikiHow dataset. Human evaluation results also showed that our methods improve the informativeness and conciseness of summaries.

1 Introduction

Current abstractive summarization models mostly utilize pre-trained language models (PLMs) (Liu and Lapata, 2019; Dou et al., 2021; Liu and Liu, 2021; Narayan et al., 2021; Liu et al., 2022a). Abstractive document summarization requires an encoder to determine the important parts in an input text and a decoder to output a non-redundant summary of the appropriate length relevant to the input. Thus, the characteristics required for an abstractive summarization model differ from those required as a language model, and are not usually considered in the pre-training for PLMs (Devlin et al., 2019; Zhang et al., 2019; Lewis et al., 2020). Hence, we need to fine-tune a PLM with a summarization dataset to treat it as an abstractive summarization model. Unlike training a randomly initialized model, this fine-tuning maintains and inherits the

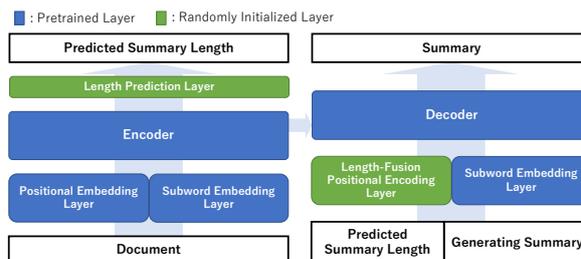


Figure 1: Overview of our methods. The length prediction layer predicts the summary length. The length-fusion positional encoding layer controls the decoder to generate a summary of the appropriate summary length.

parameters learned as an original language model. Therefore, to learn an abstractive summarization model by fine-tuning a PLM, it is necessary to suppress its characteristics as a language model while enabling it to learn the unique properties of abstractive summarization.

For this purpose, we propose two regularization methods for fine-tuning a PLM to learn abstractive summarization. Figure 1 shows an overview of our methods. The first method is a regularization method that uses the encoder’s hidden states to predict the length of an output summary. When the length is not given for a summary to be generated, we believe it is difficult to determine what volume of important key contents to select from the original document. Thus, fixing the length for a summary can make it easier to select key contents for it. We think humans can also create more informative and concise summaries when a summary length is given. The system should also be better trained for selecting key contents in the original document for a summary in case when it can be provided with the length of the summary.

The second method provides the decoder with the length predicted by the first method and enables it to learn to output a summary of the length. In addition to regularizing the training of the decoder, this method reduces the search space by searching

only for summaries of the appropriate length during generation, and so it is expected to produce a concise and informative summary. Although there have been studies on adjusting the output length of summaries, they have focused on controlling the output length for a given desired length (Kikuchi et al., 2016; Liu et al., 2018; Takase and Okazaki, 2019; Makino et al., 2019; Saito et al., 2020; Yu et al., 2021).¹ We incorporate a target-length prediction task to the encoder side and then inject the predicted length to the decoder side to generate the final summary.

In an evaluation on the WikiHow, NYT, and CNN/DM datasets, our methods improve the ROUGE scores of BART with appropriate lengths of summaries. On the WikiHow dataset, the performance improvement reached about 3.0, 1.5, and 3.1 points for ROUGE-1, -2, and -L, respectively, from GSum. Human evaluation results also showed that our methods enable the fine-tuning for a PLM to generate informative and concise summaries.

Our contributions are as follows: (1) We propose a regularization method that uses the encoder’s hidden states by predicting the length of a summary. (2) We propose a regularization method that reduces the search space by injecting the predicted length of a summary. (3) Both automatic and human evaluation results show that our novel model that combines (1) and (2) can generate a summary closer to its gold summary length by improving informativeness.

2 Our Methods

We apply our regularization methods to a transformer-based (Vaswani et al., 2017) PLM to generate a summary from a given document.

2.1 Predicting Summary Length

We impose summary-length prediction on the encoder during fine-tuning to make it easier for the encoder to determine how much important information the given document contains. The encoder converts a sequence of n tokens in a document $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ into hidden states $\{h_1, h_2, \dots, h_n\}$. Note that h_n is a hidden state of an end-of-document symbol x_n .

Then, we propose the length-prediction layer by using h_n and a 2-layer feed-forward neural network u to predict the summary length, which is the

number of subwords in the summary, as follows:

$$\ell_{pred} = u(h_n). \quad (1)$$

After that, by using the root-mean-square error (RMSE), the regularization loss for the encoder \mathcal{L}_{len} is calculated as follows:

$$\mathcal{L}_{len} = \sqrt{(\ell_{pred} - \ell_{gold})^2}, \quad (2)$$

where ℓ_{gold} is the gold length of the target summary.

2.2 Generating a Summary with the Predicted Length

We provide the decoder with the predicted summary length to generate a concise summary of the appropriate length relevant to the given document.

To encode the information of the predicted length into the decoder while keeping its pre-trained information, we insert our Length-Fusion Positional Encoding layer (LFPE), which is a transformer layer, before the decoder. Our LFPE consists of the length-ratio positional encoding (LRPE) (Takase and Okazaki, 2019) and a transformer layer. LRPE converts the position information of an output token y_t at time t to a continuous vector p_t with considering the predicted length ℓ_{pred} as follows:

$$p_t = \begin{cases} \sin(t/\ell_{pred}^{2i/dim}) & (i \equiv 0 \pmod{2}) \\ \cos(t/\ell_{pred}^{2i/dim}) & (i \equiv 1 \pmod{2}), \end{cases} \quad (3)$$

where dim is the dimension size of the embedding.

Then, the transformer layer converts $\{p_1, p_2, \dots, p_t\}$ into $E_t = \{e_1, e_2, \dots, e_t\}$ at a decoding time-step t . When adopting LFPE, we replace the original sinusoidal positional encoding of the pre-trained decoder with E_t . After that, the decoder calculates the output probability of y_t as $P(y_t|y_{t-1}, \dots, y_1, \mathbf{x}, \ell_{pred})$.

Finally, the regularization loss for the decoder \mathcal{L}_{gen} is calculated as follows:

$$\mathcal{L}_{gen} = -\sum_{t=1}^m \log P(y_t|y_{t-1}, \dots, y_1, \mathbf{x}, \ell_{pred}), \quad (4)$$

where m is the number of tokens in the target summary. Note that we replace ℓ_{pred} with ℓ_{gold} in the decoder during training.

¹Previous work assumes the desired length is given.

Dataset	Training	Valid	Test
WikiHow	168,126 (47.2)	6,000 (45.2)	6,000 (45.4)
NYT	44,382 (28.9)	5,523 (31.2)	6,495 (30.9)
CNN/DM	287,084 (20.5)	13,367 (25.1)	11,490 (22.0)

Table 1: Statistics of document summarization datasets. The value in parentheses indicates the variance of target summary lengths.

Model	R-1	R-2	R-L	VAR	AVG
WikiHow					
PEGASUS _{LARGE} *	43.06	19.71	41.35	-	
GSum*	41.74	17.73	40.09	-	
GSum	42.04	18.03	40.47	1.38	61.3
BART	<u>42.05</u>	<u>18.06</u>	<u>40.50</u>	<u>1.34</u>	57.5
BART w/ R _{enc}	44.68 [†]	19.48 [†]	43.31 [†]	0.98 [†]	51.5
BART w/ R _{enc+dec}	45.02[†]	19.53[†]	43.56[†]	0.82[†]	54.4
NYT					
GSum	<u>57.63</u>	<u>37.74</u>	41.99	1.62	151.8
BART	57.32	37.63	41.88	<u>1.55</u>	149.3
BART w/ R _{enc}	57.50	37.67	41.92	1.43 [†]	146.8
BART w/ R _{enc+dec}	58.52[†]	38.65[†]	43.48[†]	0.89[†]	129.9
CNN/DM					
PEGASUS _{LARGE} *	44.17	21.47	41.11	-	
GSum*	45.94	22.32	42.48	-	
GSum	45.79	22.21	42.37	<u>0.76</u>	69.7
BART	44.48	21.41	41.19	0.78	70.7
BART w/ R _{enc}	44.59	21.40	41.07	0.59 [†]	64.3
BART w/ R _{enc+dec}	44.65	21.60	41.25	0.36[†]	51.0

Table 2: Experimental results on WikiHow, NYT, and CNN/DM. † indicates the improvement is significant ($p < 0.05$) compared with the best baseline score (underlined) on each dataset. * indicates the reported score in the original paper. AVG indicates the average generated summary length.

2.3 Objective Function

To balance the encoder and decoder regularization, we sum the two losses through a hyperparameter λ for calculating the final loss as follows:

$$\mathcal{L} = \mathcal{L}_{gen} + \lambda \cdot \mathcal{L}_{len}. \quad (5)$$

3 Experiments

3.1 Experimental Settings

Datasets: We used WikiHow (Koupae and Wang, 2018) in the knowledge base domain and NYT² (Sandhaus, 2008) and CNN/DM (Hermann et al., 2015) in the news domain. Table 1 shows the dataset statistics.

Evaluation Metrics: We used F-scores of ROUGE-1 (R-1), -2 (R-2), and -L (R-L) in our experiments. To evaluate the quality of the predicted length and the length-controllability, we employed

²Detailed pre-processing steps are described in Appendix A.

the length variance (VAR): $\text{VAR} = 0.001 \times \frac{1}{n} \sum_{i=1}^n |y_{\text{pred}} - y_{\text{gold}}|$, where y_{pred} is the length of the generated summary and y_{gold} is the length of the reference summary in word level, respectively.

Compared Methods: We used BART-large (Lewis et al., 2020) for constructing baselines and our models by following the previous work (Dou et al., 2021). The proposed models are as follows. **BART w/ R_{enc}** employs our method only for the encoder in §2.1. **BART w/ R_{enc+dec}** employs our methods both for the encoder and the decoder. The baseline models are as follows. **BART** and **PEGASUS** (Zhang et al., 2019) are the original pre-trained BART and PEGASUS. **GSum** (Dou et al., 2021) is a BART-based combination model that utilizes extracted sentences as a guidance signal to consider extractive aspects for a summary. For the guidance signal, it uses the MatchSum model (Zhong et al., 2020).

We followed the hyperparameters of **BART** and **GSum** for training and testing the baselines and our models. We set λ to 0.1, 0.05, and 0.05 for WikiHow, NYT, and CNN/DM, respectively, on the basis of validation performances.³

3.2 Automatic Evaluation

The results are shown in Table 2. We can see that both of our models, **BART w/ R_{enc}** and **BART w/ R_{enc+dec}**, showed significant improvement in ROUGE scores over BART on WikiHow. These scores were higher than the combination model of GSum and PEGASUS (Zhang et al., 2019), which yields the current best results reported on WikiHow. We analyzed relations between lengths and ROUGE scores. When our **BART w/ R_{enc+dec}** predicted summary lengths closer to gold summary lengths than BART, 95.4% of generated summaries from ours obtained higher R-1 scores than BART. In addition, VAR and AVG scores show that our models can generate summaries closer to the gold summary lengths and can actually reduce the search space in decoding steps. These results indicate that the proposed methods enable BART to generate highly abstractive summaries of appropriate lengths.

We can also confirm that the proposed methods improved summarization performance over BART on NYT⁴ and CNN/DM. We can also see that

³Further details are described in Appendix B.

⁴There is no reported result for PEGASUS on NYT. For GSum, since the pre-processing could not be made identical, the reported and our scores were a bit different.

Model	WikiHow		CNN/DM	
	Info	Con	Info	Con
GSUM	-	-	3.97	4.02
BART	4.00	4.22	3.98	4.02
BART w/ $R_{enc+dec}$	4.09 [†]	4.19	4.05 [†]	4.07

Table 3: Human evaluation results. The notations are the same as in Table 2.

our model BART w/ $R_{enc+dec}$ showed significant improvement in ROUGE scores over GSUM on NYT. Although GSUM outperformed our BART w/ $R_{enc+dec}$ in ROUGE scores on CNN/DM, it could generate summaries closer to the gold summary lengths.

Thus, we tried to investigate what types of datasets our methods can work better on and found that the variance of reference summary lengths might be related to the performance of our models. Based on the observations from Tables 1 and 2, our BART w/ $R_{enc+dec}$ can largely improve performances on summarization datasets with a high variance of summary lengths, such as WikiHow and NYT.

3.3 Human Evaluation

For human evaluation, we sampled 100 documents each from WikiHow and CNN/DM. By using Amazon Mechanical Turk, we assigned 40 evaluators who obtained both US high school and US bachelor’s degrees to each dataset for grading the results with scores from 1 to 5 (5 is the best) in terms of informativeness (Info) and conciseness (Con).

Table 3 shows the results. These results indicate that BART w/ $R_{enc+dec}$ generated more informative summaries than BART, that is consistent with the results from the automatic evaluation. In some cases, the generated summaries with BART are just short summaries on WikiHow due to a high variance of reference summary lengths, and so the Con score is slightly lower than the one for BART w/ $R_{enc+dec}$. However, BART w/ $R_{enc+dec}$ yields the best overall Info and Con scores, which shows our regularization methods are essential for fine-tuning a PLM to learn abstractive summarization models. We also evaluated GSUM together. BART attained a 0.01 better score for Info than GSUM even on CNN/DM since GSUM focuses on generating faithful summaries with injecting outputs from an extractive summarization model.

We investigated the tendency of the length of generated summaries. Figure 2 shows the relation-

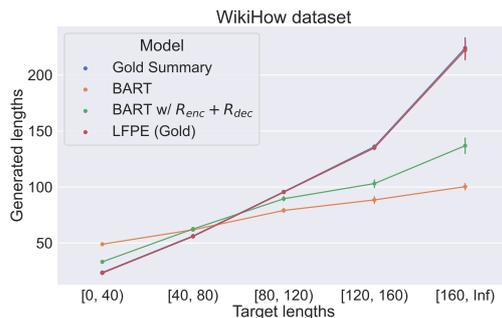


Figure 2: For x-axis, we divided the gold target lengths into 5 bins with 40 words interval. Y-axis indicates the length of generated summaries.

BART use this method if you have a digital multimeter with a diode check function. set your multimeter to resistance mode. plug the leads into the correct ports. disconnect the diode from the circuit. touch the leads in the forward-bias direction. lower the resistance range if the result is 0. test the resistance in the reverse direction. test a new diode or a working diode.

BART w/ $R_{enc+dec}$ set your multimeter to resistance mode. plug the leads into the multimeter. disconnect the diode from the circuit. touch the leads in the forward-bias direction. test in the reverse direction. try a new diode.

Gold use this method when necessary. set your multimeter to resistance mode. plug in the leads. disconnect the diode. measure the forward bias. measure the reverse bias. compare to a working diode.

Table 4: Example summaries generated from BART w/ $R_{enc+dec}$, BART, and gold summaries on WikiHow.

ship between gold and generated summary lengths for each model. We used WikiHow because it contains various target summary lengths. When we injected the gold summary length, the length of generated summaries from LFPE (Gold) was almost the same as the gold summaries. These results indicate that LFPE can precisely control various output lengths.⁵ In addition, generated summary lengths from BART w/ $R_{enc+dec}$ show that the length-prediction layer can also predict various target summary lengths.

Table 4 shows example generated summaries with BART w/ $R_{enc+dec}$, BART, and gold summaries on WikiHow. The summary length prediction is essential for creating an informative and concise summary that is closer to the gold summary length.

4 Related Work

In summary length control, previous work mostly focuses on controlling models for generating summaries with a predefined length (Kikuchi et al.,

⁵Further details are described in Appendix C.

2016; Liu et al., 2018; Takase and Okazaki, 2019; Makino et al., 2019; Saito et al., 2020; Yu et al., 2021). Our work is novel because it enables a model dynamically predicts the appropriate summary length from the input text without relying on any predefined length.

From the viewpoint of regularization, we can see such a regularization term like L_{len} in recent works of summarization tasks. Kamigaito et al. (2018); Kamigaito and Okumura (2020) in sentence compression and Ishigaki et al. (2019) in extractive document summarization incorporate dependency tree information into the attention (Kamigaito et al., 2017). Hsu et al. (2018) integrate extractive and abstractive summarization. MatchSum (Zhong et al., 2020) considers the semantic similarity between a document and its extracted summary. BRIO (Liu et al., 2022a) takes multiple similar abstractive summaries into account by contrastive learning in sequence-to-sequence (Edunov et al., 2018). Different from these works, our approach focuses on summary lengths through L_{len} and can be incorporated into these works by adding L_{len} to their loss function.

5 Conclusion

To fine-tune a pre-trained language model for abstractive document summarization, we proposed a regularization method that uses the encoder’s hidden states to predict the length of an output summary. We also proposed LFPE, that focuses on generating a summary with a given target length while keeping pre-trained information of the transformer-based model. We used LFPE to regularize the decoder during training to generate a summary with the predicted length.

Automatic evaluation results showed that the proposed methods enable BART to generate summaries of appropriate lengths while improving ROUGE scores. Human evaluation results also showed that the proposed methods enable BART to generate more informative and concise summaries.

6 Limitations

Although our models can largely improve performances on datasets with a high variance of summary lengths, the gain was small for datasets with a low variance of summary lengths. In the future, we will consider external resources to predict a summary length for the datasets with a low variance of target summary lengths. We plan to form document

clusters based on each topic since different topics may have different reference lengths. We believe this may improve performances for the datasets with a low variance of summary lengths.

Acknowledgements

We would like to gratefully acknowledge the anonymous reviewers for their helpful comments and feedbacks. This work was supported by Google AI Focused Research Award.

References

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. [GSum: A general framework for guided neural abstractive summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc’Aurelio Ranzato. 2018. [Classical structured prediction losses for sequence to sequence learning](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.
- Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15*. MIT Press.
- Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. [A unified model for extractive and abstractive summarization using inconsistency loss](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141, Melbourne, Australia. Association for Computational Linguistics.
- Tatsuya Ishigaki, Hidetaka Kamigaito, Hiroya Takamura, and Manabu Okumura. 2019. [Discourse-aware](#)

- hierarchical attention network for extractive single-document summarization. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 497–506, Varna, Bulgaria. INCOMA Ltd.
- Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2018. Higher-order syntactic attention network for longer sentence compression. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1716–1726, New Orleans, Louisiana. Association for Computational Linguistics.
- Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. 2017. Supervised attention for sequence-to-sequence constituency parsing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 7–12, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Hidetaka Kamigaito and Manabu Okumura. 2020. Syntactically look-ahead attention network for sentence compression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8050–8057.
- Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.
- Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.
- Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.
- Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.
- Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022a. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.
- Yizhu Liu, Qi Jia, and Kenny Zhu. 2022b. Length control in abstractive summarization by pretraining information selection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6885–6895, Dublin, Ireland. Association for Computational Linguistics.
- Yizhu Liu, Zhiyi Luo, and Kenny Zhu. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4110–4119, Brussels, Belgium. Association for Computational Linguistics.
- Takuya Makino, Tomoya Iwakura, Hiroya Takamura, and Manabu Okumura. 2019. Global optimization under length constraint for neural text summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1039–1048, Florence, Italy. Association for Computational Linguistics.
- Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.
- Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, Atsushi Otsuka, Hisako Asano, Junji Tomita, Hiroyuki Shindo, and Yuji Matsumoto. 2020. Length-controllable abstractive summarization by guiding with summary prototype.
- Evan Sandhaus. 2008. Ldc corpora. In *Linguistic Data Consortium*.
- Sho Takase and Naoaki Okazaki. 2019. Positional encoding to control output sequence length. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*

(*Long and Short Papers*), pages 3999–4004, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhongyi Yu, Zhenghao Wu, Hao Zheng, Zhe XuanYuan, Jefferson Fong, and Weifeng Su. 2021. [LenAtten: An effective length controlling unit for text summarization](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 363–370, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. [Pegasus: Pre-training with extracted gap-sentences for abstractive summarization](#).

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. [Extractive summarization as text matching](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

A Statistics of the datasets

NYT dataset consists of articles from the New York Times and the associated summaries.⁶ we followed the previous preprocessing step and splitting (Kedzie et al., 2018). There are two types of the reference summaries, which are archival abstracts and online teaser means. From this collection, we take all articles that have a concatenated summary length of at least 100 words.

B Model details

We introduce the detailed information of the baseline and our models.

We used Fairseq⁷ (Ott et al., 2019) for the model implementation. As the pretrained weight, we used bart-large in huggingface⁸. We used the original implementation for GSum⁹. We ran training for the models on two NVIDIA Tesla V100 with the multi-GPU setting. As described in the experimental settings, all hyperparameters were the same as for the large-scale BART in Lewis et al. (2020). Hyperparameter λ was set to 0.1, 0.05 and 0.05 for the WikiHow, CNN/DM, and NYT datasets, respectively, on the basis of validation performances.

⁶<https://catalog.ldc.upenn.edu/LDC2008T19>

⁷<https://github.com/pytorch/fairseq>

⁸<https://huggingface.co/facebook/bart-large>

⁹https://github.com/neulab/guided_summarization

Model	R-1	R-2	R-L	VAR
GOLC* (Makino et al., 2019)	38.27	16.22	34.99	5.13
PALUS* (Yu et al., 2021)	39.82	17.31	36.20	0.01
LPAS* (Saito et al., 2020)	43.23	20.46	40.00	-
PtLAAM* (Liu et al., 2022b)	44.17	20.63	40.97	-
BART	44.48	21.41	41.19	0.78
LRPE (Takase and Okazaki, 2019)	<u>45.67</u>	22.11	<u>42.20</u>	0.03
LFPE (Our)	45.93[†]	22.30	42.44[†]	0.03

Table 5: Experimental results on CNN/DM with using the gold summary length information. The notations are the same as in Table 2.

Δ	Generated Summary
+1	She and her husband are celebrating their 10th wedding anniversary.
0	She and her husband are celebrating their 10th anniversary.
-1	She and her husband are now married 10 years.

Table 6: Example summaries generated from BART with LFPE for different lengths on CNN/DM. $\Delta = +1/-1$ indicates the injected length is larger/smaller than the gold summary.

C Length-controllability

We investigated the length-controllability of our LFPE in §2.2 by comparing it with the original BART and LRPE. We also compared these methods with the previously reported scores of GOLC, PALUS, LPAS, and PtLAAM. We used CNN/DM and gave the gold summary length to the models by following the previous work. The results in Table 5 show that LFPE outperformed other methods in terms of ROUGE scores and VAR. Thus, our LFPE can control the output summary length while keeping ROUGE scores and outperform the state-of-the-art length-controllable methods.

Next, we analyzed the effect of length-controllability in actually generated summaries in CNN/DM. Table 6 shows example generated summaries with injecting different lengths into LFPE. In this example, when there is no possibility of dropping a subword, our model paraphrases “10th” to “10” while maintaining the informativeness and grammaticality. From this observation, we can understand that our LFPE controls the summary length through subword-based paraphrasing, which is supported by the decoder’s ability of abstraction.

Hierarchical Label Generation for Text Classification

Jingun Kwon^{1,3}, Hidetaka Kamigaito^{1,2}, Young-In Song³, and Manabu Okumura¹

¹Tokyo Institute of Technology

²Nara Institute of Science and Technology (NAIST)

³Naver Corporation

jingun.kwon@navercorp.com

kamigaito.h@is.naist.jp

song.youngin@navercorp.com

oku@pi.titech.ac.jp

Abstract

Hierarchical text classification (HTC) aims to assign the most relevant labels with the hierarchical structure to an input text. However, handling unseen labels with considering a label hierarchy is still an open problem for real-world applications because traditional HTC models employ a pre-defined label set. To deal with this problem, we propose a generation-based classifier that leverages a Seq2Seq framework to capture a label hierarchy and unseen labels explicitly. Because of no available social media datasets that target at HTC, we constructed a new (**Blog**) dataset using pairs of social media posts and their hierarchical topic labels. Experimental results on the **Blog** dataset showed the effectiveness of our generation-based classifier over state-of-the-art baseline models. Human evaluation results showed that the quality of generated unseen labels outperforms even the gold labels.

1 Introduction

Hierarchical text classification (HTC) aims to assign the most relevant labels with their structure for a given document. Because real-world applications categorize documents into a structured class hierarchy sequence (Silla and Freitas, 2011), such as patent collections (Tikk et al., 2005), web content collections (Dumais and Chen, 2000), and medical record coding (Cao et al., 2020), it is needed to capture the label hierarchy for better categorization.

To solve the HTC task, recent work has focused on enhancing label embeddings with a taxonomic hierarchy (Cao et al., 2020; Zhou et al., 2020; Wang et al., 2021) or considering a sequential classification approach (Rivas Rojas et al., 2020; Yang et al., 2018, 2019) that leverages a Seq2Seq framework to capture the label hierarchy. Despite the previous methods being successful, their approaches classify labels sequentially by choosing them from the pre-defined label set in the training dataset. It is still an open problem for real-world applications to handle

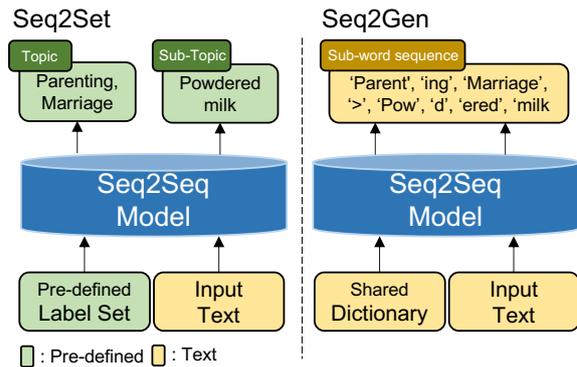


Figure 1: Different from previous Seq2Set (Rivas Rojas et al., 2020), our Seq2Gen can handle unseen labels with sub-word level generation.

unseen labels that do not appear in the pre-defined label set from the training dataset (Banerjee et al., 2019; Aly et al., 2019; Xu et al., 2021). Due to severe deficiencies in annotating data for labels in a hierarchy and handling unseen labels for real-world applications (Liu et al., 2021), we need a general modeling framework for handling unseen labels while explicitly incorporating a label hierarchy to overcome the restriction of the pre-defined label set for the development of real-world text classification applications.

For this purpose, we propose a generation-based classifier that can generate unseen labels in sub-word level. Our method can directly predict labels within a hierarchical structure by considering the label hierarchy as the order of the labels in a sequence. Because all labels are represented as sub-word strings in a shared vocabulary between labels and words, our method can predict unseen labels through generation (Sennrich et al., 2016). To expand unseen labels considerably, we also propose a method to extract knowledge of hierarchical labels from a pre-trained encoder-decoder by semi-supervised learning.

Since there are no available social media datasets

for HTC, we constructed a new blog dataset in Korean that includes a hierarchical label structure. The dataset contains up to three levels with a document. To evaluate the treatment of unseen labels in detail, we additionally constructed cross-lingual datasets, consisting of Japanese and English social media posts from the Kyoto (Hashimoto et al., 2011) and Reddit (Kim et al., 2019) datasets.

Comparisons between our generation-based and traditional classifiers on the Blog dataset showed that our method outperforms state-of-the-art models for both rank-based and ROUGE metrics. Human evaluation results showed that the quality of our generated unseen labels outperforms even the gold labels. In addition, we confirmed our generation-based classifier can handle unseen labels even on the cross-lingual datasets in a zero-shot setting, that shows the potential for tagging labels with considering a label hierarchy in unseen languages.

2 Problem Formulation

We introduce the task of traditional HTC and formulate how we solve it in our generation-based framework. The traditional HTC has been formalized as choosing labels one-by-one from a pre-defined label set in the training dataset, for example, with a sequential classification method (Seq2Set). However, handling unseen labels with considering a label hierarchy is important in designing models for real-world applications.

To solve this problem, we formulate the task as topic generation using a Seq2Seq model (Seq2Gen), such as pre-trained BART (Lewis et al., 2020). Figure 1 shows the Seq2Seq framework to generate target labels. It generates labels for an input text as a sequence of label tokens, and thus the label hierarchy can be directly considered through the Seq2Seq model. Because all the labels are represented as sub-word strings in a shared vocabulary between labels and words (Xiong et al., 2021), our model is permitted to generate even unseen labels, that are not included in the pre-defined label set (Sennrich et al., 2016). Due to the lack of diverse labels with considering their hierarchy in HTC datasets (Kowsari et al., 2017; Sinha et al., 2018), we utilize semi-supervised learning to draw the pre-trained knowledge in the pre-trained Seq2Seq model.

Topic Label	Hierarchical Template
$L = \{l_1\}$	l_1 is a topic.
$L = \{l_1, l_2\}$	l_2 is a sub-topic of l_1 .
$L = \{l_1, l_2, l_3\}$	l_2 is a sub-topic of l_1 and l_3 is a sub-topic of l_2 .

Table 1: Hierarchical template to map labels into a target topic sequence.

3 Generation-based Classifier

Considering HTC as a language generation task, we use a multi-lingual BART (mBART) (Liu et al., 2020), which is an extended version of a transformer-based pre-trained BART for multiple languages, as our Seq2Seq framework.

3.1 Seq2Seq-based Model

Our generation-based classifier can directly consider a label hierarchy. For learning, we append “>” as a special symbol representing a hierarchy between topics, $\mathbf{L} = \{l_1, l_2, l_3\}$, and concatenate them as a target topic sequence. Let w_i be the i -th token in a document $\mathbf{D} = \{w_1, w_2, \dots, w_n\}$. \mathbf{D} is fed into the encoder of the mBART, and then the generated hidden representations with the previous output token, c_{i-1} , are fed into the i -th step of the decoder. Finally, we use the cross-entropy loss between the decoder’s output and the label sequence to fine-tune the model, as follows:

$$H^{Enc} = \text{Encoder}(D), \quad (1)$$

$$H^{Dec} = \text{Decoder}(H^{Enc}, c_{i-1}), \quad (2)$$

$$\text{Loss} = - \sum_{i \in m} \log(\text{Softmax}(H^{Dec}W + b)), \quad (3)$$

where W and b indicate a learnable weight and bias, respectively, and m indicates the target length.

To show the effectiveness of directly considering a label hierarchy, we additionally consider a template-based Seq2Seq model. For learning, we manually create a hierarchical template, which has slots to map topic labels into a target topic sequence, instead of \mathbf{L} . Table 1 shows the hierarchical template to map topics into slots.

3.2 Augmentation with Semi-supervision

Since BART is a pre-trained Seq2Seq model learned with massive text corpora, we assume that we can draw pre-trained knowledge (Petroni et al., 2019) from BART to enhance the label hierarchy and expand labels considerably for dealing with unseen labels. For this purpose, we augment the

Training	Valid	Test
13,705 (1,011)	761 (254)	761 (292)

Table 2: Statistics of Blog. The number in parentheses indicates the number of different labels in each data.

dataset with a *silver* dataset, an automatically annotated dataset by using a model’s generation in a manner of semi-supervised learning. As demonstrated by He et al. (2020), we first train a model only with the *silver* dataset, generated by a model trained with the gold dataset, and then fine-tune it with the gold dataset.

4 Blog Dataset

We created a new HTC dataset (Blog) by collecting posts and their topic label sequences from Naver blogs,¹ that contain a large number of different labels compared to the previous HTC datasets (Kowsari et al., 2017; Sinha et al., 2018). The topic label sequences contain up to three hierarchical topic levels. Extracted topic label sequences can be noisy because a blogger can choose only the topic (the top-level class) from 32 classes, and the remaining topic sequence was automatically generated by the Naver blog system. Therefore, we hired experts on social media to annotate a relevance score from 0 to 3 (3 is the best) for a post and its topic label sequence. We filtered posts with scores less than 2 to ensure high quality. Then, we divided them into three parts (training: 90%, valid: 5%, and test: 5%). Table 2 shows the statistics of the created dataset.

To evaluate unseen label generation in cross-lingual few- and zero-shot settings, we additionally created Japanese (Kyoto) and English (Reddit) datasets from publicly available social media post datasets (Hashimoto et al., 2011; Kim et al., 2019). For Kyoto and Reddit, we extracted 249 and 500 posts, respectively. For each post, five human experts annotated a topic label sequence. After pre-processing, we obtained 234 and 400 posts with their label sequences for Kyoto and Reddit, respectively, and divided them into three parts (training: 10%, valid: 5%, and test: 85%). Blog, Kyoto, and Reddit are available upon request.²

¹<https://section.blog.naver.com/>

²Detailed explanations for the datasets are in Appendix A.

5 Experiments

5.1 Experimental Settings

Datasets: Blog, Kyoto, and Reddit were used to compare our generation-based and previous classification methods. To obtain silver data for semi-supervised learning, we additionally extracted 21,520 Naver blog posts. We also evaluated our models on the public HTC dataset, Web of Science (WOS) (Kowsari et al., 2017). It contains 46,985 instances with two levels, where each level consists of 7 and 134 different labels. We divided them into three parts (training: 60%, valid: 20%, and test: 20%).

Evaluation Metrics: Previous studies used a short ranked list of potentially relevant labels to evaluate the classification quality: the precision at top k ($P@k$) and the Normalized Discounted Cumulative Gain at top k ($NDCG@k$), where $k = 1, 2, 3$ (Xun et al., 2020; Zhang et al., 2021). However, these rank-based evaluation metrics could not evaluate the quality of a hierarchical label sequence, and thus, we also used ROUGE-1-F and ROUGE-2-F, that can evaluate the quality of hierarchical label sequences by taking into account label n-grams.

Compared Methods: Our methods are as follows: **Template** uses the proposed hierarchical templates to generate a topic label sequence with mBART.³ **Seq2Gen** directly generates a topic label sequence with mBART. **Self-Template** and **Self-Seq2Gen** use **Template** and **Seq2Gen** by expanding unseen labels with semi-supervised learning, respectively.

The baselines, which include state-of-the-art models that employ a tree structure of labels, are as follows: **CorNet** utilizes BERT (Devlin et al., 2019) by incorporating a feed-forward layer to consider a label hierarchy (Xun et al., 2020). **MATCH** utilizes BERT by incorporating hypernymy regularization in a loss function to consider hierarchical structures (Zhang et al., 2021).⁴ **Seq2Set** is a variant of the state-of-the-art HTC model that sequentially classifies a topic label sequence from a pre-defined label set with mBART. We replaced Bi-GRU with mBART for a fair comparison to our Seq2Gen (Rivas Rojas et al., 2020).

³Results using different templates are in Appendix B.

⁴For both CorNet and MATCH, we used a multilingual BERT instead of the original BERT for the cross-lingual setting.

⁵The paired-bootstrap-resampling (Koehn, 2004) was used ($p < 0.05$).

Model	P&N@1	P@2	P@3	N@2	N@3	R1-F	R2-F	Unseen
CorNet	77.79	50.72	36.88	70.03	72.76	47.77	8.76	-
MATCH	78.06	50.72	36.05	70.23	72.10	46.76	9.20	-
Seq2Set	92.38	<u>64.72</u>	<u>43.58</u>	88.36	88.23	<u>81.61</u>	<u>35.50</u>	-
Template	92.12	68.13 [†]	46.25 [†]	89.37	89.44	84.60 [†]	43.17 [†]	91
Seq2Gen	92.25	69.58 [†]	47.39 [†]	89.33	89.53	<u>85.51</u> [†]	<u>45.42</u> [†]	102
Self-Template	92.38	<u>68.33</u>	<u>46.30</u>	89.79	89.84	85.88	43.36	74
Self-Seq2Gen	92.77	69.84	47.48	90.23	90.36	87.69 [‡]	45.95	62

Table 3: Experimental results on Blog. Unseen indicates the number of different generated unseen labels on the test data. [†] and [‡] indicate the improvement is significant over the underlined score, respectively.⁵

Model	Kyoto		Reddit	
	R1-F	R2-F	R1-F	R2-F
Few-shot				
CorNet	47.48	15.83	20.60	0.39
MATCH	48.88	18.08	19.64	0.20
Seq2Set	63.75	51.83	19.69	3.24
Seq2Gen	56.73	35.50	33.20	7.84
Zero-shot				
Seq2Gen	41.12	13.83	17.48	4.61

Table 4: Results on Kyoto and Reddit.

Model	P&N@1	P@2	N@2	R1-F	R2-F	Unseen
CorNet	78.76	53.89	59.52	53.89	16.78	-
MATCH	74.14	51.07	56.29	51.07	13.53	-
Seq2Set	91.23	<u>85.94</u>	87.14	<u>85.94</u>	<u>80.55</u>	-
Seq2Gen	91.43	86.32 [†]	87.48	86.32 [†]	81.11 [†]	1

Table 5: Experimental results on WOS. The notations are the same as in Table 3.

5.2 Automatic Evaluation

Table 3 shows the results on Blog. Generating topic labels using the mBART-based models consistently outperformed classifying them using the mBERT-based models. Specifically, the gain was large in the ROUGE metrics. In addition, our generation-based methods, Template and Seq2Gen, outperformed the sequential classifier Set2Set. The proposed Seq2Gen outperformed Template, where the improvement in R2-F was larger than that in R1-F, that indicates Seq2Gen can capture a hierarchical sequence directly compared with the hierarchical template. Moreover, Self-Template and Self-Seq2Gen, that use the *silver* dataset to fine-tune the models, consistently improved the performances. This is because we succeeded in enhancing the label hierarchy with diverse unseen labels. For 21,520 posts in the *silver* dataset, our Seq2Gen

Model	Relevance	Taxonomy	Best
Seq2Set	2.29	2.17	0
Self-Seq2Gen	2.59	2.56 [†]	23
Gold	2.51	<u>2.46</u>	13

Table 6: Human evaluation results. The notations are the same as in Table 3.

Input: Yoon Restaurant’s Kimchi pancake. How to make kimchi pancake, recipe for kimchi pancake. It’s been a few days since spring rain has been so moist, so the air is very fresh:) ...
Gold: Cooking, Recipe
Self-Seq2Gen: Cooking, Recipe > Kimchi pancake
Input: I can’t go to the gym, I can’t exercise outside, watch diet YouTube at home. ... The problem with Home Training is that all the exercise moves go by so quickly. ...
Gold: Health, Medicine
Self-Seq2Gen: Sports > Home Training

Table 7: Examples of generated unseen labels from Self-Seq2Gen in the Blog dataset.

could generate 4,385 different unseen labels.

Table 4 shows the cross-lingual results. The R2-F scores for Seq2Gen, trained with Blog, in the zero-shot setting show that it can generate even cross-lingual unseen labels.⁶ Table 5 shows the results on WOS. We can confirm that the generation-based method outperformed the sequential classification method. Thus, our Seq2Gen can work better even for a smaller number of different labels. However, we think the improvements and the number of generated unseen labels are smaller than the ones on Blog due to the smaller number of different labels.

5.3 Human Evaluation and Analysis

We conducted a human evaluation for 50 randomly sampled posts that contain generated unseen labels from our Self-Seq2Gen. Five human annotators graded them with scores from 1 to 3 (3 is the best)

⁶Results including rank-based metrics are in Appendix C.

in terms of Relevance and Taxonomy.⁷ We additionally asked the annotators to select the best label sequence from Seq2Set, Self-Seq2Gen, and Gold label sequences. Best indicates the number of cases where the majority among the annotators judged the best. Table 6 shows the human evaluation results. The generated unseen labels from Self-Seq2Gen achieved a higher preference than the Gold labels.

Table 7 shows example generated unseen labels from Self-Seq2Gen. As we expected, our Self-Seq2Gen frequently generated unseen labels with considering the label hierarchy. In the first example, the generated unseen label, “Kimchi pancake”, can be considered as a sub-topic of “Cooking, Recipe” because the “Kimchi pancake” is a food name. In the second example, “Home training” can be considered as a sub-topic of “Sports”.

6 Conclusion

We proposed a generation-based classifier for HTC. It could handle unseen labels with considering their label hierarchy. In addition, we constructed cross-lingual HTC datasets from social media posts. Automatic evaluation results showed that our generation-based classifier could outperform state-of-the-art models. We confirmed our classifier could handle unseen labels by human evaluation.

7 Ethical Considerations

We created the new datasets of Blog, Kyoto, and Reddit for the HTC task. The created datasets have been collected in a manner which is consistent with the terms of use of any sources and the intellectual property and privacy rights of the original authors of the texts. Please note that we have confirmed by our legal team and the datasets will be available upon request for only research purpose.

8 Limitations

Although our Seq2Gen could generate unseen labels on cross-lingual datasets in the zero-shot setting, that shows the potential of tagging labels with considering their label hierarchy, it was difficult to outperform the few-shot setting. In the future, we plan to incorporate cross-lingual label trees for the zero-shot setting.

⁷Relevance and Taxonomy indicate how much the generated label sequences are related to the input context and the quality of the label hierarchy, respectively.

References

- Rami Aly, Steffen Remus, and Chris Biemann. 2019. [Hierarchical multi-label classification of text with capsule networks](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 323–330, Florence, Italy. Association for Computational Linguistics.
- Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsoulis. 2019. [Hierarchical transfer learning for multi-label text classification](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6295–6300, Florence, Italy. Association for Computational Linguistics.
- Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. [HyperCore: Hyperbolic and co-graph representation for automatic ICD coding](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Susan Dumais and Hao Chen. 2000. [Hierarchical classification of web content](#). In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00*, page 256–263, New York, NY, USA. Association for Computing Machinery.
- Chikara Hashimoto, Sadao Kurohashi, Daisuke Kawahara, Keiji Shinzato, and Masaaki Nagata. 2011. Construction of a blog corpus with syntactic, anaphoric, and sentiment annotations. *Journal of Natural Language Processing*, 18(2):175–201.
- Junxian He, Jiatao Gu, Jiajun Shen, and Marc’Aurelio Ranzato. 2020. [Revisiting self-training for neural sequence generation](#). In *International Conference on Learning Representations*.
- Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. [Abstractive summarization of Reddit posts with multi-level memory networks](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.
- Philipp Koehn. 2004. [Statistical significance tests for machine translation evaluation](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural*

- Language Processing*. Association for Computational Linguistics.
- Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. [Hdltext: Hierarchical deep learning for text classification](#). In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 364–371.
- Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. 2021. [Improving pretrained models for zero-shot multi-label text classification through reinforced label hierarchy reasoning](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1051–1062, Online. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. [Language models as knowledge bases?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.
- Kervy Rivas Rojas, Gina Bustamante, Arturo Oncevay, and Marco Antonio Sobrevilla Cabezedo. 2020. [Efficient strategies for hierarchical text classification: External knowledge and auxiliary tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2252–2257, Online. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Carlos N Silla and Alex A Freitas. 2011. [A survey of hierarchical classification across different application domains](#). In *Data Mining and Knowledge Discovery*, 22(1-2):31–72, New York, NY, USA.
- Koustuv Sinha, Yue Dong, Jackie Chi Kit Cheung, and Derek Ruths. 2018. [A hierarchical neural attention-based text classifier](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 817–823, Brussels, Belgium. Association for Computational Linguistics.
- Domonkos Tikk, György Biró, and Jae Dong Yang. 2005. [Experiment with a Hierarchical Text Categorization Method on WIPO Patent Collections](#), pages 283–302. Springer US, Boston, MA.
- Xuepeng Wang, Li Zhao, Bing Liu, Tao Chen, Feng Zhang, and Di Wang. 2021. [Concept-based label embedding via dynamic routing for hierarchical text classification](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5010–5019, Online. Association for Computational Linguistics.
- Yijin Xiong, Yukun Feng, Hao Wu, Hidetaka Kamigaito, and Manabu Okumura. 2021. [Fusing label embedding into BERT: An efficient improvement for text classification](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1743–1750, Online. Association for Computational Linguistics.
- Linli Xu, Sijie Teng, Ruoyu Zhao, Junliang Guo, Chi Xiao, Deqiang Jiang, and Bo Ren. 2021. [Hierarchical multi-label text classification with horizontal and vertical category correlations](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2459–2468, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Guangxu Xun, Kishlay Jha, Jianhui Sun, and Aidong Zhang. 2020. [Correlation networks for extreme multi-label text classification](#). KDD '20, page 1074–1082, New York, NY, USA. Association for Computing Machinery.
- Pengcheng Yang, Fuli Luo, Shuming Ma, Junyang Lin, and Xu Sun. 2019. [A deep reinforced sequence-to-set model for multi-label classification](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5252–5258, Florence, Italy. Association for Computational Linguistics.
- Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. [SGM: Sequence generation model for multi-label classification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yu Zhang, Zhihong Shen, Yuxiao Dong, Kuansan Wang, and Jiawei Han. 2021. Match: Metadata-aware text classification in a large hierarchy. In *WWW'21*, pages 3246–3257. ACM / IW3C2.

Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. [Hierarchy-aware global model for hierarchical text classification](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1106–1117, Online. Association for Computational Linguistics.

A 32 Topics for Naver blog System

Table 8 shows 32 topic classes (top-level) from Naver blog system.

For Kyoto and Reddi t, to establish the same setting as for Blog, the experts first annotated the topic label (the top-level class) from given 32 classes. Then, they annotated hierarchical label sequences up to three-levels if they consider subsequent labels are required. We deleted posts with no majority for the topic label. We obtained 234 and 400 posts with their label sequences for Kyoto and Reddi t, respectively, and divided them into three parts (training: 10%, valid: 5%, and test: 85%).

For Reddi t and Kyoto, each input text is not one-to-one matching for target labels, which is different from the **Blog** dataset. For training, we considered all different target label sequences. For the evaluation, we selected maximized scores by regrading them as multiple references. To assess the agreement between the participants for the datasets, we used Fleiss’ Kappa (L. Fleiss, 1971). We obtained Kappa scores of 0.55 for Kyoto and 0.23 for Reddi t, indicating moderate and fair agreements, respectively.

B Results using different templates.

We study the various manually created hierarchical templates using valid Blog because different hierarchical templates can express the same meaning. Table 9 shows the performance using different templates. On the basis of the valid results in terms of average ROUGE-F scores, we use the top performing template in our experiments.

C Results on Kyoto and Reddit datasets

Table 10 includes both rank-based and ROUGE metrics on Kyoto and Reddi t.

	Topic
1	Literature, Book
2	Movie
3	Art, Design
4	Performance, Exhibition
5	Music
6	Drama
7	Star, Celebrity
8	Cartoon, Anime
9	Broadcast
10	Everyday, Thoughts
11	Parenting, Marriage
12	Pet, Companion animal
13	Good article, Image
14	Fashion, Beauty
15	Interior, DIY
16	Cooking, Recipe
17	Product review
18	Horticulture, Cultivation
19	Game
20	Sports
21	Picture
22	Car
23	Hobby
24	Domestic travel
25	World travel
26	Restaurant
27	IT, Computer
28	Society, Politics
29	Health, Medicine
30	Business, Economy
31	Language, Foreign language
32	Education, Academic

Table 8: 32 topics from Blog datasets.

Topic Label	Hierarchical Template	R1-F	R2-F	Avg R-F
$L = \{l_1\}$	l_1 is a topic.			
$L = \{l_1, l_2\}$	l_2 is a sub-topic of l_1 .	86.92	45.90	66.41
$L = \{l_1, l_2, l_3\}$	l_2 is a sub-topic of l_1 and l_3 is a sub-topic of l_2 .			
$L = \{l_1\}$	l_1 is a topic.			
$L = \{l_1, l_2\}$	l_1 is a topic and l_2 is a sub-topic of l_1 .	86.72	44.88	65.80
$L = \{l_1, l_2, l_3\}$	l_1 is a topic, l_2 is a sub-topic of l_1 , and l_3 is a sub-topic of l_2 .			
$L = \{l_1\}$	l_1 is a topic.			
$L = \{l_1, l_2\}$	l_1 is a parent topic of l_2 .	87.31	43.82	65.57
$L = \{l_1, l_2, l_3\}$	l_1 is a parent topic of l_2 and l_2 is a parent topic of l_3 .			
$L = \{l_1\}$	l_1 is a topic.			
$L = \{l_1, l_2\}$	l_1 is a topic and l_1 is a parent topic of l_2 .	85.87	44.20	65.04
$L = \{l_1, l_2, l_3\}$	l_1 is a topic, l_1 is a parent topic of l_2 , and l_2 is a parent topic of l_3 .			

Table 9: Results using different hierarchical templates.

Model	Kyoto							Reddit						
	P&N@1	P@2	P@3	N@2	N@3	R1-F	R2-F	P&N@1	P@2	P@3	N@2	N@3	R1-F	R2-F
Few-shot														
CorNet	54.50	56.50	41.67	55.66	54.61	47.48	15.83	35.59	22.79	17.25	27.03	26.59	20.60	0.39
MATCH	57.50	56.75	43.00	56.92	56.65	48.88	18.08	29.71	20.44	15.78	24.02	25.68	19.64	0.20
Seq2Set	64.00	69.75	46.50	68.45	67.93	63.75	51.83	37.06	21.91	14.71	26.82	26.10	19.69	3.40
Seq2Gen	65.50	62.50	46.50	62.92	60.22	56.43	35.08	51.18	36.76	24.71	41.70	40.01	33.20	7.84
Zero-shot														
Seq2Gen	28.00	42.00	28.00	41.73	41.73	41.12	13.83	22.53	15.29	10.20	21.58	21.58	17.48	4.61

Table 10: Evaluation results on the **Kyoto** and **Reddit** datasets.

Active Learning for Multilingual Semantic Parser

Zhuang Li*, Gholamreza Haffari

Openstream.AI

{zhuang.li, reza.haffari}@openstream.com

Abstract

Current multilingual semantic parsing (MSP) datasets are almost all collected by translating the utterances in the existing datasets from the resource-rich language to the target language. However, manual translation is costly. To reduce the translation effort, this paper proposes the first active learning procedure for MSP (AL-MSP). AL-MSP selects only a subset from the existing datasets to be translated. We also propose a novel selection method that prioritizes the examples diversifying the logical form structures with more lexical choices, and a novel hyperparameter tuning method that needs no extra annotation cost. Our experiments show that AL-MSP significantly reduces translation costs with ideal selection methods. Our selection method with proper hyperparameters yields better parsing performance than the other baselines on two multilingual datasets.

1 Introduction

Multilingual semantic parsing converts multilingual natural language utterances into logical forms (LFs) using a single model. However, there is a severe data imbalance among the MSP datasets. Currently, most semantic parsing datasets are in English, while only a limited number of non-English datasets exist. To tackle the data imbalance issue, almost all current efforts build MSP datasets by translating utterances in the existing datasets from the resource-rich language (e.g. English) into other languages (Duong et al., 2017; Li et al., 2021a). However, manual translation is slow and laborious. In such cases, active learning is an excellent solution to lower the translation cost.

Active learning (AL) is a family of methods that collects training data when the annotation budgets are limited (Lewis and Catlett, 1994). Our work proposes the *first* active learning approach

for MSP. Compared to translating the full dataset, AL-MSP aims to select only a subset from the existing dataset to be translated, which significantly reduces the translation cost.

We further study which examples AL-MSP should select to optimize multilingual parsing performance. Oren et al. (2021) demonstrated that a training set with diverse LF structures significantly enhances compositional generalization of the parsers. Furthermore, our experiments show that the examples with LFs aligned with more diversified lexical variants in the training set considerably improve the performance of multilingual parsing during AL. Motivated by both, we propose a novel strategy for selecting the instances which include diversified LF structures with more lexical choices. Our selection method yields better parsing performance than the other baselines. By translating just 32% of all examples, the parser achieves comparable performance on multilingual GEO-QUERY and NLMAP as translating full datasets.

Prior works obtain the hyperparameters of the AL methods by either copying configurations from comparable settings or tuning the hyperparameters on the seed evaluation data (Duong et al., 2018). However, the former method is not suitable as our AL setting is unique, whereas the second method requires extra annotation costs. In this work, we provide a cost-free method for our AL scenario for obtaining optimal hyperparameters.

Our contributions are i) the first active learning procedure for MSP that reduces the translation effort, ii) an approach that selects examples for getting superior parsing performance, and iii) a hyperparameter tuning method for the selection that does not incur any extra annotation costs.

2 Background

Multilingual Semantic Parsing. A multilingual semantic parser is a parametric model $P_\theta(\mathbf{y}|\mathbf{x})$ that estimates the probability of the LF $\mathbf{y} \in \mathcal{Y}$ condi-

*Most of this author’s work was completed during his internship at Openstream.AI.

tioned on the natural language utterance $\mathbf{x} \in \mathcal{X}_l$ in an arbitrary language from a language set $l \in \mathcal{L}$. The model is trained on the utterance-LF pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N \in \mathcal{X}_L \times \mathcal{Y}$ where $\mathcal{X}_L = \bigcup_{l \in \mathcal{L}} \mathcal{X}_l$ includes multilingual utterances.

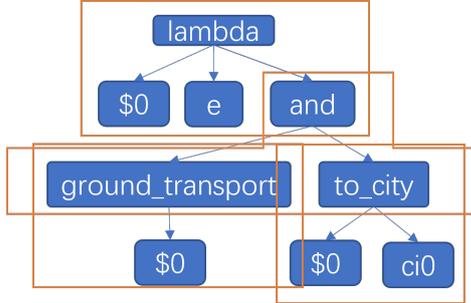


Figure 1: The example of the compounds in an LF tree, $(\text{lambda } \$0 \text{ e } (\text{and } (\text{ground_transport } \$0) (\text{to_city } \$0 \text{ ci0 })))$.

Atoms and Compounds. Each logical form sequence can be represented as a semantic tree, $\mathbf{y} = \tau_{\mathbf{y}}$. Oren et al. (2021); Shaw et al. (2021) define the nodes and sub-trees in $\tau_{\mathbf{y}}$ as the *atoms* and *compounds*, respectively. Increasing the diversity of the atoms and compounds in the training set improves the parser’s compositional generalization (Oren et al., 2021; Li et al., 2021b). For example, an LF “ $(\text{lambda } \$0 \text{ e } (\text{and } (\text{ground_transport } \$0) (\text{to_city } \$0 \text{ ci0 })))$ ” can be expressed as a tree as in Fig. 1. The atoms are nodes such as “*lambda*”, “*\$0*”, “*e*” in the LF tree. In this work, the compounds are defined as two-level sub-trees such as “ $(\text{ground_transport } \$0)$ ”, “ $(\text{to_city } \$0 \text{ ci0 })$ ”, “ $(\text{and } \text{ground_transport } \text{to_city })$ ”, and “ $(\text{lambda } \$0 \text{ e } \text{and })$ ” in the LF tree.

Data Collection for MSP. Prior data collection or active learning works annotates the LFs for the utterances (Duong et al., 2018; Sen and Yilmaz, 2020) or vice versa (Duong et al., 2018; Wang et al., 2015). But most MSP works (Susanto and Lu, 2017; Li et al., 2021a) obtain data by translating existing datasets from high-resource languages into low-resource languages, which is less costly since it does not need annotators’ expertise in LFs. Following the same annotation paradigm, our AL does not annotate LFs for multilingual utterances, but instead chooses the utterances to be translated.

3 Active Learning for MSP

AL-MSP considers only a bilingual scenario for the proof of concept, while extending our AL method to more than two languages is easy. The

goal of AL-MSP is to minimize the human effort in translating utterances while the semantic parser can still achieve a certain level of performance on the bilingual test sets. Starting from a semantic parser initially trained on the dataset $D_s = \{(\mathbf{x}_i^s, \mathbf{y}_i)\}_{i=1}^N$ whose utterances are in the high-resource language s , AL-MSP selects K_q examples $\tilde{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i)\}_{i=1}^{K_q}$ from D_s , followed by manually translating the utterances in \tilde{D}_s into a target language t , denoted by $\tilde{D}_t = t_{s \rightarrow t}(\tilde{D}_s)$, where $\tilde{D}_t = \{(\mathbf{x}_i^t, \mathbf{y}_i)\}_{i=1}^{K_q}$. The selection criterion is based on our proposed *acquisition* function $\phi(e_s)$ scoring each example, $e_s = (\mathbf{x}_s, \mathbf{y})$. The parser is re-trained on the union of \tilde{D}_t and D_s . There will be Q iterations of selection and re-training until the re-trained parser reaches a good performance on the bilingual test sets T_s and T_t . Algorithm 1 describes our experimental settings in detail.

Algorithm 1: AL-MSP

Input : Initial training set $D^0 = D_s^0$, budget size K_q , number of the selection rounds Q
Output : A well-learned multilingual parser $P_\theta(\mathbf{y}|\mathbf{x})$
 Train the parser $P_\theta(\mathbf{y}|\mathbf{x})$ on the training set D^0
for $q \leftarrow 1$ **to** Q **do**
 Estimate the acquisition $\phi(\cdot)$
 Select a subset $\tilde{D}_s^q \in D_s^{q-1}$ of the size K_q based on the acquisition function $\phi(\cdot)$
 Translate the utterances in \tilde{D}_s^q into the target language, $\tilde{D}_t^q = t_{s \rightarrow t}(\tilde{D}_s^q)$.
 Combine the training sets, $D^q = D^{q-1} \cup \tilde{D}_t^q$
 Exclude the selected examples \tilde{D}_s^q from $D_s^q = D_s^{q-1} \setminus \tilde{D}_s^q$
 Re-train the parser $P_\theta(\mathbf{y}|\mathbf{x})$ on D^q
 Evaluate parser performance on test sets T_s, T_t
end

3.1 Selection Acquisition

Our selection strategy selects the untranslated examples which maximize the acquisition scores. The acquisition comprises two individual terms, LF Structure Diversity and Lexical Choice Diversity.

LF Structure Diversity (LFSD). We give a simple technique to diversify the LF substructures (atoms and compounds) in the instances. At q th iteration, let $D_s^l = \bigcup_{i=1}^{q-1} \tilde{D}_s^i$ denotes all the translated examples and $D_s^u = D_s^{q-1}$ be the untranslated ones. We partition their union $D_s^u \cup D_s^l$ into $|D_s^l| + K_q$ clusters with Incremental K-means (Dataiku Lab, 2022). Each example $e_s = (\mathbf{x}_s, \mathbf{y})$ is featurized by extracting all the atoms and compounds in the LF tree $\tau_{\mathbf{y}}$, followed by calculating the TF-IDF (Salton and McGill, 1986) value for each atom and com-

pound. Incremental K-means considers each example of D_s^l as a fixed clustering centroids and estimates K_q new cluster centroids. For each of the K_q new clusters, we select one example closest to the centroid.

Such selection strategy is reformulated as selecting K_q examples with the highest acquisition scores one by one at each iteration:

$$\phi_s(\mathbf{e}_s) = \begin{cases} -\|f(\mathbf{y}) - \mathbf{c}_{m(\mathbf{y})}\|^2 & \text{if } m(\mathbf{y}) \notin \bigcup_{\mathbf{e}_s \in D_s^l} m(\mathbf{y}) \\ -\infty & \text{Otherwise} \end{cases} \quad (1)$$

where $f(\cdot)$ is the feature function, $m(\cdot)$ maps each LF into its cluster id and \mathbf{c}_i is the center embedding of the cluster i . As in Algo. 1, when a new example is chosen, none of its cluster mates will be selected again. The incremental mechanism guarantees the newly selected examples are structurally different from those chosen in previous iterations. Since we use batch-wise AL, we just estimate the clusters once per iteration to save the estimation cost.

Lexical Choice Diversity (LCD). LCD aims to select examples whose LFs are aligned with the most diversified lexicons. We achieve this goal by choosing the example maximizing the average entropy of the conditional probability $p(x_s|a)$:

$$\phi_c(\mathbf{e}_s) = -\frac{1}{|A_{\mathbf{y}}|} \sum_{a \in A_{\mathbf{y}}} \lambda_a \sum_{x_s \in V_s} p(x_s|a) \log p(x_s|a) \quad (2)$$

$$\lambda_a = \begin{cases} 1 & \text{if } a \in A_l \\ \beta & \text{Otherwise, } 0 \leq \beta < 1 \end{cases} \quad (3)$$

where a is the atom/compound, $A_{\mathbf{y}}$ is the set of all atoms/compounds extracted from \mathbf{y} , V_s is the vocabulary of the source language, A_l is the set of atoms/compounds in all selected examples until now, and $p(x_s|a)$ is constructed by counting the co-occurrence of a and x_s in the source-language training set. To prevent selecting structurally similar LFs, the score of each selected atom or compound is penalized by a decay weight β .

Our intuition has two premises. First, the parser trained on example pairs whose LFs have more lexical choices generalizes better. Second, LFs with more source-language lexical choices will have more target-language lexical choices as well.

LF Structure and Lexical Choice Diversity (LFS-LC-D). We eventually aggregate the two terms to get their joint benefits, $\phi(\mathbf{x}_s, \mathbf{y}) = \alpha\phi_s(\mathbf{x}_s, \mathbf{y}) + \phi_c(\mathbf{x}_s, \mathbf{y})$, where α is the weight that balances the

importance of two terms. We normalize the two terms using quantile normalization (Bolstad et al., 2003) in order to conveniently tune α .

Hyperparameter Tuning. Because our setup is unique, we can not copy hyperparameters from existing works. The other efforts (Duong et al., 2018) get hyperparameters by evaluating algorithms on seed annotated data. To tune our AL hyperparameters, α and β , a straightforward practice using seed data is to sample multiple sets of examples from the source-language data, the target-language counterparts of which are in seed data, by varying different hyperparameter configurations and reveal their translations in the target language, respectively. The parser is trained on different bilingual datasets and evaluated on the *target-side* dev set. We use the one, which results in the best parsing performance, as the experimental configuration.

Such a method still requires translation costs on the seed data. We assume if the selected examples help the parser generalize well in parsing source-language utterances, their translations should benefit the parser in parsing target languages. Given this assumption, we propose a *novel* cost-free hyperparameter tuning approach. First, we acquire different sets of source-language samples by varying hyperparameters. Then, we train the parser on each subset and evaluate the parser on the *source-side* dev set. Finally, we use the hyperparameters with the best dev set performance.

4 Experiments

Datasets. We experiment with multilingual GEOQUERY and NLMAP. GEOQUERY utterances are in English (EN), German (DE), Thai (TH), and Greek (EL); NLMAP utterances are in English and German. Neither corpora include a development set, so we use 20% of the training sets of GEOQUERY and NLMAP in each language as the development sets for tuning the hyperparameters. To simulate AL process, we consider English as the resource-rich language and others as the target languages. After the examples are selected from the English datasets, we reveal their translations in the target languages and add them to the training sets. **AL Setting.** We perform six iterations, accumulatively selecting 1%, 2%, 4%, 8%, 16% and 32% of examples from English GEOQUERY and NLMAP. **Baselines.** We compare four selection baselines and the oracle setting: i) *Random* picks English utterances randomly to be translated, ii) *S2S*

(*FW*) (Duong et al., 2018) selects examples with the lowest parser confidence on their LFs, iii) *CSSE* (Hu and Neubig, 2021) selects the most representative and diversified utterances for machine translation, iv) *Max Compound* (Oren et al., 2021) selects examples that diversify the atoms and compounds in the LFs, v) *ORACLE* trains the parser on the full bilingual training set.

Evaluation. We adopt the exact match accuracy of LFs for all the experiments. We only report the parser accuracy on the target languages as we found the influence of new data is negligible to the parser accuracy on English data (See Appendix A.2).

Base Parser. We employ BERT-LSTM (Moradshahi et al., 2020) as our multilingual parser. Please see Appendix A.1 for its detailed description.

4.1 Hyperparameter Tuning

Table 1 displays the experiment results with the hyperparameters tuned using only English data (EN) and the hyperparameters tuned using seed data on i) English data plus a small subset (10% of train data plus development data) in the target language (EN + 10%), ii) the full bilingual data (EN + full), iii) the same dataset in a different pair of languages from our experiment languages (Diff Lang), iv) a different dataset in the same languages as our experiment (Diff Data).

	GEOQUERY			NLMAP
	DE	TH	EL	DE
EN (Ours)	73.86	74.57	77.57	69.43
EN + 10%	73.86	74.57	77.57	69.02
EN + full	73.86	74.57	77.14	69.43
Diff Lang	73.86	74.04	77.57	-
Diff Data	71.36	-	-	67.72

Table 1: The parsing accuracies on GEOQUERY and NLMAP test sets in various target languages after translating 16% of the English examples selected by LFS-LC-D with the optimal hyperparameters obtained by different tuning approaches.

From Table 1, we can see our approach takes significantly fewer annotation resources than others to find optimum hyperparameters. Adding more target-language data does not help obtain better hyperparameters, validating our assumption that English data is enough for LFS-LC-D to obtain good hyperparameters. Surprisingly, the hyperparameters tuned on a different language pair do not significantly worsen the selection choices. However, tuning hyperparameters from other datasets results in inferior parsing performance, which is anticipated as different datasets include different

LFs, but the performance of LFS-LC-D is closely related to the LF structures.

4.2 Active Learning Results

Effectiveness of AL-MSP. Fig. 2 shows that only a small amount of target-language data significantly improves the parsing performance over the zero-shot performance. For example, merely 1% of training data improves the parsing accuracies by up to 13%, 12%, 15% and 6% on GEOQUERY(DE), GEOQUERY(TH), GEOQUERY(EL) and NLMAP(DE), respectively. With the best selection approach LFS-LC-D, translating 32% of instances yields parsing accuracies on multilingual GEOQUERY and NLMAP that are comparable to translating the whole dataset, with an accuracy gap of less than 5%, showing that our AL-MSP might greatly minimize the translation effort.

Effectiveness of LFS-LC-D. LFS-LC-D consistently outperforms alternative baselines on both multilingual datasets when the sampling rate is lower than 32%. In contrast, S2S(FW) consistently yields worse parser performance than the other baselines. Our inspection reveals that the parser is confident in instances with similar LFs. MAX COMPOUND diversifies LF structures as LFS-LC-D, however it does not perform well on GEOQUERY(TH). CSSE diversifies utterances yet performs poorly. We hypothesize that diversifying LF structures is more advantageous to the semantic parser than diversifying utterances. RANDOM also performs consistently across all settings but at a lesser level than LFS-LC-D.

Individual Terms of LFS-LC-D. We also inspect each individual term, LFS and LCD, in LFS-LC-D. As in Fig. 3, both terms have overall lower performance than LFS-LC-D, indicating the combination of two terms is necessary. Specifically, LFS performs poorly on NLMAP at the low sampling region. We inspect that NLMAP includes 5x more compounds than GEOQUERY. Therefore, it is difficult for the small number of chosen examples to encompass all types of compounds. LCD performs poorly on GEOQUERY(TH). We notice that Thai is an analytic language linguistically distinct from English, German or Greek, so the entropy values of the probability $p(x_s|a)$ over lexicons in Thai ($p=0.03$) is statistically more different to the ones over English than German ($p=5.80e-30$), and Greek ($p=1.41e-30$)¹. Overall, the two terms could

¹We use the Student’s t-test (Demšar, 2006).

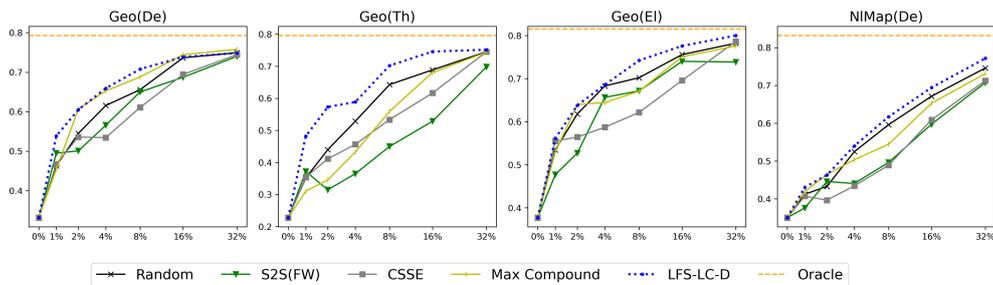


Figure 2: The parsing accuracies at different iterations on the test sets of GEOQUERY and NLMAP in German (De), Thai (Th), and Greek (El) using different selection approaches. All experiments are run five times with different seeds.

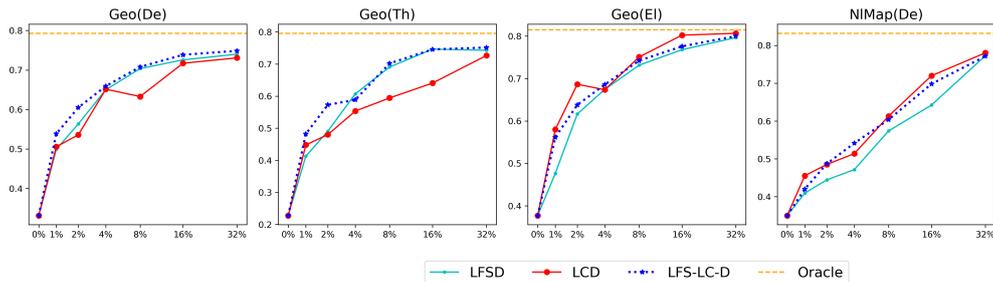


Figure 3: The parsing accuracies at different iterations on the test sets of GEOQUERY and NLMAP in German (De), Thai (Th) and Greek (El) using LFS, LCD, and LFS-LC-D, respectively.

benefit each other, so LFS-LC-D performs steadily across different settings.

Comparison with Machine Translation. We also evaluate the parsers that utilize machine translation services. The parsers are trained on a combination of English data and data translated into the target language by Google Translation (Wu et al., 2016). The accuracy of parsers evaluated on test sets of Geo(De), Geo(Th), Geo(EI), and NIMap(De) was 49%, 58%, 75%, and 75%, respectively. These parsing accuracies are significantly lower than those attained by parsers trained on data provided through human translation, which achieved 80%, 80%, 81%, and 83%, respectively. This suggests that the performance of the parser is tightly correlated to the quality of the employed machine translation system. Clearly, human translation delivers a greater output quality compared to machine translation. In addition, the results reveal that parsers employing AL methods can easily outperform those employing machine translation methods, particularly when the sampling rate for AL is more than 1%, 4%, 8%, and 32% in the four data settings.

5 Conclusion

We conducted the first in-depth empirical study to investigate active learning for multilingual seman-

tic parsing. In addition, we proposed a method to select examples that maximize MSP performance and a cost-free hyperparameter tuning method. Our experiments showed that our method with the proper hyperparameters selects better examples than the other baselines. Our AL procedure with the ideal example selection significantly reduced the translation effort for the data collection of MSP.

Limitations

To reduce annotation costs, existing data collection methods for MSP also utilize machine translation (Moradshahi et al., 2020). Despite the generally lower quality of machine-generated translations compared to human translations, the cost of machine translation services is notably more economical. Our study pioneers the investigation into the feasibility of reducing annotation costs by manually translating only selective portions of the utterance pool. In our work, we provide an initial evaluation of parsers using machine translation versus those using AL methods. Further research is necessary to thoroughly compare these cost-reduction approaches, highlighting their respective advantages and limitations, which we intend to pursue as part of our future work.

References

- Benjamin M Bolstad, Rafael A Irizarry, Magnus Åstrand, and Terence P. Speed. 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193.
- Dataiku Lab. 2022. [Cardinal](#).
- Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.
- Long Duong, Hadi Afshar, Dominique Estival, Glen Pink, Philip R Cohen, and Mark Johnson. 2017. Multilingual semantic parsing and code-switching. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 379–389.
- Long Duong, Hadi Afshar, Dominique Estival, Glen Pink, Philip R Cohen, and Mark Johnson. 2018. Active learning for deep semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 43–48.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Junjie Hu and Graham Neubig. 2021. Phrase-level active learning for neural machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1087–1099.
- David D Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier.
- Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021a. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962.
- Zhuang Li, Lizhen Qu, and Gholamreza Haffari. 2021b. Total recall: a customized continual learning method for neural semantic parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3816–3831.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Mehrad Moradshahi, Giovanni Campagna, Sina Semnani, Silei Xu, and Monica Lam. 2020. Localizing open-ontology qa semantic parsers in a day using machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5970–5983.
- Inbar Oren, Jonathan Herzig, and Jonathan Berant. 2021. Finding needles in a haystack: Sampling structurally-diverse training sets from synthetic data for compositional generalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10793–10809.
- Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.
- Priyanka Sen and Emine Yilmaz. 2020. Uncertainty and traffic-aware active learning for semantic parsing. In *Proceedings of the First Workshop on Interactive and Executable Semantic Parsing*, pages 12–17.
- Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938.
- Raymond Hendy Susanto and Wei Lu. 2017. Neural architectures for multilingual semantic parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–44.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

A Appendix

A.1 Implementation Details

BERT-LSTM BERT-LSTM is a Sequence-to-Sequence model (Sutskever et al., 2014) with the XLM-RoBERTa-base (Liu et al., 2019) as its encoder and an LSTM (Hochreiter and Schmidhuber, 1997) as its decoder.

Hyperparameters of the Parsers We tune the hyperparameters of BERT-LSTM on English data. For a fair comparison, we fix the hyperparameters of the parser while evaluating the active learning methods. Specifically, we set the learning rate to 0.001, batch size to 128, LSTM decoder layers to 2, embedding size for the LF token to 256, and epochs to 240 and 120 for the training on GEOQUERY and NLMAP, respectively.

Hyperparameters of AL For tuning the hyperparameters of the active learning method, we grid search the decay weight β in 0, 0.25, 0.5, 0.75 and the weight balance rate α in 0.25, 0.5, 0.75, 1. The optimal hyperparameters are 0.75 and 0.75 for all language pairs of GEOQUERY and 0.75 and 0.25 for multilingual NLMAP.

In the Diff Lang setting, we assume we can access the data in a language pair other than the experimental one. For selecting English utterances to be translated into German, Thai, and Greek, we tune the hyperparameters on the data of En-Th, En-EL, and En-De pairs, respectively.

In the Diff Data setting, we assume we can access the data in the same language pair as our experimental one but in a different domain with a different type of LF. For selecting English utterances in GEOQUERY for translation, we tune the hyperparameters on the bilingual NLMAP. For selecting utterances in NLMAP, we tune the hyperparameters on the GEOQUERY in the language pair, En-De.

A.2 Parser Accuracies on English Test Sets

As in Fig. 4, training the parser on the data in the target language does not significantly influence the parser’s performance on the English test sets. Therefore, in Sec. 4, we only report the experimental results on the test sets in the target languages.

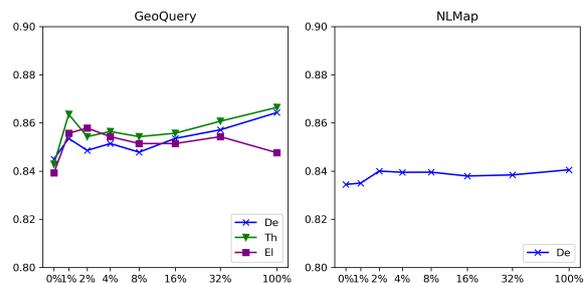


Figure 4: The parsing accuracies at different iterations on the English test sets of GEOQUERY and NLMAP after selecting data in German (De), Thai (Th) and Greek (El) using LFS-LC-D, respectively.

Joint Word and Morpheme Segmentation with Bayesian Non-Parametric Models

Shu Okabe
Univ. Paris-Saclay & CNRS
LISN, rue du Belvédère
91405 Orsay, France
shu.okabe@liscn.fr

François Yvon
Univ. Paris-Saclay & CNRS
LISN, rue du Belvédère
91405 Orsay, France
francois.yvon@liscn.fr

Abstract

Language documentation often requires segmenting transcriptions of utterances collected on the field into words and morphemes. While these two tasks are typically performed in succession, we study here Bayesian models for simultaneously segmenting utterances at these two levels. Our aim is twofold: (a) to study the effect of explicitly introducing a hierarchy of units in joint segmentation models; (b) to further assess whether these two levels can be better identified through weak supervision. For this, we first consider a deterministic coupling between independent models; then design and evaluate hierarchical Bayesian models. Experiments with two under-resourced languages (Japhug and Tsez) allow us to better understand the value of various types of weak supervision. In our analysis, we use these results to revisit the distributional hypotheses behind Bayesian segmentation models and evaluate their validity for language documentation data.

1 Introduction

In computational language documentation, unsupervised segmentation into words or morphemes¹ aims to identify boundaries between units in sequences of symbols, typically corresponding to a phonetic or orthographic transcription of an unsegmented utterance. These tasks are fundamental, as they help to identify and analyse possible dictionary entries. There is a long tradition to handle these tasks with generative probabilistic models (Brent, 1999; Venkataraman, 2001) initially designed to model the acquisition of speech by children. The most successful approaches to date rely on non-parametric Bayesian models based on Dirichlet Processes (Goldwater et al., 2006, 2009; Godard et al., 2016) and Adaptor Grammars (Johnson et al., 2007; Eskander et al., 2016; Godard et al., 2018;

¹In this paper, our position regarding the notions of ‘words’ and ‘morphemes’ is entirely empirical, as we mainly try to reproduce annotations performed by field linguists.

Eskander et al., 2019). An interesting property of these generative models is their ability to accommodate existing resources (e.g. partial list of word types) (Sirts and Goldwater, 2013; Ruokolainen et al., 2016), which are often available in actual documentation settings (Bird, 2020).

We study here a scenario where we automatically generate a two-level segmentation,² identifying simultaneously both word and morpheme boundaries. Figure 1 illustrates such a segmentation, where whitespaces separate words, while morphemes are joined with hyphens. Our main task is thus to identify two types of boundaries from the unsegmented stream of symbols (first line) to form the two-level segmented sentence (penultimate line). In this work, we only focus on *surface segmentation* (e.g. eat+ing) as opposed to *canonical segmentation* (e.g. hike+ing for hiking) (Cotterell et al., 2016).

Segmentation	Sentence
Unsegmented	uɪzokɪatɕɪpuɪwɪsɪmtoa
Word	uɪzo kɪɪ atɕɪɪ puɪwɪsɪmtoa
Morpheme	uɪzo-kɪɪ-a-tɕɪɪ-puɪ-wɪ-sɪɪ-mto-a
Two-level	uɪzo kɪɪ a-tɕɪɪ puɪ-wɪ-sɪɪ-mto-a
Translation	He let me see my son.

Figure 1: Example of two segmentation levels in Japhug: words are separated by whitespaces (‘ ’) and morphemes by hyphens (‘-’). Extract from (Jacques, 2021)

The motivation for this task is two-fold: (a) to evaluate our ability to obtain annotations such as Figure 1 in an unsupervised way; (b) to see how much the two-level model can disambiguate word from morpheme boundaries, thus improving word segmentations. Note that in actual documentation settings, the annotation of morpheme boundaries

²This ‘two-level segmentation’ is unrelated to the ‘two-level morphology’ (Koskenniemi, 1983), which describes the association between surface forms and underlying representations using the formalism of extended rational expressions.

is usually performed on utterances that are already segmented into words, hence the need to optimise this step.

A baseline for this task is a two-pass approach: first, identify putative word boundaries, then iterate the segmentation procedure on the corresponding set of word types. As we discuss below, unsupervised word segmentation procedures tend to generate units that are often halfway between morphemes and words (see e.g. (Goldwater et al., 2009) or (Godard et al., 2016) who report oversegmentation for words). This means that the first pass often delivers units that are too short and inadequate for the latter processing step. This remains true even with partial supervision information at the word level (Okabe et al., 2022).

We therefore study models that explicitly distinguish between words and morphemes, considering both the fully unsupervised and the minimally supervised settings. The research questions that we address are the following:

- RQ1: Bayesian segmentation models identify units based solely on distributional properties, identifying units that are often in between words and morphemes. Can we improve both segmentations through an explicit modelling of these two levels?
- RQ2: a simple baseline is to first segment sentences into words, then to segment each *word type*³ identified in the first step into morphemes. A second question is how much a single joint segmentation model can mitigate the error propagation of this two-step baseline.
- RQ3: there are multiple ways to implement and supervise joint segmentation models, an important distinction being between linear (flat) and hierarchical segmentation models. A third question relates to the strengths and weaknesses of these approaches, both in the presence and absence of supervision.
- RQ4: Bayesian segmentation models primarily rely on distributional properties of characters in morphemes and words, and embed specific assumptions regarding these distributions. We last question the validity of these assumptions in a low-resource language documentation context.

³Types denote unique words, as opposed to *tokens*, which encompass all running occurrences of types in a corpus.

More generally, our main goal in this study is to assess *whether statistical cues alone are sufficient to identify two distinct segmentation levels*. To answer this question, we analyse several simple joint segmentation models introduced in Section 2 and experiment with two under-resourced languages, briefly presented in Section 3. Our main results and analyses are in Section 4. From a practical perspective, our objective is *not* to devise directly-usable models for field work but to observe the effect of introducing a subword level of segmentation in Bayesian non-parametric models, especially in very low-resource situations as in language documentation: will it improve the (original) word-level segmentation quality? How can additional resources help?

2 Segmentation models

2.1 One-level segmentation

For this work, we use our own Python implementation⁴ of the unigram version of Goldwater et al.’s (2009) model: dpseg. This model relies on Dirichlet Processes to evaluate the probability of a word sequence, as we briefly recall below. In dpseg, the probability of a new occurrence w , based on the observed past words, is expressed through Equation (1) where w denotes a word $w = c_1 \dots c_L$ comprising L characters:

$$P(w|h^-; \alpha) = \frac{n_w^{(h^-)} + \alpha P_0(w|h^-)}{n^- + \alpha}. \quad (1)$$

Here, h^- denotes the rest of the text (w excluded), $n_w^{(h^-)}$ the frequency of word w in the text, and n^- the total number of words. α is the concentration parameter and P_0 , the *base distribution*, is defined by Equation (2):

$$P_0(w) = p_{\#}(1 - p_{\#})^{(L-1)} * \prod_{l=1}^L P_c(c_l), \quad (2)$$

with $p_{\#}$ the probability to terminate a word and P_c a distribution over the set of characters, assumed uniform in the dpseg model.

Observing an unsegmented character string $c_1 \dots c_T$, word segmentation can be formalised with a latent variable model, introducing unobserved boundary variables $b_1 \dots b_T$, where $b_t = 1$ (resp. $b_t = 0$) respectively denotes presence or absence of a boundary after c_t . The inference is typi-

⁴Available at <https://github.com/shuokabe/pyseg>.

cally performed with Gibbs sampling, using Equation (1) to iteratively resample the latent boundary variables values. To speed up convergence, Goldwater et al. (2009) additionally use simulated annealing.

We chose to explore dpseg over alternative segmentation models such as SentencePiece (Kudo and Richardson, 2018) or Morfessor (Creutz and Lagus, 2002; Smit et al., 2014) because of its better performance in similar language documentation contexts. It is also well suited to small data conditions and enables weak supervision (Okabe et al., 2022). Furthermore, preliminary experiments showed no major difference between using a Dirichlet process (DP), as we do, and a variant based on a Pitman-Yor process (PYP), known to better capture the underlying power-law distribution. Overall, we believe that using more sophisticated variants or faster implementations of dpseg would not substantially alter our main observations.

2.2 Pipeline model: two-step segmentation

We now turn to models computing a segmentation in words and morphemes. Our baseline two-level model combines in a pipeline two dpseg models: the first inputs unsegmented text and yields a word-level segmentation. The *word types* in this segmentation are then collected and processed by a second dpseg to get the morpheme-level segmentation. By design, in this approach, a word type is always associated with a unique morphological analysis.⁵

2.3 Flat segmentations with coupling

Two-level segmentation can also be formalised with latent variables, using two sets of variables, denoted as $\{b_1^w \dots b_T^w\}$ (resp. $\{b_1^m \dots b_T^m\}$) for word (resp. morpheme) boundaries. Obviously, using the same dpseg model to independently sample these variables will produce indistinguishable segmentations. It is, however, possible to get two-level segmentations by introducing interactions between these two models, so that the values of variables b_t^w and b_t^m are no longer independent. Deterministic interactions can be introduced in two ways which both ensure that word and morpheme segmentation hypotheses always remain consistent: by imposing either i) that word boundaries also correspond to morpheme boundaries, or ii) that morpheme internal positions are also considered word internal.

⁵This hypothesis corresponds to what we observe in our corpora, where only a few dozen words occur with more than one segmentation in morphemes.

In strategy i), we first sample boundary variables for words and then for morphemes, yielding the parallel-w approach. If a word boundary is detected ($b_t^w = 1$), then we deterministically identify a morpheme boundary at that position ($b_t^m = 1$). Otherwise ($b_t^w = 0$), we sample the value for b_t^m as usual. The net effect is to make morpheme boundaries more likely than in an independent model and generate shorter units at the morpheme level; no change is expected at the word level. In strategy ii), denoted parallel-m, morpheme variables are sampled first: if a boundary is detected ($b_t^m = 1$), an extra sample decides the value of b_t^w ; else, we readily assign $b_t^w = 0$. Here, the effect is reversed and makes word boundaries less likely, forcing the word model to generate longer units; the morpheme-level segmentation remains unchanged.

2.4 Hierarchical segmentations

Inspired by (Mochihashi et al., 2009), we also implement hierarchical segmentation models for the two-level segmentation task. These models aim to explicitly represent the structured aspect of the double segmentation process. Here, the word model is nearly identical to the basic version of dpseg, with a change in the base distribution P_0 of Equation (1). The character model (P_c) is replaced by a second non-parametric model for morphemes (hence the hierarchical nature of the model), also based on dpseg. This morpheme model has a base distribution that is, as for the original dpseg, a unigram character model.

Considering a word w (of length L) made of K morphemes, $w = m_1 \dots m_K$, by analogy to Equation (2), P_0 is therefore changed to:

$$P_0^w(w|h^-) = p_{\#}(1-p_{\#})^{(L-1)} * \prod_{k=1}^K P^m(m_k|h^-), \quad (3)$$

where $P^m(m_k)$ is the probability of morpheme m_k according to the morpheme model (the standard dpseg model), which is written as follows:

$$P^m(m_k|h^-; \alpha^m) = \frac{n_{m_k}^{(h^-)} + \alpha^m P_0^m(m_k)}{n_{\bar{m}} + \alpha_m}, \quad (4)$$

where α^m and P_0^m are, respectively, the concentration parameter and the base distribution for morphemes—the latter being a uniform character model. Sampling in this model is implemented as follows: each time a new word is hypothesised, a morpheme segmentation is obtained from the morpheme model; for words that are actually retained,

this segmentation is recorded and used for further occurrences of the same word form. A basic version of this approach (denoted *hier-type*) thus samples boundary variables for morphemes only once for each word type.⁶ Two variants are considered: in the *hier-iter* model, morpheme boundaries are iteratively resampled for all existing word types every k iterations of the word-level Gibbs sampler; in *hier-final*, this process is only performed once after convergence of the word model, to make a fair comparison with the pipeline model of Section 2.2. As in the pipeline model, these approaches ensure that all occurrences of a given word type will have the same morphological decomposition.

2.5 Unsupervised Adaptor Grammars

Another hierarchical baseline is based on the Adaptor Grammar (AG) model of (Johnson et al., 2007). This is a strong unsupervised word segmentation model that can also capture morphological structure to some extent. We use the *colloc* grammar in (Johnson, 2008), which considers the following levels: a Sentence is made of Collocations, which are made of Words, themselves composed of Characters. In the same manner as (Johnson, 2008, Section 3.2), we considered the Collocation tier to correspond to words and the Word tier to morphemes.⁷

2.6 Weak supervision

Following (Okabe et al., 2022), we further consider two types of realistically available resources that can supervise the segmentation process. The first takes the form of a small number of segmented sentences (e.g. from previously annotated texts), where the corresponding boundary variables are observed. During Gibbs sampling, we skip these positions and simply use the observed values (0 or 1). This type of supervision, which gives information at the *token* level, is denoted *sentence*.

A second type of resource corresponds to lists of lexical units (words and/or morphemes). We use them to replace in P_0 the uniform model with a bigram model, thus increasing the likelihood of known units. This supervision method which uses knowledge about *types* is denoted *dictionary*.

Observed word segmentation or the word dictionaries will be used to compute $P^w(w|h^-, \alpha)$

⁶This is slightly more subtle, as the same word can be created then deleted during the Gibbs sampling iterations.

⁷Appendix C details the hyperparameter values.

(Equation (1)). Likewise, observed morpheme boundaries or morpheme lists will be taken into account at the morpheme level (e.g. in Equation (4) for the hierarchical model). In all our experiments, we assume that weak supervision is available simultaneously at the word and morpheme levels.

2.7 Full supervision

An even more favourable situation is when boundaries are fully observed for a sufficiently large set of sentences, warranting the use of supervised learning techniques such as Conditional Random Fields (Lafferty et al., 2001). This situation is studied notably by (Moeller and Hulden, 2018; Kann et al., 2018). Our experiments with this setting show that this procedure is sample efficient. It is, however, also subject to the same confusion between word and morpheme boundaries and does not significantly outperform the weak supervision setting. Full results are reported in Appendix D.

3 Experimental protocol and material

3.1 Evaluation metrics

Following (Goldwater et al., 2006), the segmentation outputs are evaluated with F-scores on the two levels of segmentation (word and morpheme) at three tiers: BF at the boundary level obtained by comparing predicted and actual boundary values (0 or 1), WF for the token level, which focuses on the correspondence between each unit in the sentences, and LF for the lexicon level, counting the matches between unit types collected on the whole text. For finer analyses, we also report the precision and recall for all three levels in Appendix D.

In addition, some basic statistics regarding the texts will be presented for both segmentation levels. N_{utt} , N_{type} , and N_{token} respectively denote the number of utterances, unit types, and tokens in the text. We also report the inferred average token (WL) and type (TL) lengths.

3.2 Linguistic material

This work studies two low-resource languages: Japhug and Tsez.

Japhug is a Sino-Tibetan language from the Gyalrong family spoken in the Sichuan province in China. It notably has a rich morphology for both nouns and verbs. For example, verbs can use several prefixes to express tense or aspect features on top of suffixes. Japhug is currently being documented: recordings, annotated corpora, and dictio-

naries are available in Pangloss.⁸ Jacques (2021) comprehensively describes the language. The corpus is composed of all the Japhug examples from the L^AT_EX source files of this grammar book.⁹ The extraction of those sentences is made easier thanks to the `\gll` command before the Japhug sentences.

Tsez is a Caucasian language part of the Nakh-Daghestanian language family, spoken in the Republic of Dagestan in Russia. It is officially an unwritten language, transcribed and transliterated through the Avar writing system (Comrie and Polinsky, forthcoming). Nouns and verbs are mainly inflected with a variety of combined suffixes. Moreover, Tsez features a set of clitics that are merged with the words. The latest grammar is currently in the process of being published. The only substantial dictionary of the language contains around 7,500 entries. The Tsez corpus contains sentences from the Tsez Annotated Corpus of (Abdulaev and Abdulaev, 2010), used in (Zhao et al., 2020) to study the generation of interlinear glosses.

language segment	Japhug		Tsez	
	word	morph.	word	morph.
N_{utt}	3628	3628	2000	2000
WL	4.73	2.90	5.61	2.81
TL	7.30	5.41	6.93	5.21
N_{type}	6739	2731	5732	1603
N_{token}	28579	46632	20153	40229
$N_{super.}$	664	493	867	455

Table 1: Statistics for the Japhug and Tsez corpora. Both are segmented into words and morphemes (morph.).

Table 1 describes the two language corpora, reporting statistics at the two segmentation levels. Japhug word types have an average number of 2.48 morphemes, while in Tsez, that value is 2.37.

For weak supervision, the first 200 sentences of each corpus are selected as training material, used as is for boundary supervision (sentence method) or as a list of unique (word or morpheme) types for lexical supervision (dictionary method). $N_{super.}$ above summarises the number of supervision units.

3.3 Experimental settings

In our experiments, the results are obtained after 20,000 iterations of Gibbs sampling, with 10 in-

crements of simulated annealing, for quicker convergence as detailed in (Goldwater et al., 2009). The last iteration of a run returns the final boundary prediction that is considered to be the model output. To account for the variability of the sampler, we report below the average of three runs. We find that this segmentation procedure is stable with an average standard deviation of less than 1 for all metrics.

We use the default values of the base dpseg for hyperparameters: $p_{\#} = 0.5$ and $\alpha = \alpha^m = 20$. We set the same initial value of the concentration parameter for the two levels in both categories of model. Following Teh (2006) and Mochihashi et al. (2009), the two concentration parameters, which both have a Gamma posterior distribution, are re-sampled after each iteration on the corpus—thus, upon convergence, we observe $\alpha \neq \alpha^m$.

For the hierarchical models, hier-final re-segments word types into morphemes with 1,000 iterations of Gibbs sampling, while hier-iter carries out 5 iterations of morphological segmentation every 100 iterations of word segmentation.

4 Experimental results

4.1 RQ1: unsupervised two-level models

Table 2 reports segmentation results for the Japhug corpus. The corresponding results for Tsez are in Table 6 in Appendix D. As briefly stated in the introduction, the one-level dpseg segments into units that are too short for words (cf. average unit lengths WL and TL) and seems to segment units that are closer to morphemes, with higher morpheme F-scores for all three evaluation tiers. This motivated our work on two-level segmentation models, which, contrarily to the basic dpseg, make a distinction between the two types of boundaries. The more sophisticated AG model shows a similar trend, outputting words and morphemes that are too short, insufficiently diverse (low LF), and result in too many tokens (excessive N_{token}).

The pipeline approach only differs from the one-level dpseg at the morpheme level, where we see worse F-scores, with a massive drop in LF score. For the ‘parallel’ models, the expected improvements are observed: better morpheme boundaries for parallel-w, better word boundaries for parallel-m. However, these two models deliver units that remain quite close in average length, and the F-score improvements remain rather limited in magnitude. In those experiments, the hierarchical

⁸<https://pangloss.cnrs.fr/corpus/Japhug>.

⁹<https://github.com/langsci/295/>.

model level	AG		dpseg*		pipeline		parallel-w		parallel-m		hier-type		-final	hier-iter	
	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	morph.	word	morph.
BF	71.0	83.4	73.1	81.0	73.1	80.6	73.3	83.6	73.2	80.8	74.7	62.5	82.3	73.5	81.4
WF	45.8	62.5	46.2	55.1	46.2	57.8	46.5	61.3	46.0	54.7	48.7	24.5	60.9	47.2	59.1
LF	31.1	28.9	20.4	41.4	20.4	17.5	20.6	40.5	23.8	41.4	28.8	17.0	23.4	31.7	24.7
WL	4.72	2.51		3.34	3.34	2.13	3.34	2.98	3.73	3.35	3.93	1.65	2.32	4.29	2.37
TL	6.60	3.27		4.22	4.22	2.64	4.23	3.99	4.77	4.21	4.78	2.83	2.87	5.12	2.88
N_{type}	5582	1113		2260	2260	694	2257	1834	2921	2281	3806	1013	911	4925	956
N_{token}	28.6k	53.9k		40.5k	40.5k	63.4k	40.5k	45.4k	36.3k	40.4k	34.4k	82.1k	58.2k	31.5k	57.0k

Table 2: Results on the Japhug corpus for unsupervised one-level (*) and two-level dpseg models. The reference contains 6,739 words and 2,731 morphemes (N_{type}). **Bold** numbers represent the best result per metrics.

models make a stronger distinction between the two types of units, yielding well-differentiated average lengths (WL and TL). Overall, almost all two-level models but the simple-minded pipeline improve the baseline scores for at least one level of segmentation, with the unsupervised parallel-w flat model delivering the best results on average.

While our answer to RQ1 is positive, we note that the score differences between approaches are often small and that all models keep oversegmenting words, leading to a too low number of word types and yielding poor LF scores. The same trend is observed for the hierarchical models at the morpheme level: they find too few morphemes (cf. N_{type}) and result in poor type-level scores.

4.2 RQ2: error propagation

Compared to the baseline, the unsupervised pipeline approach obtains poor LF score at the morpheme level (Table 2). As the two approaches have almost identical BF and WF scores, this means that pipeline performance is mostly due to its ability to detect frequent morphemes at the expense of rarer ones. This is also reflected by the very small number of morpheme types found by this model.

This is because the pipeline model uses the word types computed by the regular dpseg to detect morpheme boundaries. As this first step obtains poor results ($WF \approx 20$), cascading errors accumulate. Wrong detections at the word level are thus counted twice: once at the word level, once at the morpheme level. The use of joint models slightly remedies this state of play, yielding improvements in the word dictionary, which then turn into improved morpheme dictionaries. This allows us to answer RQ2 positively, even though the recall for morpheme types still remains far from satisfactory. To progress on that front, the surest way seems to improve word segmentation, if only because many word types are made of one single morpheme.

4.3 RQ3: flat and hierarchical models

This section compares the flat (parallel) and hierarchical models, first analysing the differences between variants of the same family, before comparing these two approaches.¹⁰

Parallel models As explained in § 2.3, each ‘parallel’ model only improves the baseline for one type of unit: morpheme boundaries for parallel-w and word boundaries for parallel-m (Table 2). This remains true when using weak supervision. A first comparison is between the parallel models, where we see better scores for parallel-w, which outperforms parallel-m on almost all accounts and all weak supervision settings. In fact, even with the help of supervision, parallel-m obtains lower BF and WF scores at the word level than parallel-w: more word types are generated, the average length is increased, but these hypotheses are often wrong. We do not see the reverse for parallel-w, which generates fewer morphemes: the decrease in recall is almost balanced by the increase in precision, with little negative impact on the morpheme segmentation quality.

Hierarchical models First, for all three F-scores at the morpheme level, in any experimental situation, the hier-type model is consistently worse than the hier-final model, which carries out additional Gibbs sampling steps for the morpheme variables once the word boundaries have stabilised. This model finds longer units (cf. WL) with the additional iterations, which leads to significant improvements (+20 points in WF).

The hier-iter variant achieves a fair trade-off between the boundary and token F-scores on the one hand, and the type F-score on the other hand: this model is better when evaluated at the type level, while hier-final reaches better scores on the other two levels. As the hier-final model

¹⁰Full results in Appendix D.

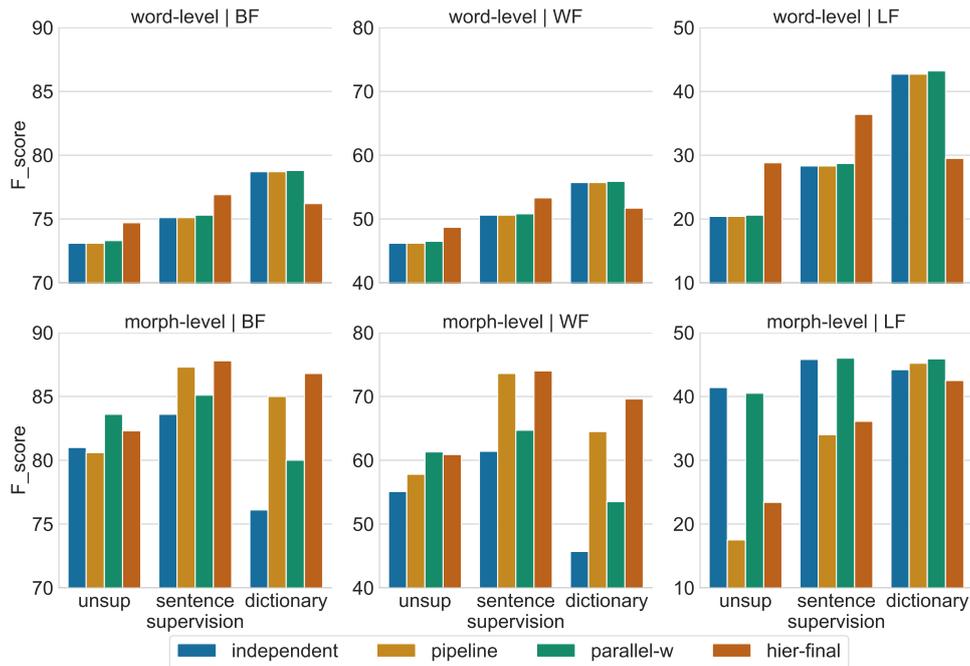


Figure 2: Results in Japhug without supervision on the left, with sentence supervision in the middle, and dictionary supervision on the right of each subplot (row: segmentation level, column: F-score). We use a different y-axis scale for each F-score.

attains a similar but higher aggregated F-score on average, we chose it to represent the hierarchical models for the following sections. This model is also slightly less computationally involved than hier-iter, another reason for choosing it in practical settings.

Comparing two types of models Figure 2 displays the results for the baselines and best performing flat and hierarchical models on Japhug with and without supervision, illustrating the impact of resources. Four models are compared: pipeline, parallel-w, hier-final, and independent. The latter corresponds to *two distinct* dpseg models, one trained for word boundaries, the other for morpheme boundaries, each supervised and evaluated at the corresponding level. It can produce inconsistent segmentations.

By design, independent, pipeline, and parallel-w generate similar word-level segmentations and improve a lot from dictionary supervision. At the morpheme level, the latter model strongly improves its LF score, equally benefiting from both weak supervision strategies.

The hierarchical model has the best results for word-level scores with sentence supervision, whereas, with dictionary supervision, it lags behind the other methods. At the morpheme level, results are less clear. When unsupervised,

hier-final is better than the baselines but worse than parallel-w; however, it always gets a strong boost from supervision, more so than its contenders. In short, sentence is more beneficial for the hierarchical model, while dictionary rather improves the others. Still, these increments remain small; we conclude that weak supervision does not seem to help the models better differentiate the two types of units. Overall, when aggregating F-scores across settings and languages, models rank as follows, from worst to best: independent, pipeline, parallel-w, and hier-final. This answers RQ3.

4.4 RQ4: distributional assumptions

4.4.1 Word distributions in CLD

The parallel and hierarchical models both rely on the same fundamental assumption: the distribution of word tokens in a natural corpus follows a power law, which was a motivation for using Dirichlet processes in (Goldwater et al., 2006). As described in (Goldwater et al., 2011), such distributions derive from the use of a two-stage model: a *generator* which focuses on creating word types (this is P_0 in the dpseg model) and an *adaptor* that produces the ‘rich-get-richer’ effect (Equation (1)).

To check how well our data matches this assumption, in Figure 3, we look at type/token curves, which display the number of word types in texts

of increasing lengths. We deem this ratio to be a reasonable proxy to observe the ‘rich-get-richer’ effect on word types. We compare the Japhug and Tsez texts, their automatic segmentations (‘dp-’), as well as their English translation (‘-en’) (as in (Godard et al., 2016)), with five languages of varying morphological complexity: English, French, Finnish, German, and Turkish. For these, we use the 2020 news data from the Leipzig corpus (Goldhahn et al., 2012), keeping only the first 2,000 sentences for comparison.

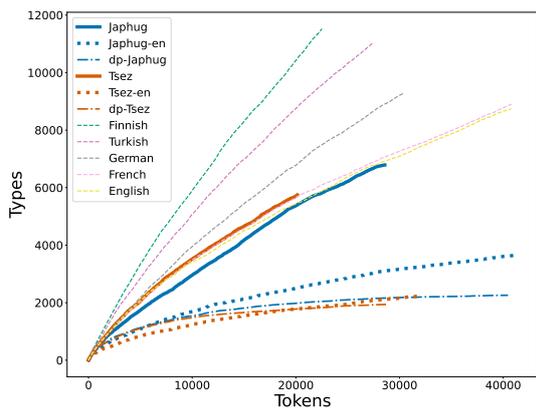


Figure 3: Type-token curves for several languages

We see that the curves for Japhug and Tsez follow the French and English trends, reflecting a lesser lexical variation than for German, Turkish, and Finnish. Looking at their English translations confirms this trend and hints that the number of word types in our corpora does not correctly mirror the actual morphological complexity of these languages. Indeed, corpora collected for language documentation may present distributional biases: sentences are often chosen to illustrate relevant linguistic properties, as in our Japhug corpus extracted from a grammar book. This reduced lexical variety is amplified in automatically segmented texts, where we fail to identify most rare words. For example, 97% of the words occurring only once are not found by the unsupervised hier-final model. See Appendix A for another view of the same phenomena.

4.4.2 Modelling morpheme distributions

Where the parallel and hierarchical versions differ is how they estimate morpheme models: parallel-w assumes a power law of morphemes in running texts, while hier-final assumes it on word types. We see the impact of these assumptions in Figure 4. This graph is based on an esti-

mation of the parameter of the Zipf distributions of words in the Tsez corpus and of morphemes in the Tsez word types (see details in Appendix A). While these parameters strongly depend on the corpus size, they are typically in the range $[-1, -1.2]$ (Baayen, 2001) — the lower value computed for the reference Tsez word distribution again hints at the peculiarity of this distribution, whereas the corresponding parameters for morpheme are in the right ballpark.

All inferred segmentations at the word level behave similarly, with values steeper than for the reference, reflecting the effect of using a power-law model. Once more, we see that supervision is hardly helping. We observe sharper differences at the morpheme level, where the hierarchical model gets much closer to the reference, further boosted by sentence supervision. This is in line with (Virpioja et al., 2011), which notes the better morpheme segmentations obtained when modelling types rather than tokens.

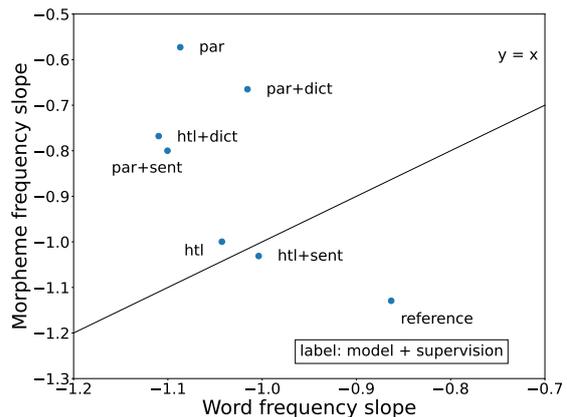


Figure 4: Zipfianity of various segmentations of Tsez. par is based on parallel-w, htl on hier-final.

5 Related work

Word segmentation and morphological segmentation are related tasks; however, both segment only at a single level. We focus here on methods or objectives of approaches comparable to ours.

Word segmentation with Bayesian non-parametric models, on the one hand, benefits from models based on Dirichlet Processes (Goldwater et al., 2006, 2009), extended with the more general Pitman-Yor Processes and a hierarchical structure (Teh, 2006; Mochihashi et al., 2009). In language documentation settings, unsupervised methods are applied (Godard et al., 2016). Morphological segmentation, on the other hand, usually focuses

on the *surface* segmentation of word *types* (Cotterell et al., 2016), with models such as Morfessor (Creutz and Lagus, 2002). Ruokolainen et al. (2016) extensively survey the task for supervised conditions. In low-resource settings, recent works include (Kann et al., 2018; Liu et al., 2021; Moeng et al., 2021).

For both tasks, the Adaptor Grammar (AG) (Johnson et al., 2007), capable of modelling hierarchical structure in sequences with trees, often yields strong results (Johnson, 2008; Eskander et al., 2016; Godard et al., 2018), especially thanks to its flexibility in incorporating minimal supervision. For instance, Sirts and Goldwater (2013) explicitly model words as a compound of one or more morphemes in their AG.

6 Conclusion

By extending a Bayesian non-parametric segmentation model, dpseg, we have proposed two models to simultaneously segment into words and morphemes: one segmenting in parallel and the other in a hierarchical manner. Using corpora of two low-resource, morphologically complex languages, we have observed improved performance with respect to the baselines. These two approaches have been contrasted in various ways, leading us to favour the hierarchical approach when supervision is available. The observed improvements are, however, modest, partly due to modelling assumptions that are not fully matched in our data. It remains that sorting words from morphemes based solely on distributional cues is difficult, if possible at all, even with the supervision considered in this work.

Further studies will need to consider other signals of ‘wordness’. Some can be extracted from the way units combine with their neighbours, using contextual word models; some will require new sources of supervision, e.g. at the phonological level. Another extension will be to distinguish between lexical and grammatical morphemes, which tend to occur and behave differently.

Limitations

The main limitation comes from the use of the unigram dpseg model. Although it has strong and stable performance on the word-level segmentation task, comparable to its bigram version in our settings (Godard et al., 2016), some weaknesses inherent to the unigram assumption appear as in Appendix B. Moreover, such an assumption at the

morpheme level means that, for example, adding a distinction between lexical and grammatical morphemes, as suggested in conclusion, will be of little use since the probability of a morpheme does not affect that of others in the word for unigram models. Nevertheless, in our language documentation setting, we deem this unigram assumption to have a small impact on the overall results due to data size.

For some of our two-level models (pipeline and hierarchical), we also relied on the assumption that a word can only have a single morphological decomposition, as stated in Sections 2.2 and 2.4. Although it may not apply in other situations, this reasonably holds in our two corpora (as briefly explained in footnote 5) since we found 51 word types with several morphological analyses in Japhug and 14 in Tsez.

Besides, our work and observation only rely on two languages. However, the two-level segmentation for very low-resourced languages, as we displayed, needs a reference text segmented with distinct boundaries for words and morphemes for evaluation in particular. Since word segmentation usually focuses on tokens in sentences and morpheme segmentation on word types, texts explicitly segmented in two levels are difficult to obtain, even so of good quality.

Finally, we reckon that our current implementation of the Gibbs sampler is not particularly optimised. For actual deployment, these models should be designed and implemented in a more computationally-efficient way or even another language than Python.

Acknowledgements

This work was partly funded by French ANR and German DFG under grant ANR-19-CE38-0015 (CLD 2025). The authors wish to thank the anonymous reviewers, Laurent Besacier for his feedback, Guillaume Jacques for the Japhug corpus, and Antonios Anastasopoulos for the Tsez corpus.

References

- Asen' K. Abdulaev and I. K. Abdulaev. 2010. *Cezjas fol'klor: (giurus mecrek^o iorno butirno) = Dido (Tsez) folklore = Didojskij (cezskij) fol'klor*. Lotos, Leipzig.
- R. Harald Baayen. 2001. *Word Frequency Distributions*, volume 18 of *Text, Speech and Language Technology*. Springer Netherlands, Dordrecht.

- Steven Bird. 2020. [Decolonising Speech and Language Technology](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3504–3519, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Michael R. Brent. 1999. [An efficient, probabilistically sound algorithm for segmentation and word discovery](#). *Machine Learning*, 34(1-3):71–105.
- Bernard Comrie and Maria Polinsky. forthcoming. Tsez. In Yuri Koryakov, Yury Lander and Timur Maisak (eds.) *The Caucasian Languages*. An International Handbook. Mouton. HSK series.
- Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016. [A joint model of orthography and morphological segmentation](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 664–669, San Diego, California. Association for Computational Linguistics.
- Mathias Creutz and Krista Lagus. 2002. [Unsupervised discovery of morphemes](#). In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.
- Ramy Eskander, Francesca Callejas, Elizabeth Nichols, Judith Klavans, and Smaranda Muresan. 2020. [MorphAGram, evaluation and framework for unsupervised morphological segmentation](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7112–7122, Marseille, France. European Language Resources Association.
- Ramy Eskander, Judith Klavans, and Smaranda Muresan. 2019. [Unsupervised morphological segmentation for low-resource polysynthetic languages](#). In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 189–195, Florence, Italy. Association for Computational Linguistics.
- Ramy Eskander, Owen Rambow, and Tianchun Yang. 2016. [Extending the use of Adaptor Grammars for unsupervised morphological segmentation of unseen languages](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 900–910, Osaka, Japan. The COLING 2016 Organizing Committee.
- Pierre Godard, Gilles Adda, Martine Adda-Decker, Alexandre Allauzen, Laurent Besacier, H el ene Bonneau-Maynard, Guy-No el Kouarata, Kevin L oser, Annie Rialland, and Fran ois Yvon. 2016. [Preliminary Experiments on Unsupervised Word Discovery in Mboshi](#). In *Proceedings of Interspeech 2016*, pages 3539–3543.
- Pierre Godard, Laurent Besacier, Fran ois Yvon, Martine Adda-Decker, Gilles Adda, H el ene Maynard, and Annie Rialland. 2018. [Adaptor Grammars for the linguist: Word segmentation experiments for very low-resource languages](#). In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 32–42, Brussels, Belgium. Association for Computational Linguistics.
- Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. [Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).
- Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. [Contextual dependencies in unsupervised word segmentation](#). In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, Sydney, Australia. Association for Computational Linguistics.
- Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2009. [A Bayesian framework for word segmentation: Exploring the effects of context](#). *Cognition*, 112(1):21–54.
- Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2011. [Producing power-law distributions and damping word frequencies with two-stage language models](#). *Journal of Machine Learning Research*, 12:2335–2382.
- Guillaume Jacques. 2021. [A grammar of Japhug](#). Number 1 in Comprehensive Grammar Library. Language Science Press, Berlin.
- Mark Johnson. 2008. [Unsupervised word segmentation for Sesotho using adaptor grammars](#). In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 20–27, Columbus, Ohio. Association for Computational Linguistics.
- Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. [Adaptor Grammars: a Framework for Specifying Compositional Nonparametric Bayesian Models](#). In *Advances in Neural Information Processing Systems 19*, pages 641–648, Cambridge, MA. MIT Press.
- Katharina Kann, Jesus Manuel Mager Hois, Ivan Vladimir Meza-Ruiz, and Hinrich Sch utze. 2018. [Fortification of neural morphological segmentation models for polysynthetic minimal-resource languages](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 47–57, New Orleans, Louisiana. Association for Computational Linguistics.
- Kimmo Koskenniemi. 1983. [Two-level morphology: A general computational model for word-form recognition and production](#), volume 11. University

- of Helsinki, Department of General Linguistics Helsinki, Finland.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. [Practical very large scale CRFs](#). In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics.
- Zoey Liu, Robert Jimerson, and Emily Prud'hommeaux. 2021. [Morphological segmentation for Seneca](#). In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 90–101, Online. Association for Computational Linguistics.
- Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. [Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling](#). In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 100–108, Suntec, Singapore. Association for Computational Linguistics.
- Sarah Moeller and Mans Hulden. 2018. [Automatic glossing in a low-resource setting for language documentation](#). In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 84–93, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Tumi Moeng, Sheldon Reay, Aaron Daniels, and Jan Buys. 2021. [Canonical and surface morphological segmentation for Nguni languages](#). *CoRR*, abs/2104.00767.
- Shu Okabe, Laurent Besacier, and François Yvon. 2022. [Weakly supervised word segmentation for computational language documentation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7385–7398, Dublin, Ireland. Association for Computational Linguistics.
- Teemu Ruokolainen, Oskar Kohonen, Kairit Sirts, Stig-Arne Grönroos, Mikko Kurimo, and Sami Virpioja. 2016. [A comparative study of minimally supervised morphological segmentation](#). *Computational Linguistics*, 42(1):91–120.
- Kairit Sirts and Sharon Goldwater. 2013. [Minimally-supervised morphological segmentation using Adaptor Grammars](#). *Transactions of the Association for Computational Linguistics*, 1:255–266.
- Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. [Morfessor 2.0: Toolkit for statistical morphological segmentation](#). In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.
- Yee Whye Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore.
- Anand Venkataraman. 2001. [A statistical model for word discovery in transcribed speech](#). *Computational Linguistics*, 27(3):352–372.
- Sami Virpioja, Oskar Kohonen, and Krista Lagus. 2011. [Evaluating the effect of word frequencies in a probabilistic generative model of morphology](#). In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 230–237, Riga, Latvia. Northern European Association for Language Technology (NEALT).
- Xingyuan Zhao, Satoru Ozaki, Antonios Anastasopoulos, Graham Neubig, and Lori Levin. 2020. [Automatic interlinear glossing for under-resourced languages leveraging translations](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5397–5408, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Word and morpheme distributions

According to Zipf’s law, for a unit of rank R , its normalised frequency f ($f = \frac{F}{N}$ with F the frequency of the unit and N the total number of units in the corpus) is computed as follows in Equation (5):

$$f = \frac{c}{R^a}, \quad (5)$$

with c a normalising constant and a the parameter of the distribution (Baayen, 2001). Hence, the relationship between the log-(normalised) frequency and the log-rank is:

$$\log(f) = -a \log(R) + \log(c) \quad (6)$$

To visualise the linear relationship shown in Equation (6), we hence fit a (least square) linear regression. Thus, Figure 4 plots the value of the slope $-a$ for words (x-axis) and morphemes (y-axis).

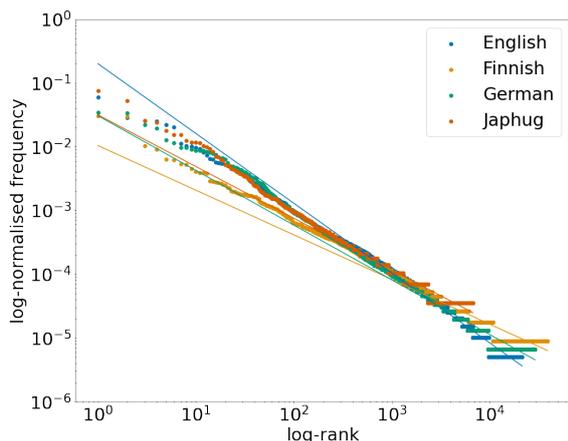


Figure 5: Log-normalised frequency of words according to their log-rank for several languages.

In Figure 5, we compare the Japhug text with three languages of varying morphological complexity (as in Section 4.4.1).

The Japhug curve lies between the English and Finnish ones, two languages with a clear contrast in morphological complexity. If for the most frequent words (i.e. low Zipf rank, on the left), the Japhug words follow the English or German trend, for rare words (i.e. high Zipf rank, on the right), it joins the Finnish trend.

B Output analysis

	supervision	sentence
dpseg	/	a mbroujme zu kszo
parallel-w	/	a mbro-ujme z ukx zo
parallel-w	sentence	a-mbro ujme zu kx-zo
reference		a-mbro u-jme zu kx-zo <i>Land on my horse's tail</i>

Figure 6: An example Japhug sentence segmented by various models, with and without supervision.

The example in Figure 6 displays a Japhug sentence segmented by two models. First, without supervision, dpseg fuses two units that should be separated by a word boundary (‘mbroujme’), and so does parallel-w with ‘ukx’. Apart from diminishing the three F-scores, this kind of error creates meaningless units. Besides, some reference morpheme boundaries are not identified: no boundary at all for ‘u-jme’ and a word boundary in ‘a-mbro’.

Once supervised, the parallel-w model corrects its initial error (‘ukx’ is segmented) and finds morpheme boundaries. Indeed, the model seems to benefit from the supervision data, which contained

the words ‘a-mbro’ and ‘kx-zo’. The remaining error (‘ujme’) can be explained by the fact that in the corpus, all occurrences of the morpheme ‘jme’ are always preceded by ‘u-’. The model thus does not identify nor recreate ‘jme’ as a unit but keeps ‘ujme’. The negative effect of collocations constitutes an inherent limit of the unigram dpseg model, already discussed by Goldwater et al. (2006).

C Reproducibility

All presented experiments have been obtained with the same three random seeds (42, 142, and 1234) for a fair comparison. Details about the hyperparameters are in Section 3.3.

The Adaptor Grammar was run with the hyperparameter values indicated for MorphAGram¹¹ (Eskander et al., 2020).

For reference, a processor of 6 cores and 12 threads takes around two days for a hierarchical model on the Tsez 2K corpus (20,000 iterations of Gibbs sampling). With the same setting, a parallel model takes approximately one day.

D Complete results

This section displays the full results for all our experimental settings: each model will be unsupervised or supervised with the sentence or dictionary supervision and will segment the Japhug and Tsez corpora. The tables also report the precision and recall for each evaluation tier (BP and BR for **B**oundary **P**recision and **B**oundary **R**ecall; WP, WR, and LP, LR, respectively for token and type evaluation). Bold values are the best score in a given experimental situation.

D.1 Japhug

Tables 3, 4, and 5 display the full results for the Japhug text.

D.2 Tsez

Similarly, Tables 6, 7, and 8 display the full results for the Tsez text.

D.3 Fully supervised model

For the sentence supervision method of Section 2.7, we also report the results of a CRF (Conditional Random Field, Lafferty et al. 2001), mainly inspired by the methodology of Moeller and Hulden (2018). Each training sentence is labelled as in Figure 7.

¹¹<https://github.com/rnd2110/MorphAGram>.

Original sentence	χ	ρ	υ	η	υ	ρ	υ
Translation (EN)							
	little				monk		

χ	ρ	υ	η	υ	ρ	υ
B-w	I	I	I	B-w	B-m	I

Figure 7: Example of Japhug sentence labelled for CRF

The ‘B-w’ label indicates the beginning of a word, while ‘B-m’ marks the start of a morpheme *inside* a word. The ‘I’ label is used for all other characters (inside a morpheme). We use Wapiti¹² (Lavergne et al., 2010) for the CRF implementation. Our feature set only includes basic unigram and bigram features.

The results in Table 4 and Table 7 show that, on average, full supervision yields better segmentation scores than weakly supervised models at the word level; contrarily, we observe worse scores at the morpheme level for both languages.

We also note that the CRF model identifies more than 4,000 morpheme types in both languages (i.e. much more than what exist in the reference or our models), which results in less than 36 in F-score on morpheme types (LF). This suggests that morphemes are difficult to distinguish from words, even in this favourable setting, confirming one of our main conclusions: statistical cues alone do not seem to be enough to correctly separate these two types of units.

¹²<https://github.com/Jekub/Wapiti>.

model level	AG		dpseg		pipeline		parallel-w		parallel-m		hier-type		-final	hier-iter	
	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	morph	word	morph.
BP	70.9	77.3	61.3	87.8	61.3	69.3	61.5	84.9	64.5	87.6	67.6	48.4	73.6	69.6	73.5
BR	71.1	90.4	90.6	75.2	90.6	96.3	90.8	82.4	84.5	75.0	83.4	88.2	93.4	77.8	91.2
BF	71.0	83.4	73.1	81.0	73.1	80.6	73.3	83.6	73.2	80.8	74.7	62.5	82.3	73.5	81.4
WP	45.8	58.3	39.4	59.3	39.4	50.1	39.7	62.1	41.1	58.9	44.6	19.2	54.9	44.9	53.7
WR	45.9	67.3	55.9	51.5	55.9	68.2	56.2	60.4	52.3	51.1	53.7	33.8	68.5	49.6	65.7
WF	45.8	62.5	46.2	55.1	46.2	57.8	46.5	61.3	46.0	54.7	48.7	24.5	60.9	47.2	59.1
LP	34.3	49.9	40.6	45.7	40.6	43.3	41.1	50.5	39.4	45.4	40.0	31.5	46.7	37.5	47.6
LR	28.4	20.3	13.6	37.8	13.6	11.0	13.8	33.9	17.1	37.9	22.6	11.7	15.6	27.4	16.7
LF	31.1	28.9	20.4	41.4	20.4	17.5	20.6	40.5	23.8	41.4	28.8	17.0	23.4	31.7	24.7
WL	4.72	2.51	3.34	3.34	2.13	3.34	2.98	3.73	3.35	3.93	1.65	2.32	4.29	2.37	
TL	6.60	3.27	4.22	4.22	2.64	4.23	3.99	4.77	4.21	4.78	2.83	2.87	5.12	2.88	
N_{type}	5582	1113	2260	2260	694	2257	1834	2921	2281	3806	1013	911	4925	956	
N_{token}	28.6k	53.9k	40.5k	40.5k	63.4k	40.5k	45.4k	36.3k	40.4k	34.4k	82.1k	58.2k	31.5k	57.0k	

Table 3: Results on the Japhug corpus for unsupervised dpseg and its two-level versions. **Bold** numbers denote the best results per metrics. Reference N_{type} : 6,739 for words and 2,731 for morphemes.

model level	CRF		dpseg		pipe.	parallel-w		parallel-m		hier-type		-final	hier-iter	
	word	morph.	word	morph.	morph.	word	morph.	word	morph.	word	morph.	morph.	word	morph.
BP	73.5	83.2	63.8	88.1	79.2	64.0	86.4	66.4	88.9	70.9	63.7	80.9	72.4	80.1
BR	80.8	85.2	91.4	79.6	97.4	91.4	83.7	86.3	77.7	84.0	92.2	96.0	80.2	94.5
BF	77.0	84.2	75.1	83.6	87.3	75.3	85.1	75.0	82.9	76.9	75.4	87.8	76.1	86.7
WP	52.6	66.4	43.7	64.3	67.2	43.9	65.6	44.8	63.6	49.6	43.8	68.5	49.8	66.9
WR	57.3	67.8	60.2	58.7	81.5	60.3	63.7	56.5	56.3	57.5	61.9	80.3	54.5	78.0
WF	54.9	67.1	50.6	61.4	73.6	50.8	64.7	50.0	59.7	53.3	51.3	74.0	52.1	72.0
LP	39.4	27.5	50.7	53.9	61.6	51.2	55.3	47.2	51.1	46.3	49.4	59.6	43.2	60.8
LR	49.5	50.3	19.6	40.2	23.5	19.9	39.4	22.3	42.7	30.0	23.2	25.9	33.4	26.4
LF	43.9	35.5	28.3	45.8	34.0	28.7	46.0	30.3	46.5	36.4	31.6	36.1	37.6	36.8
WL	4.35	2.84	3.44	3.19	2.39	3.45	2.99	3.75	3.28	4.08	2.05	2.48	4.32	2.49
TL	6.67	5.09	4.66	4.25	3.44	4.66	4.12	5.04	4.30	5.13	3.36	3.47	5.33	3.46
N_{type}	8453	4999	2610	2061	1040	2627	1946	3182	2283	4363	1285	1186	5208	1185
N_{token}	31.1k	47.6k	39.4k	42.5k	56.5k	39.2k	45.3k	36.0k	41.2k	33.2k	65.9k	54.6k	31.3k	54.4k

Table 4: Results on the Japhug corpus for dpseg and its two-level versions, supervised with dense annotations (**sentence**). 200 sentences are used as supervision data.

model level	dpseg		pipe.	parallel-w		parallel-m		hier-type		-final	hier-iter	
	word	morph.	morph.	word	morph.	word	morph.	word	morph.	morph.	word	morph.
BP	76.6	93.2	87.0	76.6	91.0	76.4	93.0	66.4	58.4	83.6	66.6	84.3
BR	81.0	64.3	83.1	81.2	71.3	74.9	64.2	89.6	89.9	90.2	90.1	90.8
BF	78.7	76.1	85.0	78.8	80.0	75.6	76.0	76.2	70.8	86.8	76.6	87.4
WP	54.4	54.8	65.9	54.5	60.1	51.6	54.4	45.6	30.4	67.2	46.0	68.7
WR	57.1	39.2	63.2	57.4	48.1	50.7	38.9	59.6	45.6	72.1	60.1	73.6
WF	55.7	45.7	64.5	55.9	53.5	51.1	45.4	51.7	36.5	69.6	52.1	71.1
LP	49.9	37.0	47.0	50.5	40.9	46.4	37.2	46.4	51.1	56.0	47.3	57.9
LR	37.3	54.8	43.5	37.8	52.3	36.8	54.8	21.6	30.2	34.3	21.9	34.9
LF	42.7	44.2	45.2	43.2	45.9	41.1	44.3	29.5	38.0	42.5	29.9	43.6
WL	4.51	4.06	3.03	4.49	3.62	4.81	4.06	3.63	1.94	2.71	3.62	2.71
TL	6.18	5.40	4.45	6.16	5.14	6.49	5.38	4.46	3.77	3.84	4.52	3.86
N_{type}	5041	4044	2524	5040	3492	5356	4027	3141	1618	1671	3116	1646
N_{token}	30.0k	33.3k	44.7k	30.1k	37.3k	28.1k	33.3k	37.3k	69.9k	50.0k	37.4k	49.9k

Table 5: Results on the Japhug corpus for dpseg and its two-level versions, supervised with a dictionary (**dictionary**). 200 sentences are used as supervision data.

model level	AG		dpseg		pipeline		parallel-w		hier-type		-final	hier-iter	
	word	morph.	word	morph.	word	morph.	word	morph.	word	morph.	morph	word	morph.
BP	67.3	78.1	59.9	91.8	59.9	69.9	59.6	89.3	64.0	47.8	74.4	64.7	74.7
BR	76.6	85.5	87.9	63.9	87.9	88.8	87.4	71.6	83.0	81.7	86.1	77.6	85.1
BF	71.6	81.6	71.3	75.3	71.3	78.2	70.9	79.5	72.2	60.3	79.8	70.6	79.6
WP	41.6	55.6	33.3	52.1	33.3	46.0	32.8	57.7	38.2	19.0	50.8	38.8	51.4
WR	46.7	60.5	47.4	37.1	47.4	57.8	46.6	46.8	48.4	31.9	58.3	45.7	58.2
WF	44.0	57.9	39.1	43.4	39.1	51.2	38.5	51.7	42.7	23.8	54.3	42.0	54.6
LP	45.9	51.3	49.6	41.4	49.6	41.2	49.0	47.7	47.3	24.2	41.1	42.3	43.1
LR	28.8	28.0	16.9	50.4	16.9	16.6	16.7	47.6	22.6	13.1	20.1	25.5	21.8
LF	35.4	36.2	25.2	45.5	25.2	23.6	25.0	47.7	30.5	17.0	27.0	31.8	29.0
WL	4.99	2.58		3.95	3.95	2.24	3.95	3.46	4.43	1.68	2.45	4.76	2.48
TL	6.52	3.52		4.53	4.53	2.89	4.52	4.32	4.95	2.87	3.07	5.35	3.08
N_{type}	3597	875		1950	1950	646	1958	1600	2732	867	786	3456	812
N_{token}	22.7k	43.8k		28.6k	28.6k	50.5k	28.6k	32.7k	25.6k	67.4k	46.2k	23.8k	45.5k

Table 6: Results on the Tsez corpus for unsupervised dpseg and its two-level versions. **Bold** numbers denote the best results per metrics. Reference N_{type} : 5,732 for words and 1,603 for morphemes.

model level	CRF		dpseg		pipe.	parallel-w		hier-type		-final	hier-iter		
	word	morph.	word	morph.	morph.	word	morph.	word	morph.	morph.	word	morph.	
BP	83.3	85.9	65.4	93.3	83.2	65.3	90.6	69.1	65.6	85.0	69.5	84.0	
BR	78.3	82.5	90.7	69.3	95.9	90.6	74.7	83.6	88.7	92.9	80.7	91.7	
BF	80.7	84.2	76.0	79.5	89.1	75.9	81.9	75.7	75.4	88.8	74.7	87.7	
WP	64.5	67.8	42.5	61.8	71.9	42.2	63.2	46.6	46.4	72.5	46.9	70.6	
WR	60.9	65.3	57.3	46.7	82.4	56.9	52.6	55.4	62.0	79.0	53.8	76.8	
WF	62.6	66.6	48.8	53.2	76.8	48.4	57.4	50.6	53.1	75.6	50.1	73.6	
LP	47.6	21.9	62.7	49.1	61.9	62.4	53.4	53.8	46.5	59.8	50.6	59.3	
LR	61.0	62.0	26.9	57.5	36.7	26.7	54.6	32.7	33.8	38.9	34.4	38.5	
LF	53.5	32.4	37.6	53.0	46.1	37.3	54.0	40.6	39.1	47.1	41.0	46.7	
WL	5.94	2.92		4.16	3.72	2.46	4.16	3.38	4.72	2.11	2.58	4.90	2.59
TL	7.83	5.98		5.02	4.58	3.67	5.03	4.40	5.43	3.49	3.70	5.61	3.67
N_{type}	7343	4537		2458	1877	950	2450	1639	3479	1165	1043	3902	1041
N_{token}	19.0k	38.7k		27.2k	30.4k	46.1k	27.2k	33.5k	24.0k	53.7k	43.8k	23.1k	43.7k

Table 7: Results on the Tsez corpus for dpseg and its two-level versions, supervised with dense annotations (**sentence**). 200 sentences are used as supervision data.

model level	dpseg		pipe.	parallel-w		hier-type		-final	hier-iter	
	word	morph.	morph.	word	morph.	word	morph.	morph.	word	morph.
BP	73.2	95.8	90.6	73.4	94.3	66.0	58.0	87.1	66.6	87.7
BR	84.9	58.5	79.6	84.9	63.1	91.2	82.0	84.5	90.5	85.4
BF	78.6	72.6	84.7	78.7	75.6	76.6	67.9	85.8	76.7	86.5
WP	50.3	49.7	66.1	50.5	53.1	43.0	29.1	65.8	43.6	67.7
WR	57.6	31.3	58.4	57.7	36.4	57.7	40.5	64.0	57.7	66.1
WF	53.7	38.4	62.0	53.9	43.2	49.3	33.8	64.9	49.7	66.9
LP	62.0	38.0	49.8	62.1	41.5	59.9	43.2	53.7	60.4	55.2
LR	37.2	64.6	54.1	37.3	63.1	26.9	36.2	44.9	27.7	45.9
LF	46.5	47.9	51.9	46.6	50.0	37.1	39.4	48.9	37.9	50.1
WL	4.91	4.47	3.18	4.92	4.10	4.18	2.02	2.89	4.24	2.88
TL	5.86	5.39	4.38	5.88	5.11	4.82	3.73	3.94	4.88	3.94
N_{type}	3442	2725	1744	3449	2441	2571	1342	1339	2624	1332
N_{token}	23.1k	25.3k	35.6k	23.0k	27.6k	27.1k	56.1k	39.1k	26.7k	39.2k

Table 8: Results on the Tsez corpus for dpseg and its two-level versions, supervised with a dictionary (**dictionary**). 200 sentences are used as supervision data. Reference N_{type} : 5,732 for words and 1,603 for morphemes.

Cross-Lingual Transfer of Cognitive Processing Complexity

Charlotte Pouw

ILLC, University of Amsterdam*
c.m.pouw@uva.nl

Nora Hollenstein

University of Copenhagen
nora.hollenstein@hum.ku.dk

Lisa Beinborn

Vrije Universiteit Amsterdam
l.beinborn@vu.nl

Abstract

When humans read a text, their eye movements are influenced by the structural complexity of the input sentences. This cognitive phenomenon holds across languages and recent studies indicate that multilingual language models utilize structural similarities between languages to facilitate cross-lingual transfer. We use sentence-level eye-tracking patterns as a cognitive indicator for structural complexity and show that the multilingual model XLM-RoBERTa can successfully predict varied patterns for 13 typologically diverse languages, despite being fine-tuned only on English data. We quantify the sensitivity of the model to structural complexity and distinguish a range of complexity characteristics. Our results indicate that the model develops a meaningful bias towards sentence length but also integrates cross-lingual differences. We conduct a control experiment with randomized word order and find that the model seems to additionally capture more complex structural information.

1 Introduction

Approximately 7,000 languages are currently spoken in the world, exhibiting differences at almost every level of linguistic organization (Eberhard et al., 2022). Nonetheless, psycholinguistic theories are predominantly supported by evidence from a handful of Indo-European languages (Norcliffe et al., 2015). Only recently, researchers have started to explore cross-linguistic differences in the neural implementation of language, uncovering both striking similarities across languages and empirical differences that cannot be explained by a unitary account (Malik-Moraleda et al., 2022).

In natural language processing, multilingual language models are optimized for tasks such as machine translation or cross-lingual information retrieval (Conneau et al., 2020) and follow a linguis-

tically naïve training regime. They are trained on dozens of languages simultaneously and do not account for typological differences between languages. Nevertheless, their cross-lingual transfer performance sets new records, even in zero-shot settings (Pires et al., 2019). The ability to transfer knowledge across languages has been attributed to the shared vocabulary that is used for all languages (Wu and Dredze, 2019) because it enables the reuse of common morphological roots for languages from the same family. However, recent studies indicate that vocabulary sharing is not a prerequisite for cross-lingual transfer (Artetxe et al., 2020) and that structural commonalities between languages play a more prevalent role in models (Karthikeyan et al., 2020).

Human sentence processing is sensitive to structural complexity. Eye movement data recorded during reading provide insights into cognitive processing patterns with a temporal accuracy of milliseconds (Winke, 2013). Structural processing difficulty materializes as regressions towards the complex region and an increase of fixations on that region (Clifton and Staub, 2011). For example, sentences with an object-relative structure trigger more regressions than sentences with more common subject-relative clauses (Gordon et al., 2006). A classical example of structural complexity are garden-path sentences which initially trigger a simplified interpretation that must be revised when reading the rest of the sentence (Bever, 1970).

On the surface level, eye movement patterns are language-specific since they are influenced by visual factors such as orthography and word length (Kliegl et al., 2004). For example, the Chinese script is much more visually dense than the alphabetic script, resulting in longer fixations and saccades that move to positions relatively close to the current word (Liversedge et al., 2016). On a deeper processing level, reading patterns seem to converge across languages. Predictability effects

This research was developed when the first author was affiliated to Vrije Universiteit Amsterdam.

have been demonstrated in multiple languages (Al-Jassmi et al., 2022; Laurinavichyute et al., 2019) and sentences that are matched for content are read at a similar speed in Chinese, English, and Finnish (Liversedge et al., 2016).

Sarti et al. (2021) find that the representations of an English pre-trained transformer-based language model encode structural complexity more prominently when they are fine-tuned to predict English eye-tracking patterns. Interestingly, Rama et al. (2020) claim that structural similarity between languages is only weakly represented in multilingual models. Nevertheless, Hollenstein et al. (2021) show that multilingual models are able to predict eye movement patterns of reading even for languages that are not seen during fine-tuning, which indicates a general learnability of the relationship between structural complexity and eye movement patterns. Their results are restricted to four languages (three of them are from the Germanic family), and it remains unclear which structural cues are leveraged for the cross-lingual prediction because the test sentences are not aligned across languages.

Contributions We examine whether the multilingual model XLM-RoBERTa (henceforth XLM-R) is sensitive to the structural complexity patterns that can be found in eye-tracking data. We use data from the newly released Multilingual Eye-tracking Corpus (Siegelman et al., 2022) to predict eye movement patterns for parallel texts in 13 typologically diverse languages. This allows us to specifically target the model’s sensitivity towards structural information and rules out the possibility that the results are influenced by differences in semantics or dataset sizes.

We show that XLM-R can apply cross-lingual transfer to predict eye-tracking patterns for all 13 languages while being fine-tuned only on English eye-tracking data. Our results indicate that the model develops a meaningful bias towards sentence length, but also integrates cross-lingual differences. For a more detailed analysis of structural sensitivity, we probe the model’s final layer for complexity features. Based on a control experiment with randomized word order, we conclude that the model seems to additionally capture more complex structural information. All our experimental code is publicly available at <https://github.com/CharlottePouw/crosslingual-complexity-transfer>.

2 Related Work

We introduce recent findings on the role of structural information for cross-lingual transfer in multilingual models and motivate the use of eye-tracking data as a proxy for cognitive processing complexity.

2.1 Cross-lingual Transfer in Multilingual Models

Massive multilingual language models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) are trained on more than a hundred languages simultaneously. Wu and Dredze (2019) show that this approach leads to surprisingly strong performances in cross-lingual transfer settings and attribute the improvements to the shared subword vocabulary. Pires et al. (2019) note that the model’s ability to generalize "cannot be attributed solely to vocabulary memorization". Complementary, Artetxe et al. (2020) and Liu et al. (2020) find that a shared vocabulary is not necessary for cross-lingual transfer. Instead, the multilingual model seems to exploit structural similarity between the training and the target language to facilitate transfer (Karthikeyan et al., 2020).

Structural similarity is loosely defined as an overlap on a subset of typological characteristics which seem to be better reflected in multilingual language models explicitly optimizing for cross-lingual transfer (Beinborn and Choenni, 2020; Choenni and Shutova, 2022). In language-agnostic models such as mBERT and XLM-R, the multilingual representations of the input can be separated into language-specific and language-neutral components (Tanti et al., 2021; Libovický et al., 2020; Gonen et al., 2020). While Rama et al. (2020) find that structural similarity between languages is only weakly represented in these models, Bjerva et al. (2019) observe that structural similarity between languages correlates most with representational similarity. Experiments with artificial languages indicate that multilingual models are sensitive to hierarchical structure (De Varda and Zamparelli, 2022) and to word order (Chai et al., 2022; Deshpande et al., 2022). Ahmad et al. (2021) show that cross-lingual transfer can be improved by explicitly encoding structural information via an auxiliary syntactic objective and Guarasci et al. (2022) find that structural complexity knowledge can even be transferred across languages without explicit training.

2.2 Predicting Processing Complexity

Recent studies indicate that transformer-based language models are sensitive to structural characteristics of the input sentence when predicting eye-tracking patterns. [Hollenstein et al. \(2021\)](#) find a correlation between the Flesch reading ease score and eye-tracking prediction accuracy of pre-trained multilingual transformer models which disappears after fine-tuning. [Wiechmann et al. \(2022\)](#) detect similar correlations between the prediction accuracy of English transformer models and a wider range of readability features. Finally, [Hollenstein et al. \(2022b\)](#) find that eye-tracking metrics predicted by multilingual transformer models correlate in a similar way with readability features as eye-tracking metrics recorded from human readers.

Sensitivity to structural complexity also seems to increase when incorporating eye-tracking data in NLP models. Learning eye movement behavior as an auxiliary task has been shown to facilitate the prediction of text complexity in English and Portuguese ([González-Garduño and Søgaard, 2017](#); [Evaldo Leal et al., 2020](#)). [Barrett et al. \(2016\)](#) show that English eye-tracking features improve the performance a French part-of-speech tagger, suggesting that information learned from monolingual eye-tracking data is transferable across languages.

In this work, we explicitly test for sensitivity to a range of structural characteristics in multilingual models and analyze if structural sensitivity increases by learning to predict eye-tracking patterns. We extend previous analyses to a much wider range of languages from five different families (Indo-European, Koreanic, Semitic, Turkic, and Uralic).

3 Methodology

We fine-tune a pre-trained multilingual transformer model to predict eye-tracking metrics in a setting of zero-shot cross-lingual transfer.

3.1 Data

We use the aligned multilingual eye-tracking corpus MECO for testing. As the multilingual data consists of only few samples, we use the larger monolingual English eye-tracking dataset GECO for training. Size statistics of both corpora can be found in the appendix in Table 3.

Multilingual Eye-tracking Corpus (MECO)
The Multilingual Eye-tracking Corpus contains par-

allel eye-tracking data of reading in 13 different languages ([Siegelman et al., 2022](#)).¹ The reading material consists of 12 short Wikipedia-style texts about various topics, which participants read in their native language. The texts were either directly translated or carefully matched for topic, genre, and readability. Each of the 12 texts was presented on a single screen and in the same fixed order in all languages. The number of participants ranged from 29 to 54 per language (45 on average).

Ghent Eye-tracking Corpus (GECO) The Ghent Eye-tracking Corpus contains eye-tracking data from 14 monolingual English readers ([Cop et al., 2016](#)). They were reading the entire novel *The Mysterious Affair at Styles* by Agatha Christie which was presented on the screen one paragraph at a time.

3.2 Experimental Setup

We use multi-task learning for predicting four sentence-level eye-tracking metrics.

Sentence-Level Eye-Tracking Metrics [Liversedge et al. \(2016\)](#) find that eye movement patterns are more comparable across languages at the sentence level than at the word level. We select four sentence-level eye-tracking metrics that cover both early and late language processing in line with [Sarti et al. \(2021\)](#). For each sentence s , we consider:

1. *Fixation count*: number of fixations on s
2. *Total fixation duration*: total duration of all fixations on s
3. *First-pass duration*: duration of the first reading pass over s
4. *Regression duration*: total duration of all regressions within s .

Duration values are measured in milliseconds. To obtain generalized eye movement patterns, we average all eye-tracking metrics over participants and scale each eye-tracking feature to fall in the range 0–100, so that the loss can be calculated uniformly for durations and counts ([Hollenstein et al., 2021](#)). The distribution of the four metrics is shown in the appendix in Figure 7.

Model We use XLM-R ([Conneau et al., 2020](#)) as our multilingual transformer model since it achieved the best zero-shot results in the CMCL 2022 Shared Task on Multilingual and Crosslingual

¹Dutch, English, Estonian, Finnish, German, Greek, Hebrew, Italian, Korean, Norwegian, Russian, Spanish, Turkish.

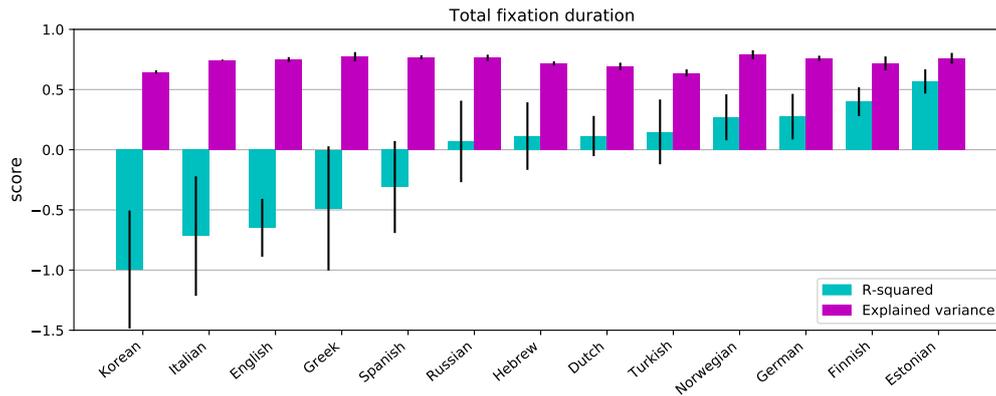


Figure 1: Cross-lingual transfer results for predicting cognitive processing complexity (i.e. sentence-level fixation duration). Prediction performance is evaluated with explained variance and R^2 for each language in MECO. The results are averaged over 5 folds; error bars denote the standard deviation over folds.

Prediction of Human Reading Behaviour (Srivastava, 2022; Hollenstein et al., 2022a). The model was pre-trained on 2.5TB CommonCrawl data containing 100 languages using the Masked Language Modelling objective and uses SentencePiece subword tokenization (Kudo and Richardson, 2018). We select the Huggingface checkpoint *xlm-roberta-base* and add a linear dense layer to predict four sentence-level eye-tracking metrics.

Multi-Task Learning We employ multi-task learning with hard parameter sharing to fine-tune the model on all eye-tracking metrics simultaneously in line with Sarti et al. (2021). This means that all model parameters are shared except for the task-specific regression heads in the final prediction layer. More specifically, the same sentence representation is fed into each of the four regression heads which predict their respective eye-tracking metric. The model parameters are optimized jointly for all regression tasks by summing the individual MSE losses in line with previous work (Hollenstein et al., 2021, 2022a; Wiechmann et al., 2022).

Training Parameters We fine-tune XLM-R for 15 epochs with early stopping after 5 epochs without an improvement in the validation accuracy. We use 10% of the training data as validation data and evaluate every 40 steps. We employ a batch size of 32 and a learning rate of $1e-5$. The sentence representation is obtained by mean pooling over token representations. We train the model on the GECCO data using 5-fold cross-validation and report the average over the folds for each language in MECO.

Evaluation We report explained variance and R-Squared (R^2) to capture the proportion of variance

in the dependent variable that can be explained by our model in line with Sarti et al. (2021). Explained variance uses the biased variance to determine what fraction of the variance is explained. R^2 uses the raw sums of squares instead and provides complementary information about systematic offsets in the predictions. We report both metrics and evaluate the performance of the fine-tuned model individually for each of the four eye-tracking metrics.²

4 Cross-Lingual Transfer Results

Figure 1 shows the explained variance and R^2 scores of the fine-tuned model for total fixation duration across languages. In terms of explained variance, we see that the model achieves a similar performance across languages, i.e. it captures 60 to 80 percent of the variance in the original eye-tracking signal for all languages. The R^2 scores, on the other hand, vary much more depending on the language. Similar results were observed for two of the other eye-tracking metrics, i.e. fixation count and first-pass duration, but the model is worse at predicting regression duration (see Figure 8 in the appendix). To better control for spurious correlations, we ran the experiment on permuted input-output pairs, i.e., we paired input sentences with eye-tracking values corresponding to another random sentence and averaged the results over 5 folds. For this random baseline setup, both explained variance and R^2 are always strictly negative for all languages.

²In previous work on token-level eye-tracking prediction, the mean absolute error was reported instead but it is less informative for sentence-level predictions because sentence-level eye-tracking metrics are generally more centered around the mean.

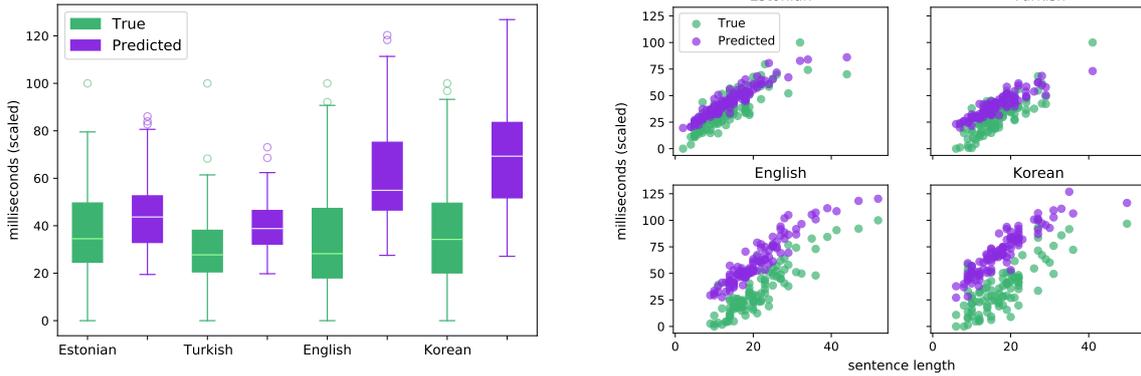


Figure 2: The left plot shows the distribution of true and predicted values for total fixation duration for Estonian, Turkish, English and Korean sentences in MECO. The right figure shows the distribution of values with respect to sentence length.

To better understand the varied R^2 scores for different languages, we show the distribution of the true and predicted values for total fixation duration for two languages with high R^2 (Estonian, Turkish) and two languages with low R^2 (English, Korean) in Figure 2. We see that the low R^2 for English and Korean is caused by predictions that are consistently too high. For Estonian and Turkish, the difference between true and predicted values is clearly smaller, resulting in a higher R^2 . Nevertheless, the model is able to predict a significant amount of the variance in the eye-tracking signal of all languages, as expressed by the stable explained variance scores across languages.

Interestingly, the model performs slightly better for most zero-shot languages than for the fine-tuning language English. Recall that this performance difference cannot be attributed to cross-lingual differences in semantics, since all sentences are parallel with respect to content. On the right side of Figure 2, we analyze the predictions with respect to sentence length and find that both the model predictions and the true values for fixation duration correlate with sentence length in all languages. As sentence length is an indicator of structural complexity, we further dissect this phenomenon and conduct an analysis of a range of structural characteristics in the following section.

5 Sensitivity to Structural Complexity

We explore four categories of sentence-level complexity features: length, frequency, morpho-syntactic, and syntactic. Word frequencies are obtained as standardized Zipf frequencies using the Python package wordfreq (Speer et al., 2018). The

package combines several frequency resources, including SUBTLEX lists (e.g. Brysbaert and New (2009)) and OpenSubtitles (Lison and Tiedemann, 2016). The morpho-syntactic and syntactic features are computed using the Profiling-UD tool (Brunato et al., 2020).

Cross-Lingual Differences We showcase an individual example sentence in Table 1 to compare the predicted fixation duration for English, Finnish and Turkish. We observe that the highest value is predicted for the English version. This is most likely caused by its length, as the sentence is less complex than the Finnish and Turkish versions in terms of all other linguistic features.

Interestingly, the model predicts that Finnish readers will fixate on the sentence longer than Turkish readers, even though both sentences have the same length. The Turkish sentence contains longer, less frequent words, and is lexically more dense, but the Finnish sentence contains longer dependency links. This indicates that the model is more sensitive to dependency structure than to low-level complexity (i.e. word length and frequency) when predicting eye-tracking values for sentences of the same length.

5.1 Sensitivity to Fine-Tuning Input

To analyze the model’s sensitivity to the structural complexity of the fine-tuning data, we compare the performance of the fine-tuned model for in-domain data (English GECO) and cross-domain data (English MECO). Table 2 shows the explained variance and R^2 scores of the fine-tuned model predictions for each eye-tracking metric for both domains. We see that the model consistently yields

Example		Prediction
English	<i>In ancient Roman religion and myth, Janus is the god of beginnings and gates.</i>	42.96
Finnish	<i>Muinaisen roomalaisen mytologian mukaan Janus oli alkujen ja porttien jumala.</i>	38.91
Turkish	<i>Antik Roma inanışlarında ve mitlerinde, Janus başlangıçların ve kapıların tanrısıdır.</i>	32.28

Structural Complexity		English	Finnish	Turkish
Length	Sentence length (tokens)	14	10	10
	Avg. word length (characters)	4.57	6.80	7.60
Frequency	Avg. word frequency (Zipf)	5.63	4.36	3.46
	# low frequency words	2	6	6
	Lexical density	0.57	0.70	0.73
Morpho-Syntactic Syntactic	Parse tree depth	3	3	3
	Avg. dependency link length	2.15	2.78	1.90
	Max. dependency link length	7	7	4
	# verbal heads	1	1	1

Table 1: Predicted values for total fixation duration for the same example sentence in English, Finnish, and Turkish (top), and the respective values for the nine structural complexity features (bottom).

more accurate predictions for the in-domain data than for the cross-domain data.

	MECO		GECO	
	EV	R^2	EV	R^2
FC	.78 (.02)	-.63 (.35)	.93 (.00)	.93 (.01)
TFD	.75 (.02)	-.65 (.24)	.92 (.00)	.92 (.01)
FPD	.50 (.03)	-.87 (.27)	.95 (.00)	.95 (.01)
RD	-.28 (.14)	-.96 (.45)	.44 (.04)	.45 (.05)

Table 2: Explained variance (EV) and R^2 -scores of the fine-tuned model predictions for four eye-tracking metrics from the English parts of MECO and GECO: fixation count (FC), total fixation duration (TFD), first-pass duration (FPD), and regression duration (RD). The results are averaged over 5 folds; standard deviations are indicated in parentheses.

To better understand why the model does not generalize well across domains for English, we visualize the Spearman correlation between complexity features and eye-tracking metrics for English GECO and MECO sentences in Figure 3. We see that the predicted values for the MECO sentences exhibit a similar correlation pattern with the complexity features as the GECO sentences. The true values of MECO are less consistent with this pattern. Literary texts contain very different words than encyclopedic texts, which might influence fixation durations and trigger regressions that cannot solely be explained by structural complexity. In addition, MECO is significantly smaller than GECO (99 vs 4,041 English sentences) and contains data from a higher number of participants (46 vs 14). The smaller amount of sentences and the larger amount of readers increase the effect of individ-

ual differences³ which might obscure correlations between structural complexity and eye movement patterns. Directly applying the learned correlations from GECO to MECO might explain why the fine-tuned model fails to generalize across domains.

The average sentence length is considerably higher in GECO than in MECO (21 vs 13 words, see Table 3). As the model predictions strongly correlate with sentence length, we speculate that the model overestimates eye-tracking values for sentences that are longer than the majority of fine-tuning sentences which would explain the higher mean of the predictions in Figure 2.

Multi-Task Learning Effect Figure 3 further shows that regression duration is only weakly correlated with the complexity metrics in contrast to the other eye-tracking metrics. Nevertheless, the correlations between the model predictions and the complexity features are similar for all four metrics. This indicates a drawback of multi-task learning: since the loss is computed jointly over all tasks, accurate predictions for three out of four tasks already yield a small loss. The model seems to overfit to first-pass duration, total fixation duration and fixation count, which can all be predicted from similar complexity features, and does not learn the deviat-

³A higher number of participants leads to more diversity across readers with respect to individual factors that could influence reading strategies (e.g. age, education level). The GECO data came from 14 English readers who were all undergraduate students with an age range of 18-26. The MECO data came from 29 to 54 readers per language (45 on average), who had more diverse educational backgrounds and a wider age range (18-45). Based on these statistics, we assume that the increased heterogeneity of the MECO participants influences the correlations observed in Figure 3.

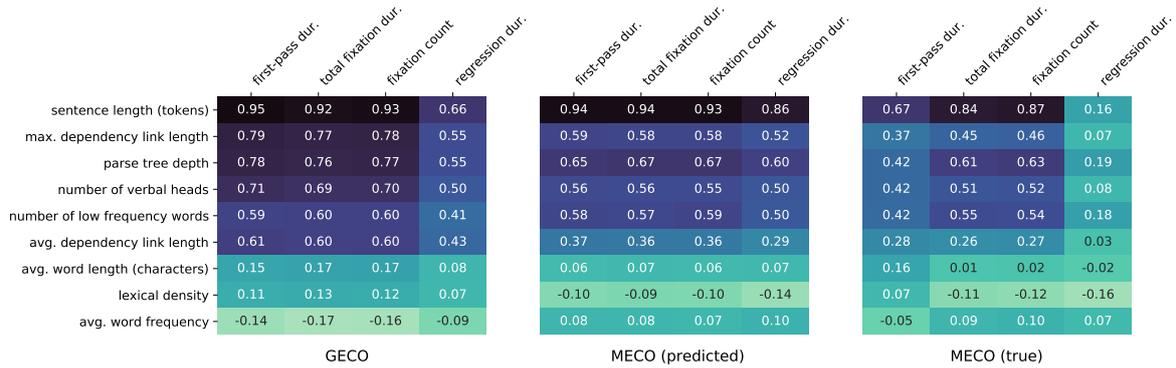


Figure 3: Spearman correlations between complexity features and eye-tracking metrics of GECO and the English part of MECO (predicted versus true). A darker color represents a stronger correlation. All GECO correlations are significant ($p < 0.001$); MECO correlations above 0.2 are significant ($p < 0.01$).

ing patterns to predict regression duration. Further research is needed to better understand the linguistic features underlying regression duration.

5.2 Feature-Based Prediction

To further establish which complexity features are good predictors for each individual eye-tracking metric, we examine the extent to which the four eye-tracking metrics can be predicted from explicit features. Since multi-task learning seems to have a negative impact on learning the structural features underlying each individual eye-tracking metric, we train a separate feature-based model for each eye-tracking metric individually. We use support vector machines (SVM) with a linear kernel as our feature-based regression models. We employ the SVR implementation from scikit-learn (Pedregosa et al., 2011) with all default parameters and use different subsets of features from Table 1: 1) only the two length features, 2) only the two frequency features, 3) only the five structural (i.e., morpho-syntactic and syntactic) features, and 4) all nine features.

As the SVM models predict a simpler problem (a single eye-tracking metric), it is not surprising that they outperform the fine-tuned multi-task model with respect to the absolute predictions (as measured by R^2 , see appendix Figure 9). More interestingly, Figure 4 shows that the multi-task model is able to capture a similar amount of variance as the length-based SVM. Furthermore, we see that the length-based SVM performs almost identically to the SVM trained on *all* complexity features, outperforming the SVMs trained on frequency features and structural features. This shows that length is a strong predictor for sentence-level eye-tracking metrics, and suggests that structural and frequency

features do not provide much additional information. We further investigate if length is the main factor affecting the predictions of the fine-tuned model in the following section.

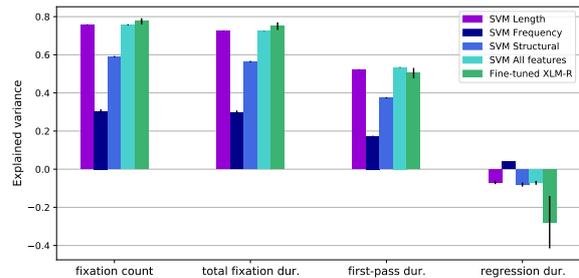


Figure 4: Explained variance of the four feature-based SVM models and the fine-tuned XLM-R model. The models are trained on GECO using 5-fold cross-validation and evaluated on the English part of MECO; error bars denote the standard deviation over folds.

6 The Role of Sentence Length

To test whether the fine-tuned XLM-R model captures more sophisticated structural information than sentence length, we conduct two additional experiments. First, we probe the final-layer representations of the model for the complexity features from Table 1, both before and after fine-tuning on eye-tracking data. Second, we compare the performance of the fine-tuned model to a control condition: we randomize the word order within each MECO sentence to analyze the prediction performance on scrambled input.

6.1 Probing Set-up

We train regressors g_i to predict a value for each of the nine latent factors of structural complexity

$Z = z_1, \dots, z_9$ using XLM-R’s final-layer representation $\theta(x)$ of our input sentence x . The prediction accuracy of g_i is an indication of how prominently the linguistic property z_i is encoded in θ . We analyze this both for the pre-trained and fine-tuned representations of XLM-R to quantify the relative increase of sensitivity to z_i after fine-tuning on eye-tracking metrics.

We conduct the probing experiments for three typologically different languages to analyze if the structural sensitivity that was acquired from English eye-tracking data transfers to other languages. As input, we use 1,000 parallel sentences from the English, Korean and Turkish parts of the Parallel Universal Dependencies (PUD) treebanks which were randomly selected from Wikipedia and news articles (Zeman et al., 2017). We apply a 5-fold cross-validation setting with 800 sentences for training the probing regressors for each language and the remaining 200 for testing. We use the same architecture as described in Section 3.2, but freeze the encoder model and only update the final regression layer during training. The regression layer contains nine probing heads (one for each linguistic feature) and is trained for 5 epochs.⁴

6.2 Results

We report the results of the probing experiments and the model performance on scrambled inputs.

Probing Figure 5 shows the relative probing performance for each complexity feature. We see that fine-tuning yields the largest improvements for probing sentence length and average dependency link length. For the other complexity features, we see that the fine-tuned representations yield little to no improvement in probing accuracy compared to the pre-trained representations. This mostly concerns the features for which sentence length is factored out, i.e., average word frequency, average word length and lexical density. Sarti et al. (2021) report similar results and show that increased probing performance for dependency features persists for sentences of the same length. This provides additional evidence that structural information is learned in addition to low-level length information.

We observe only minor differences in probing accuracy for individual complexity features of En-

⁴We report results for a multi-task set-up for probing in line with Sarti et al. (2021) and use the same hyperparameters as for the fine-tuning experiments but without intermediate evaluation on a development set. We also ran single-task probing as a sanity check and obtained similar results.

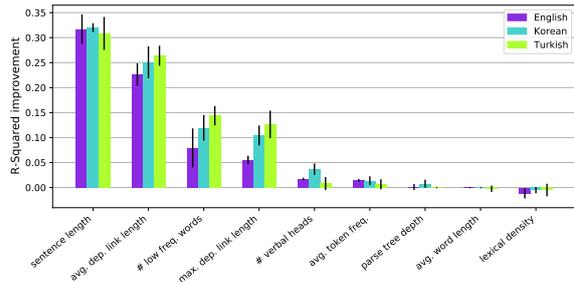


Figure 5: Relative improvement in R^2 for complexity features of English, Korean and Turkish sentences in fine-tuned XLM-R sentence representations over pre-trained representations. The results are calculated using probing regressors and averaged over 5 folds.

glish, Korean and Turkish sentences. The general pattern is consistent for all languages: features related to the structural complexity of sentences are more easily predicted after fine-tuning on eye-tracking metrics. This indicates that the fine-tuned model is able to transfer structural complexity knowledge acquired from English eye-tracking data to other languages.

Influence of Word Order We compare the performance of the fine-tuned model on sentences with normal versus scrambled word order, both in terms of explained variance and R^2 . We measure similar explained variance scores for both input types. This indicates that the model is able to account for a large portion of the variance in our eye-tracking data by merely considering sentence length. The R^2 scores, on the other hand, are consistently lower for scrambled inputs, as shown for total fixation duration in Figure 6 (see appendix Figure 10 for the other eye-tracking metrics). We conclude that the model is sensitive to word order and bases its eye-tracking predictions not only on sentence length but also on more complex structural characteristics.

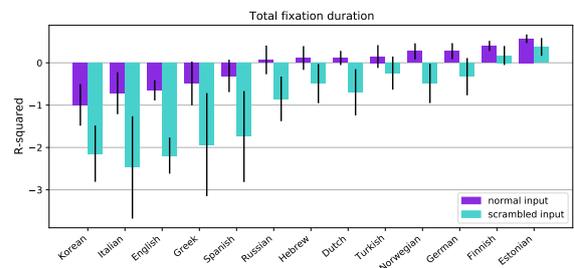


Figure 6: R^2 scores for total fixation duration for each language in MECO, both for sentences with normal and scrambled word order. The results are averaged over 5 folds; error bars denote the standard deviation.

7 Conclusion

We find that XLM-R can apply cross-lingual transfer to predict cognitive processing difficulty with similar performance across 13 typologically diverse languages, despite being fine-tuned only on English data. We conducted a range of experiments to quantify the model’s sensitivity to structural complexity and find that the fine-tuned model prominently encodes sentence length, but also considers more complex structural information such as dependency structure and word order for the prediction of eye-tracking metrics.

Our analyses suggest that domain differences in training and testing data have a greater impact on model performance than language differences within the same domain. More specifically, XLM-R performs better on in-domain GECO data than cross-domain MECO data, but within MECO, XLM-R shows similar performance across languages. This aligns with the findings of [Morger et al. \(2022\)](#), who show that the correlation between relative importance metrics and total fixation duration is influenced by text domain. Our study highlights the significance of controlling for text domain and size, as it allows to evaluate cross-lingual generalization that is independent of dataset characteristics.

In future work, we plan to better account for individual differences between readers ([Brandl and Hollenstein, 2022](#)) and spill-over effects across sentence boundaries ([Wiechmann et al., 2022](#)). The modeling approach for learning eye-tracking patterns also needs further exploration. We find that sentence-level prediction of eye-tracking patterns works well for learning about structural complexity, but that it is not optimal for capturing lexical complexity. Token-level measures, as predicted in [Hollenstein et al. \(2021\)](#), are more likely to be informative about lexical phenomena. A joint loss for sentence and token-level eye-tracking metrics might lead to sensitivity to a wider range of linguistic complexity features.

8 Limitations

The main limitation of our work is the use of relatively small datasets for testing our models due to limited availability of eye-tracking data in multiple languages. The dataset used for testing cross-lingual transfer (MECO) contains approximately 100 sentences per language. For probing structural complexity, we used a sample of 1,000 sentences

per language.

As in related work, we averaged the eye-tracking metrics over readers to obtain a more robust indication of human reading behavior. This approach disregards the fact that reading is a highly individual process that is dependent on cognitive factors and experience. A computational model might develop a better sense of linguistic complexity when it learns about the linguistic properties that lead to variation across readers and we are working towards methods for integrating this information.

Acknowledgements

We thank the anonymous reviewers for their insightful feedback. L. Beinborn’s research was supported by the Dutch National Science Organisation (NWO) through the projects CLARIAHPLUS (CP-W6-19-005) and VENI (VI.Veni.211C.039).

References

- Wasi Ahmad, Haoran Li, Kai-Wei Chang, and Yashar Mehdad. 2021. [Syntax-augmented multilingual BERT for cross-lingual transfer](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4538–4554, Online. Association for Computational Linguistics.
- Maryam AlJassmi, Kayleigh Warrington, Victoria McGowan, Sarah White, and Kevin Paterson. 2022. [Effects of word predictability on eye movements during Arabic reading](#). *Attention, Perception, & Psychophysics*, 84(1):10–24.
- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. [On the cross-lingual transferability of monolingual representations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.
- Maria Barrett, Frank Keller, and Anders Søgaard. 2016. [Cross-lingual transfer of correlations between parts of speech and gaze features](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1330–1339, Osaka, Japan. The COLING 2016 Organizing Committee.
- Lisa Beinborn and Rochelle Choenni. 2020. [Semantic drift in multilingual representations](#). *Computational Linguistics*, 46(3):571–603.
- Thomas Bever. 1970. *The Cognitive Basis for Linguistic Structures*, pages 279–352. Cognition and the Development of Language.

- Johannes Bjerva, Robert Östling, Maria Han Veiga, Jörg Tiedemann, and Isabelle Augenstein. 2019. [What Do Language Representations Really Represent?](#) *Computational Linguistics*, 45(2):381–389.
- Stephanie Brandl and Nora Hollenstein. 2022. [Every word counts: A multilingual analysis of individual human alignment with model attention.](#) In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 72–77, Online only. Association for Computational Linguistics.
- Dominique Brunato, Andrea Cimino, Felice Dell’Orletta, Giulia Venturi, and Simonetta Montemagni. 2020. [Profiling-UD: a tool for linguistic profiling of texts.](#) In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7145–7151, Marseille, France. European Language Resources Association.
- Marc Brysbaert and Boris New. 2009. [Moving beyond Kucera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English.](#) *Behavior research methods*, 41:977–90.
- Yuan Chai, Yaobo Liang, and Nan Duan. 2022. [Cross-lingual ability of multilingual masked language models: A study of language structure.](#) In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4702–4712, Dublin, Ireland. Association for Computational Linguistics.
- Rochelle Choenni and Ekaterina Shutova. 2022. [Investigating language relationships in multilingual sentence encoders through the lens of linguistic typology.](#) *Computational Linguistics*, 48(3):635–672.
- Charles Clifton and Adrian Staub. 2011. [Syntactic influences on eye movements during reading.](#) In *The Oxford Handbook of Eye Movements*, pages 896–909. Oxford University Press.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale.](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2016. [Presenting GECO: An eyetracking corpus of monolingual and bilingual sentence reading.](#) *Behavior Research Methods*, 49.
- Andrea De Varda and Roberto Zamparelli. 2022. [Multilingualism encourages recursion: a transfer study with mBERT.](#) In *Proceedings of the 4th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 1–10, Seattle, Washington. Association for Computational Linguistics.
- Ameet Deshpande, Partha Talukdar, and Karthik Narasimhan. 2022. [When is BERT multilingual? isolating crucial ingredients for cross-lingual transfer.](#) In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3610–3623, Seattle, United States. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding.](#) In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- David Eberhard, Gary Simons, and Charles Fenig (eds.). 2022. *Ethnologue: Languages of the World*, twenty-fifth edition. SIL International, Dallas, Texas.
- Sidney Evaldo Leal, João Marcos Munguba Vieira, Erica dos Santos Rodrigues, Elisângela Nogueira Teixeira, and Sandra Aluísio. 2020. [Using eye-tracking data to predict the readability of Brazilian Portuguese sentences in single-task, multi-task and sequential transfer learning approaches.](#) In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5821–5831, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. 2020. [It’s not Greek to mBERT: Inducing word-level translations from multilingual BERT.](#) In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 45–56, Online. Association for Computational Linguistics.
- Ana Valeria González-Garduño and Anders Søgaard. 2017. [Using gaze to predict text readability.](#) In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 438–443, Copenhagen, Denmark. Association for Computational Linguistics.
- Peter Gordon, Randall Hendrick, Marcus Johnson, and Yoonhyoung Lee. 2006. [Similarity-based interference during language comprehension: Evidence from eye tracking during reading.](#) *Journal of experimental psychology. Learning, memory, and cognition*, 32:1304–21.
- Raffaele Guarasci, Stefano Silvestri, Giuseppe De Pietro, Hamido Fujita, and Massimo Esposito. 2022. [BERT syntactic transfer: A computational experiment on Italian, French and English languages.](#) *Comput. Speech Lang.*, 71(C).

- Nora Hollenstein, Emmanuele Chersoni, Cassandra Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. 2022a. [CMCL 2022 shared task on multilingual and crosslingual prediction of human reading behavior](#). In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 121–129, Dublin, Ireland. Association for Computational Linguistics.
- Nora Hollenstein, Itziar Gonzalez-Dios, Lisa Beinborn, and Lena Jäger. 2022b. [Patterns of text readability in human and predicted eye movements](#). In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, pages 1–15, Taipei, Taiwan. Association for Computational Linguistics.
- Nora Hollenstein, Federico Pirovano, Ce Zhang, Lena Jäger, and Lisa Beinborn. 2021. [Multilingual language models predict human reading behavior](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 106–123, Online. Association for Computational Linguistics.
- K. Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. [Cross-lingual ability of multilingual BERT: An empirical study](#). In *International Conference on Learning Representations*.
- Reinhold Kliegl, Ellen Grabner, Martin Rolfs, and Ralf Engbert. 2004. [Length, frequency, and predictability effects of words on eye movements in reading](#). *European Journal of Cognitive Psychology*, 16(1-2):262–284.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Anna K. Laurinavichyute, Irina A. Sekerina, Svetlana Alexeeva, Kristine Bagdasaryan, and Reinhold Kliegl. 2019. Russian Sentence Corpus: Benchmark measures of eye movements in reading in Russian. *Behavior Research Methods*, 51:1161–1178.
- Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. [On the language neutrality of pre-trained multilingual representations](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1663–1674, Online. Association for Computational Linguistics.
- Pierre Lison and Jörg Tiedemann. 2016. [OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).
- Chi-Liang Liu, Tsung-Yuan Hsu, Yung-Sung Chuang, and Hung-Yi Lee. 2020. [A study of cross-lingual ability and language-specific information in multilingual bert](#). *arXiv preprint arXiv:2004.09205*.
- Simon P. Liversedge, Denis Drieghe, Xin Li, Guoli Yan, Xuejun Bai, and Jukka Hyönä. 2016. [Universality in eye movements and reading: A trilingual investigation](#). *Cognition*, 147:1–20.
- Saima Malik-Moraleda, Dima Ayyash, Jeanne Gallée, Josef Affourtit, Malte Hoffmann, Zachary Mineroff, Olessia Jouravlev, and Evelina Fedorenko. 2022. [An investigation across 45 languages and 12 language families reveals a universal language network](#). *Nature Neuroscience*, 25:1–6.
- Felix Morger, Stephanie Brandl, Lisa Beinborn, and Nora Hollenstein. 2022. [A cross-lingual comparison of human and model relative word importance](#). In *Proceedings of the 2022 CLASP Conference on (Dis)embodiment*, pages 11–23, Gothenburg, Sweden. Association for Computational Linguistics.
- Elisabeth Norcliffe, Alice C. Harris, and T. Florian Jaeger. 2015. [Cross-linguistic psycholinguistics and its critical role in theory development: early beginnings and recent advances](#). *Language, Cognition and Neuroscience*, 30(9):1009–1032.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. [Scikit-learn: Machine learning in Python](#). *Journal of Machine Learning Research*, 12:2825–2830.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. [How multilingual is multilingual BERT?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.
- Taraka Rama, Lisa Beinborn, and Steffen Eger. 2020. [Probing multilingual BERT for genetic and typological signals](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1214–1228, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Gabriele Sarti, Dominique Brunato, and Felice Dell’Orletta. 2021. [That looks hard: Characterizing linguistic complexity in humans and language models](#). In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 48–60, Online. Association for Computational Linguistics.
- Noam Siegelman, Sascha Schroeder, Cengiz Acartürk, Hee-Don Ahn, Svetlana Alexeeva, Simona Amenta, Raymond Bertram, Rolando Bonandrini, Marc Brysbaert, Daria Chernova, Sara Maria Da Fonseca, Nicolas Dirix, Wouter Duyck, Argyro Fella, Ram Frost,

- Carolina A Gattei, Areti Kalaitzi, Nayoung Kwon, Kaidi Lõo, Marco Marelli, Timothy C Papadopoulos, Athanassios Protopapas, Satu Savo, Diego E Shalom, Natalia Slioussar, Roni Stein, Longjiao Sui, Analí Taboh, Veronica Tønnesen, Kerem Alp Usal, and Victor Kuperman. 2022. [Expanding horizons of cross-linguistic research on reading: The multilingual eye-movement corpus \(meco\)](#). *Behavior Research Methods*, page 1–21.
- Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2018. [Luminosinsight/wordfreq:v2.2](#).
- Harshvardhan Srivastava. 2022. [Poirot at CMCL 2022 shared task: Zero shot crosslingual eye-tracking data prediction using multilingual transformer models](#). In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 102–107, Dublin, Ireland. Association for Computational Linguistics.
- Marc Tanti, Lonneke van der Plas, Claudia Borg, and Albert Gatt. 2021. [On the language-specificity of multilingual BERT and the impact of fine-tuning](#). In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 214–227, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Daniel Wiechmann, Yu Qiao, Elma Kerz, and Justus Mattern. 2022. [Measuring the impact of \(psycho\)linguistic and readability features and their spill over effects on the prediction of eye movement patterns](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5276–5290, Dublin, Ireland. Association for Computational Linguistics.
- Paula M. Winke. 2013. *Eye-Tracking Technology for Reading*, chapter 62. John Wiley & Sons, Ltd.
- Shijie Wu and Mark Dredze. 2019. [Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.
- Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Drohanova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke,

A Additional Tables and Figures

Dataset	Language	#Words	#Sentences	Avg. sent. length	Avg. word length
GECO	English	52131	4041	12.90	4.60
MECO	English	2092	99	21.13	5.32
	Dutch	2226	112	19.88	5.54
	German	2019	115	17.56	6.38
	Finnish	1462	110	13.29	8.19
	Estonian	1542	112	13.77	7.35
	Norwegian	2106	116	18.16	5.62
	Italian	2111	90	23.46	5.70
	Spanish	2412	98	24.61	5.01
	Greek	2082	99	21.03	5.67
	Turkish	1696	104	16.31	6.92
	Russian	1827	101	18.09	6.53
	Hebrew	1943	121	16.06	4.89
	Korean	1699	101	16.82	3.21

Table 3: Size characteristics for the reading materials of GECO and MECO. GECO sentences which are shorter than five words are removed to ensure that the model sees an adequate amount of complex structures during training.

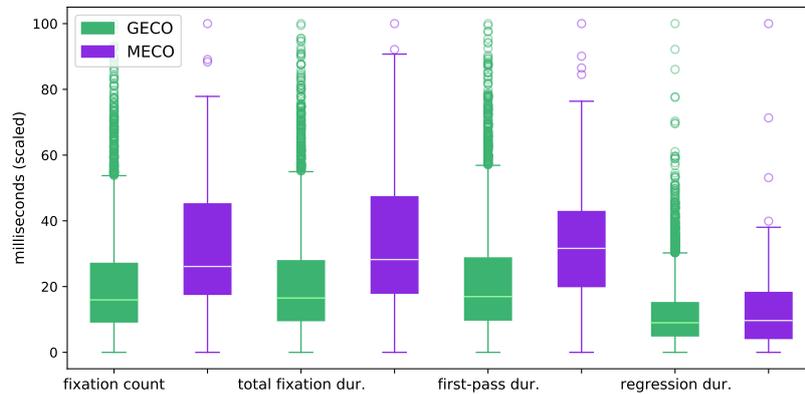


Figure 7: Distribution of four sentence-level eye-tracking metrics in English parts of GECO and MECO. All metrics are scaled between 0-100.

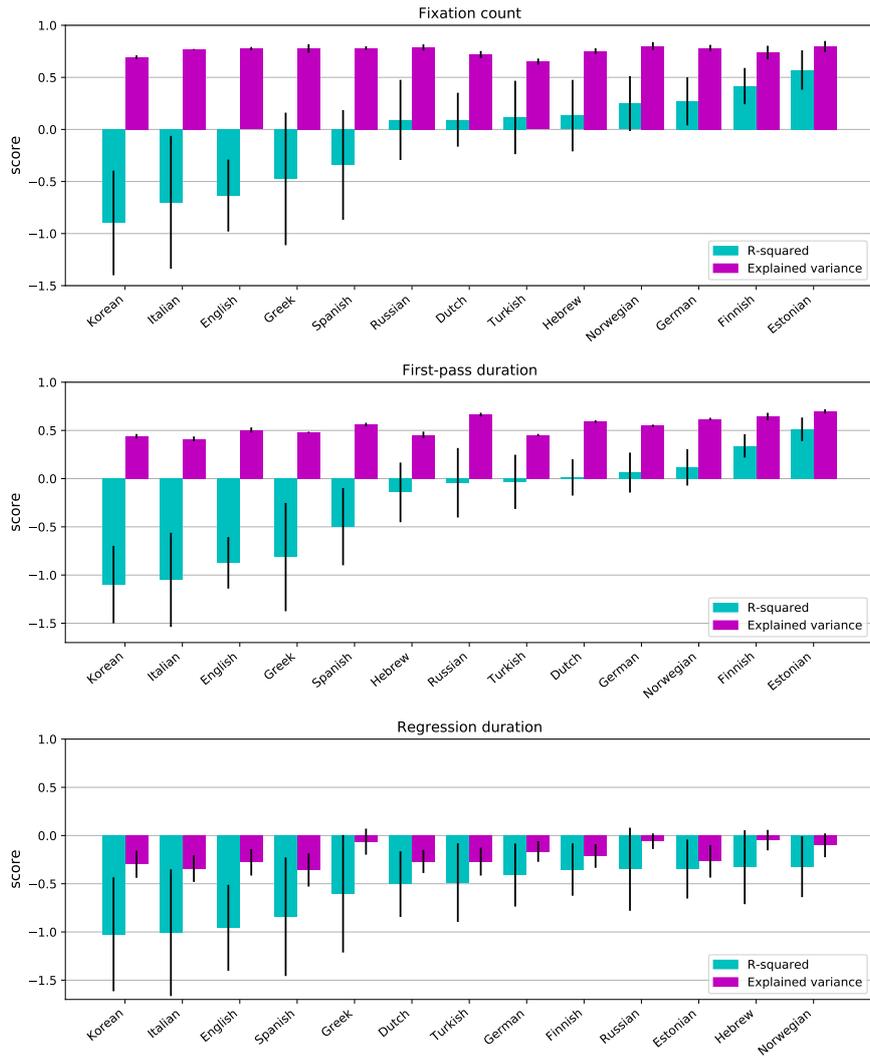


Figure 8: Cross-lingual transfer results for predicting cognitive processing complexity (i.e. fixation count, first-pass duration and regression duration). Prediction performance is evaluated with explained variance and R^2 for each language in MECO. The results are averaged over 5 folds; error bars denote the standard deviation over folds.

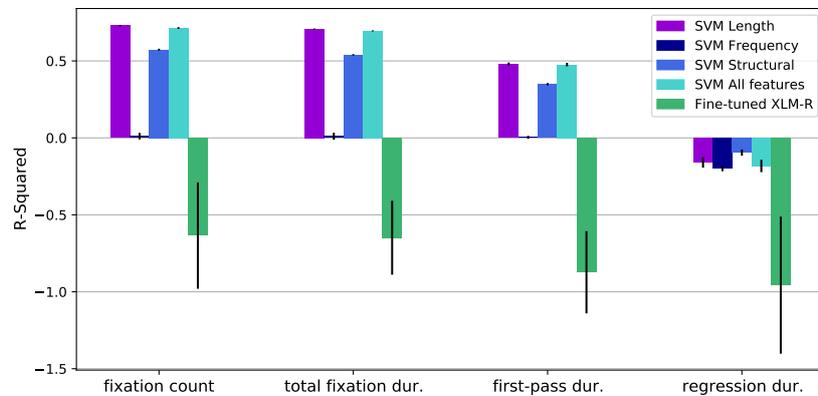


Figure 9: R^2 of the four feature-based SVM models and the fine-tuned XLM-R model. The models are trained on GECO using 5-fold cross-validation and evaluated on the English part of MECO; error bars denote the standard deviation over folds.

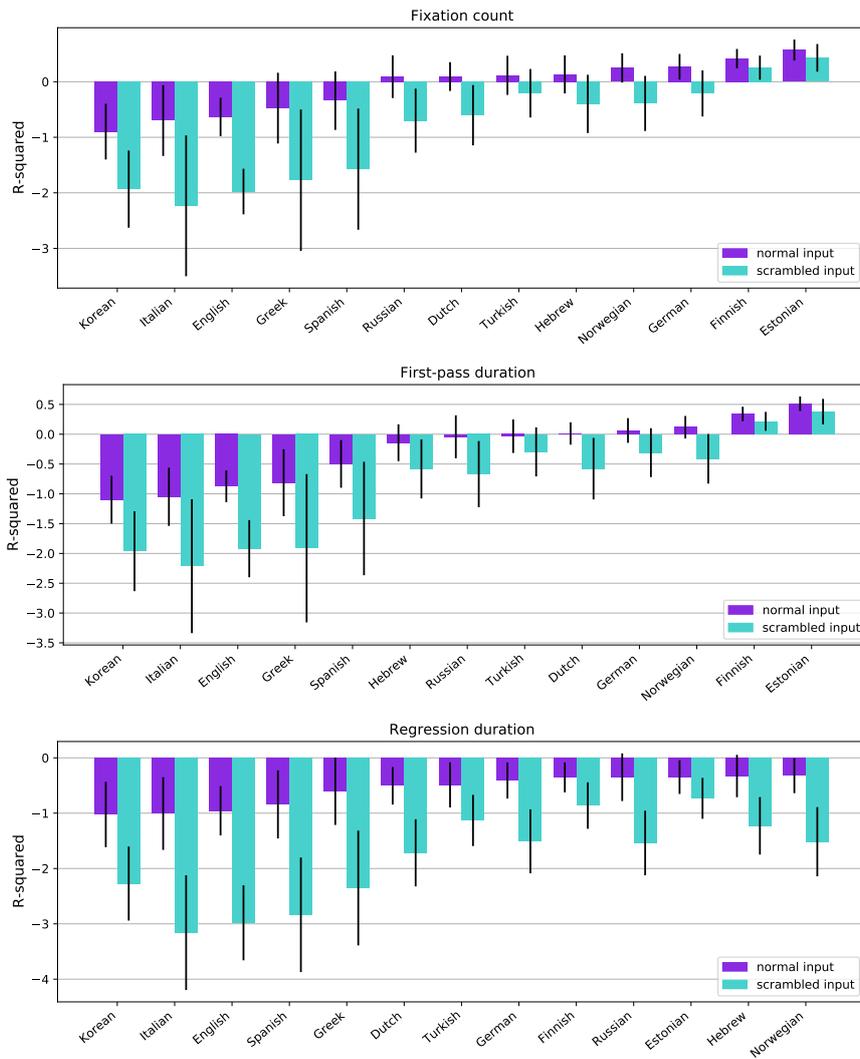


Figure 10: R^2 for fixation count, first-pass duration and regression duration for each language in MECO, both for sentences with normal and scrambled word order. The results are averaged over 5 folds; error bars denote the standard deviation.

Does Transliteration Help Multilingual Language Modeling?

Ibraheem Muhammad Moosa
Pennsylvania State University
ibraheem.moosa@psu.edu

Mahmud Elahi Akhter and Ashfia Binte Habib
Dept. of Electrical and Computer Engineering,
North South University, Dhaka, Bangladesh
{mahmud.akhter01, ashfia.habib}@northsouth.edu

Abstract

Script diversity presents a challenge to Multilingual Language Models (MLLM) by reducing lexical overlap among closely related languages. Therefore, transliterating closely related languages that use different writing scripts to a common script may improve the downstream task performance of MLLMs. We empirically measure the effect of transliteration on MLLMs in this context. We specifically focus on the Indic languages, which have the highest script diversity in the world, and we evaluate our models on the IndicGLUE benchmark. We perform the Mann-Whitney U test to rigorously verify whether the effect of transliteration is significant or not. We find that transliteration benefits the low-resource languages without negatively affecting the comparatively high-resource languages. We also measure the cross-lingual representation similarity of the models using centered kernel alignment on parallel sentences from the FLORES-101 dataset. We find that for parallel sentences across different languages, the transliteration-based model learns sentence representations that are more similar.

1 Introduction

In the last few years, we have seen impressive advances in many NLP tasks. These advances have been primarily led by the availability of large representative corpora and improvement in the architecture of large language models. While improving model architectures, training methods, regularization techniques, etc., can help advance the state of NLP in general, the unavailability of large, diverse corpora is the bottleneck for most languages (Joshi et al., 2020). Thus to inclusively advance the state of NLP across languages, it is crucial to develop techniques for training MLLMs that can extract the most out of existing multilingual corpora. Here, we focus on the issue of diverse writing scripts used by

closely related languages that may prevent MLLMs from learning good cross-lingual representations. Previous papers (Pfeiffer et al., 2021) have noted that low-resource languages that use unique scripts tend to have very few tokens representing them at the tokenizer. As a result, these languages tend to have more *UNK* tokens, and the words in these languages tend to be more split up by subword tokenizers. Often we can easily transliterate from one script to another using rule-based systems. For example, there are established standards that can be used to transliterate Greek (ISO 843), Cyrillic (ISO 9), Indic scripts (ISO 15919), and Thai (ISO 11940) to the Latin script.

In this paper, we focus on the Indic languages, which have the highest script diversity in the world. Many South Asian and Southeast Asian languages are intimately connected linguistically, historically, phonologically (Littell et al., 2017) and phylogenetically. However, due to different scripts, it is difficult for MLLMs to fully exploit this shared information. Among the Indic languages we considered in this study we encounter eleven different scripts. These are shown in Table 1. Nevertheless, these scripts have shared ancestry from the ancient Brahmic script (Hockett et al., 1997; Coningham et al., 1996) and have similar structures that we can easily use to transliterate them to a common script. Also, many of these languages heavily borrow from Sanskrit, and due to its influence, many words are shared among these languages. Therefore, due to their relatedness and highly diverse script barrier, the Indic languages presents a unique opportunity to analyze the effects of transliteration on MLLMs.

We empirically measure the effect of transliteration on the downstream performance of MLLMs. We pretrain ALBERT (Base, 11M Parameters) (Lan et al., 2020) and RemBERT (Base, 192M Parameters) (Chung et al., 2020) models from scratch

on Indic languages. We pretrain two variants of each model, one with the original writing scripts and the other after transliterating to a common writing script. Henceforth, we will refer to the transliterated script model as uni-script model and the other as a multi-script model. We evaluate the models on downstream tasks from the IndicGLUE benchmark dataset (Kakwani et al., 2020). In order to rigorously compare the two models, we finetune using nine random seeds on all downstream tasks. Then we perform the Mann-Whitney U test (MWU) between the uni-script and multi-script models. Using the MWU test, we conclude that transliteration significantly benefits the low-resource languages without negatively affecting the comparatively high-resource languages.

We also measure the Cross-Lingual Representation Similarity (CLRS) to understand why the uni-script model performs better than the multi-script model. To measure the CLRS, we use the centered kernel alignment (CKA) (Kornblith et al., 2019) similarity score. We measure the CKA similarity score between the hidden representations of the models on the parallel sentences of the Indic languages from the FLORES-101 dataset (Goyal et al., 2022). We find that, compared to the multi-script models, the uni-script models achieve a higher CKA score, and it is more stable throughout the hidden layers of the models. Based on this, we conclude that the uni-script models learn better cross-lingual representation than the multi-script models. In summary, our contributions are primarily three-fold:

1. We find that transliteration significantly benefits the low-resource languages without negatively affecting the comparatively high-resource languages.
2. We establish this finding through rigorous experiments and show the statistical significance along with the effect size of transliteration using the Mann-Whitney U test.
3. Using CKA on the FLORES-101 dataset, we show that transliteration helps MLLMs learn better cross-lingual representation.

Our code is available at Github¹ and our model

¹<https://github.com/ibraheem-moosa/XLM-Indic>

weights can be downloaded from HF Hub^{2 3 4 5}.

2 Motivation and Background

2.1 Motivation

In their study, Joshi et al. (2020) showed the resource disparity between low-resource and high-resource languages, and Ruder (2020) discussed the necessity of working with low-resource languages. A large body of work suggests that language-relatedness can help MLLMs achieve better performance on low-resource languages by leveraging related high-resource languages. For instance, Pires et al. (2019) found that lexical overlap improved mBERT’s multilingual representation capability even though it learned to capture multilingual representations with zero lexical overlaps. Dabre et al. (2017) showed that transfer learning in the same or linguistically similar language family gives the best performance for NMT. Lauscher et al. (2020) found that language relatedness is crucial for POS-tagging and dependency parsing tasks. Although, corpus size is much more important for NLI and Question Answering tasks. Wu and Dredze (2020) showed that bilingual BERT outperformed monolingual BERT on low-resource languages when the languages were linguistically closely related. Nevertheless, mBERT outperformed bilingual BERT on low-resource languages.

2.2 Script Barrier in Multilingual Language Models

One of the major challenges in leveraging transfer between high-resource and low-resource languages is overcoming the script barrier. Script barrier exists when multiple closely related languages use different scripts. Anastasopoulos and Neubig (2019) found that for morphological inflection, script barrier between closely related languages impedes cross-lingual learning, and language relatedness improved cross-lingual transfer. Transliteration and phoneme-based techniques have been proposed to solve this issue. For example, Murikinati et al. (2020) expanded upon Anastasopoulos and Neubig (2019) and showed that both transliter-

²<https://huggingface.co/ibraheemmoosa/xlmindic-base-uniscript>

³<https://huggingface.co/ibraheemmoosa/xlmindic-base-multiscript>

⁴<https://huggingface.co/ibraheemmoosa/xlmindic-rembert-uniscript>

⁵<https://huggingface.co/ibraheemmoosa/xlmindic-rembert-multiscript>

ation and grapheme to phoneme (g2p) conversion removes script barrier and improves cross-lingual morphological inflection and Rijhwani et al. (2019) showed that pivoting low-resource languages to their closely related high-resource languages results in better zero shot entity linking capacity and used phoneme-based pivoting to overcome the script barrier. Bharadwaj et al. (2016) showed that phoneme representation outperformed orthographic representations for NER. Chaudhary et al. (2018) also used phoneme representation to resolve script barriers and adapt word embeddings to low-resource languages.

2.3 Transliteration in Language Modeling

Different works have applied transliteration in different aspect for language models. For instance, Goyal et al. (2020) and Song et al. (2020) both utilized transliteration and showed that language relatedness was required for improving performance on NMT. Amrhein and Sennrich (2020) studied how transliteration improved NMT and came to the conclusion that transliteration offered significant improvement for low-resource languages with different scripts.

Khemchandani et al. (2021) showed on Indo-Aryan languages that language relatedness could be exploited through transliteration along with bilingual lexicon-based pseudo-translation and aligned loss to incorporate low-resource languages into pretrained mBERT. Muller et al. (2021) showed that for unseen languages, the script barrier hindered transfer between low-resource and high-resource languages for MLLMs and transliteration removed this barrier. They showed that transliterating Uyghur, Buryat, Erzya, Sorani, Meadow Mari, and Mingrelian to Latin script and finetuning mBERT on the respective corpus with masked language modeling objective improved their downstream POS performance significantly. In contrast, K et al. (2020) and Artetxe et al. (2020) proposes that mBERT can learn cross-lingual representations without any lexical overlap, a shared vocabulary, or joint training. However, these works focus on zero-shot cross-lingual transfer learning only. From the literature, it can be seen that many in the community believe transliteration to be a potential solution for script barriers. However, most of the work shows the benefits of transliteration for NMT. Nevertheless, there is no solid empirical analysis of the effects of transliteration for MLLMs

apart from Dhamecha et al. (2021); Muller et al. (2021). Hence, the motivation behind this paper is to provide a solid empirical analysis of the effect of transliteration for MLLMs with statistical analysis and determine whether or not it helps models learn better cross-lingual representation.

It should also be noted that, even though our idea seems to be similar to Muller et al. (2021) and Dhamecha et al. (2021), there are major differences. For instance, Muller et al. (2021) adapted existing pretrained model to very low-resource languages. Whereas, we focus on training the models with transliteration from scratch. We also train our models on 20 languages and evaluate on more than 50 tasks. Unlike Dhamecha et al. (2021), we also include Dravidian Languages in our analysis. Furthermore, we focused on the issue of script barrier while Dhamecha et al. (2021) focused on multilingual fine-tuning. Whereas, we adopt multilingual fine-tuning on all our models. Thus the improvement we see comes only from circumventing the script barrier. Moreover, we have provided statistical testing to show the significance of transliteration instead of just showing better metrics. We also performed cross-lingual representation similarity analysis to show the benefits of transliteration.

2.4 Cross Lingual Similarity Learning in Language Modeling

Several techniques have recently been used to study the hidden representations of multilingual language models. Kudugunta et al. (2019) study CLRS of NMT models using SVCCA (Raghu et al., 2017). Singh et al. (2019) used PWCCA (Morcos et al., 2018) to study the CLRS of mBERT and found that it drastically fell with depth. (Conneau et al., 2020) have used CKA to study the CLRS of bilingual BERT models. They found that similarity is highest in the first few layers and drops moderately with depth. Müller et al. (2021) used CKA to study CLRS of mBERT before and after finetuning on downstream tasks. They found in all cases that CLRS increases steadily in the first five layers, then it decreases in the later layers. From this, they concluded that mBERT learns multilingual alignment in the early layers and preserves it throughout finetuning. Del and Fishel (2021) applied various similarity measures to understand CLRS of various multilingual masked language models. Their results also show that CLRS increases in the first half of the models, while in the later layers, this

similarity steadily falls.

3 Experiment and Results

3.1 Mann–Whitney U test

We perform Mann–Whitney U test (MWU) (Mann and Whitney, 1947; Wilcoxon, 1945) to determine if the performance differences between the multi-script and the uni-script models are significant. In short, it tells us the effect of transliteration on model performance. MWU is a non-parametric hypothesis test between two groups/populations. MWU is chosen because it has weak assumptions. The only assumptions of MWU are that the samples of the two groups are independent of each other, and the samples are ordinal. Under the MWU, our null hypothesis or \mathbf{h}_0 is that the performances of the uni-script (group 1) and the multi-script (group 2) models are similar, and the alternative hypothesis or \mathbf{h}_a is that the performances (groups) are different. We set our confidence interval α at 0.05 and reject the \mathbf{h}_0 for the p-values $< \alpha$. We also report three test statistics as the p-value only gives statistical significance, which can be misleading at times (Sullivan and Feinn, 2012).

The test statistics are three different effect sizes that convey three different information. These test statistics are absolute effect size (δ), common language effect size (ρ), and standardized effect size (r). The absolute effect size δ is the difference between the mean of the models’ performance metric, which is given as,

$$\delta = \mu_{\text{uni-script}} - \mu_{\text{multi-script}}$$

for any given task and language. When the \mathbf{h}_0 is rejected for any given task, a positive δ indicates the uni-script model is better, and a negative δ indicates the multi-script model is better. The details and results of common language effect size (ρ), and standardized effect size (r) are presented in appendix D.

3.2 Dataset

The ALBERT models were pretrained on a subset of the OSCAR corpus containing Indo-Aryan languages. We use the unshuffled deduplicated version of OSCAR corpus (Ortiz Su’arez et al., 2019) available via Huggingface datasets library (Lhoest et al., 2021). We pretrain on Panjabi, Hindi, Bengali, Oriya, Assamese, Gujarati, Marathi, Sinhala, Nepali, Sanskrit, Goan Konkani, Maithili, Bihari, and Bishnupriya portion of the OSCAR corpus.

The RemBERT models were trained on a significantly larger pretraining corpus with additional languages. We pretrained the RemBERT models on a combination of Wikipedia (Foundation), mC4 (Raffel et al., 2019), OSCAR2109 (Abadji et al., 2021) and OSCAR corpus. These datasets are also available via the Huggingface datasets library. In addition to the languages in the ALBERT pretraining corpus, we consider English, four Dravidian languages Kannada, Telugu, Malayalam, and Tamil, and an Indo-Aryan language Dhivehi. We evalu-

Lang.	Sub-family	Script	Size(GB)
en	Germanic	Latin	131
hi	Central Indo-Aryan	Devanagari	43
mr	Southern Indo-Aryan	Devanagari	35
bn	Eastern Indo-Aryan	Bengali	28
ta	South Dravidian	Tamil	22
ml	South Dravidian	Malayalam	10
te	South-Central Dravidian	Telugu	7
kn	South Dravidian	Kannada	6
si	Insular Indo-Aryan	Sinhala	5
ne	Northern Indo-Aryan	Devanagari	4
gu	Western Indo-Aryan	Gujarati	3.5
pa	Northwestern Indo-Aryan	Gurmukhi	2
or	Eastern Indo-Aryan	Oriya	0.5
sa	Sanskrit	Devanagari	0.2
as	Eastern Indo-Aryan	Bengali	0.1
dv	Insular Indo-Aryan	Thaana	0.1
bpy	Eastern Indo-Aryan	Bengali	< 0.1
gom	Southern Indo-Aryan	Devanagari	< 0.1
bh	Eastern Indo-Aryan	Devanagari	< 0.1
mai	Eastern Indo-Aryan	Devanagari	< 0.1

Table 1: Languages in our pretraining corpus and their writing scripts and the pretraining corpus sizes used for the RemBERT model

ate the models on four downstream tasks from IndicGLUE (Kakwani et al., 2020), which are News Article Classification, WSTP, CSQA, and NER. We use the balanced Wikiann dataset from Rahimi et al. (2019) for NER. In addition, we evaluate the models on other publicly available datasets that are part of the IndicGLUE benchmark. These are BBC Hindi News Classification, Soham Bengali News Classification, INLTK Headlines Classification, IITP Movie, and Product Review Sentiment Analysis (Akhtar et al., 2016), MIDAS Discourse Mode Classification (Dhanwal et al., 2020) and ACTSA Sentiment Classification (Mukku and Mamidi, 2017) datasets.

3.3 Transliteration Method

We transliterate Indic language texts to Latin script using the ISO 15919 transliteration scheme. We tested with two publicly available implementations of this scheme, Aksharamukha (Rajan, 2015) and PyICU (PyICU). We found the quality of translit-

eration of the Aksharamukha library to be better. Thus we use this library for transliterating the inputs to the ALBERT uni-script model. However, the Aksharamukha implementation is very slow compared to the PyICU implementation. As we significantly expanded our pretraining corpus for the RemBERT model, we switched to PyICU for the RemBERT uni-script model.

3.4 Downstream Finetuning

We finetune the models on each downstream task independently. The specific hyperparameters used for each task are reported in the appendix B. On all tasks, we finetune with nine random seeds and report the average and standard deviation of the metrics. In Table 2 and Table 4, we report the performances on IndicGLUE benchmark tasks and in Table 3 on other publicly available datasets. Here, we discuss the results on each of the the models on each of the tasks. Furthermore, in appendix D, we show the test statistics for all the datasets.

Wikipedia Section Title Prediction: For both RemBERT and ALBERT models, the uni-script model performed better on all languages except Malayalam (ml). We noticed that a letter of Malayalam script is not properly transliterated by the PyICU library. This introduced some artifacts in the form of unnecessary splitting of words by the subword tokenizer.

News Category Classification: It is interesting that on this task the uni-script models performed better for Panjabi (pa) and Oriya (or) languages. It is clear from Table 1 that these two languages are low-resource compared to Bengali (bn) and Marathi (mr). On Bengali and Marathi we see slight performance degradation which is not statistically significant. This shows the validity of our first finding.

Named Entity Recognition: On this task we see that the uni-script model performs much better for Assamese (as), Oriya(or), Panjabi (pa) and Gujarati (gu). These languages are low-resource and here again the uni-script model shines. The large performance improvement on this task can be explained by the fact that Named Entities usually have the same spelling after transliteration for Indian languages. Thus the uni-script model has better chances for learning various named-entities during pre-training.

Article Genre, Sentiment & Discourse Mode Classification: We evaluate the models on various

other sequence classification datasets that are part of the IndicGLUE benchmark. Here again the uni-script model usually performs better than the multi-script model. However for two tasks in Malayalam (ml) and Tamil (ta) we see better performance for the multi-script model. We already mentioned that there is some tokenization issue for Malayalam which can explain the results for Malayalam. The results for Tamil suggests that it may be a good idea to try both uni-script and multi-script model if they are available to see which performs best on a particular task. However this is the only instance of a task where we see the multi-script model perform better.

3.5 Zero Shot Capability Testing

We use the CSQA task to test the zero-shot capability of the models as we can use the models without finetuning. This task is designed to test whether language models can be used as knowledge bases (Petroni et al., 2019). In Table 4 we report the results. We note that the RemBERT models perform much better than the ALBERT models on this task. This is expected as the ALBERT models' memorization capability is hampered by weight sharing.

The ALBERT uni-script model is better on all languages compared to the ALBERT multi-script model. This shows the potential of a uni-script model in a restricted low parameter situation. For the RemBERT models, the results are mixed. However, on average the uni-script model performs better than the multi-script model. The worst results are for Malayalam (ml) which as we mentioned before has some tokenization issues.

4 Cross-lingual Representation Similarity

In this section, we analyze why the uni-script model performs better than the multi-script model from the perspective of Cross-Lingual Representation Similarity. Following (Müller et al., 2021), (Conneau et al., 2020) and (Del and Fishel, 2021) we apply CKA to measure CLRS. We use the CKA implementation from the Ecco library (Alammar, 2021). We use parallel sentences on thirteen languages from the FLORES-101 (Goyal et al., 2022) dataset. For the ALBERT models, which are trained on only the Indo-Aryan languages, we only consider Panjabi, Hindi, Bengali, Oriya, Assamese, Gujarati, Marathi, and Nepali sentences. For the RemBERT models, we additionally consider Kannada, Telugu, Malayalam, Tamil, and English sentences.

Model	pa	hi	bn	or	as	gu	mr	kn	te	ml	ta	avg
Wikipedia Section Title Prediction												
RemBERT _{MS}	68.42±0.92	70.90±0.39	72.58±0.45	69.92±0.90	68.37±1.37	72.93±0.58	73.23±0.61	71.67±0.41	92.98±0.19	69.03±0.57	69.77±0.45	73.00
RemBERT _{US}	71.01±0.22	72.45±0.29	73.65±0.21	75.37±0.69	72.50±0.91	76.35±0.29	74.58±0.72	74.21±0.29	93.66±0.09	69.33±0.35	70.63±0.22	74.89
δ	2.59	1.55	1.07	5.45	4.13	3.42	1.34	2.54	0.68	0.31	0.86	1.89
p -value	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0035	0.0004	0.0004	0.2505	0.0006	-
ALBERT _{MS}	74.33±0.83	78.18±0.33	81.18±0.28	74.35±1.2	76.70±0.83	76.37±0.53	79.10±0.84	-	-	-	-	77.17
ALBERT _{US}	77.55±0.61	82.24±0.18	84.38±0.29	81.47±0.99	81.74±0.82	82.39±0.27	82.74±0.52	-	-	-	-	81.78
δ	3.22	4.06	3.20	7.12	5.04	6.02	3.64	-	-	-	-	4.61
p -value	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	-	-	-	-	-
News Category Classification												
RemBERT _{MS}	95.67±0.38	-	97.90±0.17	96.59±0.18	-	98.22±0.58	99.16±0.16	97.23±0.10	99.03±0.12	91.25±0.43	97.33±0.18	96.93
RemBERT _{US}	96.92±0.29	-	97.78±0.12	97.55±0.14	-	99.02±0.14	99.14±0.21	97.10±0.12	99.03±0.66	92.08±0.40	97.49±0.20	97.34
δ	1.24	-	-0.11	0.95	-	0.80	-0.03	-0.13	0.00	0.83	0.16	0.41
p -value	0.0003	-	0.0981	0.0004	-	0.0040	0.7783	0.0995	0.7548	0.0014	0.0814	-
ALBERT _{MS}	96.83±0.19	-	98.14±0.14	98.09±0.16	-	98.80±0.43	99.58±0.25	-	-	-	-	98.30
ALBERT _{US}	97.90±0.17	-	97.99±0.22	98.77±0.12	-	99.40±0.54	99.47±0.21	-	-	-	-	98.70
δ	1.07	-	-0.15	0.68	-	0.60	-0.18	-	-	-	-	0.40
p -value	0.0003	-	0.181	0.0004	-	0.03084	0.1683	-	-	-	-	-
Named Entity Recognition (F1-Score)												
RemBERT _{MS}	69.47±1.72	90.95±0.33	95.51±0.18	87.92±1.26	79±0.22	69±0.94	90.72±0.17	72.65±1.81	81.82±1.81	89.17±0.25	90.07±0.33	83.40
RemBERT _{US}	81.91±1.93	91.73±0.39	96.19±0.21	88.92±2.88	83.50±2.75	80.25±1.42	90.75±0.35	78.98±1.50	84.97±0.45	89.26±0.46	90.18±0.27	86.97
δ	12.44	0.78	0.68	1.00	4.28	10.31	0.02	6.33	3.15	0.01	0.12	3.56
p -value	0.00004	0.0005	0.00001	0.1615	0.0019	0.00004	0.6665	0.00004	0.00004	0.7304	0.2973	-
ALBERT _{MS}	76.69±1.5	91.80±0.42	96.39±0.19	84.18±1.8	75.45±1.8	69.10±2.9	88.72±0.40	-	-	-	-	83.19
ALBERT _{US}	85.42±1.9	92.93±0.21	97.31±0.22	93.54±0.58	89.06±2.2	80.16±0.15	90.56±0.44	-	-	-	-	89.85
δ	8.73	1.13	0.92	9.36	13.61	11.06	1.84	-	-	-	-	6.66
p -value	0.0004066	0.0004066	0.0003983	0.0004038	0.000401	0.0004066	0.0004095	-	-	-	-	-

orange indicates the multi-script and uni-script models are equal and blue indicates the uni-script model is better

Table 2: Results on Classification Tasks from IndicGLUE Benchmark

Language	Dataset	RemBERT _{MS}	RemBERT _{US}	δ	p -value	ALBERT _{MS}	ALBERT _{US}	δ	p -value
Article Genre Classification									
hi	BBC News	76.80±0.84	77.78±0.92	0.98	0.0466	77.28±1.51	79.14±0.60	1.86	0.0088
bn	Soham News Article Classification	92.86±0.10	93.69±0.20	0.83	0.0004	93.22±0.49	93.89±0.48	0.67	0.0090
gu	INLTK Headlines	90.27±0.47	91.60±0.28	1.33	0.0004	90.41±0.69	90.73±0.75	0.32	0.6249
mr	INLTK Headlines	91.24±0.50	92.27±0.39	1.03	0.0008	92.21±0.23	92.04±0.47	-0.17	0.3503
ml	INLTK Headlines	94.11±0.49	93.33±0.22	-0.78	0.003	-	-	-	-
ta	INLTK Headlines	95.59±0.70	94.93±0.30	-0.65	0.013	-	-	-	-
Sentiment Analysis									
hi	IITP Product Reviews	72.17±1.98	72.85±0.63	0.68	0.9646	76.33±0.84	77.18±0.77	0.85	0.04099
hi	IITP Movie Reviews	58.66±1.09	62.65±2.74	3.99	0.0023	65.91±2.2	66.34±0.16	0.15	0.8941
te	ACTSA	61.18±1.38	60.53±0.85	-0.66	0.1981	-	-	-	-
Discourse Mode Classification									
hi	MIDAS Discourse	78.07±0.83	79.46±0.67	1.39	0.0415	78.39±0.33	78.54±0.91	0.15	0.7561

orange indicates the multi-script and uni-script models are equal, cyan indicates multi-script is better than uni-script models and blue indicates vice versa

Table 3: Accuracy on Public Datasets

Model	pa	hi	bn	or	as	gu	mr	ta	te	ml	kn	avg
Cloze-style QA (Zero Shot)												
RemBERT _{MS}	33.93	39.06	38.93	37.32	37.66	84.21	46.15	37.02	34.42	38.45	40.75	42.53
RemBERT _{US}	33.92	40.10	39.62	38.28	39.26	85.37	45.92	36.68	34.36	37.16	44.29	43.17
δ	-0.01	1.04	0.69	0.96	1.6	1.16	-0.23	-0.34	-0.06	-1.29	3.54	0.64
ALBERT _{MS}	31.04	36.72	35.19	34.63	33.92	59.86	36.14	-	-	-	-	38.21
ALBERT _{US}	32.77	38.52	36.38	36.00	37.36	70.22	39.53	-	-	-	-	41.54
δ	1.73	1.8	1.19	1.37	3.44	10.36	3.39	-	-	-	-	3.33

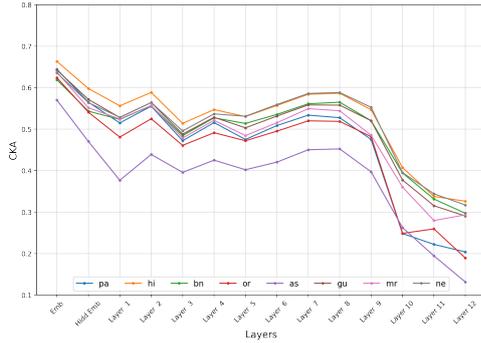
cyan indicates multi-script is better than uni-script models and blue indicates vice versa

Table 4: Test accuracy on CSQA

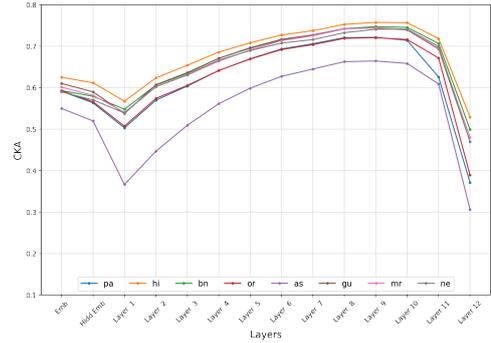
First, we calculate the sentence embeddings of these parallel sentences from the models. Sentence embedding is calculated by averaging the hidden state representations of the tokens. Then, we calculate the CKA similarity score between the sentence embeddings for each language pair. For each language, we average its CKA similarity scores. In Figure 1 we plot this average CKA similarity for

each layer of the models.

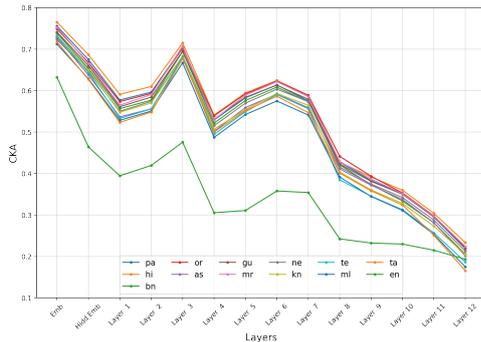
We see that CLRS score drops significantly at the last layer for all models. However, the uni-script models retain high CLRS score until the eleventh layer, whereas the multi-script models have low CLRS score from the ninth layer. Overall the CLRS score of the uni-script models are more stable. This indicates that the uni-script models have learned



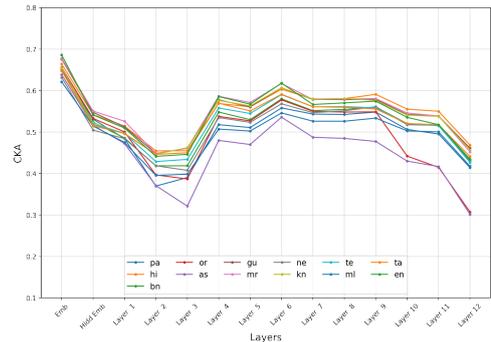
(a) ALBERT_{MS}



(b) ALBERT_{US}



(c) RemBERT_{MS}



(d) RemBERT_{US}

Figure 1: CKA Similarity Score for the multi-script and uni-script models

better cross-lingual representations.

5 Tokenizer Quality Analysis

In terms of performance, we expect the transliteration model to exploit better tokenization across the languages. Following (Ács, 2019) and (Rust et al., 2021), we measure the subword fertility (average number of tokens per word) and the ratio of words unbroken by the tokenizer. From figure 2, we can see that transliteration reduces the splitting of words. This indicates that many words that were represented by different tokens in the multi-script model are represented by a single token in the transliteration model. On average, the ALBERT uni-script tokenizer has a lower subword fertility score of 1.55 compared to the multi-script tokenizer’s 1.825. The uni-script tokenizer also has a lower proportion of continued word score of 0.36 while the multi-script tokenizer has a score of 0.45.

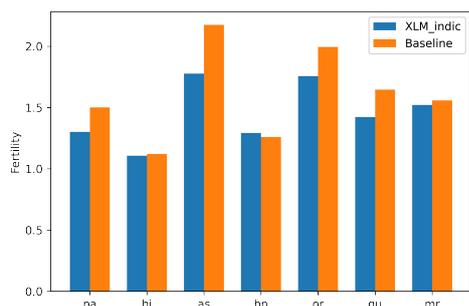
6 Conclusion and Future Work

In this paper, we show that transliterating closely related languages to a common script improves multilingual language model performance and leads to better cross-lingual representations. We conducted

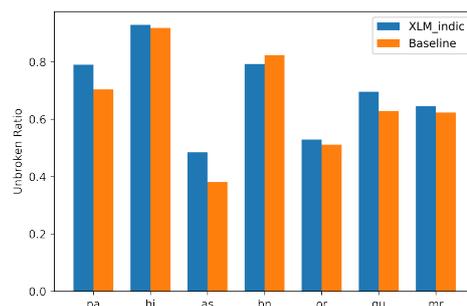
rigorous statistical analysis to quantify the significance and effect size of transliteration on downstream task performance. We found that transliteration especially improves performance on comparatively low-resource languages and did not hurt the performance on high-resource languages. This findings are in agreement with (Dhamecha et al., 2021; Muller et al., 2021). Our results indicate that in other scenarios where closely related languages use different scripts, transliteration can be used to improve the performance of language models. For example, Slavic and Turkic languages present similar scenarios. We would like to extend our study to models at different scales and more languages in the future. Also, another interesting future direction would be to just use the transliteration for pretraining signal but give the model the ability to deal with the original scripts.

Limitations

A limitation of our work is that it introduces a transliteration step into the model pipeline. Thus we need a stable implementation of the transliteration scheme. Thus the model can become tied to a specific version of the transliteration library. Also



(a) Subword Fertility.



(b) Unbroken Ratio.

Figure 2: Subword fertility (lower is better) and unbroken ratio (higher is better)

the transliteration scheme is not perfect as we saw for Malayalam, it introduced some artifacts. Finally given our limited computational budget, we could not run experiments with a lot of models at different scales. Thus the impact of transliteration over different model scales has not been explored. Even though our work has these limitations, it clearly shows transliteration as an important tool for training better multilingual models.

Ethics Statement

In their study, Joshi et al. (2020) showed the resource disparity between low-resource and high-resource languages, and (Ruder, 2020) also highlighted the necessity of working with low-resource languages. However, creating representative and inclusive corpora is a difficult task and an ongoing process and is not always possible for many low-resource languages. Thus to inclusively advance the state of NLP across languages, it is crucial to develop techniques for training MLLMs that can extract the most out of existing multilingual corpora. Hence, we believe our analysis might help MLLMs with low-resource languages in real-world applications. However, there is one ethical issue that we want to state explicitly. Even though we pre-train on a comparatively large multilingual corpus, the model may exhibit harmful gender, ethnic and political bias. If the model is fine-tuned on a task where these issues are important, it is necessary to take special consideration when relying on the model’s decisions.

References

Julien Abadji, Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2021. [Ungoliant: An optimized pipeline for the generation of a very large-](#)

[scale multilingual web corpus](#). In *CMLC-9, Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-9) 2021*. Limerick, 12 July 2021 (Online-Event), pages 1–9, Mannheim. Leibniz-Institut für Deutsche Sprache.

Judit Ács. 2019. [Exploring BERT’s Vocabulary](#). *Blog Post*.

Md. Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. [Aspect based sentiment analysis: Category detection and sentiment classification for hindi](#). In *proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2016)*, April 3-9, 2016, Konya, Turkey, pages 246–257. Association for Computational Linguistics.

J Alammur. 2021. [Ecco: An open source library for the explainability of transformer language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 249–257, Online. Association for Computational Linguistics.

Chantal Amrhein and Rico Sennrich. 2020. [On Romanization for model transfer between scripts in neural machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2461–2469, Online. Association for Computational Linguistics.

Antonios Anastasopoulos and Graham Neubig. 2019. [Pushing the limits of low-resource morphological inflection](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 984–996, Hong Kong, China. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. [On the cross-lingual transferability of monolingual representations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

- Akash Bharadwaj, David Mortensen, Chris Dyer, and Jaime Carbonell. 2016. [Phonologically aware neural model for named entity recognition in low resource transfer settings](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1472, Austin, Texas. Association for Computational Linguistics.
- Aditi Chaudhary, Chunting Zhou, Lori Levin, Graham Neubig, David R. Mortensen, and Jaime Carbonell. 2018. [Adapting word embeddings to new languages with morphological and phonological subword representations](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3285–3295, Brussels, Belgium. Association for Computational Linguistics.
- Hyung Won Chung, Thibault Févry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2020. [Rethinking embedding coupling in pre-trained language models](#). *CoRR*, abs/2010.12821.
- R.A.E. Coningham, F.R. Allchin, C.M. Batt, and D. Lucy. 1996. [Passage to india? anuradhapura and the early use of the brahmi script](#). *Cambridge Archaeological Journal*, 6(1):73–97.
- Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Emerging cross-lingual structure in pretrained language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6022–6034. Association for Computational Linguistics.
- Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. 2017. [An empirical study of language relatedness for transfer learning in neural machine translation](#). In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, pages 282–286. The National University (Phillippines).
- Maksym Del and Mark Fishel. 2021. [Establishing interlingua in multilingual language models](#). *CoRR*, abs/2109.01207.
- Tejas Dhamecha, Rudra Murthy, Samarth Bharadwaj, Karthik Sankaranarayanan, and Pushpak Bhattacharyya. 2021. [Role of Language Relatedness in Multilingual Fine-tuning of Language Models: A Case Study in Indo-Aryan Languages](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8584–8595, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Swapnil Dhanwal, Hritwik Dutta, Hitesh Nankani, Nilay Shrivastava, Yaman Kumar, Junyi Jessy Li, Debanjan Mahata, Rakesh Gosangi, Haimin Zhang, Rajiv Ratn Shah, and Amanda Stent. 2020. [An annotated dataset of discourse modes in Hindi stories](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1191–1196, Marseille, France. European Language Resources Association.
- Wikimedia Foundation. [Wikimedia downloads](#).
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. [The Flores-101 evaluation benchmark for low-resource and multilingual machine translation](#). *Transactions of the Association for Computational Linguistics*, 10:522–538.
- Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. 2020. [Efficient neural machine translation for low-resource languages via exploiting related languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 162–168, Online. Association for Computational Linguistics.
- Charles F. Hockett, Peter T. Daniels, and William Bright. 1997. [The world's writing systems](#). *Language*, 73(2):379.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. [The state and fate of linguistic diversity and inclusion in the NLP world](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.
- Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. [Cross-lingual ability of multilingual BERT: an empirical study](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. [IndicNLPsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.
- Yash Khemchandani, Sarvesh Mehtani, Vaidehi Patil, Abhijeet Awasthi, Partha Talukdar, and Sunita Sarawagi. 2021. [Exploiting language relatedness for low web-resource language model adaptation: An Indic languages study](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1312–1323, Online. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. [Similarity of neural network representations revisited](#). *CoRR*, abs/1905.00414.

- Taku Kudo and John Richardson. 2018. [Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.
- Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. [Investigating multilingual NMT representations at scale](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.
- Anoop Kunchukuttan. 2020. The Indic-NLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. [ALBERT: A lite BERT for self-supervised learning of language representations](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. [From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.
- Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. [Datasets: A community library for natural language processing](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. [URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.
- Daniel Lüdecke. 2020. *sjstats: Statistical Functions for Regression Models (Version 0.18.0)*.
- H. B. Mann and D. R. Whitney. 1947. [On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other](#). *The Annals of Mathematical Statistics*, 18(1):50 – 60.
- Ari S. Morcos, Maithra Raghu, and Samy Bengio. 2018. [Insights on representational similarity in neural networks with canonical correlation](#). In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5732–5741.
- Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. [On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Sandeep Sricharan Mukku and Radhika Mamidi. 2017. [ACTSA: Annotated corpus for Telugu sentiment analysis](#). In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 54–58, Copenhagen, Denmark. Association for Computational Linguistics.
- Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. [When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.
- Benjamin Müller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. [First align, then predict: Understanding the cross-lingual ability of multilingual BERT](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2214–2231. Association for Computational Linguistics.
- Nikitha Murikinati, Antonios Anastasopoulos, and Graham Neubig. 2020. [Transliteration for cross-lingual morphological inflection](#). In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 189–197, Online. Association for Computational Linguistics.
- Pedro Javier Ortiz Suárez, Benoit Sagot, and Laurent Romary. 2019. [Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures](#). In *Proceedings of the Workshop on Challenges in the Management of Large Corpora*

- (CMLC-7) 2019, Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. **Language models as knowledge bases?** In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.
- Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2021. **Unks everywhere: Adapting multilingual language models to new scripts.** In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10186–10203. Association for Computational Linguistics.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. **How multilingual is multilingual BERT?** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.
- PyICU. Pyicu transliteration tool. <https://pypi.org/project/PyICU/>. Accessed: 2022-03-15.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. **Exploring the limits of transfer learning with a unified text-to-text transformer.** *arXiv e-prints*.
- Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. **SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability.** In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6076–6085.
- Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. **Massively multilingual transfer for NER.** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.
- Vinodh Rajan. 2015. Aksharamukha transliteration tool. <https://github.com/virtualvinodh/aksharamukha-python>. Accessed: 2021-10-04.
- Shruti Rijhwani, Jiateng Xie, Graham Neubig, and Jaime G. Carbonell. 2019. **Zero-shot neural transfer for cross-lingual entity linking.** In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6924–6931. AAAI Press.
- Sebastian Ruder. 2020. **Why You Should Do NLP Beyond English.** <http://ruder.io/nlp-beyond-english>.
- Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. **How good is your tokenizer? on the monolingual performance of multilingual language models.** In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.
- Jasdeep Singh, Bryan McCann, Richard Socher, and Caiming Xiong. 2019. **BERT is not an interlingua and the bias of tokenization.** In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP 2019, Hong Kong, China, November 3, 2019*, pages 47–55. Association for Computational Linguistics.
- Haiyue Song, Raj Dabre, Zhuoyuan Mao, Fei Cheng, Sadao Kurohashi, and Eiichiro Sumita. 2020. **Pre-training via leveraging assisting languages for neural machine translation.** In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 279–285, Online. Association for Computational Linguistics.
- G. M. Sullivan and R. Feinn. 2012. **Using Effect Size-or Why the P Value Is Not Enough.** *J Grad Med Educ*, 4(3):279–282.
- Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. **SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.** *Nature Methods*, 17:261–272.
- Frank Wilcoxon. 1945. **Individual comparisons by ranking methods.** *Biometrics Bulletin*, 1(6):80.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu,

Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Trans-formers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. [Are all languages created equal in multilingual BERT?](#) In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

A Cloze Style QA Evaluation Method

Since a word can be tokenized to multiple tokens by the subword tokenizer, correctly evaluating the model on this task requires special care. Specifically, we have to use the same number of mask tokens as the number of subword tokens that a word gets split into. Then we calculate the probability for the word by multiplying the probability of the subword tokens predicted by the masked language model.

B Pretraining Details

Corpus Preparation: Since the OSCAR corpus contains raw text from the Web, we apply a few filtering and normalization. First, we discard entries where the dominant script does not match the language tag provided by the OSCAR corpus. Then we use the IndicNLP normalizer (Kunchukuttan, 2020) to normalize the raw text. For the uniscript model, we then transliterate all the text to ISO-15919 format using the Aksharamukha (Rajan, 2015) library.

For the RemBERT models we do not perform any of the filtering mentioned above since our pretraining corpus is comparatively very large. In this case, we use the PyICU library (PyICU) for transliterating to ISO-15919 format.

Tokenizer Training: For the ALBERT models, we train two SentencePiece tokenizers (Kudo and Richardson, 2018) on the transliterated and the non-transliterated corpus with a vocabulary size of 50,000. For the RemBERT models we train Unigram tokenizers from the Tokenizers library (Wolf et al., 2020) with a vocabulary size of 65,536.

ALBERT Model Training: We first pretrained an ALBERT base model from scratch on the non-transliterated corpus as our baseline. Afterward, we pretrained another ALBERT base from scratch on the transliterated corpus. We chose the base model due to computing constraints. We trained

the models on a single TPUv3 VM. Both models were trained using the same hyperparameters. We followed the hyperparameters used in (Lan et al., 2020) except for batch size and learning rate. The pretraining objective is also the same as (Lan et al., 2020). We used a batch size of 256, which is the highest that fits into TPU memory, whereas the ALBERT paper used a batch size of 4096. As our batch size is $1/16^{\text{th}}$ of the ALBERT paper, we use a learning rate of $1e-3/8$, which is approximately $1/16^{\text{th}}$ of the learning rate used in the ALBERT paper ($1.76e-2$). Additionally, we use the Adam optimizer (Kingma and Ba, 2015) instead of the LAMB optimizer. The rest of the hyperparameters were the same as the ALBERT paper. Specifically, we use a sequence length of 512 with absolute positional encoding, weight decay of $1e-2$, warmup steps of 5000, max gradient norm of 1.0, and Adam epsilon of $1e-6$. The models were trained for 1M steps. Each model took about 7.5 days to train. We use the ALBERT implementation from the Huggingface Transformers Library (Wolf et al., 2020).

RemBERT Model Training: We pretrained an RemBERT base models similar to the ALBERT models. We trained the models on a single TPUv3 VM. Both models were trained using the same hyperparameters. We followed the hyperparameters used in (Chung et al., 2020) except for batch size and learning rate. The pretraining objective is also the same as (Chung et al., 2020). We used a batch size of 256, which is the highest that fits into TPU memory, whereas the RemBERT paper used a batch size of 2048. As our batch size is $1/8^{\text{th}}$ of the RemBERT paper, we use a learning rate of $2e-4/8$, which is $1/8^{\text{th}}$ of the learning rate used in the RemBERT paper. Similar to the ALBERT model, we use the Adam optimizer (Kingma and Ba, 2015). The rest of the hyperparameters were the same as the RemBERT paper. Specifically, we use a sequence length of 512 with absolute positional encoding, weight decay of $1e-2$, warmup steps of 15000, max gradient norm of 1.0, and Adam epsilon of $1e-6$. The models were trained for 1M steps. Each model took about 7.5 days to train. We use the RemBERT implementation from the Huggingface Transformers Library (Wolf et al., 2020).

C Downstream Hyperparameters

Hyperparameters for downstream tasks are presented in Table 5 and Table 6.

Task	TPU	Batch Size	Learning Rate	Weight Decay	Dropout	Epochs	Warmup Ratio
News Category Classification	False	16	2e-5	0.01	0.1	20	0.10
Wikipedia Section-Title Prediction	True	256	2e-5	0.01	0.1	3	0.10
Named Entity Recognition	True	512	2e-5	0.01	0.1	20	0.10
BBC Hindi News Classification	False	16	2e-5	0.01	0.1	20	0.10
Soham Bengali News Classification	False	16	2e-5	0.01	0.1	8	0.10
INLTK Headlines Classification	False	256	2e-5	0.01	0.1	20	0.10
IITP Movie Review	False	64	5e-5	0.01	0.25	20	0.10
IITP Product Review	False	16	5e-5	0.01	0.5	20	0.10
MIDAS Discourse Mode	False	32	2e-5	0.01	0.5	20	0.10

Table 5: Hyperparameters for ALBERT models

Task	TPU	Batch Size	Learning Rate	Weight Decay	Dropout	Steps	Label Smoothing
News Category Classification	False	16	1e-5	0.1	0.1	2500	0.0
Wikipedia Section-Title Prediction	True	256	8e-6	0.1	0.1	12500	0.0
Named Entity Recognition	False	16	5e-5	0.1	0.1	10000	0.0
BBC Hindi News Classification	False	16	1e-5	0.01	0.1	2500	0.0
Soham Bengali News Classification	False	16	1e-5	0.1	0.1	2500	0.1
INLTK Headlines Classification	False	16	1e-5	0.1	0.1	5000	0.0
IITP Movie Review	False	16	1e-5	0.1	0.1	5000	0.0
IITP Product Review	False	16	1e-5	0.1	0.1	5000	0.0
ACTSA Sentiment Classification	False	16	1e-5	0.1	0.1	5000	0.0
MIDAS Discourse Mode	False	16	8e-6	0.1	0.1	2500	0.1

Table 6: Hyperparameters for RemBERT models

For the ALBERT models batch size was chosen to be the maximum that fits in memory. This was done so that each batch contains approximately the same number of tokens. Otherwise the hyperparameters were chosen following the recommendations of (Mosbach et al., 2021). On the highly skewed IITP Movie Review, IITP Product Review and MIDAS Discourse we found that this default setting resulted in worse performance compared to the independent baselines. So we finetuned the learning rate and classifier dropout on the validation set of these tasks.

For the RemBERT models learning rate, weight decay, dropout, steps and label smoothing were chosen based on grid search with a few values.

D Test Statistics Results

ρ gives us the probability of one group being better than the other group. That is the probability that a random performance sample of the uni-script model is greater than a random performance sample of the multi-script model. The last test statistic is r which indicates the magnitude of difference

between the performance values of the uni-script model (group 1) and the multi-script model (group 2). r shows us how realistically significant the performance differences are between models even if the performance difference is statistically significant. It gives us a value between 0 to 1 and its ranges are: **small effect** ($0 \leq r \leq 0.3$), **medium effect** ($0.3 < r \leq 0.5$) and **large effect** ($0.5 < r$). We performed MWU on all downstream tasks except CSQA. On CSQA, we only report the δ . The MWU is performed using the SciPy library (Virtanen et al., 2020), and the results are further validated using R (Lüdecke, 2020). These statistics are reported in Table 7 for the IndicGLUE classification tasks and in Table 8 for the public dataset classification tasks.

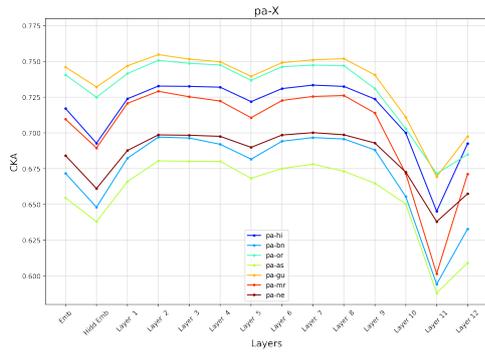
E Cross-lingual Similarity of ALBERT Models on All Language Pairs

Model	pa	hi	bn	or	as	gu	mr	kn	te	ml	ta
Wikipedia Section Title Prediction											
RemBERT _{ρ}	1	1	1	1	1	1	0.91	1	1	0.67	0.99
RemBERT _{r}	0.83	0.83	0.83	0.84	0.83	0.84	0.69	0.83	0.83	0.27	0.81
ALBERT _{ρ}	1	1	1	1	1	1	1	-	-	-	-
ALBERT _{r}	0.83	0.83	0.83	0.83	0.83	0.83	0.83	-	-	-	-
News Category Classification											
RemBERT _{ρ}	1	-	0.27	1	-	0.87	0.46	0.27	0.45	0.94	0.75
RemBERT _{r}	0.85	-	0.39	0.84	-	0.68	0.07	0.39	0.07	0.75	0.41
ALBERT _{ρ}	1	-	0.31	1	-	0.80	0.31	-	-	-	-
ALBERT _{r}	0.86	-	0.32	0.83	-	0.51	0.32	-	-	-	-
Named Entity Recognition											
RemBERT _{ρ}	1.00	0.95	0.99	0.70	0.91	1.00	0.57	1.00	1.00	0.56	0.65
RemBERT _{r}	0.83	0.75	0.81	0.33	0.69	0.83	0.10	0.83	0.83	0.08	0.25
ALBERT _{ρ}	1	1	1	1	1	1	1	-	-	-	-
ALBERT _{r}	0.83	0.83	0.83	0.83	0.83	0.83	0.83	-	-	-	-

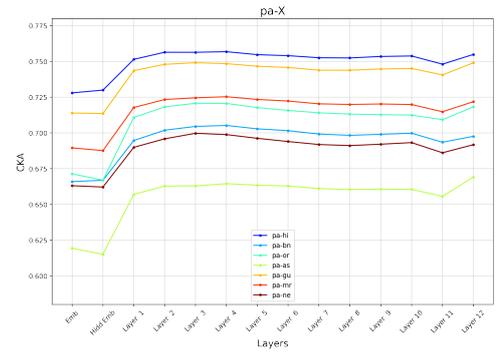
Table 7: Test Statistics on Classification Tasks from IndicGLUE Benchmark

Language	Dataset	RemBERT _{ρ}	RemBERT _{r}	ALBERT _{ρ}	ALBERT _{r}
Article Genre Classification					
hi	BBC News	0.78	0.47	0.87	0.62
bn	Soham News Article Classification	1	0.84	0.87	0.62
gu	INLTK Headlines	1	0.84	0.57	0.12
mr	INLTK Headlines	0.98	0.79	0.36	0.22
ml	INLTK Headlines	0.08	0.70	-	-
ta	INLTK Headlines	0.15	0.59	-	-
Sentiment Analysis					
hi	IITP Product Reviews	0.51	0.01	0.79	0.48
hi	IITP Movie Reviews	0.93	0.72	0.52	0.03
te	ACTSA	0.31	0.30	-	-
Discourse Mode Classification					
hi	MIDAS Discourse	0.79	0.48	0.45	0.07

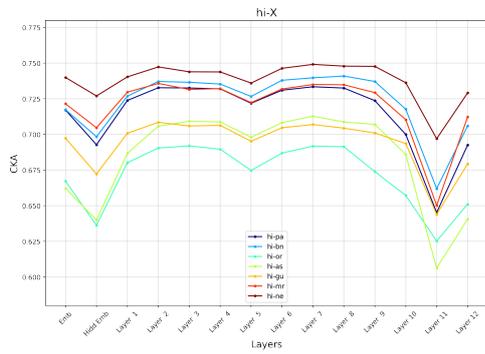
Table 8: Test Statistics on Public Datasets



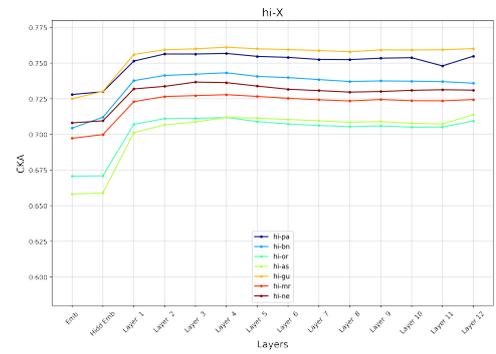
(a) multi-script PA-X



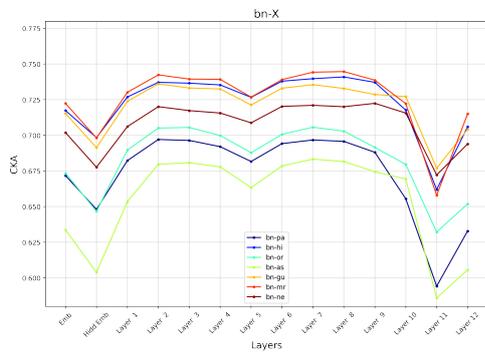
(b) uni-script PA-X



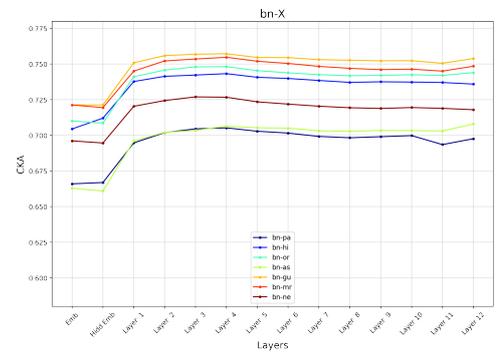
(c) multi-script HI-X



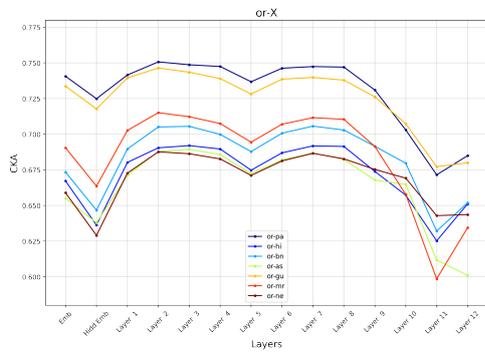
(d) uni-script HI-X



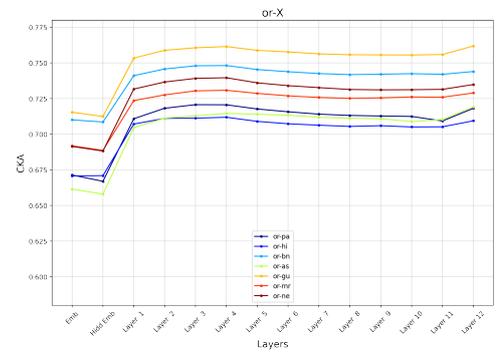
(e) multi-script BN-X



(f) uni-script BN-X

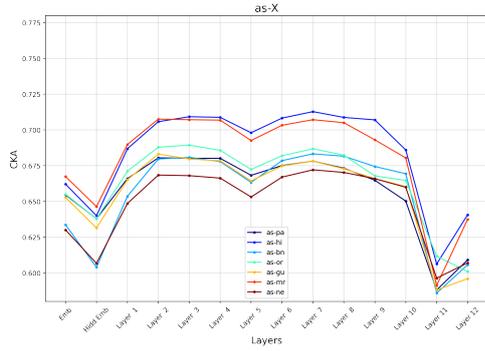


(g) multi-script OR-X

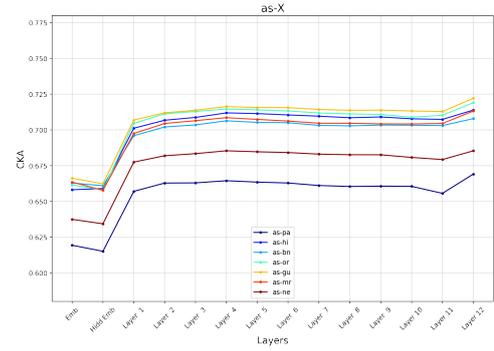


(h) uni-script OR-X

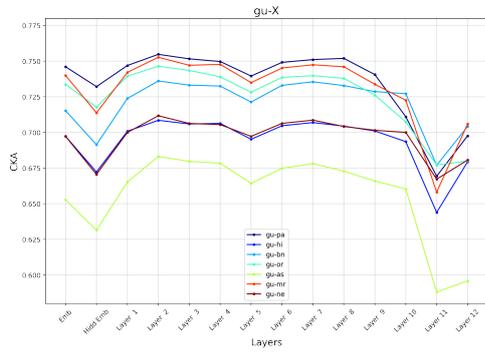
Figure 3: CKA of multi-script and uni-script on all language pairs for pa, hi, bn and or



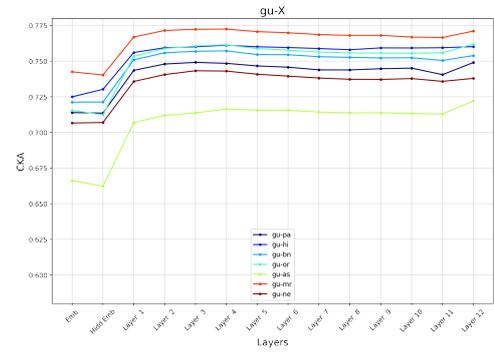
(a) multi-script AS-X



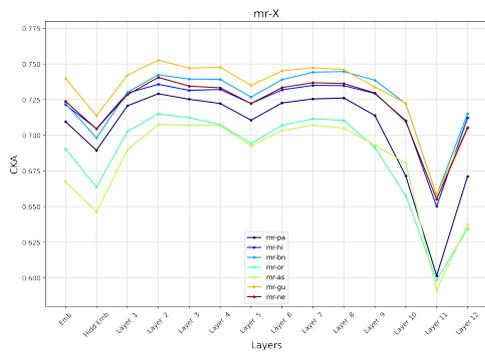
(b) uni-script AS-X



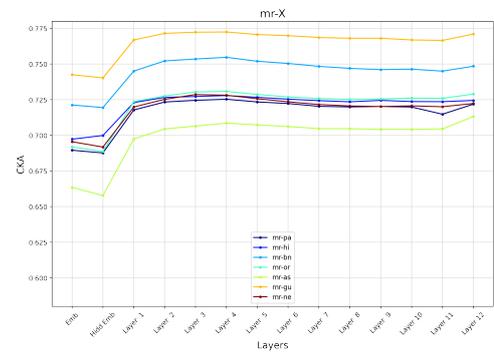
(c) multi-script GU-X



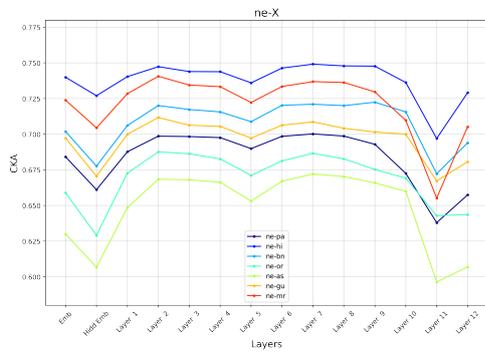
(d) uni-script GU-X



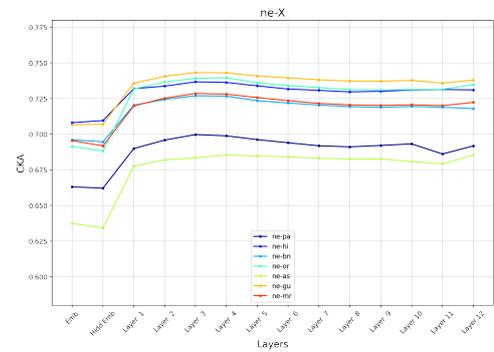
(e) multi-script MR-X



(f) uni-script MR-X



(g) multi-script NE-X



(h) uni-script NE-X

Figure 4: CKA of multi-script and uni-script on all language pairs for AS, GU, MR and NE

A Multilingual Dataset of Racial Stereotypes in Social Media Conversational Threads

Tom Bourgeade^{1,*}, Alessandra Teresa Cignarella^{2,*}, Simona Frenda^{2,*},
Mario Laurent¹, Wolfgang S. Schmeisser-Nieto³, Farah Benamara^{1,4},
Cristina Bosco², Véronique Moriceau¹, Viviana Patti² and Mariona Taulé³

1. IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, France

2. Dipartimento di Informatica, Università degli Studi di Torino, Italy

3. CLiC research group, UBICS, Universitat de Barcelona, Spain

4. IPAL, CNRS-NUS-ASTAR, Singapore

Warning: *This paper contains examples of potentially offensive content.*

Abstract

In this paper, we focus on the topics of misinformation and racial hoaxes from a perspective derived from both social psychology and computational linguistics. In particular, we consider the specific case of anti-immigrant feeling as a first case study for addressing racial stereotypes. We describe the first corpus-based study for multilingual racial stereotype identification in social media conversational threads. Our contributions are: (i) a multilingual corpus of racial hoaxes, (ii) a set of common guidelines for the annotation of racial stereotypes in social media texts, and a multi-layered, fine-grained scheme, psychologically grounded on the work by Fiske et al., including not only stereotype presence, but also contextuality, implicitness, and forms of discredit, (iii) a multilingual dataset in Italian, Spanish, and French annotated following the aforementioned guidelines, and cross-lingual comparative analyses taking into account racial hoaxes and stereotypes in online discussions. The analysis and results show the usefulness of our methodology and resources, shedding light on how racial hoaxes are spread, and enable the identification of negative stereotypes that reinforce them.

1 Introduction

Racial Hoaxes (RHs) are “a communicative act oriented to spread fallacious information against a social group” (Russell, 1998). As social media have become a dominant means of communication, investigating them is crucial for tackling the spread of RHs. We approach this task combining

psychological and computational linguistics methods with a multilingual, cross-cultural perspective (Italian, Spanish, and French).

In particular, RHs can contribute to the diffusion of stereotypes about people belonging to the *outgroup*, i.e., a social group with features that differ from the *ingroup* (Rooduijn et al., 2021) and are, thus, more vulnerable. Even common, naive users are as likely to become spreaders of RHs as malicious users (Papapicco et al., 2022). In this paper, we cover a specific theme: anti-immigrant stereotypes. The discursive construction of immigrants and refugees in user interaction on social media has been studied by Ekman (2019), who has shown how racial expressions and overt racism are becoming increasingly normalized, thus leading to prejudices and racial stereotypes and, eventually, even harmful acts.

Overall, the attention to these topics is relatively new in the NLP community, and thus, there is still a meaningful lack of annotated resources for the development of automatic tools to detect stereotypes and related phenomena. Among the few research contributions in this direction, Sanguinetti et al. (2020) organized the second edition of HaSpeeDe at EVALITA 2020, asking participants to automatically detect hate speech and stereotypes in Italian tweets and headlines. Similarly, in the DETESTS shared task at IberLEF 2022, Ariza-Casabona et al. (2022) proposed a 10-label classification task for the identification of stereotypes in Spanish; and finally, during IROSTEREO at PAN/CLEF 2022, Ortega-Bueno et al. (2022) proposed an author profiling task regarding stereotype spreaders and studied the link with irony in English. Recently, for French, Chiril et al. (2021) investigated how to improve gender hate speech classification by leveraging stereotype detection based on multitask architectures.

* The first three authors contributed equally.

However, such related works only focus on monolingual contents, without considering multilingual settings from which cross-cultural differences and similarities in the expression of stereotypes can emerge. Furthermore, most of the related work limits the scope of investigation to the mere presence/absence of stereotypes in a single text, without diving into the finer-grained features that arise from psychological studies (Allport et al., 1954; Fiske et al., 2007; Cuddy et al., 2008), and without taking into account their propagation in social media conversational threads. Considering the gaps in current related work, we propose a cross-cultural, and multilingual perspective for studying racial hoaxes and stereotypes. In this work, our original contributions are:

- A Multilingual Racial Hoaxes Corpus that was manually created, extracting fake news about migration and racial content from fact-checking web sites. The list of hoaxes has been employed as the core knowledge-base for extracting texts from social media that spread RHs and the *reactions* to them.¹
- A methodology that makes it possible to collect a full conversational thread, with replies and comments that are written under the post spreading the main racial hoax.
- A multi-layered annotation scheme for the annotation of racial stereotypes in social media texts, which allows us to study how the presence of a racial hoax interacts with the surrounding textual context. The scheme, based on psychological work by Fiske (1998), includes four layers: (a) stereotype presence, (b) contextuality, (c) implicitness and (d) forms of discredit.
- A multilingual dataset annotated according to this scheme. For this first study, we chose to retrieve data in languages that are spoken in three countries on the maritime coast of the Mediterranean basin, where migration is widespread and has been made a particular issue in local politics: Italy, Spain and France.²

¹By ‘reactions’ we refer to replies and comments to the main thread that is spreading a racial hoax.

¹To guarantee anonymity and protect the privacy of Twitter users, throughout this paper, instead of using direct quotations from the tweets, we only provide their English translations and/or adaptations.

²The annotated dataset will be available for research pur-

- Qualitative and quantitative analyses from a comparative perspective of the three language subsets, focusing in particular on the interactions between the topics of RHs, stereotypes and discredit in conversations.

2 Related Work

2.1 RH and stereotypes in Psychology

Hoaxes are a form of ‘misinformation’ that aims to disseminate false information with the intention of making it viral in social media (Wardle and Derakhshan, 2018). In particular, ‘Racial Hoaxes’ are fallacious discursive acts that contribute to the spread of information against a social group because of race, religion or origin, such as ‘immigrants’ (Cerase and Santoro, 2018).

From a psychological point of view, RHs have become an important object of study since, firstly, they help to spread misinformation by attacking, discrediting and damaging immigrants’ image; secondly, they can increase the formation of people’s prejudices and stereotypes towards the *outgroup* (Fiske, 1998). In fact, while the stereotype is the cognitive nucleus of prejudice, which contains a set of beliefs and social images; prejudice is a preconceived attitude that is based on common voices and opinions. RHs, therefore, appear to install a stereotype facilitating a categorization in which there is a generalization through a label referring to an entire group, e.g., ‘all immigrants are thieves’ (Allport et al., 1954).

The manifestations of stereotypes can range from a more explicit to a more implicit expression. It is possible, in fact, to distinguish an EXPLICIT stereotype content when identifying a direct association between immigrants and a particular quality, e.g., ‘immigrants bring us diseases’ (Fiske and Taylor, 2013). IMPLICIT stereotypes can be expressed through evaluative utterances and figures of speech such as metaphors, humor, and irony. For instance, Schmeisser-Nieto et al. (2022) present criteria to identify and annotate implicit stereotypes focusing on immigration.

2.2 Stereotypes in Computational Linguistics

The computational linguistics community has only recently focused on modeling stereotypes in order to automatically recognize them, e.g., within political debates (Sánchez-Junquera et al., 2021a) or

poses upon request, together with the complete set of annotation guidelines.

social media (Sanguinetti et al., 2020; Chiril et al., 2021), but without considering the conversational threads in which they occur, nor their reinforcement or confirmation through RHs.

Recently, Sánchez-Junquera et al. (2021a) proposed a taxonomy of stereotypes about immigrants and approached the problem of the automatic classification of stereotypes in Spanish by focusing on the narrative *frames* that spread the stereotypes. Similarly, Fokkens et al. (2018) approached stereotype detection by extracting the *microporraits* and Card et al. (2016) by extracting *stories* about individuals from text. Beukeboom and Burgers (2019) propose a framework which looks at how stereotypes are shared through language: bias in labels and bias in the description of characteristics and behaviors.

Fraser et al. (2022) rely on the Stereotype Content Model (SCM) and present a computational method to mine large datasets and then map sentences to the two-dimensional plane of perceived *warmth* and *competence* (Fiske et al., 2007). Other common computational approaches in NLP mainly focused on measuring and quantifying social bias towards different groups, especially using techniques of word representation, such as word embedding (Bolukbasi et al., 2016), transformers (Card et al., 2016), techniques of natural language inference (Dev et al., 2020) and masking BERT for racial stereotype detection (Sánchez-Junquera et al., 2021b). In this context, this multidisciplinary study on the stereotypes related to RHs from a multilingual, cross-cultural perspective represents an interesting, novel opportunity to understand the expression, perception, and reinforcement of stereotypes, stemming from RHs, against immigrants in conversations on Twitter.

3 From Racial Hoaxes to Reactions

In order to collect reactions to racial hoaxes on social media, we first created the Multilingual Racial Hoaxes Corpus (MRHC), a list of 239 RHs in three languages: Italian, Spanish, and French. Given the difficulty of spotting them automatically, we collected the entries of the MRHC manually.

Depending on the language, different fact-checking websites or newspapers commenting on hoaxes were used as a source for manually extracting the MRHC between 2019 and 2021. For instance, for Italian we used the debunking sites bufale.net and butac.it; for Spanish [\[ita.es\]\(http://ita.es\) and \[newtral.es\]\(http://newtral.es\); and finally for French \[factuel.afp.com\]\(http://factuel.afp.com\) and \[lemonde.fr/les-decodeurs\]\(http://lemonde.fr/les-decodeurs\).](http://mald</p></div><div data-bbox=)

3.1 Topics of the MRHC

Inspired by the taxonomy of stereotypes proposed in Sánchez-Junquera et al. (2021a); Ariza-Casabona et al. (2022), we defined five macro categories of topics, in which immigrants are perceived as threat by the society.

Table 1 contains some examples for each topic: **(a) Security** for events related to citizen safety, such as murder, sexual assault, fights, terrorist attacks, theft, and public disorder; **(b) Public Health** related to health issues that may potentially affect the population, mainly infectious diseases (e.g., COVID-19); **(c) Migration Control** covers migratory flows, arrivals, disembarkation, border control and the regulation of immigration; **(d) Benefits** describe situations in which the outgroup (immigrants) receives more help, social assistance and welfare benefits than the ingroup; **(e) Religion** covers religious and cultural differences of the out group that threaten the traditions of the ingroup (even though terrorism and religion are closely associated in RH, the former category has been considered under the security topic), and finally, **(f) Others** includes RHs about other topics not included in the previous categories.

In terms of a cross-cultural analysis, we observed variations among the different types of RHs. As shown in Table 2, the most common topic of RHs in Italian is related to Security, accounting for 58.76% of the total, while in Spanish and French, RHs are related to Benefits, accounting for 29.16% and 50% respectively. Another relevant result is that the topic Religion has no representation in the Italian subset, which is also the case of Public Health in the French subset.

3.2 Reactions to RHs

We started the collection procedure by retrieving texts from Twitter that contained one of the RHs from the MRHC, or texts that presented a high similarity to one of those. We searched for texts containing the same URL as the RH, or same title of news of the RH on the debunking sites, or even keywords extracted from the textual body of RH by using the Twitter APIs v2 for Academia.³ In

³<https://developer.twitter.com/en/docs/twitter-api/tools-and-libraries/v2>

Example	Topic
Immigrants out of control: they flee and injure an officer	Security
Migrant with Covid repatriated. And now 100 agents are in quarantine	Public Health
The electoral roll increases because the Government nationalizes 200,000 "illegals"	Migration Control
A foreign minor, 4,700€ per month, your grandmother, 426€ pension per month	Benefits
In Aubervilliers, the sheep ready to be slaughtered for #Eid on their way to the butcher. Mind boggling! #Ramadam	Religion

Table 1: Examples of different topics of RHs. All tweets were originally written either in Italian, Spanish or French. They have been translated to English and adapted to ensure anonymity and guarantee privacy to users.

Language	Benefit	Security	Migration Control	Public Health	Religion	Others	Total
Italian	4.12%	58.76%	15.46%	20.62%	0.00%	1.03%	97
Spanish	29.16%	25.00%	16.66%	12.50%	13.88%	2.77%	72
French	50.00%	25.00%	19.44%	0.00%	5.56%	0.00%	70

Table 2: Percentages of Types of RHs in the three language subsets.

Figure 1 we show the full pipeline employed for the collection of “reactions to racial hoaxes”.

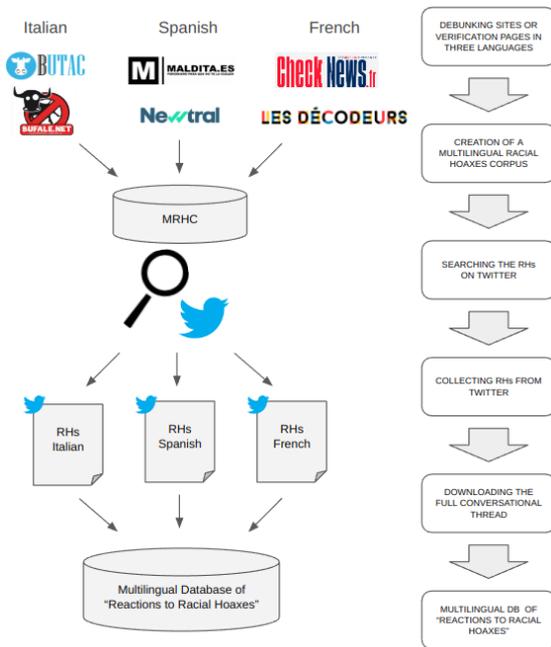


Figure 1: Pipeline for the creation of the Multilingual Racial Hoaxes Corpus (MRHC) and reactions to them.

As can be seen from the picture above, when a racial hoax from the MRHC was found on Twitter, we referred to it as the ‘Conversational Head’, because it was the first text in the conversational thread. Then, for each language, we retrieved all the conversational heads and, in order to study the conversational context, we further collected all the direct replies, and the replies-to-replies.

After the collection and cleaning of data, we

obtained a total of 2,850 unique tweets stemming from Conversational Heads for Italian, 4,751 tweets for Spanish, and 9,305 tweets for French. In Table 3 we display the information on the three subsets of the multilingual dataset. We show the number of the original RHs that we searched for on Twitter and from which we were able to extract the Conversational Heads. In the other columns, we display the number of direct replies, the number of replies-to-replies, and the total of reactions (tweets). In many cases, we had to discard the original RH because it did not originate a conversational thread on Twitter but rather contained just images, videos or recording from other platforms that have not been commented on Twitter with textual content (see the difference between the numbers in the first two columns of Table 3).

4 Annotating Reactions to Racial Hoaxes

4.1 A Multi-layered Annotation Scheme

The annotation scheme designed for the multilingual dataset is inspired by studies regarding stereotypes in the psychological and linguistic literature (Fiske et al., 2007; Cuddy et al., 2008; Sánchez-Junquera et al., 2021a). The outcome of such research is a scheme that consists of four layers, organized in two levels:

1. The first level refers to the presence of a **racial stereotype** as a binary category (*yes/no*).
2. The second level can be annotated only if the

Lang.	Original RHs	RHs found on Twitter	Conversational Heads	Direct Replies	Replies to Replies	Total of Reactions
Italian	97	50	273	597	2,253	2,850
Spanish	72	24	353	85	4,313	4,751
French	70	36	36	3,927	5,378	9,305

Table 3: Number of RHs and details about conversational threads.

precedent level is annotated as *yes*, and it includes three categories:

- (a) **Contextuality.** It encodes whether, in order to understand the meaning of the racial stereotype expressed, you need to look through the context (such as Twitter thread, the RH that triggered the conversation, URLs and images). It is annotated as a binary category (*yes/no*).
- (b) **Implicitness.** It encodes whether the stereotype is expressed explicitly in the message (i.e., a clear span of text where lexical items can be selected) or whether at least one inference needs to be made for the stereotype to be understood). It is annotated as a binary category (*explicit/implicit*).
- (c) **Forms of Discredit.** It encodes the precise form in which the text spreads a racial or anti-migration stereotype, attributing a type of behavior to the discriminated target. The values that can be applied are six: Affective Competence (AC), Attack to Benevolence (B), Competence (C), Dominance Down (DD), Dominance Up (DU) and Physical (P).

These six categories inspired by the Stereotype Content Model proposed by Fiske (1998), can in turn be encompassed in two: COMPETENCE (including C, DD, P) and WARMTH (including AC, B, DU). In the SCM, these macro-categories are respectively referred to as “*agency*” and “*communion*”. For instance, Cuddy et al. (2008) show how, depending on the emotion that is elicited primarily by the form of discredit, different ways of sorting and grouping could be admissible. Furthermore, they underline that the main dimensions of COMPETENCE and WARMTH can be seen as a two-dimensional array for sorting groups. This is an ideal solution that includes at least four clusters which significantly differ regarding warmth and competence.

This motivates our strategy in which Competence (C) is grouped with Physical (P) (both forms of discredit with HIGH COMPETENCE), and Attack to Benevolence (B) with Dominance Up (DU) (both forms of discredit with LOW WARMTH), resulting in the following four clusters for forms of discredit: C+P, DD, B+DU, AC.⁴

4.2 Annotation and Agreement

The data were entirely annotated on locally adapted versions of the LabelStudio⁵ open source platform, in which the questions and labels of the annotation scheme were translated into all the three languages.

The Italian portion of the dataset was annotated by two trained native speakers. Concerning the main dimension of stereotype, they obtained an inter-annotator agreement (IAA) of $\kappa = 0.48$, as calculated by Cohen’s kappa coefficient (moderate). The remaining disagreement was solved by a third expert. The Spanish subset was annotated by three annotators, two of whom are Linguistics students trained for the task, along with a researcher. The IAA was calculated by Fleiss’ Kappa coefficient, resulting in $\kappa = 0.76$. The French subset was annotated by a total of four annotators: an expert and three Linguistics students. Due to the larger quantity of data to annotate, most of the subset was annotated separately by two annotators (two sets of $\sim 4,250$ tweets). The rest was annotated in three sets, each by two annotators, at different stages of the annotation process, to ensure no degradation in IAA was occurring. The Cohen’s Kappa for the French stereotype annotations is $\kappa = 0.73$.

Comparing the scores in the three subsets, it can be noticed that in Italian the IAA is lower with respect to those obtained in French and Spanish. Our hypothesis to explain this is linked to the fact that, in Italian conversational threads, the discussions among users tends to shift quickly to other sub-

⁴Please note that the dataset has been annotated according to the six forms of discredits and that this grouping has been designed with a computational perspective in mind.

⁵<https://labelstud.io/>

jects that are unrelated to RHs. We think that this conversational drift in a large number of tweets created doubt among the annotators and lowered the overall IAA.

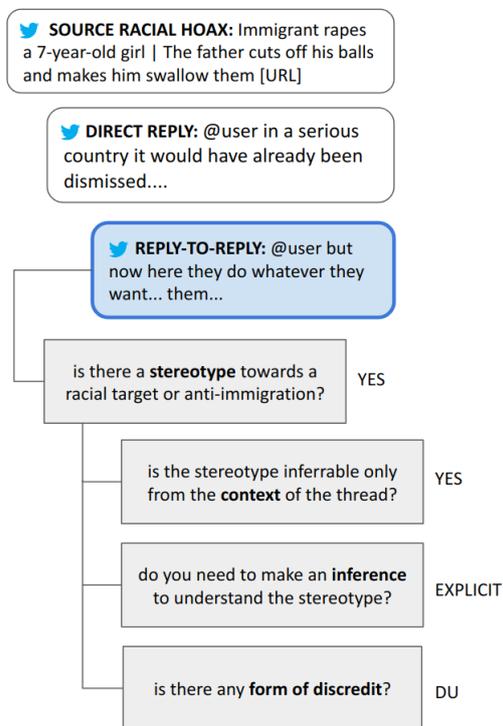


Figure 2: Triplet of tweets from a conversational thread, with the decision tree of the annotation scheme.

We conclude this section with a commented example. Figure 2 shows a Twitter conversational thread and the application of the annotation scheme on it. By looking at the third tweet of the triplet –in the blue box– it can be observed how the user reinforces the stereotypical distinction between “us” and “them”, which highlights the concept of the *ingroup* as different from the *outgroup*. The anaphorically referenced “them” is the group (*outgroup*) to which the immigrant cited in the SOURCE RACIAL HOAX belongs, and for this reason the text has been annotated as containing a **racial stereotype**.

In order to grasp the presence of the stereotype and understand its content, the annotator also had to read the previous textual context (DIRECT REPLY and SOURCE RACIAL HOAX), so the dimension of **contextuality** was annotated as positive. As for the implicitness dimension, the tweet clearly states that “they do whatever they want”, and because this sentence is a clear lexical expression of generalization, the stereotype is annotated as **explicit**. Finally, according to what the user

wrote, the immigrant exercises a sort of forceful dominion and displays aggressive behavior, breaking the law. For this reason, the text was annotated as containing the form of discredit labelled **Dominance Up**.

5 Cultural and Linguistic Analyses

In this section, we describe the comparative analyses we performed to extract analogies and differences in the expression of stereotypes and the forms of discredit in the reactions to RHs among the three subsets.

5.1 Quantitative Results

In Table 4 we report the distribution and percentage of each annotated dimension. As can be seen, in the Italian and French data, stereotypes are found more rarely than in the Spanish subset, which contains about 30% of stereotypes. Another commonality between the Italian and French subsets is the distribution of contextuality and implicitness. In contrast, the Spanish subset contains a higher percentage of explicit stereotypes. Finally, the distribution of forms of discredit is similar in the French and Spanish subsets. In these two subsets, stereotypes are mainly concerned with the provision of social and economic benefits by governments (DD), as well as criminality, illegality and fear of invasion (B+DU). In Italian, this last form of discredit is present with a higher percentage, followed by discredit regarding the competences of immigrants and their physical attributes (C+P).

In our dataset, the number of tweets containing stereotype is lower than in other datasets labelling the presence of this phenomenon (Sanguinetti et al., 2020; Ortega-Bueno et al., 2022). Rather than a purposely balanced dataset created in the context of shared tasks, our multilingual dataset is a reflection of users’ reactions to RHs in social media.

5.2 Stereotype, Discredit, and Types of RHs

In this section, we report some observations regarding the reactions to RHs retrieved from Twitter in the three languages. For Italian, we were able to retrieve a total of 67 RHs on Twitter from the original 97 taken from fact-checking websites (see Table 2). However, after the annotation process and discussion, the gold dataset contains reactions to only 50 RHs. Those RHs that foster the

Language	Tweets	Forms of Discredit									
		Stereotype		Contextual		Implicitness		Agency		Communion	
		yes	no	yes	no	explicit	implicit	C+P	DD	B+DU	AC
Italian	2,850	234 8.21%	2,616 91.79%	177 75.64%	57 25.36%	95 40.60%	138 59.40%	71 23.75%	40 13.38%	176 58.86%	12 4.01%
Spanish	4,751	1,449 30.50%	3,302 69.50%	549 37.89%	900 62.11%	1,344 92.75%	105 7.25%	23 1.74%	761 57.48%	421 31.79%	119 8.99%
French	9,305	1,093 11.75%	8,211 88.25%	818 74.84%	275 25.16%	114 10.43%	979 89.57%	43 3.76%	609 53.23%	395 34.53%	97 8.48%

Table 4: Number of texts and label distribution for the categories annotated in the three language subsets. The numbers in the last four columns do not sum up to the total of the tweets containing stereotype. Indeed, discredit could be annotated with more than one label per tweet, and tweets could therefore be counted more than once.

	Language	Benefit	Security	Migration Control	Public Health	Religion	Others
Stereotype	Italian	0.21%	51.69%	-	48.09%	-	-
	Spanish	38.79%	20.01%	10.97%	0.07%	30.16%	-
	French	70.91%	9.70%	10.43%	-	8.97%	-

Table 5: Percentage co-occurrence of the presence of racial stereotypes and topic of the RH originally spread.

most stereotyped conversations are mainly nine, describing immigrants as threats to public health and security (see Table 5), as shown in the following examples:

- (1) Coronavirus spreads, Government goes to secretly take illegal immigrants in Africa
- (2) He kills an old Jewish woman at the cry of Allah Akbar. Acquitted because he was drugged.

The special attention paid to these two topics is also evident in the analysis of the most used hashtags in the tweets labelled with the presence of negative stereotypes, such as: #Crimes-Immigrants, #SALVINI, #PD, #M5S, #hospitality. By using these hashtags, the users discuss the adopted policies of hospitality and control of immigration of various political parties (#SALVINI, #PD, #M5S), or depict immigrants as criminals (#CrimesImmigrants). The tweets containing these hashtags tend to be labelled with the B+DU form of discredit.

For Spanish, we were able to retrieve 24 RHs on Twitter, out of the 72 RHs originally collected from the fact-checking websites. The most prevalent topic within the Spanish context is related to benefits and the “illegality” of the immigrant. Those topics are associated directly to the forms of discredit DD and B+DU. These topics are also reflected in the use of hashtags such as: #StopIllegalImmigration or #Pensions.

Regarding French, from the 70 RHs identified at the start on the fact-checking website, we extracted 36 instances published on Twitter. As mentioned in Section 3.2, in some cases, we discarded the original RH because it did not originate a conversational Twitter thread or only contained images and videos without textual content. This was common in all the three languages considered.

Overall, French RHs had two common themes: attributing the role of victims to the representatives of Western civilization and the role of perpetrator to immigrants, as in Example (3) below; and pointing the finger at political decisions, real or fantasized, which would favor migrant populations at the expense of the “good French” such as farmers and students, as in Example (4).

- (3) Immigrants burn down a refugee center because there’s not enough Nutella: [URL]

- (4) An immigrant who has never paid taxes in France receives 820 euros per month from the state, in the meantime some farmers get only 360 euros, how do you expect French people not to be angry?

The tweets similar to Example (3) are mainly associated with reactions containing B+DU types of discredit (around 35% of the total) while those similar to Example (4) are linked to DD (~53%).

5.3 Contextuality and Implicitness

Focusing on the textual context, we analyzed how the stereotypes are propagated from the starting

point of the conversations throughout the thread, and if the context is needed to infer implicit forms of stereotypes in the three languages.

In the Italian subset, the majority of the tweets (87%) that are conversational heads (see Table 3) contain negative stereotypes against immigrants. However, even though a conversational head is deemed stereotypical, it is not correct to assume that all the tweets within its thread also contain stereotypes. Indeed, only about 17% of direct replies contain stereotypes and only 6% of the remaining threads are labelled with the presence of stereotypes. This is due mainly to two factors: 1) the tweets spreading fake news or offensiveness tend to be deleted by Twitter; 2) some of the tweets in the conversational threads tend to unveil the inaccuracy of the hoax.

Similarly to what happens in the Italian conversations, in the French subset, all conversational heads contain stereotypes, while 14% of direct replies and 10% of replies-to-replies contain stereotypes. Indeed, the fact that the RHs were debunked by fact-checking websites leads many comments to be criticisms of the conversational head, and this phenomenon is accentuated even more when the RH is shared by accounts with many followers. For the Spanish dataset, only 54% of conversational heads contain stereotypes, with the vast majority of stereotypes contained in replies, accounting for 90% of them.

	Implicitness	
	Language	χ^2
Contextuality	Italian	45.954
	Spanish	41.169
	French	11.419

Table 6: Association between contextuality and implicitness. The χ^2 tests are statically significant at $p < 0.001$ for the three languages.

The results reported in Table 6 show a statically significant association between the dimensions of implicitness and contextuality. As defined in Section 4, annotators labeled the necessity to use the context to understand the message or infer the presence of stereotypes. As expected, in the three datasets, the inference of stereotypes is especially facilitated by access to the textual context.

5.4 Lexical Analysis

To better understand the similarities and differ-

ences at the linguistic and cultural level between the languages, we performed a linguistic analysis, looking at the discriminative lexica used in texts containing stereotypes and labeled with specific forms of discredit. In particular, for all datasets, we listed: the most relevant n-grams⁶ of the data annotated with stereo = *yes* (comparing them with the n-grams of the data annotated with stereo = *no*), and the most relevant n-grams from the data annotated with the four forms of discredit.

By looking at the resulting lists of words, we noticed that, in **Italian**, the words extracted from texts that do not contain stereotypes are related to the emotional sphere (“feeling”, “feel ashamed”, “hope”), in contrast to those extracted from texts containing stereotypes, which are related mainly to the negative actions of immigrants (“immigrant rape”, “kill”, “spit”). Regarding the various forms of discredit, we observed interesting differences. In general, words such as “invasion”, “occupation” and “commanding” or expressions like “walk in underwear” or “laugh in court” are typical in texts annotated with the labels grouped under *communism*. In contrast, words such as “lux”, “gratis”, “withdraw”, “euro”, “gene” or expressions like “psychological disorder”, “return to pre-history” are present in texts annotated with the labels grouped under *agency*.

For the **French** subset, we noticed similar patterns for the terms linked to instances containing stereotypes, with links to violence (“knife”), but also to school (“schooling”, “student”), which are often brought up in instances labeled with discredit under the *agency* group (more particularly, DD), in claims that children of immigrants receive disproportionate financial aid from the state. For instances which do not contain stereotypes, we notably find terms related to misinformation (“fake”, “fake news”, “ridiculous”), which are often levelled against tweets containing stereotypes linked to racial hoaxes.

This underlines the polarization found in the reactions to RHs, by which one section of the users oppose ideas embodied in the RH since they are spread by a proven fake news, thereby avoiding playing the game of attributing certain characteristic to the population designated by the label of

⁶The n-grams are weighted using the TF-IDF measure on normalized texts; the phase of preprocessing involved: the deletion of all user mentions, stop-words, punctuation and URLs, leaving only words that were lexically significant; the tokenization, and the lemmatization with the SpaCy library.

"immigrants"; while another section of the users deliberately ignores the fact that the news has been diverted to focus on the designation of immigrants as the source of a problem. Immigrants are blamed either by their mere presence, which would represent a competition for limited resources, or by their acts, essentializing them as individuals all alike, violent and imposing their foreign culture.

The **Spanish** dataset also present interesting patterns in line with the topics of the grouped data. Firstly, the most prevalent words from texts with no stereotypes are related mainly to politics and economy ("unemployment", "reform", "communist"), whereas texts containing stereotypes show representative words used in RHs ("illegal immigrant", "tradition", "pay health care"). In relation to the categories of discredit, the main characteristic of *communism* is the perception of immigrants as violent, but also as victims, a fact that we can observe in the words like "invasion", "security", "serious" on the negative view, and "poor", "foreigner" and "right" on the patronizing view. On the other hand, *agency* takes a rather derogatory point of view of immigrants, which is displayed in words such as "idiot", "inferior race" and "dumb". The lexica in all languages reflect the stereotypes used against immigrants and the different forms of discredit.

6 Conclusion and Future Work

In this paper, we presented the first outcomes of a study of the stereotypes that are spread through racial hoaxes, with the aim of creating NLP resources and tools to automatically detect them. In order to address this challenging task, we started with an examination of the psychological and computational literature on fake news and stereotypes. This helped us to build the MRHC, the first multilingual corpus of racial hoaxes, which includes RHs in Italian, Spanish, and French, classified according to the topic of the news they spread. We designed a multi-layered annotation scheme for the annotation of racial stereotypes that takes into consideration the conversational thread extracted from social media. We applied it for the first time to a newly created multilingual dataset of Twitter reactions to RHs. Thanks to the outcomes of the annotation procedure, we were able to perform cross-cultural and cross-language analyses of these texts that are shaped in a Twitter conversational structure.

The results show that the presence of stereotype is, in general, lower within the RHs domain, with respect to its percentage in other pre-existing more general-purpose datasets (e.g., the ones developed within shared tasks). Other relevant findings show that, even if the first source RH contains a stereotype, in the following replies in the conversational thread, the presence of stereotypes decreases. Additionally, the dimension of implicitness was shown to be highly dependent on the dimension of contextuality in this domain. Content-wise, from an observation of RHs' topics, crossed with a lexical analysis (counting the most relevant tokens and expressions in each language subset), the outcomes show how the presence of stereotypes is linked to words that are typically grounded within the specificities of a certain language or culture. Finally, it can be observed that people who continue to spread a stereotypical view, originated in the source tweet and throughout the replies-to-replies, typically use polarized expressions that are in line with the original RH that generated the full conversational thread.

Thanks to the resources and framework elaborated in this study, it will be possible to investigate the spread of racial stereotypes on social media in a finer-grained way from a computational perspective and in a multilingual context. Furthermore, these steps are essential for developing computational tools for the automatic detection and classification of racial stereotypes in real-life scenarios.

Limitations

In this work we presented, for the first time, a multi-layered scheme for the annotation of racial stereotypes in social media data in three different languages and in conversational threads. This work can, therefore, be considered pioneering and its multi-layered annotation scheme might require adaptation if applied to datasets with very different characteristics. The Stereotype Content Model inspired the annotation and analysis of stereotypes, by providing a socio-psychological theoretical framework. However, when being as faithful as possible to it during the annotation process, a computational setting can benefit from the integration of a more data-driven perspective.

Furthermore, the three subsets of the multilingual dataset of "reactions to racial hoaxes" now have very different sizes and present many unbalanced dimensions and high data sparsity. If in the

future they will be used for computational tasks, as it is intended, they should be made more balanced and more inclusive in terms of data sources.

Finally, cultural and geographical differences between the three languages of this study need to be taken into account and investigated in a deeper fashion, as it emerged that they are not trivial.

Ethics Statement

The authors have carefully considered the ethics of conducting this kind of study regarding racial stereotypes on social media, and include here their assessment of the ethical issues raised and how to approach them. Throughout the project, they will be critically reflexive about unanticipated ethical issues arising from its sensitive, qualitative and digital nature.

The research presented in this work does not include any studies with human participants carried out by any of the authors. Furthermore, the data that was used is textual content from social media extracted from datasets publicly available to the research community and which also conform to the Twitter Developer Agreement and Policy, which allows for the unlimited distribution of the numeric identification number of each tweet. Each tweet in Italian, Spanish, and French has been translated and adapted into English in order to ensure the anonymity of the author.

Hiring policy: beside the authors of this article, other researchers were involved. We hired two Italian native-speaking annotators (one male and one female: a master's degree student in Linguistics, and a pre-doctoral student). We hired two Spanish native-speaking annotators (one male and one female, both Linguistics undergraduate students in their last year). We hired three French native-speaking annotators (three females, two master's degree students in "Linguistics, Communication and Gender", and one Linguistics undergraduate student). All hired workers have either received a monetary compensation or university credits valid for their career.

Acknowledgements

This work is supported by the International project 'STERHEOTYPES - Studying European Racial Hoaxes and sterEOTYPES' funded by the Compagnia di San Paolo and VolksWagen Stiftung under the 'Challenges for Europe' call for

Project (CUP: B99C20000640007). The research of Farah Benamara is also partially supported by DesCartes: The National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program. Furthermore, the authors would like to acknowledge Francesca D'Errico, Marinella Paciello, Giuseppe Corbelli, Paolo Giovanni Cicirelli and Concetta Papapicco, who contributed to the definition of the theoretical framework for the annotation of racial stereotypes and to the data collection procedure.

References

- Gordon Willard Allport, Kenneth Clark, and Thomas Pettigrew. 1954. *The nature of prejudice*. Addison-wesley Reading, MA.
- Alejandro Ariza-Casabona, Wolfgang S. Schmeisser-Nieto, Montserrat Nofre, Mariona Taulé, Enrique Amigó, Berta Chulvi, and Paolo Rosso. 2022. Overview of DETESTS at IberLEF 2022: DETEc-tion and classification of racial STereotypes in Span-ish. *Procesamiento del Lenguaje Natural*, 69:217–228.
- Camiel J Beukeboom and Christian Burgers. 2019. How stereotypes are shared through language: A review and introduction of the social categories and stereotypes communication (SCSC) framework. *Review of Communication Research*, 7:1–37.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in neural information processing systems*, 29:4349–4357.
- Dallas Card, Justin Gross, Amber Boydston, and Noah A. Smith. 2016. [Analyzing framing through the casts of characters in the news](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1410–1420, Austin, Texas. Association for Computational Linguistics.
- Andrea Cerase and Claudia Santoro. 2018. *From racial hoaxes to media hypes: Fake news' real consequences.*, pages 333–354. Amsterdam University Press.
- Patricia Chiril, Farah Benamara, and Véronique Moriceau. 2021. ["Be nice to your wife! the restaurants are closed": Can gender stereotype detection improve sexism classification?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2833–2844, Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Amy J.C. Cuddy, Susan T. Fiske, and Peter Glick. 2008. Warmth and competence as universal dimensions of social perception: The stereotype content model and the BIAS map. *Advances in experimental social psychology*, 40:61–149.
- Sunipa Dev, Tao Li, Jeff M Phillips, and Vivek Srikumar. 2020. On measuring and mitigating biased inferences of word embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7659–7666.
- Mattias Ekman. 2019. Anti-immigration and racist discourse in social media. *European Journal of Communication*, 34(6):606–618.
- Susan Fiske. 1998. Stereotyping, prejudice, and discrimination. In D. T. Gilbert, S. T. Fiske, & G. Lindzey (Eds.). *The handbook of social psychology*, pages 357–411.
- Susan T. Fiske, Amy J.C. Cuddy, and Peter Glick. 2007. Universal dimensions of social cognition: Warmth and competence. *Trends in cognitive sciences*, 11(2):77–83.
- Susan T. Fiske and Shelley E. Taylor. 2013. *Social cognition: From brains to culture*. Sage.
- Antske Fokkens, Nel Ruigrok, Camiel Beukeboom, Gagstein Sarah, and Wouter van Atteveldt. 2018. [Studying muslim stereotyping through microportrait extraction](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Kathleen C. Fraser, Svetlana Kiritchenko, and Isar Nejadgholi. 2022. Computational modeling of stereotype content in text. *Frontiers in artificial intelligence*, 5.
- Reynier Ortega-Bueno, Berta Chulvi, Francisco Rangel, Paolo Rosso, and Elisabetta Fersini. 2022. Profiling Irony and stereotype spreaders on Twitter (IROSTEREO) at PAN 2022. *CEUR-WS.org*.
- Concetta Papapicco, Isabella Lamanna, and Francesca D’Errico. 2022. Adolescents’ Vulnerability to Fake News and to Racial Hoaxes: A Qualitative Analysis on Italian Sample. *Multimodal Technologies and Interaction*, 6(3):20.
- Matthijs Rooduijn, Bart Bonikowski, and Jante Parlevliet. 2021. [Populist and nativist attitudes: Does ingroup-outgroup thinking spill over across domains?](#) *European Union Politics*, 22(2):248–265.
- Katheryn K. Russell. 1998. *The color of crime: Racial hoaxes, white fear, black protectionism, police harassment, and other macroaggressions*. New York University Press New York.
- Juan Javier Sánchez-Junquera, Berta Chulvi, Paolo Rosso, and Simone Paolo Ponzetto. 2021a. [How Do You Speak about Immigrants?](#) *Taxonomy and StereoImmigrants Dataset for Identifying Stereotypes about Immigrants*. *Applied Sciences*, 11(8).
- Juan Javier Sánchez-Junquera, Paolo Rosso, Manuel Montes, Berta Chulvi, et al. 2021b. Masking and BERT-based models for stereotype identification. *Procesamiento del Lenguaje Natural*, 67:83–94.
- Manuela Sanguinetti, Gloria Comandini, Elisa Di Nuovo, Simona Frenda, Marco Antonio Stranisci, Cristina Bosco, Caselli Tommaso, Viviana Patti, Russo Irene, et al. 2020. HaSpeeDe 2 @ EVALITA2020: Overview of the EVALITA 2020 Hate Speech Detection Task. In *EVALITA 2020 Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, pages 1–9. CEUR.
- Wolfgang Schmeisser-Nieto, Montserrat Nofre, and Mariona Taulé. 2022. [Criteria for the annotation of implicit stereotypes](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 753–762, Marseille, France. European Language Resources Association.
- Claire Wardle and Hossein Derakhshan. 2018. Thinking about ‘information disorder’: formats of misinformation, disinformation, and mal-information. *Ireton, Cheryl; Posetti, Julie. Journalism, ‘fake news’ & disinformation. Paris: Unesco*, pages 43–54.

Detecting Contextomized Quotes in News Headlines by Contrastive Learning

Seonyeong Song¹ Hyeonho Song^{2,3} Kunwoo Park¹ Jiyoung Han² Meeyoung Cha^{3,2}

¹Soongsil University ²KAIST ³Institute of Basic Science
kunwoo.park@ssu.ac.kr, jiyoung.han@kaist.ac.kr

Abstract

Quotes are critical for establishing credibility in news articles. A direct quote enclosed in quotation marks has a strong visual appeal and is a sign of a reliable citation. Unfortunately, this journalistic practice is not strictly followed, and a quote in the headline is often “contextomized.” Such a quote uses words out of context in a way that alters the speaker’s intention so that there is no semantically matching quote in the body text. We present QuoteCSE, a contrastive learning framework that represents the embedding of news quotes based on domain-driven positive and negative samples to identify such an editorial strategy. The dataset and code are available at <https://github.com/ssu-humane/contextomized-quote-contrastive>.

1 Introduction

A direct quotation, a verbatim replication of a speaker’s words as opposed to offering news reporters’ own opinions, manifests news stories’ neutrality, factuality, and objectivity (Zelizer, 1989). Quoting others also adds color to the news with authentic expressions and conveniently establishes authority based on the speakers’ reputation (The Missouri Group, 2013). Therefore, a direct quotation constitutes an integral element of news reporting (Nylund, 2003).

More studies have found a link between the use of direct quotations and fake news. Content analyses of news stories document evidence such that deceptive (*versus* trustworthy) news articles contain more direct quotations (Dalecki et al., 2009; Govaert et al., 2020). An equally problematic but less studied concern involving direct quotations is *contextomy*, quoting words out of context in a way that alters the speaker’s intention. A previous study argued that contextomy is a “common spin tactic” of news reporters promoting their political agenda (McGlone, 2006, p. 332).

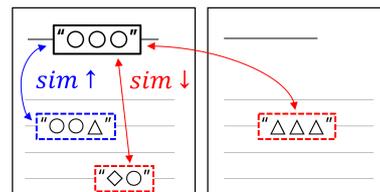


Figure 1: The central idea of QuoteCSE is based on journalism principles, where quotes from news headlines and body text should be matched. The proposed contrastive learning framework maximizes the semantic similarity between the headline quote and the matched quote in the body text while minimizing the similarity for other unmatched quotes in the same or other articles.

Some news outlets have been notorious for editorializing and sensationalizing their stories with contextomized quotes in news headlines (Han and Lee, 2013). The first example in Table 1 illustrates contextomy. This example has a headline, "A government handing out money ... ‘A debt crisis, like Greece, is on the horizon’." The quoted sentence rephrased a financial expert saying in the body text, "If we do not maintain our fiscal health, we may end up like Greece." This is far from word-for-word replication. Instead, the headline reduced the expert’s normative claim about government spending and fiscal distress to a blurb that blasted the national leadership, which was on the opposite side of the political spectrum. As such, a contextomized quote in a news headline can serve as an editorial slogan, misinforming public opinion.

We propose a new problem of identifying contextomized quotes in news headlines. In contrast to a modified quote, which corrects grammar, replaces unheralded pronouns with proper names, removes unnecessary phrases, and substitutes synonyms, a **contextomized quote** refers to the excerpt of words with semantic changes from the original statement (McGlone, 2006). Hence, the task is to classify whether a headline quote is semantically matched by comparing quotes in the

News headline quote	Body-text quotes	Label
"이대론 그리스처럼 파탄" (A debt crisis, like Greece, is on the horizon)	"건강할 때 재정을 지키지 못하면 그리스처럼 될 수도 있다" (If we do not maintain our fiscal health, we may end up like Greece) "강력한 ‘지출 구조조정’을 통해 허투루 쓰이는 예산을 아껴 필요한 곳에 투입해야 한다" (Wasted budgets should be reallocated to areas in need through the reconstruction of public expenditure)	Contextomized
"불필요한 모임 일절 자제" (Avoid unnecessary gatherings altogether)	"저도 백신을 맞고 해서 여름에 어디 여행이라도 한번 갈 계획을 했었는데..." (Since being vaccinated, I had planned to travel somewhere in the summer, but ...) "행사가 일단 다 취소됐고요..." (Events have been canceled...) "어떤 행위는 금지하고 어떤 행위는 허용한다는 개념이 아니라 불필요한 모임과 약속, 외출을 일제 자제 하고..." (It is not a matter of prohibiting or permitting specific activities, but of avoiding unnecessary gatherings, appointments, and going out altogether...)	Modified

Table 1: Dataset examples in Korean and their English translations

news headline and body text.

To tackle the detection task, we propose using contrastive learning for quote representation, which trains a model to maximize the similarity of samples that are expected to be similar (known as *positive* samples). Simultaneously, the model tries to reduce the similarity between samples that should be dissimilar (aka *negative* samples). Following the recent research in contrastive sentence embedding (Gao et al., 2021; Chuang et al., 2022), we introduce a positive and negative sample selection strategy that is suited to the problem.

Our key idea is illustrated in Figure 1. If a direct quotation appears in a news headline, there should be a quote with the same semantics in the body text. Furthermore, the title quote must be distinct from other quotes in the same article or from quotes in other (randomly chosen) news articles. Since quotes from the same article share common topics, it is more challenging to distinguish a headline quote from those in its body text than to understand semantic differences between quotes from distinct articles. Adopting the ‘hard’ negatives in contrastive loss can help a model learn an effective representation, thereby capturing nuanced semantic differences between quotes. Evaluation experiments show its effectiveness at the target problem as well as its high quality in terms of theoretical measures, such as alignment and uniformity.

Our main contributions are three-fold:

1. Based on journalism research and principles, we present a new NLP problem of detecting contextualized quotes in news headlines.
2. We release a dataset for the detection problem based on a guideline constructed by annotators with journalism expertise. The label

annotation by three workers achieved Krippendorff’s alpha of 0.93.

3. We present QuoteCSE, a contrastive quote embedding framework that is designed based on journalism ethics. A QuoteCSE-based detection model outperformed existing methods, including SimCSE and fine-tuned BERT.

2 Related Work

Following the recent success in computer vision (Chen et al., 2020a; He et al., 2020; Grill et al., 2020; Chen and He, 2021), previous studies on contrastive sentence embedding focused on how to construct a positive pair by employing data augmentation methods to an anchor sentence (Fang et al., 2020; Giorgi et al., 2021; Wu et al., 2020; Yan et al., 2021). A recent study showed that a simple dropout augmentation (unlike complex augmentations) with BERT to construct a positive pair could be an effective strategy known as SimCSE (Gao et al., 2021). Another study improved the performance by combining SimCSE with masked token detection (Chuang et al., 2022). This study proposes a strategy for selecting positive and negative samples according to journalistic ethics.

3 Problem and Data

Research Problem Let a given news article be $X : (T, B)$, where T is the news title, and B is the body text. Our task is to predict a binary label Y indicating whether the headline quote in T is either contextomized (1) or modified (0) by referring to the body-text quotes. The detection target is news articles that use at least one direct quotation in the headline and body text.

News Data Collection We gathered a nationwide corpus of Korean news articles published through Naver, a popular news aggregator service. Direct quotes in news articles were identified via regular expression. The dataset contains around 0.4 million news stories published until 2019.

Label Annotation Two journalism-major undergraduates were trained to manually label whether a direct quote in the headline is contextomized or modified. The *contextomized* quote refers to the excerpt of words with semantic changes from the original statement. The *modified* quote in a headline keeps the semantics of the original expression but is a different phrase or sentence. A faculty member in mass communication drafted annotation guidelines that stipulated the definitions of contextomized and modified quotations with multiple examples. The annotators reviewed the guidelines and labeled 70 (up to 200) news articles per training session. Inconsistent cases were discussed to reach a consensus. After the eighth iterative training practice over two weeks, the annotators achieved high inter-coder reliability (Krippendorff’s alpha = 0.93 for 200 articles). Then the annotators split the rest and labeled the news articles separately.

We randomly sampled 2,000 news articles for the manual annotation. We ignored cases where the body text includes an identical quote to the one in the headline because its detection can be achieved by a string-matching method without learning. As a result, the final dataset comprises 814 contextomized and 786 modified samples, leaving a total N of 1,600. Table 1 presents examples. We investigate contrastive embedding approaches to utilize the 381,206 news articles that remained unlabeled.

4 Methods

To predict the label L of $X : (T, B)$, we utilize contrastive embedding and measure the semantic relationship between quotes in the headline and body text. We introduce the main framework.

4.1 Background: SimCSE

SimCSE (Gao et al., 2021) is a contrastive learning method that updates a pretrained bidirectional transformer language model to represent the sentence embedding. Its loss function adapts InfoNCE (van den Oord et al., 2018), which considers identical text with a different dropout mask as a positive sample and the other text within the same batch as negative samples. Formally, the SimCSE loss of

i -th text x_i is

$$-\log \frac{e^{\text{sim}(\mathbf{h}_i, \tilde{\mathbf{h}}_i)/\tau}}{\sum_{j=1}^N e^{\text{sim}(\mathbf{h}_i, \tilde{\mathbf{h}}_j)/\tau}}, \quad (1)$$

where \mathbf{h}_i is x_i ’s embedding¹, $\tilde{\mathbf{h}}_i$ is the embedding of positive sample, τ is temperature hyperparameter, N is the batch size, and $\text{sim}(\cdot, \cdot)$ is the cosine similarity between embedding vectors.

4.2 Proposed Method: QuoteCSE

We propose **QuoteCSE**, a domain-driven contrastive embedding framework on news quotes. Its contribution is in defining positive and hard negatives according to journalism principles. This framework identifies positive and negative samples for a news headline quote according to the golden rules of journalism: When a direct quotation appears in a news headline, its body text should include a quote that is either identical or semantically similar to the headline quote. The latter form can be a good candidate for contrastive learning, where semantically identical yet lexically different quotes serve as ‘positive’ samples. The other quotes in the body text represent different semantics yet cover the same topic, serving as *hard negative* samples.

We define the QuoteCSE loss of i -th sample $X^{(i)} : (T^{(i)}, B^{(i)})$ as

$$-\log \frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^N \{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau} + e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^-)/\tau}\}}, \quad (2)$$

where \mathbf{h}_i is embedding of headline quote for i -th sample. \mathbf{h}_i^+ and \mathbf{h}_i^- are embedding of positive and negative quotes in the same body text $B^{(i)}$. \mathbf{h}_j^+ and \mathbf{h}_j^- are embeddings of $X^{(j)}$, other news articles in the same batch ($i \neq j$), which are negative samples.

We applied SentenceBERT (SBERT) (Reimers and Gurevych, 2019) to make initial assignments on positive (i.e., semantically identical) and negative (i.e., dissimilar) samples among quotes in the body text. A quote is deemed positive if it appears the most similar to the quote in the news headline. After excluding the positive sample, one quote from the body text was chosen randomly as the negative sample. We removed news articles where the cosine similarity between the anchor and the positive sample is below 0.75 because the news headline quote might be contextomized. Additionally, news

¹We applied a 2-layer MLP projection head to the hidden representation corresponding to the [CLS] token in the pretrained BERT.

	F1	AUC
BERT	0.665±0.007	0.662±0.006
SBERT	0.44±0.083	0.591±0.020
SimCSE-Quote	0.69±0.009	0.686±0.009
SimCSE-NLI	0.617±0.008	0.623±0.008
BERT fine-tune	0.754±0.006	0.749±0.006
QuoteCSE	0.77±0.007	0.768±0.008

Table 2: Performance comparison with baselines.

articles that did not contain at least two quotes in the body text were eliminated. The remaining 86,275 articles were divided into 69,020, 8,627, and 8,628 for training, validation, and testing of contrastive learning methods.

We compared QuoteCSE with three baseline embedding methods, (i) BERT (Devlin et al., 2019)², (ii) SBERT³, and (iii) SimCSE. For BERT and SBERT, we used the model checkpoint that was pretrained on a Korean corpus. For SimCSE, we tested two versions. The first version is to train BERT on our news corpus by minimizing Eq. 1 on headline quotes (SimCSE-Quote). The second version is a publicly available SimCSE embedding pretrained on a corpus on natural language inference in Korean (SimCSE-NLI)⁴. For QuoteCSE and SimCSE-Quote, we used SBERT for the initial assignments of positive and negative samples. The assignments iteratively get updated for every training step using the target embedding being trained (e.g., QuoteCSE). QuoteCSE and SimCSE-Quote were trained on the 69,020 sizes of the unlabeled corpus with a batch size of 16, which is the upper limit under the computing environment.

To assess the role of contrastive learning, we implemented a binary MLP classifier with a 64-dimensional hidden layer, following an embedding evaluation framework (Conneau and Kiela, 2018). The model takes \mathbf{u} , \mathbf{v} , $|\mathbf{u} - \mathbf{v}|$, and $\mathbf{u} * \mathbf{v}$ as input, where \mathbf{u} and \mathbf{v} are the embeddings of a news headline quote and the body-text quote most similar to the \mathbf{u} , respectively. In deciding \mathbf{v} , cosine similarity is used along with the target embedding. The classifier predicts whether the headline quote is contextomized based on a vector relationship between \mathbf{u} and \mathbf{v} .

For evaluation, we report the mean F1 and AUC scores by repeating the split process 15 times on

²huggingface.co/monologg/kobert

³huggingface.co/jhgan/ko-sbert-sts

⁴github.com/BM-K/KoSimCSE-SKT

Positive	Hard Negative	F1	AUC
QuoteCSE	QuoteCSE	0.77±0.007	0.768±0.008
SimCSE	QuoteCSE	0.7±0.005	0.69±0.004
QuoteCSE	—	0.674±0.006	0.673±0.006

Table 3: Ablation results confirm the role of both positive and negative samples in the model.

the labeled dataset with a ratio of 8:2. As a strong baseline, we also tested a fine-tuned BERT classifier (BERT fine-tune) that takes '[CLS] q_t [SEP] $q_{b,1}, \dots, q_{b,N_b}$ [SEP]' where q_t is the headline quote, $q_{b,i}$ is the i -th quote in the body text, and N_b is the number of body-text quotes. Details of the model configuration and computing environment are in Section A.1.

5 Evaluation Results

Table 2 presents the evaluation results for the contextomized quote detection. We report the average performance along with standard errors by repeating the experiments using each different random seed. QuoteCSE obtained an F1 of 0.77 and an AUC of 0.76, outperforming the fine-tuned BERT and other contrastive learning methods. Among the baseline models, the fine-tuned BERT model achieved the best F1 of 0.754, which is significantly higher than the performance of the standard contrastive learning methods. The results point to the effectiveness of journalism-driven contrastive quote embedding for the detection problem.

Ablation experiment We examined the importance of positive and negative samples in the QuoteCSE framework by removing each component. The first model is to replace QuoteCSE’s positive sample with that of SimCSE, which is an embedding of the anchor text with a different dropout mask. The second model is to ignore the hard negative sample from QuoteCSE. It only differs from SimCSE in the selection of the positive sample. We trained two contrastive embeddings using the 69,020-size unlabeled corpus. Table 3 presents the results. The detection performance of QuoteCSE was reduced significantly by the ablation of the positive and negative samples. The hard negative sample turned out to be more critical to the detection performance, as F1 of the corresponding model decreased by 0.096. The results confirm the necessity of both positive and negative samples in the QuoteCSE framework.

	Alignment (title-title)	Alignment (title-body)	Uniformity
BERT	0.638	0.738	-0.711
SBERT	0.227	0.329	-1.356
SimCSE-Quote	0.503	0.38	-2.176
SimCSE-NLI	0.319	0.26	-3.257
QuoteCSE	0.15	0.194	-3.562

Table 4: Results of alignment (i.e., closeness of positive samples) and uniformity (i.e., even distribution) scores

Embedding quality We employed two metrics to evaluate the quality of contrastive sentence embeddings (Wang and Isola, 2020). The first is *alignment*, which measures how closely positive pairs are located in the embedding space. The next is *uniformity*, which measures how evenly distributed the target data is. A smaller value denotes a higher embedding quality for both metrics, and their formal definitions are given in Section A.2. We examined two alignments: (i) between two embeddings from the same headline quote with a different dropout mask (title-title) and (ii) between a headline quote and a positive quote in the body text (title-body). We measured the three metrics on the test split of unlabeled data. Table 4 shows that QuoteCSE achieves the best result for all types of theoretical measures, implying a high embedding quality.

Error analysis We identified a common pattern of false positives where a model deems a quote contextomized, which turned out to be modified. They corresponded to instances in which a quote in the headline represents a claim that combines multiple quotes in the body text. For example, in a news article, a headline quote was “감옥 같은 생활... 음식 엉망 (Prison-like conditions... Poor food)” which could be referred to multiple quotes in the body text “삿포로 생활은 감옥처럼 느껴진다 (Living in Sapporo feels like being in prison)” and “음식도 엉망이다 (food is poor).” Since the current detection framework compares a headline quote and another quote in the body text, it could not detect the corner case of a modified quote. Future studies could investigate an approach that considers multiple quotes in the body text.

6 Conclusion

Inspired by the importance of direct quotations in news reporting and their widespread misuse, this study proposed a new NLP problem of detecting contextomized news quotes. While there had been

studies on quote identification (Pavlo et al., 2018) and speaker attribution (Vaucher et al., 2021), this study is the first to discern a specific type of headline news quote that distorts the speaker’s intention and is cut out of context. Not only does it violate journalism ethics (The Missouri Group, 2013; Nylund, 2003), but it can also mislead public opinion (McGlone, 2006). Therefore, tackling the problem of detecting contextomized quotes in news headlines can significantly aid the existing efforts to nurture healthy media environments using NLP techniques (Oshikawa et al., 2020).

Understanding the subtle semantic differences between quotes from news headlines and those from body text is a prerequisite for detecting contextomized news quotes. To assist with this, we introduce QuoteCSE, a contrastive learning framework for quote representation. We specifically tailored SimCSE (Gao et al., 2021) to the detection of the editorial slogan by proposing a positive and negative sample selection strategy consistent with journalism ethics. In the evaluation experiments, we confirmed the effectiveness of both positive and hard negative samples in the journalism-driven contrastive learning framework. Altogether, the findings imply the crucial role of domain knowledge in tackling computational social science problems.

Limitations and Future Directions

First, since this study was done on a monolingual corpus in Korean, the generalizability of the method to other languages is unknown. Future research could replicate this study in other languages to test its broad applicability. Second, the contrastive learning techniques were only tested to a batch size of 16 due to the particular computing environment. To address this limitation, we also tested MoCo-based methods that mitigate the memory limitation (Chen et al., 2020b); however, the results were unsatisfactory (Section A.3.1). The effect of large batch sizes might be examined in future studies. Third, there may be corner cases that the current detection framework is unable to handle. Even if a direct quotation in the headline is schematically consistent with a quote in the body text, this by no means guarantees the authenticity of the quoted remark. It could have been made up by the speaker in the first place. Accordingly, future research warrants considering labels on veracity in conjunction with labels on whether they are contextomized or modified.

Ethics and Impact Statement

Despite the limited headline space, journalism textbooks underscore that direct quotations should meet the strict verbatim criterion (Brooks et al., 2001; The Missouri Group, 2013; Cappon, 1982). This verbatim rule renders news stories with direct quotations more credible and factual. The aforementioned instances of contextomized quotes, however, violate this public trust in journalism. We thus propose a new NLP problem of detecting contextomized quotes and aim to better contribute to the development of responsible media ecosystems. This study is an example of how social science theories can be incorporated with NLP techniques. Thus it will have a broader impact on future studies in NLP and computational social science.

We used public news dataset published through a major web portal in South Korea. Our data is considered clean regarding misinformation because the platform implements a strong standard in deciding which news outlets to admit. However, the considered news data is not free from media bias, and the learned embedding may learn such political bias. Therefore, users should be cautious about applying the embedding to problems in a more general context. We have fewer privacy concerns because our study used openly accessible news data following journalistic standards.

Acknowledgement

K. Park and J. Han are the corresponding authors. This research was supported by the National Research Foundation of Korea (2021R1F1A1062691), the Institute of Information & Communications Technology Planning & Evaluation (IITP-2023-RS-2022-00156360, 2019-0-00075: Artificial Intelligence Graduate School Program (KAIST)), and the Institute for Basic Science (IBS-R029-C2). We are grateful to Seung Eon Lee for putting together the dataset and to the reviewers for their detailed comments that helped improve the paper.

References

Brian S Brooks, Jack Zanville Sissors, and Floyd K Baskette. 2001. *The art of editing*. Allyn & Bacon.

René Jacques Cappon. 1982. *The Associated Press guide to good writing*. Addison-Wesley.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. [A simple framework for](#)

[contrastive learning of visual representations](#). In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 1597–1607.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020b. [Improved baselines with momentum contrastive learning](#). *arXiv e-prints*.

Xinlei Chen and Kaiming He. 2021. [Exploring simple siamese representation learning](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. [DiffCSE: Difference-based contrastive learning for sentence embeddings](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4207–4218.

Alexis Conneau and Douwe Kiela. 2018. [SentEval: An evaluation toolkit for universal sentence representations](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*.

Linden Dalecki, Dominic L Lasorsa, and Seth C Lewis. 2009. [The news readability problem](#). *Journalism Practice*, 3(1):1–12.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. [CERT: Contrastive self-supervised learning for language understanding](#). *arXiv e-prints*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. [DeCLUTR: Deep contrastive learning for unsupervised textual representations](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 879–895.

Charlotte Govaert, Luuk Lagerwerf, and Céline Klemm. 2020. [Deceptive journalism: Characteristics of untrustworthy news items](#). *Journalism Practice*, 14(6):697–713.

- Jean-Bastien Grill, Florian Strub, Florent Alché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. 2020. [Bootstrap your own latent a new approach to self-supervised learning](#). In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 21271–21284.
- Jiyoung Han and Gunho Lee. 2013. [A comparative study of the accuracy of quotation-embedded headlines in chosun ilbo and the new york times from 1989 to 2009](#). *Korea Journal*, 53(1):65–90.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. [Momentum contrast for unsupervised visual representation learning](#). In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 9729–9738.
- Matthew S McGlone. 2006. [Quoted out of context: Contextomy and its consequences](#). *Journal of Communication*, 55(2):330–346.
- Mats Nylund. 2003. [Quoting in front-page journalism: Illustrating, evaluating and confirming the news](#). *Media, Culture & Society*, 25(6):844–851.
- Ray Oshikawa, Jing Qian, and William Yang Wang. 2020. [A survey on natural language processing for fake news detection](#). In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 6086–6093.
- Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Ji Yoon Han, Jangwon Park, Chisung Song, Junseong Kim, Youngsook Song, Taehwan Oh, et al. 2021. [KLUE: Korean language understanding evaluation](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Dario Pavllo, Tiziano Piccardi, and Robert West. 2018. [Quootstrap: Scalable unsupervised extraction of quotation-speaker pairs from large news corpora via bootstrapping](#). *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, 12(1).
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.
- The Missouri Group. 2013. *News Reporting and Writing*. Bedford/St. Martin’s; Eleventh edition.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. [Representation Learning with Contrastive Predictive Coding](#). *arXiv e-prints*.
- Timoté Vaucher, Andreas Spitz, Michele Catasta, and Robert West. 2021. [Quotebank: A corpus of quotations from a decade of news](#). In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM)*, page 328–336.
- Tongzhou Wang and Phillip Isola. 2020. [Understanding contrastive representation learning through alignment and uniformity on the hypersphere](#). In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 9929–9939.
- Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022. [ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding](#). In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pages 3898–3907.
- Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. [CLEAR: Contrastive learning for sentence representation](#). *arXiv e-prints*.
- Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. [ConSERT: A contrastive framework for self-supervised sentence representation transfer](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 5065–5075.
- Barbie Zelizer. 1989. ‘Saying’ as collective practice: [Quoting and differential address in the news](#). *Text - Interdisciplinary Journal for the Study of Discourse*, 9(4):369–388.

A Appendix

A.1 Details of model configuration and computing environment

We ran experiments on a machine with an Intel(R) Xeon(R) CPU E5-2620 v4 running at 2.10GHz, four TitanXP 12GB GPUs, and 130GB RAM. All models were evaluated on Python 3.9 with the Transformers library (ver. 4.19.4). We ran contrastive learning experiments with the batch size of 16 using Adam with a learning rate of 0.01, and the maximum number of epochs was 10. The parameter size of KoBERT is 92m, and that of the MLP projection head is 87k with a hidden dimension of 100. The temperature of the softmax is 0.05, which is the same as [Gao et al. \(2021\)](#). It took 10 and 13 hours to finish SimCSE and QuoteCSE contrastive training, respectively. For the detection task, we trained models with the same configuration. We did not conduct hyperparameter optimization since the dataset is small. Instead, we reported summary statistics of performance by repeating the data split,

model training, and evaluation process while varying random seeds (0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140).

A.2 Formal definition of alignment and uniformity

Alignment is

$$\mathbb{E}_{(x,x^+)\sim P_{pos}} \|f(x) - f(x^+)\|^2 \quad (3)$$

, where x is an anchor text, x^+ is positive sample, and $f(\cdot)$ is an embedding function. P_{pos} is the distribution of positive pairs.

Uniformity is

$$\log \mathbb{E}_{(x,y)\sim P_{data}} e^{-2\|f(x) - f(y)\|^2} \quad (4)$$

, where P_{data} is the distribution of the anchor text.

A.3 Additional evaluations

A.3.1 Momentum-based methods

	F1	AUC
MoCo: SimCSE	0.658±0.011	0.667±0.008
MoCo: QuoteCSE	0.756±0.005	0.753±0.006

Table A1: Momentum-based methods underperform their corresponding general methods.

Our computing environment is limited, such that all models were trained with a batch size of 16. Since the batch size decides the number of negatives for InfoNCE-based contrastive learning frameworks, it was reported that a larger batch size can result in better performance (Chen et al., 2020a). To approximate the effects of a larger number of negatives in a batch, we evaluated MoCo-based approaches that keep samples in previous batches as additional negatives with momentum updates (He et al., 2020). We set the queue size to be 40 according to the observation on the effect of queue size in a previous study (Wu et al., 2022). We make two observations from Table A1 on the evaluation results of contextomized quote detection. QuoteCSE still outperformed SimCSE, but the MoCo versions performed worse than the general version.

A.3.2 STS benchmark

To see if the learned embeddings are generalizable, we tested the baseline and proposed models on the KLUE benchmark on sentence similarity (Park et al., 2021). Using the same model architecture for the contextomized quote detection, we trained

	F1	AUC
KoBERT	0.636	0.659
SimCSE-Quote	0.633	0.662
QuoteCSE	0.775	0.796

Table A2: Evaluation based on the KLUE-STS benchmark indicates the generality of the proposed method.

a model to predict a binary label on whether two given sentences are similar.

The evaluation results based on the valid dataset are shown in Table A2. QuoteCSE outperforms KoBERT and SimCSE-Quote, suggesting that our model can produce better semantic embedding.

A.3.3 Filtering scenarios in the wild

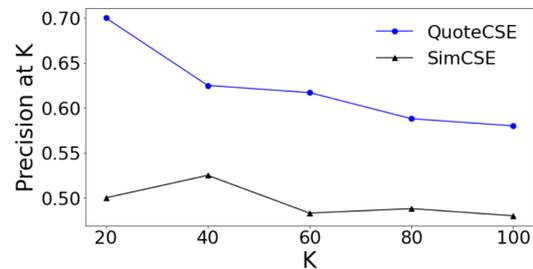


Figure A1: The precision at k results of QuoteCSE and SimCSE suggest QuoteCSE’s effectiveness in filtering contextomized quotes in the wild.

We collected 10,055 news articles published in July and August 2021. To test the proposed model’s effectiveness in the wild, we manually evaluated the top-100 news articles regarding the prediction scores of SimCSE-Quote and QuoteCSE, respectively. A high prediction score indicates that a model consider the given news article containing a contextomized quote in headline with a high confidence, therefore this evaluation assumes a scenario of filtering news articles with contextomized quotes.

Figure A1 presents the precision at k of the two models, indicating how many instances turned out to be correct among the top- k examples, which are predicted to be contextomized by a model with a high confidence. Results indicate that QuoteCSE can achieve a high precision value of 0.7 for the top-20 examples. The precision decreases as its confidence gets lowered, reaching a plateau around 0.6. On the contrary, SimCSE achieved a precision lower than 0.55 even when its confidence is high. The results suggest the potential of QuoteCSE-based detection model for filtering contextomized quotes in the real-world scenario.

Zero-Shot On-the-Fly Event Schema Induction

Rotem Dror*, Haoyu Wang*, and Dan Roth

Department of Computer and Information Science

University of Pennsylvania

{rtmdrr, why16gzl, danroth}@seas.upenn.edu

Abstract

What are the events involved in a pandemic outbreak? What steps should be taken when planning a wedding? The answers to these questions can be found by collecting many documents on the complex event of interest, extracting relevant information, and analyzing it. We present a new approach¹ in which large language models are utilized to generate source documents that allow predicting, given a high-level event definition, the specific events, arguments, and relations between them to construct a schema that describes the complex event in its entirety. Using our model, complete schemas on any topic can be generated on-the-fly without any manual data collection, i.e., in a zero-shot manner. Moreover, we develop efficient methods to extract pertinent information from texts and demonstrate in a series of experiments that these schemas are considered to be more complete than human-curated ones in the majority of examined scenarios. Finally, we show that this framework is comparable in performance with previous supervised schema induction methods that rely on collecting real texts and even reaching the best score in the prediction task.

1 Introduction

Event processing refers to tracking, analyzing, and drawing conclusions from streams of information about events. This event analysis aims at identifying meaningful events (such as opportunities or threats) in real-time situations and responding appropriately. Event processing can also be utilized to gain a deep understanding of the specific steps, arguments, and relations between them that are involved in a complex event. The information above can be consolidated into a graphical representation called an *event schema* (Li et al., 2021). For instance in Fig. 1, the graph representation of events

and participants assists in gaining an understanding of the complex event of kidnapping and could help composing a reaction plan if needed.

The NLP community has devoted much effort to understanding events that are described in a document or in a collection of documents for this purpose. These efforts include identifying event triggers (Lu and Roth, 2012; Huang et al., 2018; Wadden et al., 2019; Han et al., 2019), extracting event arguments (Punyakanok et al., 2008; Peng et al., 2016; Lin et al., 2020; Zhang et al., 2021a), and predicting the relations between events, e.g., temporal, coreferential, causal or hierarchical relations (Do et al., 2012; Lee et al., 2012; Glavaš et al., 2014; Ning et al., 2018; Wang et al., 2020; Zhang et al., 2020a; Trong et al., 2022).

Previous works on event schema induction relied on the information extracted from manually collected documents to build the schema graph. For instance, Li et al. (2020) learn an auto-regressive language model (LM) over paths in the instance graphs depicting events, arguments and relations of instances of the complex events, and then construct a schema graph by merging the top k ranked paths. Their approach, however, requires access to many documents on each topic of interest, which can be extremely laborious and time consuming to obtain.

In this paper, our goal is to allow creating schemas on-the-fly by taking as input only the name of the complex event of interest (like a “pandemic outbreak” or an “armed robbery”). To avoid manually collecting many documents on the topic of the schema, we utilize pre-trained text generators, e.g., GPT-3 (Brown et al., 2020), to obtain documents of diverse genres on the desired topic (examples presented in Fig. 2). These documents are then processed to extract pertinent information from which a schema is constructed. The fact that we do not collect any data makes our learning framework zero-shot since we do not rely on any human-collected articles or example schemas.

* Indicating equal contribution.

¹https://cogcomp.seas.upenn.edu/page/publication_view/995

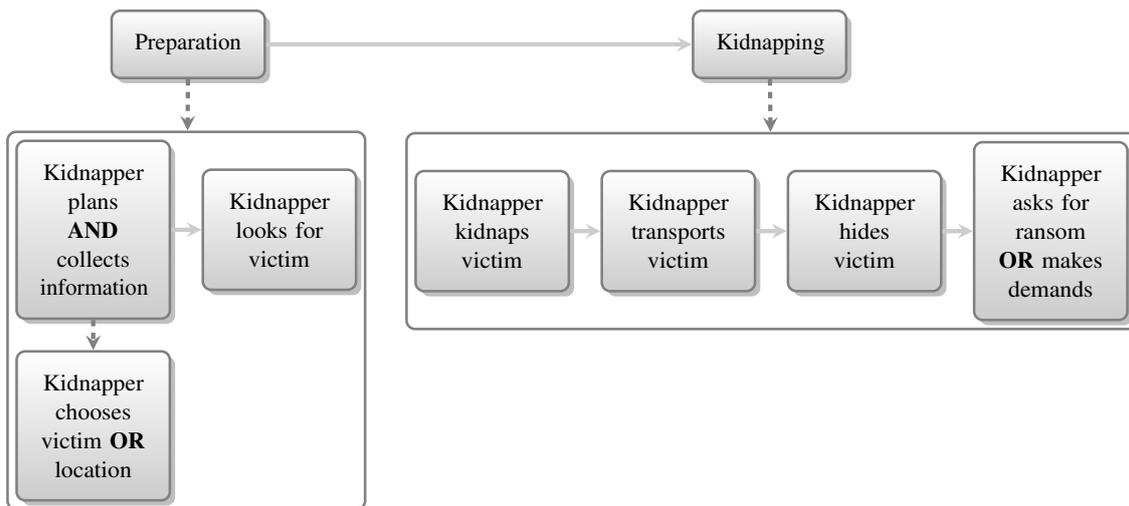


Figure 1: An example schema for the event of Kidnapping. The regular arrows represent temporal relations and the dashed arrows represent hierarchical relations (PARENT-CHILD).

In addition to eliminating the need to collect data, we also made the information extraction process faster by implementing new and efficient methods for identifying temporal and hierarchical relations between events mentioned in the text. These two steps are the most time consuming in the process of schema induction and could take up to 2 hours each using state-of-the-art models proposed by Zhou et al. (2021); Wang et al. (2021). Sending the whole text as input instead of two sentences at each time, our proposed model shortens the inference time significantly to several minutes without enduring a major loss in performance.

The process of generating texts is explained in Section §3, and the process of extracting relevant and salient information is described in Section §4, then we introduce the construction of schema graphs in Section §5. To evaluate our zero-shot schema generator we conduct experiments on a benchmark dataset for schema induction, LDC2020E25, and provide a new dataset for further evaluation called Schema-11. Additionally, we design a subject-matter expert Turing test, a.k.a. Feigenbaum test (Feigenbaum, 2003), to determine whether our algorithm could mimic experts’ response. We also demonstrate that documents generated by GPT-3 are informative and useful for the task of schema induction. The experiments and results are presented in Section §6. The contributions of our work include:

1. Predicting an entire schema given the name of a complex event without collecting data.
2. Implementing a novel and efficient One-Pass

approach for identifying temporal and hierarchical relations between events.

3. Presenting a method for automatically inducing logical relations between events based on temporal relations.
4. Offering a Feigenbaum test for evaluation on a new schema dataset, Schema-11.

2 Related Work

Schema Induction: Early schema induction efforts focused on identifying the triggers and participants of atomic events without considering relations between atomic events that comprise complex schemas (Chambers, 2013; Cheung et al., 2013; Nguyen et al., 2015; Sha et al., 2016; Yuan et al., 2018). More recent work focuses on inducing schemas for pairs of events (Li et al., 2020) and multiple events (Zhang et al., 2021b; Li et al., 2021), but they require access to large corpora for the induction process. In this work, we induce schemas on-the-fly in a zero-shot manner. As is standard in state-of-the-art (SOTA) works (Li et al., 2020, 2021; Wen et al., 2021), we output all the essential information about relations between events and arguments extracted from the text, in addition to logical and hierarchical relations not studied previously in schema induction.

Script Learning: Early script learning work concentrated on chains of events with a single protagonist (Chambers and Jurafsky, 2008, 2009; Jans et al., 2012; Rudinger et al., 2015; Granroth-Wilding and Clark, 2016) and later extended to

multiple protagonists (Pichotta and Mooney, 2014; Peng and Roth, 2016; Pichotta and Mooney, 2016; Modi, 2016; Weber et al., 2018, 2020; Zhang et al., 2020b). All of these works assume there exists a single line of events that describes all occurrences within a complex event. This work does not limit itself to generating single-chained schemas. We also consider more complex graphs as schema outputs. In addition, none of these works deal with zero-shot scenarios that do not require training data.

Pre-Trained Generation Models: Large-scale pre-trained text generation models such as GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), BART (Lewis et al., 2020), T5 (Raffel et al., 2020), i.a. have been used in many NLP tasks. These models are often seen as few-shot learners (Brown et al., 2020) and therefore used as inference methods. However, these text generation models are not explicitly trained to perform inference, but to produce the most likely sequence of words to proceed a certain prompt, similar to language models. In our work, we use these large pre-trained LMs as text generators. The generated documents on a particular topic are leveraged as a corpus for extracting the schema of the given topic. We rely on the intuition that the generated text will include salient and stereotypical information that is expected to be mentioned in the context of the topic (e.g., for the topic of “planning a wedding,” we assume most documents will include “order catering”).

3 Data Generation

The schema induction process begins with generating texts using large LMs as text generation models. These texts are joined to form a knowledge base for the schema, including all of the potential information that the schema may present. One could, of course, create this knowledge base by crawling the web for real news articles or Wikipedia entries related to a certain topic.

We argue, however, that in addition to the obvious advantages of not having to rely on the availability of data online and not having to crawl the entire web for relevant documents on each topic, the generated data from these large generative models is more efficient in reporting salient events than random events described in the news, i.e., generated texts are more likely to mention important information than real documents do.

Our analysis shows that the generated stories contain a higher percentage of relevant tokens than

	Generated Text	Real Text
# events / # tokens	12.52%	6.31%
# arguments / # tokens	5.45%	3.01%

Table 1: The ratio of relevant events and relevant argument roles identified in generated texts and real texts for the scenario of IED attack.

real news articles that are used for schema induction. To demonstrate this phenomenon, we compare manually collected documents with those that are automatically generated using GPT-3 for the event of Improvised Explosive Device (IED) Attack (Li et al., 2021). To identify salient events and arguments concerning IED attacks, we adopt the DARPA KAIROS Phase 1 (v3.0) ontology² — a fine-grained ontology for schema learning, with 24 entity types, 67 event types, and 85 argument roles.

We calculate the number of relevant event triggers and arguments identified in the text, where a relevant mention is one whose type appears in the ontology. The results shown in Table 1 demonstrate that the quality of the generated texts in terms of conciseness and appearance of important details is higher than that of real texts. For example, the ratio of relevant events per token is more than twice as high in generated texts as it is in real texts. Hence we are able to not only generate a schema for every given topic without putting any effort in searching the web, but the information we generate is also better suited for our end task of depicting all of the important aspects of a complex event.

Given a topic for which we want to create a schema, we generate multiple texts that discuss the topic event using the OpenAI GPT-3 API³ with the Davinci-instruct-beta-v3 model and we also experiment with the Hugging Face GPT-2 API⁴. We use three prompting methods to generate documents of diverse genres as follows:

News Articles: We begin by generating a headline using the prompt: “Write a news headline about *topic*.” The output from this prompt is then used in the following prompt: “Write a news story titled *headline*.” The output from the second prompt is added to the pool of generated texts. The process is repeated 30 times. See example in Fig. 2b.

How-To Articles: We use the prompt: “Describe how to *topic*.” to generate wikiHow-like instruction

²The full ontology definition can be accessed at this link: <https://bit.ly/3mIWJoN>.

³<https://openai.com/blog/openai-api/>.

⁴<https://huggingface.co/gpt2>

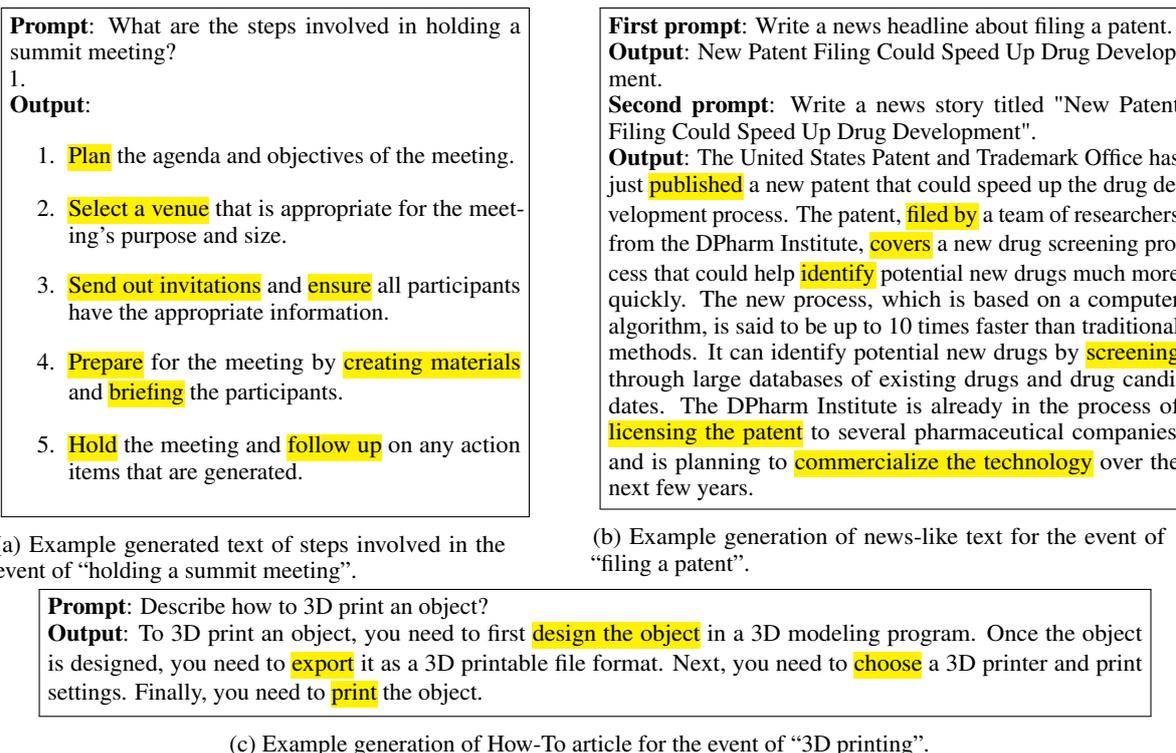


Figure 2: Examples of generated texts using different prompting methods. The highlighted tokens display relevant events that will be extracted in the information extraction step.

articles. The process is repeated 30 times. See example in Fig. 2c.

Direct Step-by-Step Schema: We use the prompt: "What are the steps involved in *topic*? 1."⁵ to directly generate a schema. We run this process once. See example in Fig. 2a.

Generating documents of various genres enables our model to induce comprehensive schemas on any given topics. Considering that some events are more likely to be in the news (e.g., elections, pandemic outbreaks) while others are more technical in nature and are hence less newsworthy (such as earning a Ph.D. degree or planning a wedding), we generate diverse texts and then use a ranking model to choose the most relevant documents.

The ranking process includes embedding the texts and the topic with the model proposed in Reimers and Gurevych (2019), and then calculating the cosine similarity between each text and the topic embeddings. Only the 30 texts closest to the topic are selected, together with the output from the direct step-by-step schema. The following section describes the next step in generating a schema of extracting relevant information from the texts.

⁵The "1." in the prompt is for the LM to automatically complete the steps.

4 Information Extraction

For each document, we extract event triggers, arguments and relations between the events that are important and relevant to the schema topic. We do not work with a predefined ontology that defines what events and arguments are salient in advance because we allow generating a schema on any topic. Instead, we employ a statistical approach by extracting all the information and later filter it down to include just frequent items. Here are the steps involved in our information extraction pipeline:

Semantic Role Labeling (SRL): We use the SOTA SRL system⁶ trained on CoNLL12 (Pradhan et al., 2012) and Nombank dataset (Meyers et al., 2004) to extract both verb and nominal event triggers and arguments.

Named Entity Recognition (NER): We employ the SOTA NER model (Guo and Roth, 2021) to extract and map entities (potential arguments of events) into entity types defined in the CoNLL 2002 dataset (Tjong Kim Sang, 2002) and the LORELEI project (Strassel and Tracey, 2016).

⁶ https://cogcomp.seas.upenn.edu/page/demo_view/SRLEnglish

Constituency Parsing: The arguments extracted by SRL can be clauses and long phrasal nouns, hence we employ the AllenNLP⁷ constituency parsing model for argument head word extraction.

Coreference Resolution: We use the SOTA model (Yu et al., 2022) for event and entity coreference resolution to identify within-document coreferential relations.

Temporal Relation Extraction: We first try to use SOTA models (Ning et al., 2019; Zhou et al., 2021) to predict the temporal relations⁸ between all possible pairs of extracted events but since the SOTA models accept two sentences containing events as input, the inference time⁹ for an n -event document is $\mathcal{O}(n^2)$, making the schema induction process several hours long.

One-Pass Model: We develop a One-Pass model that takes the document as input and uses the contextual representation of events to predict relations between them. A document D is represented as a sequence of tokens $D = [t_1, \dots, e_1, \dots, e_2, \dots, t_n]$ where some of the tokens belong to the set of annotated event triggers, i.e., $\mathcal{E}_D = \{e_1, e_2, \dots, e_k\}$, whereas the rest are other lexemes. We employ the transformer-based language model Big Bird (Zaheer et al., 2020) to encode a whole document and obtain the contextualized representations for all the event mentions. These representations are fed into a multi-layer perceptron in a pairwise fashion and the cross-entropy loss for each pair is calculated and accumulated for a batch of documents. As shown in Tab. 2, the inference time is shortened 63-186 times on average, while the performance of the One-Pass model is comparable to SOTA models.

Hierarchical Relation Extraction: The extremely long inference time of SOTA models for predicting hierarchical relations (PARENT-CHILD, CHILD-PARENT, COREF, NOREL) (Zhou et al., 2020; Wang et al., 2021) also impairs the efficiency of our schema induction system. Thus we use the same One-Pass methodology to extract hierarchical relations. We observe that the inference time is greatly shortened, and the One-Pass model

⁷<https://demo.allennlp.org/constituency-parsing>.

⁸The possible temporal relations (start-time comparison) are: BEFORE, AFTER, EQUAL and VAGUE.

⁹The inference time is mostly spent on obtaining the contextual representation of events using large fine-tuned LMs.

Corpus	Model	Metrics		
		F_1 score	Speed	GPU Memory
HiEve	Zhou et al. (2020)	0.489	-	-
	Wang et al. (2021)	0.522	41.68s	4515MiB
	One-Pass model	0.472	0.65s	2941MiB
MATRES	Ning et al. (2019)	0.767	30.12s	4187MiB
	Zhou et al. (2021)	0.821	89.36s	9311MiB
	One-Pass model	0.768	0.48s	2419MiB

Table 2: Performance comparison between the One-Pass model and SOTA models for event temporal and hierarchical relation extraction. We report F_1 scores on benchmark datasets (HiEve for hierarchical relations, MATRES for temporal relations), speed (average inference time for 100 event pairs), and required GPU memory during inference. The One-Pass models are 63-186 times faster than SOTA models and take up only 26%-65% of the GPU memory required by SOTA models.

achieves comparable results to previous models while taking up less GPU memory (see Tab. 2).

After processing the data using the procedure described above, we get a list of events, their arguments, and relations between the events. We concentrate on events and relations that frequently appear in the generated texts since we assume those are the most important to add to the schema (without any other source of information that could identify what is salient). We describe the process of building a schema in the following section.

5 Schema Induction

To consolidate the information extracted from the previous step, we build a schema as follows:

Make a list of events and relations: To compare similar event mentions in different texts, we compare the event trigger itself (whether they are the same verb or coreferential verbs¹⁰) and the NER types of its arguments. For example, the trigger “(take) precautions” appeared in 5 documents generated for the topic of Pandemic Outbreak. In two documents the subject of the verb phrase “take precautions” was “residents”, in another two it was “people” and in the last one, it was “public”. Nevertheless, the NER type is identical in all cases (PER), and thus we set the frequency of “(take) precautions” to 5. Similarly, we calculate the frequency of the temporal and hierarchical relations. We only consider relations and events that appeared in more than one document.

Construct timelines: We construct the longest timelines from the list of temporal relations. This

¹⁰We only consider coreferential and hierarchical relations if they appear in more than 2 documents.

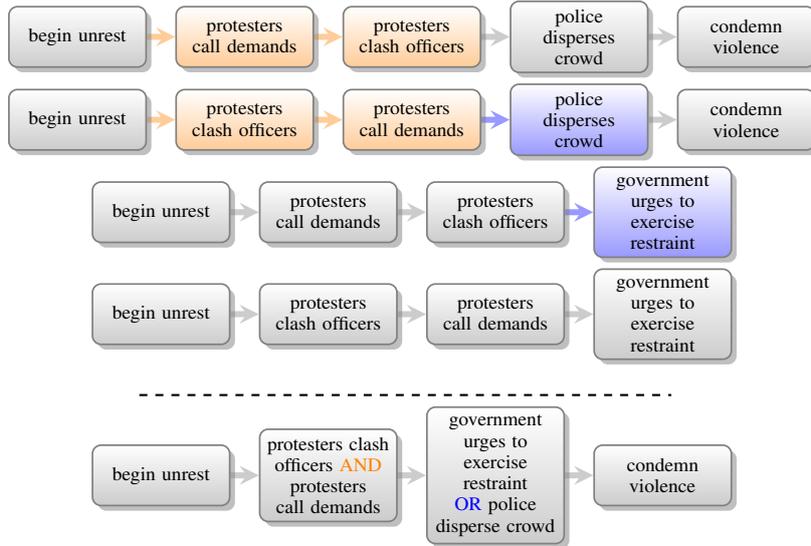


Figure 3: An example of integrating timelines and logical relations in the schema of Civil Unrest. The four upper timelines are the ones extracted from the generated texts and the lower one is their merger into a single timeline with logical relations.

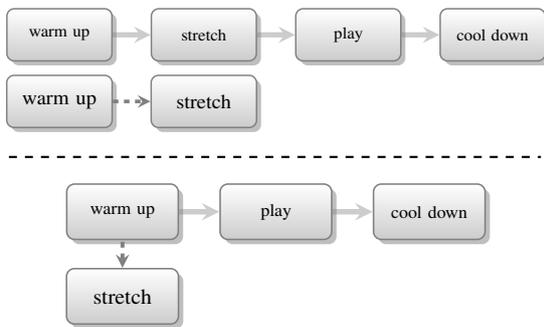


Figure 4: Example of the procedure to amend a timeline in the schema of “Sports Games”. The timeline at the top that includes events from different levels (“warm up” is the parent of “stretch”) is fixed below. Gray arrows mark temporal relations, and dashed arrows mark PARENT-CHILD.

list is a list of tuples (A, B) , indicating that event A happened before event B . To construct a timeline, we search recursively for the longest chains of the following form $(A, B), (B, C), (A, C)$ and so on.

Fix timelines according to hierarchical relations: We build a hierarchy of the events using the hierarchical relation list¹⁰ and change the timelines so that they will only include events that appear in the same level of hierarchy (see example in Fig. 4).

Add logical relations: The final step is to combine the timelines and hierarchies into a single schema graph using logical relations (AND/OR). When observing two timelines with discrepancies between the order of events, we place a logical AND between them, since we interpret this discrepancy as

both events occurring at the same time or there is no significance to the order between them. We use a logical OR to mark events that can occur simultaneously but not necessarily. See Fig. 3 for example of both logical relations.

The final output is a schema graph that contains all the events, arguments, and temporal, hierarchical and logical relations between the events. It is noteworthy that our proposed schema generation model can be easily used to *extend the scope of existing schemas* by further querying the model on more specific topics. For example, the schema in Fig. 1 does not cover the consequences of kidnapping, probably because the LM did not attend to this aspect. Hence an analyst can input another topic (e.g., consequences of kidnapping) to further develop the schema. Similarly, analysts can generate schemas for very specific events (e.g., kidnapping in a political setting). Next, we provide an in-depth experimentation for the proposed schema induction framework.

6 Experiments

6.1 Data

We conduct experiments on a dataset for general schema learning released by LDC (LDC2020E25). The corpus includes 84 types of complex events, such as Cyber Attack, Farming and Recycling. This dataset includes ground-truth schemas created by LDC annotators. In addition, we also collected human generated schemas for 11 newsworthy sce-

narios¹¹. The schemas were generated by four human experts who were instructed to write a schema on each topic based on their commonsense knowledge that includes a list of event triggers, event arguments and their NER types¹², and relations¹³.

6.2 Evaluation Protocols

We follow Li et al. (2021) to use instance coverage and last event prediction to evaluate our method on the LDC dataset. For the Schema-11 dataset, we ask human testers to assess the completeness and soundness of both human- and automatically-generated schemas.

Coverage and Prediction A common evaluation method in schema induction and script prediction is to calculate the recall of events and relations predicted by the model, assuming the human annotations are gold labels (coverage), and to calculate the accuracy in predicting the final outcome of a scenario (prediction). For instance, the accuracy of predicting the last event type of the LDC schemas is reported in Li et al. (2021). Here we present the results of predicting the last events using event triggers instead of event types since our schemas do not use an ontology of event types.

Feigenbaum Test We show human testers two schemas on each topic in the Schema-11 dataset (see example in Appx. §A). One schema is automatically generated by our model, and the other is randomly sampled from the Schema-11 corpus¹⁴. Then, we ask the testers to determine which events and relations are valid to appear in the schema (soundness), and answer the following questions: which schema is more complete in the sense of including all the events needed to describe the topic, and which schema, in their opinion, was generated by a human expert (as opposed to a machine).

6.3 Results

Coverage We calculate the intersection between events in the generated schemas and the gold

¹¹The topics are: Bombing Attack, Business Change, Civil Unrest, Disaster and Rescue, Elections, International Conflict, Kidnapping, Mass Shooting, Pandemic Outbreak, Sports Games, and Terrorism Attack.

¹²The annotators are familiar with SRL annotations (e.g., ARG0, ARG1, etc.) and NER types (e.g., PER, ORG, etc.). See additional details in App. C

¹³No restrictions were placed for the annotators. For example, in one case, an annotator mentioned causal relations that are not covered in our framework.

¹⁴In some cases we combine two randomly sampled schemas because the length of the human schemas tend to be shorter than the automatically generated ones.

schemas in two ways: (a) the matching of event triggers, and (b) the matching of event triggers and synonyms of the events in the gold schemas (synonym coverage)¹⁵. We believe that synonym coverage is a better evaluation metric to avoid errors such as considering different verbs describing the same action as different (e.g., “buy” and “acquire”) than using a predefined ontology of event types such as the one used in Li et al. (2021). The reason is twofold: firstly, any predefined ontology is limited to certain scenarios and it may impair the variety of events extracted; and secondly the typing mechanism may also inflict errors to the schema. In the calculation of coverage of relations we only take into account relations (a, b) where both events, a and b , appear in the generated schema.

From the results in Table 3, we observe that despite the difficulty of exact matching, our model with GPT-3 covers 23.73% of the gold events, showing that generated texts are useful. If we use synonym coverage as our metric, we achieve a promising coverage of 37.84% while the SOTA supervised event graph model (Li et al., 2021) covers 54.84% using limited event types. In addition, we calculated an average number of 26.19 additional events that appeared in the generated schemas and not in the LDC schema, pointing to the potential of using generated documents for expending existing schemas. With the high quality event representations obtained from the One-Pass model and the proposed logical relation induction algorithm, our method can successfully cover a high percentage of multiple types of relations.

Prediction In the prediction task, our schemas are able reach SOTA performance and predict the final outcome in 63.1% of the cases for the LDC schemas (see Tab. 4). This result is extremely impressive when it is compared with Li et al. (2021) since they predict event types instead of verbs, which is a much easier task due to the fact that the set of possible answers is limited.

Schema-11 In the soundness experiments, where the testers are asked to decide which events and relations are valid to appear in the schema, it turns out that human-schemas contain 7.14% invalid events and 15.4% invalid relations on average. For the automatically-generated schemas, 6.06% of the events and 22.9% of the relations are considered to be invalid on average, meaning that the average

¹⁵Implemented using the NLTK WordNet Python package.

	GPT2		GPT3		Li et al. (2021)
	Coverage	Coverage (Syn)	Coverage	Coverage (Syn)	Coverage
Event Match	14.88	29.55	23.73	37.84	54.84
Temporal Relations	10.80	33.31	31.07	49.99	
Hierarchical Relations	33.33	33.33	11.11	13.88	-
Logical Relations	4.16	24.99	43.76	49.81	

Table 3: Coverage results for the LDC dataset. The first row presents the percentage of events that appeared in both the LDC schemas and the automatically generated schemas (out of events in LDC schemas), and the three bottom rows present the same metric for relations of different types.

Model	Accuracy
Event Language Model	49.7
Sequential Pattern Mining	47.8
Human Schema	20.5
Event Graph Model	52.0
Zero-Shot Schema GPT2	25.0
Zero-Shot Schema Synonym GPT2	45.2
Zero-Shot Schema GPT3	35.7
Zero-Shot Schema Synonym GPT3	63.1

Table 4: Experimental results for last event prediction in the LDC dataset. The top 4 results are from (Li et al., 2021), and the metric is HITS@1 where the events are typed based on a predefined ontology.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Human	4	0	1	1	1	2	1	1	0	3	1
Automatic	2	3	4	2	1	1	1	1	4	0	1

Table 5: Distribution of votes for which is the more complete schema for Schema-11 dataset.

percentage of legitimate events is even higher in machine-generated schemas. For the completeness results presented in Tab. 5, in 4 cases the testers agreed that the automatically generated schemas are more complete; in 3 cases they claimed that the human schemas are more complete; and a tie in the remaining 4 cases. Hence our automatically generated schemas are of comparable quality to human generated ones in the sense of completeness.

Finally, in the Feigenbaum test, where testers are asked to decide whether a schema is generated by a human or a machine, eight out of eleven times they correctly identify the human-generated schema, one incorrectly, and two ties. Some of the testers who succeeded in their guesses mentioned that it was easy to determine which schema was automatically generated since it tends to be longer and comprehensive. The full results from the Feigenbaum test are shown in Appx. §B.

Wizard of Oz Experiment There seems to be a discrepancy between the low event coverage results and the quality of generated texts that were presented in Section §3. We, therefore, conducted

another experiment to identify if the problem stems from the quality of the generated documents. In this experiment, one of the authors sampled 10 complex event names from the LDC dataset and generated, using GPT-3 text davinci-002 model, 3 texts for each scenario using the prompting methods presented in Section §3. Then, the author manually extracted all relevant events and relations from each document and built a schema based solely on those events and relations.

This experiment, in which the author pretends to be the IE and schema generator models, aims to demonstrate that if we had perfect IE and schema induction systems, then the texts generated by GPT-3 would be sufficient and even superior to other corpora collected manually. The macro-average coverage of events in this experiment is 68% and the micro-average is 74%. Furthermore, GPT-3 texts generated schemas that included, on average, 6.5 additional events not mentioned in LDC schemas but relevant to the scenario at hand. As a result, we conclude that the generated texts from GPT3 contain much of the necessary information to generate schemas in a variety of topics, and can even be used to enrich existing schemas generated by other models or humans. Two example scenarios and more details appear in Appx. §D.

7 Conclusion

We propose a method to generate schemas given the sole input of a topic. We use GPT-3 to generate texts of diverse genres and a pipeline of information extraction tools to obtain relevant information before inducing logical relations and integrating the events and relations into a schema graph. To improve the efficiency of the pipeline, we implement One-Pass models for identifying temporal and hierarchical relations that achieve comparable performances with SOTA models but require far less inference time and memory space. To evaluate our framework, we conduct experiments on a

benchmark LDC dataset to show that our schemas cover a decent amount of pertinent information and display comparable ability for event prediction with supervised approaches. We observe a high percentage of valid events and relations generated for the Schema-11 dataset and the testers endorsed the completeness of our machine-generated schemas.

8 Acknowledgments

The authors would like to thank the anonymous ACL ARR reviewers for their insightful feedback on our work. This work was supported by Contract FA8750-19-2-1004 with the US Defense Advanced Research Projects Agency (DARPA). Approved for Public Release, Distribution Unlimited. This research is also based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the BETTER Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

9 Limitations

The paper presents a method for building an event schema without manually collecting documents from sources such as news articles or Wikipedia. In order to generate diverse and informative documents on any topic, we rely on large pre-trained language models. Our model, which uses GPT-3, generates schemas that are comparable to those generated by manually searching the web for documents, however, when we use inferior LMs such as GPT-2, we see a decline in performance (see [Tab. 3](#) and [Tab. 4](#)).

Our assumption is that the quality of the generated schema depends on the quality of the LM and the level of coverage of the selected topic in the LM training data. If, for instance, we were to ask our model to generate a schema for a unique topic such as "conducting an archaeological dig in an unexplored territory" we doubt that the results would be as useful to an archaeologist as if they were looking for information themselves due to the low coverage of this topic in the corpus the model was trained on.

Despite our model's reliance on pre-trained LMs, we believe the generated schemas can always serve as a good basis for further development.

10 Ethical Consideration

The proposed schema induction method does not present any direct societal implications. As is observed in [Abid et al. \(2021\)](#), the text generated by GPT-3 might include undesired social bias. Extracting events and relations from text with such social bias might potentially propagate the bias to the induced schemas. Besides, there are risks of malicious or unintended harmful uses of the generated schemas, for instance, the system might be used to inquire about making a bomb or contriving a terrorist attacks. Yet we believe that the proposed method can benefit various downstream NLP/NLU tasks like event prediction, task-oriented dialogue agents ([Andreas et al., 2020](#)) and risk detection ([Pohl et al., 2012](#)).

References

- Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Large language models associate muslims with violence. *Nature Machine Intelligence*, 3(6):461–463.
- Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitriy Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. [Task-oriented dialogue as dataflow synthesis](#). *Transactions of the Association for Computational Linguistics*, 8:556–571.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

- Nathanael Chambers. 2013. [Event schema induction with a probabilistic entity-driven model](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Seattle, Washington, USA. Association for Computational Linguistics.
- Nathanael Chambers and Dan Jurafsky. 2008. [Unsupervised learning of narrative event chains](#). In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.
- Nathanael Chambers and Dan Jurafsky. 2009. [Unsupervised learning of narrative schemas and their participants](#). In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore. Association for Computational Linguistics.
- Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. [Probabilistic frame induction](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 837–846, Atlanta, Georgia. Association for Computational Linguistics.
- Quang Do, Wei Lu, and Dan Roth. 2012. [Joint inference for event timeline construction](#). In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687, Jeju Island, Korea. Association for Computational Linguistics.
- Edward A Feigenbaum. 2003. Some challenges and grand challenges for computational intelligence. *Journal of the ACM (JACM)*, 50(1):32–40.
- Goran Glavaš, Jan Šnajder, Marie-Francine Moens, and Parisa Kordjamshidi. 2014. [HiEve: A corpus for extracting event hierarchies from news stories](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3678–3683, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Mark Granroth-Wilding and Stephen Clark. 2016. [What happens next? event prediction using a compositional neural network model](#). *Proceedings of the AAI Conference on Artificial Intelligence*, 30(1).
- Ruohao Guo and Dan Roth. 2021. [Constrained labeled data generation for low-resource named entity recognition](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4519–4533, Online. Association for Computational Linguistics.
- Rujun Han, Qiang Ning, and Nanyun Peng. 2019. [Joint event and temporal relation extraction with shared representations and structured prediction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 434–444, Hong Kong, China. Association for Computational Linguistics.
- Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. [Zero-shot transfer learning for event extraction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.
- Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. 2012. [Skip n-grams and ranking functions for predicting script events](#). In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344, Avignon, France. Association for Computational Linguistics.
- Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. [Joint entity and event coreference resolution across documents](#). In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500, Jeju Island, Korea. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. 2021. [The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5215, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020. [Connecting the dots: Event graph schema induction with path language modeling](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–695, Online. Association for Computational Linguistics.
- Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. [A joint neural model for information extraction with global features](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

- Wei Lu and Dan Roth. 2012. [Automatic event extraction with structured preference modeling](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 835–844, Jeju Island, Korea. Association for Computational Linguistics.
- Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. [Annotating noun argument structure for NomBank](#). In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).
- Ashutosh Modi. 2016. [Event embeddings for semantic script modeling](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 75–83, Berlin, Germany. Association for Computational Linguistics.
- Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2015. [Generative event schema induction with entity disambiguation](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 188–197, Beijing, China. Association for Computational Linguistics.
- Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. [Joint reasoning for temporal and causal relations](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.
- Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. [An improved neural baseline for temporal relation extraction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6203–6209, Hong Kong, China. Association for Computational Linguistics.
- Haoruo Peng and Dan Roth. 2016. [Two discourse driven language models for semantics](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 290–300, Berlin, Germany. Association for Computational Linguistics.
- Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. [Event detection and co-reference with minimal supervision](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, Austin, Texas. Association for Computational Linguistics.
- Karl Pichotta and Raymond Mooney. 2014. [Statistical script learning with multi-argument events](#). In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 220–229, Gothenburg, Sweden. Association for Computational Linguistics.
- Karl Pichotta and Raymond Mooney. 2016. [Learning statistical scripts with lstm recurrent neural networks](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).
- Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2012. Automatic sub-event detection in emergency management using social media. In *Proceedings of the 21st international conference on world wide web*, pages 683–686.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. [CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes](#). In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.
- Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. [The importance of syntactic parsing and inference in semantic role labeling](#). *Computational Linguistics*, 34(2):257–287.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. [Script induction as language modeling](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686, Lisbon, Portugal. Association for Computational Linguistics.
- Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. [Joint learning templates and slots for event schema induction](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 428–434, San Diego, California. Association for Computational Linguistics.

- Stephanie Strassel and Jennifer Tracey. 2016. [LORELEI language packs: Data, tools, and resources for technology development in low resource languages](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3273–3280, Portorož, Slovenia. European Language Resources Association (ELRA).
- Erik F. Tjong Kim Sang. 2002. [Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition](#). In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.
- Hieu Man Duc Trong, Nghia Ngo Trung, Linh Van Ngo, and Thien Huu Nguyen. 2022. [Selecting optimal context sentences for event-event relation extraction](#). *Association for the Advancement of Artificial Intelligence*.
- David Wadden, Ulme Wennberg, Yi Luan, and Hananeh Hajishirzi. 2019. [Entity, relation, and event extraction with contextualized span representations](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.
- Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020. [Joint constrained learning for event-event relation extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 696–706, Online. Association for Computational Linguistics.
- Haoyu Wang, Hongming Zhang, Muhao Chen, and Dan Roth. 2021. [Learning constraints and descriptive segmentation for subevent detection](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5216–5226, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Noah Weber, Niranjan Balasubramanian, and Nathanael Chambers. 2018. [Event representations with tensor-based compositions](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Noah Weber, Rachel Rudinger, and Benjamin Van Durme. 2020. [Causal inference of script knowledge](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7583–7596, Online. Association for Computational Linguistics.
- Haoyang Wen, Ying Lin, Tuan Lai, Xiaoman Pan, Sha Li, Xudong Lin, Ben Zhou, Manling Li, Haoyu Wang, Hongming Zhang, Xiaodong Yu, Alexander Dong, Zhenhailong Wang, Yi Fung, Piyush Mishra, Qing Lyu, Dídac Surís, Brian Chen, Susan Windisch Brown, Martha Palmer, Chris Callison-Burch, Carl Vondrick, Jiawei Han, Dan Roth, Shih-Fu Chang, and Heng Ji. 2021. [RESIN: A dockerized schema-guided cross-document cross-lingual cross-media information extraction and event tracking system](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 133–143, Online. Association for Computational Linguistics.
- Xiaodong Yu, Wenpeng Yin, and Dan Roth. 2022. [Pairwise representation learning for event coreference](#). In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 69–78, Seattle, Washington. Association for Computational Linguistics.
- Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. [Open-schema event profiling for massive news corpora](#). In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 587–596.
- Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. [Big bird: Transformers for longer sequences](#). *Advances in Neural Information Processing Systems*, 33.
- Hongming Zhang, Muhao Chen, Haoyu Wang, Yangqiu Song, and Dan Roth. 2020a. [Analogous process structure induction for sub-event sequence prediction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1541–1550, Online. Association for Computational Linguistics.
- Hongming Zhang, Haoyu Wang, and Dan Roth. 2021a. [Zero-shot Label-aware Event Trigger and Argument Classification](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1331–1340, Online. Association for Computational Linguistics.
- Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020b. [Reasoning about goals, steps, and temporal ordering with WikiHow](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639, Online. Association for Computational Linguistics.
- Yi Zhang, Sujay Kumar Jauhar, Julia Kiseleva, Ryan White, and Dan Roth. 2021b. [Learning to decompose and organize complex tasks](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2726–2735, Online. Association for Computational Linguistics.
- Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. [Temporal common sense acquisition with minimal supervision](#). In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pages 7579–7589, Online. Association for Computational Linguistics.

Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2021. [Temporal reasoning on implicit events from distant supervision](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1361–1371, Online. Association for Computational Linguistics.

A Feigenbaum Test Details

The experiment took place online through filling a Google Form and involved 11 volunteer annotators. Each annotator got 3-4 scenarios to annotate. The instructions for the survey appear in Figure 5. An example scenario and the questions of the survey are presented in Fig. 6, Fig. 7, Fig. 8, and Fig. 9.

B Feigenbaum Test Results

In this section we present all the results from the experiments on the dataset Schema-11. Tab. 6 shows the distribution of answers for the question “which schema is more complete?” (same as depicted in Tab. 5), Tab. 7 presents the distribution of answers for the question “which schema was generated by a human?” together with the correct answer written in the bottom row, and Tab. 8 presents the percentage of invalid events and relations determined by the majority vote of the annotators in the automatic schema and the human schema.

C Details on Human Schema Curation

Here are the instructions that were given to the annotators that generated the human schemas for the Schema-11 dataset. All the annotators are graduate students that previously were involved in research projects that include schema induction, SRL, NER or other relevant tasks:

We are developing a system that generates schemas automatically given a topic. We want to compare our automatically-generated schema to schemas derived by people using their common-sense (without relying on texts). To do this, we need expert human annotators and would appreciate your assistance.

A schema is defined as a list of events with their argument types, and the relationships between the events. For example, here is a schema I wrote that describes the event of “armed robbery”:

List of events and arguments:

- intend: arg0 - perpetrator [PER], arg1 - commit a felony
- acquire: arg0 - perpetrator [PER], arg1 - weapon [WEA]
- arrive: arg0 - perpetrator [PER], arg-loc - crime scene [LOC]
- assault: arg0 - perpetrator [PER], arg1 - [PER]

- threaten: arg0 - perpetrator [PER], arg1 - [PER]
- get: arg0 - perpetrator [PER], arg1 - money or goods
- injure: arg0 - perpetrator [PER], arg1 - [PER]
- kill: arg0 - perpetrator [PER], arg1 - [PER]
- flee: arg0 - perpetrator [PER], arg-loc - crime scene [LOC]
- call: arg0 - [PER], arg1 - police [ORG]
- chase: arg0 - police [ORG], arg1 - perpetrator [PER]
- catch: arg0 - police [ORG], arg1 - perpetrator [PER]
- manage to escape: arg0 - perpetrator [PER]

Temporal and logical relations (in the form of a timeline):

- a perpetrator (PER) **intent** to commit a felony ->
- the perpetrator (PER) **acquires** weapon (WEA) ->
- the perpetrator (PER) **arrives** at the scene (LOC) ->
- perpetrator (PER) **assault** victim (PER) with weapon (WEA) at the scene (LOC) **OR** perpetrator (PER) **threatens** a person (PER) with the weapon (WEA) at the scene (LOC) ->
- perpetrator (PER) **gets** money or goods from the person (PER) **OR** victim **injured OR** victim **killed** ->
- perpetrator **flees** the scene of the crime (LOC) **AND** someone (PER) **calls** the police (ORG) ->
- the police (ORG) are **chasing** the criminal (PER) ->
- the police (ORG) **catches** the perpetrator (PER) **XOR** the criminal (PER) **manages to escape**.

The complex events we are interested in are the following: (1) Disease Outbreak (2) IED Bombing (3) Civil Unrest (4) International Invasion (5) Disaster and Rescue (6) Terrorism Attacks (7) Election (8) Kidnapping (9) Business Change (10) Mass Shooting.

Feigenbaum Test - Scenario 11

This form mainly focuses on the evaluation of machine generated schema. Given a certain scenario, the schema includes stereotypical events and the relations between them, for instance, within scenario "acquiring a PhD degree", a schema would typically includes "publish papers," "attend conferences," "write PhD thesis" and "defend PhD thesis." And there is also a "before" relation between "write PhD thesis" and "defend PhD thesis." Besides, we also have "SuperSub" relation that means hierarchical relation between events, and "AND"/"OR" relation that means the two events must happen together/either of the events may happen.

We've asked a group of people to generate schemas from their commonsense knowledge. Given two schemas per scenario, your task is to determine whether you can distinguish the machine generated schema from the human generated one. And also provide your insights on the completeness and soundness of each schema.

For completeness, we would like you to tell us which schema is more complete.

For soundness, we would like you to tell us for each event and relation listed, whether it is valid for this scenario.

Most importantly, we would like to know which schema you think is generated by human.

Figure 5: Instructions for the Feigenbaum test.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Human	4	0	1	1	1	2	1	1	0	3	1
Automatic	2	3	4	2	1	1	1	1	4	0	1

Table 6: Completeness results. The table presents the number of votes that were recorded for which schema is more complete - the human generated schema or the automatically generated schema. The majority vote is highlighted in yellow.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
A	1	1	3	0	0	2	0	2	2	1	1
B	5	2	2	3	2	1	2	0	2	2	1
Correct Answer	B	B	B	B	B	A	B	A	A	B	B

Table 7: Feigenbaum test results. The annotators guesses which schema (A or B) was generated by humans. The number of votes for each option appear along with the correct answer in the bottom row. The correct majority guesses are marked with green and incorrect with red.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Invalid Events (Auto.)	0	0	0	0	0	8.33	0	7.69	0	14.28	0
Invalid Relations (Auto.)	46.15	16.66	25	25	0	23.52	0.4	11.76	12.5	22.22	46.15
Invalid Events (Human)	0	0	14.28	14.28	0	0	0	0	0	0	0
Invalid Relations (Human)	7.69	50	15.38	15.38	0	6.25	0	11.11	0	10	7.69

Table 8: Invalidity results. The table presents the percentage of invalid events and relations determined by the human annotators for each schema and scenario.

Scenario 11: Terrorism Attack (A)

Events:

1. event: kill, arg0: {PER, ORG, VEH, WEA}, arg1: PER
2. event: injure, arg0: {PER, ORG, VEH, WEA}, arg1: PER
3. event: detonate, arg0: PER, arg1: WEA
4. event: come, arg1: attack
5. event: open, arg0: {PER, ORG}, arg1: fire
6. event: wound, arg0: {PER, ORG, VEH, WEA}, arg1: PER
7. event: strike, arg0: {PER, ORG, WEA}
8. event: claim, arg0: ORG, arg1: responsibility
9. event: leave, arg0: {PER, VEH}
10. event: attack, arg0: {PER, ORG}
11. event: choose, arg0: {PER, ORG}, arg1: {PER, ORG, GPE}
12. event: select, arg0: {PER, ORG}, arg1: method
13. event: acquire, arg0: {PER, ORG}, arg1: WEA
14. event: carry out, arg0: PER

Relations:

1. before: 3->8
2. before: 3->5
3. before: 1->4
4. before: 1->9
5. before: 2->4
6. before: 2->9
7. before: 6->4
8. before: 6->9
9. before: 11->12->13->14->10
10. OR: 8,5
11. OR: 4,9
12. AND: 1,2,6
13. supersub: 10->7->1,2

Figure 6: An example schema in the topic of Terrorism Attack. This schema was generated automatically (information that was unknown to the annotators).

Scenario 11: Terrorism Attack (B)

Events:

1. event: find, arg0: PER, arg1: ORG, arg-loc:LOC
2. event: emerge, arg0: ORG, arg1: ORG
3. event: fade, arg0: ORG, arg-tmp: TMP
4. event: reemerge, arg0: ORG, arg-tmp: TMP, arg-loc: LOC
5. event: lead, arg0: ORG, arg1: losses
6. event: lost, arg0: ORG, arg1: LOC
7. event: declare, arg0: GPE, arg1: ORG
8. event: kill, arg0: GPE, arg1: PER
9. event: plan, arg0: PER
10. event: executes, arg0: PER
11. event: injures, arg0: the attack, arg1: PER
12. event: kills, arg0: the attack, arg1: PER
13. event: damages, arg0: the attack, arg1: infrastructure
14. event: calls, arg0: PER, arg1: PER
15. event: arrive, arg0: PER
16. event: treat, arg0: PER, arg1: PER
17. event: take, arg0: PER, arg1: PER
18. event: reports, arg0: PER
19. event: claims, arg0: the group, arg1: responsibility

Relations:

1. before: 9->10
2. before: 10->11
3. before: 10->12
4. before: 10->13
5. before: 10->14
6. before: 14->15->16->17
7. before: 10->18
8. before: 10->19
9. before: 1->3
10. before: 3->4
11. AND: 1->2
12. cause: 5->6
13. cause: 8->7

Figure 7: An example schema in the topic of Terrorism Attack. This schema was generated by a human (information that was unknown to the annotators).

Which schema is more complete? *

(A)

(B)

Which one do you think is generated by human? *

(A)

(B)

Figure 8: Questions that were asked about the completeness of the schemas and the generator of the schema.

For each EVENT in schema (B), select if it is valid to appear in the scenario. *

	valid	invalid
Row 1	<input type="radio"/>	<input type="radio"/>
Row 2	<input type="radio"/>	<input type="radio"/>

Figure 9: Questions about the validity of the events appearing in one of the schemas. This question was asked on both schemas and on the relations appearing in the schemas too.

D Wizard of Oz Experiment

Two examples of the Wizard of Oz experiment for the events of invasion and remodel a facility. The examples show the three generated texts using the three prompting methods, the schema created by the author, and the LDC schema together with a matching between the events in the two schemas. The entire list of sampled topics is: print a 3D object, cyber attack, make a website, religious conversion, invasion, shipping and mailing, remodel a facility, obtain and provide medical treatment, build a drone, and nonviolent protest march.

D.1 Example 1: Invasion

Prompts and Outputs:

1. Prompt: describe how to perform an invasion.
Output: 1. Choose a target: Identify a target population that you want to invade. This target population can be a group of people, a city, or a country. 2. Prepare your troops: Assemble your troops and supplies in preparation for the invasion. 3. Launch your attack: Attack the target population using whatever means necessary to take control. 4. Consolidate your power: Once you have taken control of the target population, consolidate your power by establishing your authority and eliminating any resistance.
2. Prompt: what are the steps involved in an invasion?
Output: 1. The first step in an invasion is to gather intelligence about the enemy. This includes learning about their strengths, weaknesses, and their territory. 2. The next step is to plan the invasion. This includes deciding what resources will be needed and how they will be used. 3. The third step is to launch the invasion. This is when the actual fighting begins. 4. The fourth step is to consolidate the gains. This is when the invading force secures their position and prepares for the next phase of the operation.
3. Prompt: write a news headline about invasion.
Output: U.S. troops invade Canada in surprise attack.
Prompt: write a news article titled: "U.S. troops invade Canada in surprise attack."
Output: In a stunning move, the United States has invaded Canada in a surprise attack. The

troops came across the border early this morning, catching the Canadians by surprise. The Canadians are currently fighting back, but it is unknown how long they will be able to hold out against the might of the United States military. This is a developing story, and more details will be released as they become available.

Schema generated based on the three outputs

1. Choose target.
2. Gather intelligence on target: learn strengths and weaknesses and territory.
3. Plan invasion: decide on resources and how the resources will be used.
4. Prepare troops: Assemble troops and supplies.
5. Launch attack: come across the border.
6. Fight or use whatever means possible.
7. Take control.
8. Consolidate power: establish authority and eliminate resistance.
9. Prepare for the next phase of operation.

LDC Schema

1. Step 1: PlanInvasion: Plan[3] for the invasion.
 - (a) Step 1.1: IdentifyTargets[1]: Leadership builds a list of entry points for the invasion and selects one or more as the target(s) of the initial invasion.
 - (b) Step 1.2: DevelopPlan[3]: A team develop a master invasion plan.
 - (c) Step 1.3: OrderDissemination [not matched]: Planners send out detailed instructions to military forces about the invasion plan.
2. Step 2: DeployTroops[4]: Amass and deploy troops for invasion.
3. Step 3: Invasion[5]: Military force launch the invasion and enter the target territory.
4. Step 4: DisperseTroops[6]: Dispersion of invading troops across territory.

Results: The GPT3 schema covers 6 out of 7 events that are mentioned in the LDC schema. The matching between LDC events and schema events are marked in [] next to the event on the LDC schema. GPT3 schema has 7 additional events: gather intelligence, learn strengths, decide on resources, assemble troops and supplies, come across border, take control, consolidate power, establish authority, eliminate resistance, prepare for next phase.

D.2 Example 2: Remodel A Facility

Prompts and Outputs:

1. Prompt: describe how to remodel a facility.
Output: There is no one-size-fits-all answer to this question, as the best way to remodel a facility depends on the specific needs and goals of the organization. However, some tips on how to remodel a facility include: 1. Define the scope of the project. 2. Create a budget and timeline. 3. Hire an experienced architect or designer. 4. Work with a reputable contractor. 5. Use high-quality materials. 6. Follow building codes and regulations. 7. Get input from stakeholders. 8. Test the new space before moving in.
2. Prompt: what are the steps involved in remodeling a facility?
Output: 1. The first step in remodeling a facility is to develop a plan. This plan should include the goals of the remodel, the budget, the timeline, and the scope of work. 2. The next step is to obtain the necessary permits. Depending on the scope of work, this may require a building permit, a zoning permit, and/or a permit from the local health department. 3. Once the permits are obtained, the next step is to select a contractor. The contractor should be licensed and insured, and should have experience with the type of work that needs to be done. 4. The contractor will then begin the work, which may include demolition, construction, painting, and/or installation of new fixtures and appliances. 5. Once the work is completed, the facility will need to be inspected to ensure that it meets all applicable codes and regulations. 6. Finally, the facility can be reopened to the public.
3. Prompt: write a news headline about remodeling a facility.

Output: Facility Remodeling Underway

Prompt: write a news article titled: "Facility Remodeling Underway"

Output: In an effort to better serve the needs of the community, the local library is undergoing a remodel. The project is expected to take several months to complete, but the end result will be a more user-friendly space that meets the needs of today's patrons. Some of the changes include a new layout for the adult fiction section, additional study rooms, and updated furniture throughout the facility. The children's area will also be expanded to create a more inviting and interactive space. The library staff is excited about the changes and can't wait to see the finished product. They are confident that the updates will make the library a more popular destination for both residents and visitors.

Schema generated based on the three outputs

1. Develop a plan: define scope and goals and create budget and timeline.
2. Hire an architect or designer and obtain necessary permits.
3. Select a contractor (preconditions: contractor licensed, insured, have experience).
4. Contractor works: use high quality materials AND follow building code and regulations and demolition or construction or painting or installation.
5. Get input from stakeholders.
6. Inspect facility.
7. Test space.
8. Reopen facility.
9. Facility is user-friendly and meets needs of patrons.

LDC Schema

1. Step 1: Acquisition [not matched]: Acquire facility.
2. Step 2: Planning[1]: Research and plan expected usage, budget, changes, legal issues, dependencies.

3. Step 3: Labor[2,3]: Arrange for skills, or laborers, or both.
 - (a) Step 3.1: AcquireSkills [not matched]: Acquire skills or knowledge required for remodeling.
 - (b) Step 3.2: HireLaborers[2,3]: Hire skilled person or organization to perform remodeling work.
4. Step 4: AcquireMaterials[4.1]: Acquire materials and tools.
5. Step 5: Remodel[4]: Facility is remodeled.
 - (a) Step 5.1: Demolition[4.2]: Deconstruction or demolition of portions of building and/or equipment installations.
 - (b) Step 5.2: DebrisRemoval [not matched]: Hauling away/dumping of debris.
 - (c) Step 5.3: Modification[4.2]: Modification, addition, or installation of building or systems/equipment in building.
6. Step 6: Inspection[6,7]: Inspect and/or test new portions of facility and/or new systems of facility for functionality and compliance with laws and regulations.

Results: The GPT3 schema covers 8 out of 11 events that are mentioned in the LDC schema. The matching between LDC events and schema events are marked in [] next to the event on the LDC schema. GPT3 schema has 9 additional events: contractor works, follow building code and regulations, preconditions on contractor, painting, installation, construction.

BanglaNLG and BanglaT5: Benchmarks and Resources for Evaluating Low-Resource Natural Language Generation in Bangla

Abhik Bhattacharjee¹, Tahmid Hasan¹, Wasi Uddin Ahmad², Rifat Shahriyar¹

Bangladesh University of Engineering and Technology (BUET)¹,

University of California, Los Angeles²

abhik@ra.cse.buet.ac.bd, {tahmidhasan,rifat}@cse.buet.ac.bd

Abstract

This work presents ‘BanglaNLG,’ a comprehensive benchmark for evaluating natural language generation (NLG) models in Bangla, a widely spoken yet low-resource language. We aggregate six challenging conditional text generation tasks under the BanglaNLG benchmark, introducing a new dataset on dialogue generation in the process. Furthermore, using a clean corpus of 27.5 GB of Bangla data, we pretrain ‘BanglaT5’, a sequence-to-sequence Transformer language model for Bangla. BanglaT5 achieves state-of-the-art performance in all of these tasks, outperforming several multilingual models by up to 9% absolute gain and 32% relative gain. We are making the new dialogue dataset and the BanglaT5 model publicly available at <https://github.com/csebuetnlp/BanglaNLG> in the hope of advancing future research on Bangla NLG.

1 Introduction

The emergence of pretrained language models (Devlin et al., 2019; Radford et al., 2019; Liu et al., 2019) has brought about a revolutionary change in natural language processing (NLP). With little task-specific fine-tuning, these models have achieved state-of-the-art results on many NLP tasks (Wang et al., 2018; Rajpurkar et al., 2016; Tjong Kim Sang and De Meulder, 2003). However, the focus of these models has predominantly been on natural language understanding (NLU). Even models pretrained with generative objectives (Raffel et al., 2020) concern themselves with NLU tasks more than natural language generation (NLG) tasks. Although there have been recent efforts to uplift NLG (Gehrmann et al., 2021), they are primarily geared towards high- and mid-resource languages. For example, despite being the sixth most spoken language in the world with over 230 million native speakers comprising 3% of the world’s total population,¹ Bangla has remained an underrepresented

¹<https://w.wiki/Psq>

language in the NLP literature (Joshi et al., 2020). There have been only a handful of benchmark studies on Bangla NLG (Dabre et al., 2022; Kumar et al., 2022), and that too without Bangla being the main focus. This can be attributed to the lack of diverse NLG tasks under a single benchmark and strong pretrained Bangla NLG models.

To this end, we present ‘BanglaNLG,’ a comprehensive benchmark for Bangla language generation comprising six representative tasks on machine translation, text summarization, question answering, dialogue generation, headline generation, and cross-lingual summarization. To our knowledge, BanglaNLG is the first NLG benchmark exclusively for a low-resource language.

To establish a strong baseline for this benchmark, we pretrain **BanglaT5** – a sequence-to-sequence Transformer model (Vaswani et al., 2017) pretrained on a 27.5 GB clean Bangla text corpus covering a broad range of domains. In summary:

- We develop the BanglaNLG benchmark bringing together six NLG tasks.
- We introduce a Multi-turn Dialogue dataset.
- We pretrain BanglaT5 and evaluate it on the six NLG tasks, showing strong results.

BanglaT5 outperforms similar-sized multilingual models, achieving new state-of-the-art results on three tasks with a 4% gain on average. We are releasing the BanglaT5 model and a live leaderboard to promote future research on Bangla NLG.

2 The Bangla Natural Language Generation (BanglaNLG) Benchmark

There have been sporadic works on Bangla NLG, mostly catered to machine translation (Hasan et al., 2020; Mumin et al., 2019a,b) and text summarization (Bhattacharjee et al., 2021b; Dhar et al., 2021). However, Bangla NLG lacks a unified study comprising diverse and challenging tasks. Motivated by the popular benchmarks like GLUE (Wang

Task	Corpus	Train	Dev	Test	Metric	Domain
Machine Translation	BanglaNMT, FLoRes	2,751,315	997	1,012	SacreBLEU	Misc.
Text Summarization	XL-Sum	8,102	1,012	1,012	ROUGE-2	BBC
Question Answering	BQA	127,771	2,502	2,504	EM/F1	Wikipedia
Multi-turn Dialogue	DailyDialog	76,052	7,069	6,640	BLEU-1	Misc.
News Headline Generation	XL-Sum	8,102	1,012	1,012	ROUGE-2	BBC
Cross-lingual Summarization	CrossSum	1241	153	155	ROUGE-2	BBC

Table 1: Dataset statistics and basic characteristics of BanglaNLG. Machine translation and cross-lingual summarization datasets include examples of Bangla ↔ English.

et al., 2018), XTREME (Hu et al., 2020), GEM (Gehrmann et al., 2021), that have facilitated the training/evaluation of NLP models, we establish the first-ever Bangla Natural Language Generation (BanglaNLG) Benchmark.

2.1 Task Selection Criteria

We consider the following factors while choosing the evaluation tasks:

1. Diversity: The tasks should focus on evaluating the model’s generalization capabilities. Therefore, they should vary in task nature – the input and output length, the type of generated text, the target domain, and the dataset size.

2. Practical Applicability: The choice of tasks should be driven by their practical implications. Rather than being used in abstract situations, NLG models trained on these tasks should be able to aid/reduce human effort in real-world scenarios.

3. Difficulty: The tasks should be challenging while not being unsolvable. There should be clear room for improvement to foster future research.

4. Accessibility: The selected datasets for these tasks should be openly accessible to encourage researchers to design better NLG models.

5. Evaluation: The selected tasks should have reliable automated metrics for evaluating the focused abilities of an NLG model.

2.2 Selected Tasks

Considering the criteria mentioned above, we design BanglaNLG as an aggregation of six tasks:

1. Machine Translation (MT): MT is perhaps the most studied NLG task in Bangla and the most commonly benchmarked NLG task in general. We use the BanglaNMT parallel corpus (Hasan et al., 2020), the largest Bangla-English MT dataset curated, with 2.75 million parallel pairs for training. The sentence pairs originate from various domains such as Wikipedia, news articles, religious and law

documents, etc. We evaluate the NLG models using FLoRes-100 (Goyal et al., 2022) in both directions on this dataset, i.e., Bangla to English and English to Bangla. This task is particularly challenging since it assesses an NLG model’s bilingual generation capabilities. Following standard practice, we use detokenized SacreBLEU (Post, 2018) as the evaluation metric for this task.

2. Text Summarization (TS): This task aims to generate a short and fluent summary given a long text document. We chose the Bangla portion of XL-Sum (Hasan et al., 2021) for this task. XL-Sum is a large comprehensive dataset for abstractive TS where the article and summaries are written by professional editors of BBC News. The articles cover various topics such as entertainment, politics, science, sports, etc. For this task, we use ROUGE-2² (Lin, 2004) as the evaluation metric.

3. Question Answering (QA): This is a fundamental NLP task that can be modeled as both an NLU and NLG task. We use the BQA (Bhattacharjee et al., 2022) dataset for this task. The training data is machine translated from SQuAD 2.0 (Rajpurkar et al., 2018), while the evaluation data come from the human-annotated question-answer pairs of the TyDi-QA (Clark et al., 2020) secondary gold passage task. Although TyDi-QA only contains answerable questions, BQA introduced unanswerable questions to make the task more challenging. Following SQuAD 2.0, we use Exact Match (EM) and F1 as the evaluation metrics.

4. Multi-turn Dialogue (MTD): Conversational AI is a crucial task for NLG (Chen et al., 2017). However, there is no public dataset for dialogue generation in Bangla. As such, we curate a new multi-turn dialogue dataset by translating the DailyDialog (Li et al., 2017) dataset using the English to Bangla translation model introduced by Hasan

²We use Bangla stemming supported ROUGE implementation from https://github.com/csebuetnlp/xl-sum/tree/master/multilingual_rouge_scoring.

et al. (2020). Unlike standard QA-style conversation datasets, DailyDialog reflects real-life conversations in various social situations rich in emotion, making it a perfect candidate for our benchmark. We automatically translate the training data following the same procedure described in [Bhattacharjee et al. \(2022\)](#) and have the evaluation sets manually translated by expert human translators. We use BLEU-1 as the evaluation metric for this task to properly differentiate between models since averaged BLEU scores of up to 4-gram tend to be quite low in dialogue evaluation ([Zhang et al., 2020](#)).

5. News Headline Generation (NHG): Automating headline generation can help news editors write compelling headlines to draw readers’ attention. We consider NHG as a complementary task to TS. Given an article, the objective is to generate an appropriate headline that accurately depicts the article. We repurpose the XL-Sum ([Hasan et al., 2021](#)) dataset for this task since it also includes the titles of the articles. Like TS, we use ROUGE-2 as the evaluation metric.

6. Cross-lingual Summarization (XLS): As another task for evaluating models’ bilingual generation capabilities, we consider XLS. In this task, given a piece of text in a source language, we have to generate the corresponding summary in a target language. This is potentially harder than both MT and TS considering it combines both in a single task. We consider the English-Bengali portion of the CrossSum ([Bhattacharjee et al., 2021a](#)) dataset for this task. It is curated by aligning identical articles written in different languages from the XL-Sum dataset. For evaluation, we use ROUGE-2.

We present detailed statistics of the BanglaNLG benchmark in Table 1.

3 BanglaT5

We introduce BanglaT5, a sequence-to-sequence Transformer model ([Vaswani et al., 2017](#)), to establish a strong baseline for BanglaNLG benchmark. In this section, we describe the pretraining data, objectives, and model architecture of BanglaT5.

3.1 Pretraining Data

We chose Bangla2B+ ([Bhattacharjee et al., 2022](#)) as the pretraining corpus for BanglaT5. This is a 27.5 GB dataset containing 5.25 million documents collected from a meticulously selected list of web sources. While larger sources like CCNet ([Wenzek et al., 2020](#)) and mC4 ([Xue et al., 2021](#)) are

available, these contain a lot of noise and offensive texts that are difficult to remove. For a generative model, even small amounts of unwanted texts in pretraining could lead to potentially dangerous biases in generated text ([Luccioni and Viviano, 2021](#)). Therefore, we decided not to use them.

3.2 Data Pre-processing

Following [Hasan et al. \(2020\)](#), we preprocessed the texts using their normalization pipeline³. We trained a SentencePiece ([Kudo and Richardson, 2018](#)) vocabulary of 32k subword tokens on the normalized corpus with a character coverage of 0.99995. While creating a training sample, we limited the maximum sequence length to 512 tokens for both input and output and discarded documents with a token count below 7. After tokenization, we had 4.8 million data points with an average sequence length of 402.32 tokens.

3.3 Pretraining Objective

For generative language modeling, two standard choices are decoder-only models ([Mikolov et al., 2010](#)) and encoder-decoder models ([Sutskever et al., 2014](#)). [Radford et al. \(2019\)](#) trained a decoder-only Transformer ([Vaswani et al., 2017](#)) pretrained on the conditional continuation objective. However, to provide more flexibility on generation and possible usage on understanding tasks, we only consider encoder-decoder models following the original design of the Transformer. They are generally trained with different denoising objectives to increase the encoder’s and decoder’s capacity. For instance, BART ([Lewis et al., 2020b](#)), and mBART ([Liu et al., 2020](#)) use a text-infilling-based objective. In contrast, MARGE ([Lewis et al., 2020a](#)) is a multilingual encoder-decoder model trained to reconstruct a document in one language by retrieving documents in other languages. Following [Raffel et al. \(2020\)](#), we pretrained BanglaT5 using a "span-correction" objective, empirically shown to be an optimal choice for encoder-decoder models. In this objective, consecutive spans of input tokens are replaced with a mask token, and the model is trained to reconstruct them.

3.4 Model Architecture & Hyperparameters

We pretrained the base variant of the T5 model: 12 layers, 12 attention heads, 768 hidden size, 2048 feed-forward size with GeGLU activation ([Shazeer,](#)

³<https://github.com/csebuetnlp/normalizer>

Model	Parameters	MT	TS	QA	MTD	NHG	XLS
mT5 (base)	582M	30.1/ 17.2	10.3	59.0/65.3	17.5	9.6	2.7/0.7
XLM-ProphetNet	616M	27.5/15.4	7.8	53.0/57.3	20.0	9.5	6.2/2.7
mBART-50	611M	29.7/15.5	10.4	53.4/58.9	18.5	11.2	5.4/ 3.7
IndicBART (unified)	244M	28.1/16.6	8.9	59.6/65.6	14.8	7.9	6.3/2.5
IndicBART (separate)	244M	27.5/15.7	9.2	55.3/61.2	14.1	9.1	5.3/2.4
BanglaT5	247M	31.3/17.4	13.7	68.5/74.8	19.0	13.8	6.4/4.0

Table 2: Performance comparison of the pretrained models on different BanglaNLG tasks. Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling (Koehn, 2004).

2020) with a batch size of 65536 tokens for 3 million steps on a v3-8 TPU instance on GCP. We used the Adam (Kingma and Ba, 2015) optimizer with a $3e-4$ learning rate, linear warmup of 10k steps, and ‘inverse square root’ learning rate decay.

4 Experiments & Results

We compared BanglaT5 it with four multilingual models: mT5 (base) (Xue et al., 2021), mBART-50 (Tang et al., 2020), XLM-ProphetNet (Qi et al., 2021), and IndicBART (both unified and separate script variants) (Dabre et al., 2022).⁴ All pretrained models were fine-tuned for 3-15 epochs with batch size 32 (128 for MT). We used linear warmup with a ratio of 0.1, label smoothing of 0.1 (Szegedy et al., 2016), and weight decay of $1e-6$ with the Adam optimizer (Kingma and Ba, 2015). The learning rate was tuned from the set $\{5e-5, 1e-4, 5e-4\}$. The best model was evaluated based on the validation performance after each epoch.

During inference, we used beam-search (Hayes-Roth et al., 1976) with beam size 5 (on all tasks except QA), removed duplicated trigrams during beam search (Fan et al., 2018), and used a length penalty (Wu et al., 2016) of 0.6. For QA, we used greedy decoding, i.e., picking the most probable token during each decoding step.

The evaluation results are presented in Table 2. In all the tasks, BanglaT5 outperformed all multilingual models by a considerable margin, on average 4% over the second-best, mT5. In all monolingual tasks except MTD, BanglaT5 achieves a big performance gain over others (up to 9.54% in QA), which can be attributed to the quality of the pretraining data. In MD, BanglaT5 lags marginally behind XLM-ProphetNet. We hypothesize this is due to the lack of colloquial data in Bangla2B+ since Bhat-tacharjee et al. (2022) left out such sources to avoid

⁴Due to computational budget limitations, we do not benchmark on billion-parameter models like large mT5 variants.

toxic and biased conversations.

We find the MT results particularly interesting, where BanglaT5 outperforms larger multilingual models in both directions. This suggests that despite having very little English data in the pretraining corpus, BanglaT5 can generalize well to a new translation language, given high-quality fine-tuning data. We explore this more in the Appendix. Conspicuously, all the models achieve relatively poor scores on the XLS task. This can be attributed to the smaller amount of training data.

BanglaT5 proves its superiority in compute and memory efficiency along with its performance due to its smaller size (less than half the parameters of all multilingual models except IndicBART). In practice, we observe 2-2.5x faster training and inference times with BanglaT5 than these larger multilingual models.

5 Related Works

Pretrained models NLP has witnessed a sea of change with the advent of pretrained language models like ULMfit (Howard and Ruder, 2018), ELMo (Peters et al., 2018), and most notably BERT (Devlin et al., 2019), achieving state-of-the-art results in many NLU benchmarks. Besides these NLU models, more and more pretrained models designed for NLG tasks have been proposed. Rothe et al. (2020) adopted pretrained NLU model checkpoints for generative tasks. GPT-2 (Radford et al., 2019), and later GPT-3 (Brown et al., 2020) showed that pretrained generative language models can perform remarkably well in zero-shot transfer tasks. More recently, Qi et al. (2020) proposed ProphetNet, which introduces the future n-gram prediction mechanism for language generation. Dabre et al. (2022) introduced IndicBART, which is pretrained on 11 Indic languages, including Bangla.

NLG Benchmarks Recently, many multi-task benchmarks have been proposed to drive the

progress of NLG models. [Moussallem et al. \(2020\)](#) proposed the BENG benchmark for NLG and knowledge extraction. GLGE ([Liu et al., 2021](#)) is a similar benchmark with a different set of tasks and difficulty levels. However, these benchmarks are limited to English only. [Gehrmann et al. \(2021\)](#) introduced the GEM benchmark for various tasks such as summarization ([Narayan et al., 2018](#)), data-to-text generation ([Nan et al., 2021](#)) across different languages. [Cahyawijaya et al. \(2021\)](#) introduced different tasks and baselines for 3 Indonesian languages. More recently, [Kumar et al. \(2022\)](#) introduced IndicNLG, a benchmark with five tasks in 11 Indic languages, including Bangla.

6 Conclusion & Future Works

NLP research in low-resource languages is lagging behind due to the lack of reliable benchmarks and datasets. To facilitate the development, evaluation, and comparison of new NLG models, we introduced a multi-task evaluation benchmark for Bangla NLG, a widely spoken yet low-resource language. We presented BanglaT5, a pretrained NLG model in Bangla, setting new state-of-the-art results with BanglaT5. We strongly believe that our contributions in this work will help the Bangla NLP community benchmark NLG tasks more easily under a unified setup.

In future work, we plan to introduce new tasks to BanglaNLG, such as personalized dialogue generation ([Zhang et al., 2018](#)), conversational question-answering ([Reddy et al., 2019](#)). We will also add more recent multilingual models to our comparison to BanglaT5, e.g., DeltaLM ([Ma et al., 2021](#)).

Limitations

Although [Bhattacharjee et al. \(2022\)](#) claimed that Bangla2B+, the pretraining corpus for BanglaT5, had been carefully filtered for offensive or unwanted texts, they alerted that there might be small amounts of these contents may be present, which can result in bias or toxicity in the pretrained model. We, therefore, recommend using BanglaT5 with caution, especially for real-world deployment.

Ethics Statement

License The TyDiQA dataset ([Clark et al., 2020](#)) is released under the Apache License 2.0, allowing modifications and distribution. All other pretraining and fine-tuning datasets are released under the

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0), which allows modifications and distributions for non-commercial research purposes. We strictly adhere to these licenses and will release BanglaT5 and BanglaNLG benchmark resources under CC BY-NC-SA 4.0.

Annotation Expert translators who provide translation services for renowned Bangla newspapers were hired to translate the evaluation sets of the dialogue dataset. Each translated sentence was further assessed for quality by another expert. It was again translated by the original translator if found to be of low quality. If the re-translation was found to be of low quality, it was then translated by the other expert. The experts were paid hourly as per standard rates in local currency.

Hallucinated Text It is well-known that text generation models can hallucinate outputs that may not necessarily be faithful to the original input ([Maynez et al., 2020](#)). Though the texts may be fluent and human-like, the hallucinations may be factually inconsistent and impact the outputs negatively. BanglaT5 may be susceptible to the same kinds of hallucinations.

Carbon Footprint We avoided using large models for pretraining and fine-tuning, reducing their environmental impacts. BanglaT5 was trained for about 30 days on Google v3 TPUs. Google’s TPUs are specifically designed for machine learning, which makes them up to five times more efficient than GPUs. Assuming 0.080kg carbon emission per kWh,⁵ the pretraining would emit fewer than 100kg carbon into the environment, far below most computationally demanding models. All fine-tuning experiments were done on a desktop machine with an 8-core Intel Core-i7 11700k CPU and NVIDIA RTX 3090 GPU, and no single run except machine translation took more than 12 hours, which amounts to fewer than 0.5kg carbon emission. On average, machine translation runs took three days each, emitting less than 3kg of carbon.

Acknowledgements

We would like to thank the Research and Innovation Centre for Science and Engineering (RISE), BUET, for funding the project and Google TPU Research Cloud (TRC) program for providing cloud support.

⁵<https://blog.google/technology/ai/minimizing-carbon-footprint/>

References

- Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Yuan-Fang Li, Yong bin Kang, and Rifat Shahriyar. 2021a. [Crosssum: Beyond english-centric cross-lingual abstractive text summarization for 1500+ language pairs](#). *CoRR*, abs/2112.08804.
- Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad Uddin, Kazi Mubasshir, Md. Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. [BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla](#). In *Findings of the North American Chapter of the Association for Computational Linguistics: NAACL 2022*.
- Prithwiraj Bhattacharjee, Avi Mallick, Md. Saiful Islam, and Marium-E-Jannat. 2021b. [Bengali abstractive news summarization \(bans\): A neural attention approach](#). In *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*, pages 41–51, Singapore. Springer Singapore.
- Terra Blevins and Luke Zettlemoyer. 2022. Language contamination explains the cross-lingual capabilities of english pretrained models. *arXiv preprint arXiv:2204.08110*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Samuel Cahyawijaya, Genta Indra Winata, Bryan Wilie, Karissa Vincentio, Xiaohong Li, Adhiguna Kuncoro, Sebastian Ruder, Zhi Yuan Lim, Syafri Bahar, Masayu Khodra, Ayu Purwarianti, and Pascale Fung. 2021. [IndoNLG: Benchmark and resources for evaluating Indonesian natural language generation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8875–8898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.
- Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. [TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages](#). *Transactions of the Association for Computational Linguistics*, 8:454–470.
- Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh Khapra, and Pratyush Kumar. 2022. [IndicBART: A pre-trained model for indic natural language generation](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1849–1863, Dublin, Ireland. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Nobel Dhar, Gaurob Saha, Prithwiraj Bhattacharjee, Avi Mallick, and Md Saiful Islam. 2021. [Pointer over attention: An improved bangla text summarization approach using hybrid pointer generator network](#). In *2021 24th International Conference on Computer and Information Technology (ICCIT)*, pages 1–5.
- Angela Fan, David Grangier, and Michael Auli. 2018. [Controllable abstractive summarization](#). In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.
- Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjana Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezero, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. [The GEM benchmark: Natural language generation, its evaluation and metrics](#). In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. [The Flores-101 evaluation](#)

- benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.
- Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. **XLsum: Large-scale multilingual abstractive summarization for 44 languages**. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.
- Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. **Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2612–2623, Online. Association for Computational Linguistics.
- P Hayes-Roth, M Fox, G Gill, DJ Mostow, and R Reddy. 1976. Speech understanding systems: Summary of results of the five-year research effort. *Carnegie-Mellon University, Computer Science Department Interim Report*.
- Jeremy Howard and Sebastian Ruder. 2018. **Universal language model fine-tuning for text classification**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. **XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation**. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. **The state and fate of linguistic diversity and inclusion in the NLP world**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. **Adam: A method for stochastic optimization**. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Philipp Koehn. 2004. **Statistical significance tests for machine translation evaluation**. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.
- Taku Kudo and John Richardson. 2018. **SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Aman Kumar, Himani Shrotriya, Prachi Sahu, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, Amogh Mishra, Mitesh M. Khapra, and Pratyush Kumar. 2022. **Indicnlg suite: Multilingual datasets for diverse nlg tasks in indic languages**.
- Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. **Pre-training via paraphrasing**. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20*, Red Hook, NY, USA. Curran Associates Inc.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. **BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. **DailyDialog: A manually labelled multi-turn dialogue dataset**. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Chin-Yew Lin. 2004. **ROUGE: A package for automatic evaluation of summaries**. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Dayiheng Liu, Yu Yan, Yeyun Gong, Weizhen Qi, Hang Zhang, Jian Jiao, Weizhu Chen, Jie Fu, Linjun Shou, Ming Gong, Pengcheng Wang, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, Ruofei Zhang, Winnie Wu, Ming Zhou, and Nan Duan. 2021. **GLGE: A new general language generation evaluation benchmark**. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 408–420, Online. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. **Multilingual denoising pre-training for neural machine translation**. *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. **Roberta: A robustly optimized bert pretraining approach**. *arXiv preprint arXiv:1907.11692*.

- Alexandra Luccioni and Joseph Viviano. 2021. [What’s in the box? an analysis of undesirable content in the Common Crawl corpus](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 182–189, Online. Association for Computational Linguistics.
- Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. [Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders](#). *CoRR*, abs/2106.13736.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. *Interspeech*, 2(3):1045–1048.
- Diego Moussallem, Paramjit Kaur, Thiago Ferreira, Chris van der Lee, Anastasia Shimorina, Felix Conrads, Michael Röder, René Speck, Claire Gardent, Simon Mille, Nikolai Ilinykh, and Axel-Cyrille Ngonga Ngomo. 2020. [A general benchmarking framework for text generation](#). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 27–33, Dublin, Ireland (Virtual). Association for Computational Linguistics.
- Mohammad Abdullah Al Mumin, Md Hanif Seddiqui, Muhammed Zafar Iqbal, and Mohammed Jahirul Islam. 2019a. [Neural machine translation for low-resource english-bangla](#). *Journal of Computer Science*, 15(11):1627–1637.
- Mohammad Abdullah Al Mumin, Md Hanif Seddiqui, Muhammed Zafar Iqbal, and Mohammed Jahirul Islam. 2019b. [shu-torjoma: An english-bangla statistical machine translation system](#). *Journal of Computer Science*, 15(7):1022–1039.
- Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangu Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. [DART: Open-domain structured data record to text generation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Weizhen Qi, Yeyun Gong, Yu Yan, Can Xu, Bolun Yao, Bartuer Zhou, Biao Cheng, Daxin Jiang, Jiusheng Chen, Ruofei Zhang, Houqiang Li, and Nan Duan. 2021. [ProphetNet-X: Large-scale pre-training models for English, Chinese, multi-lingual, dialog, and code generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 232–239, Online. Association for Computational Linguistics.
- Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. [ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI blog*, 1(8).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21:1–67.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. [Know what you don’t know: Unanswerable questions for SQuAD](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for](#)

- machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. **CoQA: A conversational question answering challenge**. *Transactions of the Association for Computational Linguistics*, 7:249–266.
- Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. **Leveraging pre-trained checkpoints for sequence generation tasks**. *Transactions of the Association for Computational Linguistics*, 8:264–280.
- Noam Shazeer. 2020. **Glu variants improve transformer**.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. **Sequence to sequence learning with neural networks**. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS 2014)*, pages 3104–3112, Montreal, Canada.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. **Rethinking the inception architecture for computer vision**. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. **Multilingual translation with extensible multilingual pretraining and finetuning**. *arXiv preprint arXiv:2008.00401*.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. **Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition**. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. **Attention is all you need**. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, page 6000–6010, Long Beach, California, USA.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. **GLUE: A multi-task benchmark and analysis platform for natural language understanding**. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. **CCNet: Extracting high quality monolingual datasets from web crawl data**. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. **Google’s neural machine translation system: Bridging the gap between human and machine translation**. *CoRR*, abs/1609.08144.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. **mT5: A massively multilingual pre-trained text-to-text transformer**. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. **Personalizing dialogue agents: I have a dog, do you have pets too?** In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.
- Yizhe Zhang, Siqu Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. **DIALOGPT: Large-scale generative pre-training for conversational response generation**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Supplementary Material: Appendices

A Multi-turn Dialogue Scores

In Table 3, we mention BLEU-1, BLEU-2, BLEU-3, and BLEU-4 scores for different models in the multi-turn dialogue generation task.

Model	B-1	B-2	B-3	B-4
mT5 (base)	17.54	3.67	1.25	0.43
XLNet-ProphetNet	19.98	6.06	2.98	1.86
mBART-50	18.54	5.56	2.97	2.09
IndicBART (unified)	14.75	3.18	1.06	0.37
IndicBART (separate)	14.05	3.23	1.18	0.49
BanglaT5	19.00	5.02	2.04	0.92

Table 3: Performance comparison of the pretrained models on the dialogue generation task. Scores in bold texts have statistically significant ($p < 0.05$) difference from others with bootstrap sampling (Koehn, 2004).

B Cross-lingual Capabilities of BanglaT5

Despite being a monolingual model pretrained on heavily filtered Bangla data, BanglaT5 exhibits strong cross-lingual abilities, particularly in the machine translation (MT) task. In addition to the quality and size of the fine-tuning dataset, this performance can also be attributed to the presence of a significant amount of non-Bangla tokens ($\sim 10.3\%$) in the BanglaT5 vocabulary.

Since Bhattacharjee et al. (2022) curated the Bangla2B+ corpus by document-level language filtering, these documents preserve foreign text sequences occurring in the Bangla documents. We deliberately maintain these tokens while training the vocabulary of BanglaT5, using a relatively high character coverage. Our rationale behind doing this was to capture code-switching and allow better generalization across languages co-occurring with Bangla, as well as romanized forms of Bangla texts during fine-tuning, which is reflected in the MT results. However, it should be noted that the quality and size of fine-tuning data are essential for a strong cross-lingual performance since the mere existence of foreign tokens in the vocabulary is not enough to produce meaningful generation performance, as demonstrated by the poor performance in the cross-lingual summarization (XLS) task.

This phenomenon has been studied in-depth by Blevins and Zettlemoyer (2022) in the context of pretrained language models in English, where they showed that these models develop strong

cross-lingual transfer capabilities due to the non-negligible amount of foreign text present in the pretraining data and robustness to UNK tokens during fine-tuning.

It’s about Time: Rethinking Evaluation on Rumor Detection Benchmarks using Chronological Splits

Yida Mu, Kalina Bontcheva, Nikolaos Aletras

Department of Computer Science, The University of Sheffield
{y.mu, k.bontcheva, n.aletras}@sheffield.ac.uk

Abstract

New events emerge over time influencing the topics of rumors in social media. Current rumor detection benchmarks use random splits as training, development and test sets which typically results in topical overlaps. Consequently, models trained on random splits may not perform well on rumor classification on previously unseen topics due to the temporal concept drift. In this paper, we provide a re-evaluation of classification models on four popular rumor detection benchmarks considering chronological instead of random splits. Our experimental results show that the use of random splits can significantly overestimate predictive performance across all datasets and models. Therefore, we suggest that rumor detection models should always be evaluated using chronological splits for minimizing topical overlaps.

1 Introduction

Unverified false rumors can spread faster than news from mainstream media, and often can disrupt the democratic process and increase hate speech (Vosoughi et al., 2018; Zubiaga et al., 2018). Automatic detection of rumors is an important task in computational social science, as it helps prevent the spread of false rumors at an early stage (Ma et al., 2017; Zhou et al., 2019; Karmakharm et al., 2019; Bian et al., 2020).

Current rumor detection approaches typically rely on existing annotated benchmarks consisting of social media data, e.g., Twitter 15 (Ma et al., 2017), Twitter 16 (Ma et al., 2017), Weibo (Ma et al., 2016), and PHEME (Zubiaga et al., 2016) that cover a wide range of time periods. These benchmarks use random splits for train, development and test sets which entail some topical overlap among them (see Table 1 for recent previous work). However, the distribution of topics in various NLP benchmarks (e.g., news, reviews, and biomedical) can be significantly affected by time (Huang and

Paper	Twitter 15	Twitter 16	PHEME	Weibo
Tian et al. (2022)	✓	✓	-	✓
Zeng and Gao (2022)	-	✓	✓	-
Sheng et al. (2022)	-	-	-	✓
Mukherjee et al. (2022)	-	-	✓	-
Sun et al. (2022)	✓	✓	✓	-
de Silva and Dou (2021)	✓	✓	-	-
Ren et al. (2021)	-	-	✓	-
Wei et al. (2021)	✓	✓	✓	-
Li et al. (2021)	-	-	✓	-
Rao et al. (2021)	✓	✓	-	✓
Lin et al. (2021)	✓	✓	✓	-
Farinneya et al. (2021)	-	-	✓	-
Sun et al. (2021)	-	-	✓	-
Qian et al. (2021)	-	-	✓	-
Song et al. (2021)	✓	✓	✓	-
Kochkina and Liakata (2020)	✓	✓	✓	-
Yu et al. (2020)	-	-	✓	-
Xia et al. (2020)	-	✓	-	✓
Bian et al. (2020)	✓	✓	-	✓
Lu and Li (2020)	✓	✓	-	-

Table 1: Recent work on rumor detection using random splits.

Paul, 2018, 2019). This is the phenomenon of temporal concept drift which can be induced by the changes in real-world events. Specifically, this also affects benchmarks on social media with new events such as elections, emergencies, pandemics, constantly creating new topics for discussion.

Gorman and Bedrick (2019) and Søggaard et al. (2021) have showed that using different data split strategies affects model performance in NLP downstream tasks. Previous work has demonstrated that text classifiers performance significantly drops in settings where chronological data splits are used instead of random splits in various domains, e.g., hate speech, legal, politics, sentiment analysis, and biomedical (Huang and Paul, 2018; Lukes and Søggaard, 2018; Huang and Paul, 2019; Florio et al., 2020; Chalkidis and Søggaard, 2022; Agarwal and Nenkova, 2022; Zhao et al., 2022). To minimize topical overlaps, a Leave-One-Out (LOO) evaluation protocol has been proposed (Lukasik et al., 2015, 2016). While this topic split strategy could potentially mitigate temporal concept drift, it still yields temporal overlaps between each subset and is practically not applicable to most common ru-

Dataset	id	Post	Label	Leven
Twitter 15	407231*	r.i.p to the driver who died with paul walker that no one cares about because he wasn't famous.	Rumor	3
	407236*	r.i.p to the driver that died with paul walker that no one cares about because he wasn't famous.	Rumor	
Twitter 16	594687*	the kissing islands, greenland. URL	Non-Rumor	0
	604628*	the kissing islands, greenland. URL	Non-Rumor	
PHEME	498483*	happening now in #ferguson URL	Non-Rumor	9
	499402*	Right now in #ferguson URL	Non-Rumor	
Weibo	349863*	【喝易拉罐一定要吸管】一妇女喝了罐饮料，被送进医院，离开了世界。研究显示罐上面的毒菌很多 请转给你关心的朋友 。 Translation: Please forward to your friends you care about.	Rumor	10
	350023*	【喝易拉罐一定要吸管】一妇女喝了罐饮料，被送进医院，离开了世界。研究显示罐上面的毒菌很多！！ 这些你知道么 Translation: Do you know about this?	Rumor	

Table 2: Four pairs of posts from train and test data with similar or identical text content sampled from four rumor detection benchmarks. Post ids with close values indicate that two posts are published in the same period. **Leven** denotes the Levenshtein distance (Levenshtein et al., 1966) on character-level between the two posts with the same label (i.e., lower values indicate higher text similarity and vice versa).

rumor detection benchmarks with a large number of topics (e.g., Twitter 15, Twitter 16, Weibo, etc.). We observe that the LOO protocol can be used for a few specific rumor detection benchmarks, such as (PHEME (Zubiaga et al., 2016)), where each post is associated with a corresponding event, e.g. *Ottawa Shooting* and *Charlie Hebdo shooting*.

Using random splits also results into posts with almost identical textual content shared during the same period. Table 2 displays four pairs of posts with **similar or identical** text content sampled from four different rumor detection benchmarks. This potential information leakage, results in classifying data almost identical to ones already being present in the training set. For practical application reasons, we believe that in order to evaluate a rumor detection system, it is necessary to detect not only long-standing rumors, but also emerging ones.

In this paper, we design a battery of controlled experiments to explore the hypothesis that whether temporality affects the predictive performance of rumor classifiers. To this end, we re-evaluate models on popular rumor detection benchmarks using chronological data splits i.e., by training the model with earlier posts and evaluating the model performance with the latest posts. Results show that the performance of rumor detection approaches trained with random data splits is significantly overestimated than chronological splits due to temporal concept drift. This suggests that rumor detection approaches should be evaluated with chronological data for real-world applications, i.e., to automatically detect emerging rumors.

2 Methodology

2.1 Data

We use four most popular rumor detection benchmarks, three in English and one in Chinese. Note

that most related work is currently evaluating their rumor detection systems on two or three of these four benchmarks. (see Table 1).

Twitter 15 and Twitter 16: These datasets contain 1,490 and 818 tweets labeled into four categories including Non-rumor (NR), False Rumor (FR), True Rumor (TR), and Unverified Rumor (UR) introduced by Ma et al. (2017).

PHEME: This benchmark contains 5,802 verified tweets collected from 9 real-world breaking news events (e.g., Ottawa Shooting, Ferguson Unrest, etc.) associated with two labels, i.e., 1,972 Rumor and 3,830 Non-Rumor (Zubiaga et al., 2016).

Weibo: This dataset includes 4,664 verified posts in Chinese including 2,313 rumors debunked by the Weibo Rumor Debunk Platform¹ and 2,351 non-Rumors from Chinese media (Ma et al., 2016).

Data Pre-processing We opt for the binary setup (i.e., re-frame all benchmarks as rumor detection) to distinguish true/false information following Lu and Li (2020); Rao et al. (2021). We pre-process the posts by replacing @mention and hyperlinks with @USER and URL respectively. We also lowercase the tweets from three Twitter benchmarks.

2.2 Data Splits

Standard Chronological Splits For Twitter 15 and PHEME, we first sort all posts chronologically and then divide them into three subsets including a training set (70% of the earliest data), a development set (10% of data after train and before test), and a test set (20% of the latest data). There is no temporal overlap between the three subsets.

¹<https://service.account.weibo.com/?type=5&status=4>

Splits	Benchmarks Subsets	Twitter 15			Twitter 16			PHEME			Weibo		
		Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
Standard Chronological	# of Rumors	285	35	52	-	-	-	1,420	72	480	-	-	-
	# of Non-Rumors	234	40	96	-	-	-	2,641	508	681	-	-	-
Stratified Chronological	# of Rumors	260	37	75	144	21	40	1,380	197	394	1,645	235	470
	# of Non-Rumors	259	37	74	144	21	40	2,681	383	766	1,619	231	463
Random Splits	# of Rumors	260	37	75	144	21	40	1,380	197	394	1,645	235	470
	# of Non-Rumors	259	37	74	144	21	40	2,681	383	766	1,619	231	463

Table 3: Statistics of subsets. Note that using random splitting yields the same percentage of examples in each category as in the stratified chronological splits.

Stratified Chronological Splits On the other hand, we observe that there is no temporal overlap between rumors and non-rumors in Twitter 16 and Weibo datasets. This suggests that it is not possible to use standard chronological splits as in Twitter 15 and PHEME.

Therefore, we apply a **stratified chronological split** strategy for all benchmarks. We first split rumors and non-rumors separately in chronological order. We then divide them into three subsets (a total of six subsets), i.g., all rumors are split into a training set (70% of the earliest rumors), a development set (10% of data after train and before test), and a test set (20% of the latest rumors). Finally, we merge the six subsets into the final three train, development and test sets. Note that this approach will result in no temporal overlap for **each label (i.e., rumor or non-rumor)** among the three final sets. We show the number of each split in Table 3.

Random Splits Following standard practice (e.g., [Bian et al. 2020](#); [Lin et al. 2021](#); [Rao et al. 2021](#)), we **randomly** split data using a 5-fold cross-validation. Note that these splits are made by preserving the percentage of posts in each category. Each split contains a training set (70%), development set (10%) and a test set (20%) with the same ratio as in our chronological splits.

Leave-One-Out (LOO) Splits For reference, we also provide the results of using the LOO evaluation protocol on PHEME dataset (see Table 5).

2.3 Models

The main purpose of our experiments is to improve model evaluation by investigating the effects of temporal drifts in rumor detection by providing an extensive empirical study. Therefore, we opted using strong text classifiers that are generic and can be applied to all of our benchmarks:

- **LR** We train a LR classifier using BOW to represent posts weighted by TF-IDF using a vocabulary of 5,000 n-grams.

- **BERT** We directly fine-tune the BERT base model by adding a linear prediction layer on the top of the 12-layer transformer architecture following ([Devlin et al., 2019](#)).
- **BERT+ (BERTweet and ERNIE)** We also experiment with two domain specific models: BERTweet ([Nguyen et al., 2020](#)) and ERNIE ([Sun et al., 2020](#)) pre-trained on social media data using the same fine-tune strategy as the original BERT model.

2.4 Hyperparameters and Implementation Details

We train the model on the training set, perform model tuning and selecting on the development set, and evaluate performance on the test set. To evaluate the chronological data splits, we run the model five times with different random seeds for consistency. All chronological splits are available for reproducibility.²

For logistic regression, we use word-level and character-level tokenizers for Twitter and Weibo datasets respectively and only consider uni-gram, bi-grams, and tri-grams that appear in more than two posts for each dataset. For BERT, we set learning rate $lr = 2e - 5$, batch size $bs = 32$, and maximum input length as 256 covering the max tokens of all posts. All BERT-style models are trained for 10 epochs using the early stopping method based on the loss on the development set. The best checkpoint model is saved for evaluation on the test set. The average run time of 10 epochs for the BERT model is less than 2 minutes. We employ Bert-Base-Uncased, Bertweet-Base and Chinese-Bert-WWM, Ernie-1.0 models from the HuggingFace library ([Wolf et al., 2020](#)). All experiments are conducted on a single NVIDIA V100 GPU with 32GB memory.

²https://github.com/YIDAMU/Rumor_Benchmarks_Temporality

Model	Strategy	Twitter15			PHEME		
		P	R	F1	P	R	F1
LR	Random	86.7 ± 2.1	85.2 ± 1.8	85.0 ± 1.8	84.1 ± 1.2	79.3 ± 1.0	80.9 ± 1.0
	Standard Chronological	56.6 ± 0.8	56.3 ± 0.7	56.4 ± 0.7	67.3 ± 0.1	64.0 ± 0.1	63.9 ± 0.1
	Stratified Chronological	56.3 ± 2.5	51.9 ± 0.7	41.4 ± 0.4	64.5 ± 0.2	63.0 ± 0.3	63.5 ± 0.3
BERT	Random	88.2 ± 2.4	87.9 ± 2.2	87.9 ± 2.2	84.8 ± 0.5	84.8 ± 1.2	84.8 ± 0.8
	Standard Chronological	54.8 ± 4.0	55.1 ± 4.3	52.9 ± 3.6	74.8 ± 1.1	75.1 ± 0.8	73.7 ± 0.4
	Stratified Chronological	58.2 ± 7.3	56.1 ± 4.5	52.8 ± 5.6	75.5 ± 0.6	77.7 ± 0.5	75.7 ± 1.1
BERT+	Random	90.8 ± 1.2	90.4 ± 1.2	90.4 ± 1.2	84.6 ± 1.0	85.5 ± 0.9	85.0 ± 0.8
	Standard Chronological	58.6 ± 1.9	58.8 ± 2.1	57.4 ± 2.5	76.1 ± 1.1	74.8 ± 1.5	71.6 ± 2.2
	Stratified Chronological	61.8 ± 6.5	57.9 ± 2.4	55.2 ± 1.5	75.3 ± 0.9	76.9 ± 2.1	71.0 ± 3.5
Model	Strategy	Twitter16			Weibo		
LR	Random	89.9 ± 1.2	89.3 ± 1.5	89.3 ± 1.5	90.1 ± 0.9	90.1 ± 0.9	90.1 ± 0.9
	Stratified Chronological	62.1 ± 6.9	55.8 ± 4.7	48.7 ± 11.4	79.1 ± 0.1	78.1 ± 0.1	77.9 ± 0.1
	BERT	Random	91.9 ± 1.0	91.5 ± 0.8	91.5 ± 0.8	92.3 ± 1.2	92.2 ± 1.2
BERT	Stratified Chronological	61.0 ± 11.2	54.3 ± 4.3	47.2 ± 3.5	89.0 ± 2.5	87.6 ± 2.6	87.5 ± 2.6
	BERT+	Random	89.8 ± 2.8	89.3 ± 3.2	89.3 ± 3.3	92.5 ± .4	92.5 ± .4
BERT+	Stratified Chronological	49.8 ± 1.7	49.9 ± 0.9	45.1 ± 2.9	88.1 ± 2.5	87.6 ± 1.4	88.5 ± 1.5

Table 4: Rumor detection prediction results across different data split methods. Green cells indicate that the model trained on random splits performs significantly better than both standard chronological splits and stratified chronological splits ($p < 0.05$, t-test).

Model	PHEME		
	P	R	F1
LR	68.3 ± 3.8	65.1 ± 6.3	63.2 ± 6.3
BERT	73.4 ± 3.1	71.9 ± 6.1	70.7 ± 4.9
BERT+	75.3 ± 2.2	72.6 ± 8.1	71.4 ± 7.0

Table 5: Leave-One-Out evaluation protocol on PHEME dataset.

2.5 Evaluation Metrics

For all tasks, we report the averaged macro Precision, Recall and F1 values across five runs using different random seeds.

3 Results

Random Splits vs. Chronological Splits Table 4 shows the experimental results across all models and rumor detection benchmarks using **chronological splits** and random **5-fold cross-validation**. Overall, we observe that the use of random splits always leads to a significant overestimation of performance compared to chronological splits (t-test, $p < 0.05$) across all models. Our results corroborate findings from previous work on studying temporal concept drift (Huang and Paul, 2018; Chalkidis and Sogaard, 2022). This suggests that chronological splits are necessary to more realistically evaluate rumor detection models.

We also note that the effect of temporality varies in datasets of different size. For both data splitting strategies, we observe that the difference in performance is 50% higher for the two datasets with hundreds of posts (e.g., Twitter 15 and Twitter 16) and around 10% in ones with thousands of posts (e.g., PHEME and Weibo). For rumor detection

tasks, temporality may have a greater impact on small-scale benchmarks than on large-scale benchmarks. For Twitter 16 and Weibo, the use of stratified chronological splits demonstrates significant performance drops compared to random splits due to the temporal concept drift.

For chronological splits, we observe that pre-trained language models (i.e., BERT and BERT+) significantly outperform (t-test, $p < 0.05$) logistic regression in all benchmarks. This is due to the fact that BERT-style models (i) outperform simpler linear models by a large margin in various NLP tasks (Devlin et al., 2019); and (ii) have been trained after the development of these four benchmarks implying some information leakage.

Standard vs. Stratified Chronological Splits

Note that dividing the datasets into standard chronological splits results in subsets that do not preserve the sample percentages for each category (see Table 3). The upper part of Table 4 displays the difference in model performance between two types of chronological splits on Twitter 15 and PHEME. We observe that using both standard and stratified chronological splits results in similar model predictive performance (t-test, $p > 0.05$). Even though stratified chronological splits contain temporal overlap, it is still not sufficient to improve model performance compared to random splits. This suggests that the temporal drift affects particular classes rather than the entire data set.

4 Error Analysis

Finally, we perform an error analysis to further investigate the type of errors made by BERT us-

Benchmark		Twitter 15			Twitter 16			PHEME			Weibo		
Splits	Test set	total	#	%	total	#	%	total	#	%	total	#	%
Chrono.	all posts	148	3	2%	82	6	7%	1161	39	3%	933	41	4%
	# of wrong predictions	63	2	1%	34	2	2%	301	5	<1%	99	7	<1%
	# of correct predictions	85	1	1%	48	4	5%	860	34	3%	834	34	4%
Random	all posts	149	35	23%	83	26	30%	1161	181	16%	933	129	14%
	# of wrong predictions	12	0	0%	5	1	1%	150	14	1%	65	4	<1%
	# of correct predictions	137	35	23%	78	25	30%	1011	167	14%	868	125	14%

Table 6: Error Analysis for all benchmarks. # denotes the number of posts that are similar to posts from training set, i.e., known data. % denote the percentage of similar posts in the test set. We set the threshold value to 20, which indicates that there are two or three different words between the two tweets.

	Example	Test	Train	Correct	Wrong
Twitter 15	#rip to the driver who died with #paulwalker that no one cares about because he wasn't famous.	4	6	4	0
Twitter 16	steve jobs was adopted. his biological father was abdufatah jandali, a syrian muslim	2	13	2	0
PHEME	Police are leaving now . #ferguson HTTPURL	4	11	4	0
Weibo	【交通新规】2013年1月1日施行:1... 扩散给大家! [广州日报] Translation: [New driving laws] From 1 Jan 2013: Running a red light will result in a fine of 100 RMB and 6 points. ... Spread the news to everyone! [Guangzhou Daily]	2	6	2	0

Table 7: Four examples of correct predictions using random splits, which artificially removes temporal concept drift. For example, in Twitter 15, there are 4 and 6 similar posts about rumors related to Paul Walker in the test set and the training set respectively.

ing both random and chronological splits. Table 6 shows the number of correct and wrong predictions for each of the two data splitting strategies. We also use the Levenshtein distance³ to calculate the quantity of posts in the test set that are similar to posts in the corresponding train set.

- We first observe that the temporal concept drift is evident in all rumor detection benchmarks. Most of the rumors on the same topic are posted in a very short time span.
- In addition, long-standing rumors are only a small part of the data (less than 5%). Second, we note that using random splits leads to topical overlap between the training and test sets (see Table 7) resulting in higher model performance.
- Finally, for both random and chronological splits, most of the posts in the test set with overlapping topics in the training set are predicted correctly. In contrast, wrong predictions are often posts with emerging or different topics compared to the posts in the train set.

5 Conclusion

We have shed light on the impact of temporal drift on computational rumor detection. Results from our controlled experiments show that the use of chronological splits causes substantially drops in predictive performance across widely-used rumor

³We set the threshold value to 20.

detection benchmarks. This suggests that random splits rather overestimate the model predictive performance. We argue that the temporal concept drift needs to be considered when developing real-world rumor detection approaches. In the future, we plan to study the impact of temporal concept drift on other NLP tasks, such as detecting user reactions to untrustworthy posts on social media (Glenski et al., 2018; Mu and Aletras, 2020; Mu et al., 2022).

Limitations

We provide the first re-evaluation of four standard rumor detection benchmarks in two languages (English and Chinese) from two platforms (Twitter and Weibo). We acknowledge that further investigation is needed in rumor detection datasets in other languages. We provide an error analysis in Section 4.

Acknowledgments

We would like to thank Ahmed Alajrami, Danae Sánchez Villegas, Mali Jin, Xutan Peng and all the anonymous reviewers for their valuable feedback.

References

- Oshin Agarwal and Ani Nenkova. 2022. Temporal effects on pre-trained models for language processing tasks. *Transactions of the Association for Computational Linguistics*, 10:904–921.
- Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 549–556.

- Ilias Chalkidis and Anders Søgaard. 2022. [Improved multi-label classification under temporal concept drift: Rethinking group-robust algorithms in a label-wise setting](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2441–2454, Dublin, Ireland. Association for Computational Linguistics.
- Nisansa de Silva and Dejing Dou. 2021. [Semantic oppositionness assisted deep contextual modeling for automatic rumor detection in social networks](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 405–415, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Parsa Farinneya, Mohammad Mahdi Abdollah Pour, Sardar Hamidian, and Mona Diab. 2021. Active learning for rumor identification on social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4556–4565.
- Komal Florio, Valerio Basile, Marco Polignano, Pierpaolo Basile, and Viviana Patti. 2020. Time of your hate: The challenge of time in hate speech detection on social media. *Applied Sciences*, 10(12):4180.
- Maria Glenski, Tim Weninger, and Svitlana Volkova. 2018. [Identifying and understanding user reactions to deceptive and trusted social news sources](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 176–181, Melbourne, Australia. Association for Computational Linguistics.
- Kyle Gorman and Steven Bedrick. 2019. [We need to talk about standard splits](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.
- Xiaolei Huang and Michael J. Paul. 2018. [Examining temporality in document classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 694–699, Melbourne, Australia. Association for Computational Linguistics.
- Xiaolei Huang and Michael J. Paul. 2019. [Neural temporality adaptation for document classification: Diachronic word embeddings and domain adaptation models](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4113–4123, Florence, Italy. Association for Computational Linguistics.
- Twin Karmakharm, Nikolaos Aletras, and Kalina Bontcheva. 2019. [Journalist-in-the-loop: Continuous learning as a service for rumour analysis](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 115–120, Hong Kong, China. Association for Computational Linguistics.
- Elena Kochkina and Maria Liakata. 2020. [Estimating predictive uncertainty for rumour verification models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6964–6981, Online. Association for Computational Linguistics.
- Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, pages 707–710. Soviet Union.
- Jiawen Li, Shiwen Ni, and Hung-Yu Kao. 2021. [Meet the truth: Leverage objective facts and subjective views for interpretable rumor detection](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 705–715, Online. Association for Computational Linguistics.
- Hongzhan Lin, Jing Ma, Mingfei Cheng, Zhiwei Yang, Liangliang Chen, and Guang Chen. 2021. [Rumor detection on Twitter with claim-guided hierarchical graph attention networks](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10035–10047, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yi-Ju Lu and Cheng-Te Li. 2020. [GCAN: Graph-aware co-attention networks for explainable fake news detection on social media](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514, Online. Association for Computational Linguistics.
- Michal Lukasik, Trevor Cohn, and Kalina Bontcheva. 2015. [Classifying tweet level judgements of rumours in social media](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2590–2595, Lisbon, Portugal. Association for Computational Linguistics.
- Michal Lukasik, P. K. Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. 2016. [Hawkes processes for continuous time sequence classification: an application to rumour stance classification in Twitter](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–398, Berlin, Germany. Association for Computational Linguistics.
- Jan Lukes and Anders Søgaard. 2018. [Sentiment analysis under temporal shift](#). In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages

- 65–71, Brussels, Belgium. Association for Computational Linguistics.
- Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 3818–3824.
- Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. [Detect rumors in microblog posts using propagation structure via kernel learning](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 708–717, Vancouver, Canada. Association for Computational Linguistics.
- Yida Mu and Nikolaos Aletras. 2020. Identifying twitter users who repost unreliable news sources with linguistic information. *PeerJ Computer Science*, 6:e325.
- Yida Mu, Pu Niu, and Nikolaos Aletras. 2022. [Identifying and characterizing active citizens who refute misinformation in social media](#). In *14th ACM Web Science Conference 2022, WebSci '22*, page 401–410, New York, NY, USA. Association for Computing Machinery.
- Rajdeep Mukherjee, Uppada Vishnu, Hari Chandana Peruri, Sourangshu Bhattacharya, Koustav Rudra, Pawan Goyal, and Niloy Ganguly. 2022. [Mtlts: A multi-task framework to obtain trustworthy summaries from crisis-related microblogs](#). In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 755–763.
- Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. [BERTweet: A pre-trained language model for English tweets](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.
- Shengsheng Qian, Jinguang Wang, Jun Hu, Quan Fang, and Changsheng Xu. 2021. Hierarchical multi-modal contextual attention network for fake news detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 153–162.
- Dongning Rao, Xin Miao, Zhihua Jiang, and Ran Li. 2021. [STANKER: Stacking network based on level-grained attention-masked BERT for rumor detection on social media](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3347–3363, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiaoying Ren, Jing Jiang, Ling Min Serena Khoo, and Hai Leong Chieu. 2021. [Cross-topic rumor detection using topic-mixtures](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1534–1538, Online. Association for Computational Linguistics.
- Qiang Sheng, Juan Cao, Xueyao Zhang, Rundong Li, Danding Wang, and Yongchun Zhu. 2022. Zoom out and observe: News environment perception for fake news detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4543–4556.
- Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. [We need to talk about random splits](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832, Online. Association for Computational Linguistics.
- Yun-Zhu Song, Yi-Syuan Chen, Yi-Ting Chang, Shao-Yu Weng, and Hong-Han Shuai. 2021. [Adversary-aware rumor detection](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1371–1382, Online. Association for Computational Linguistics.
- Mengzhu Sun, Xi Zhang, Jianqiang Ma, and Yazheng Liu. 2021. [Inconsistency matters: A knowledge-guided dual-inconsistency network for multi-modal rumor detection](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1412–1423, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Tiening Sun, Zhong Qian, Sujun Dong, Peifeng Li, and Qiaoming Zhu. 2022. [Rumor detection on social media with graph adversarial contrastive learning](#). In *Proceedings of the ACM Web Science Conference 2022, WWW '22*, page 2789–2797, New York, NY, USA. Association for Computing Machinery.
- Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8968–8975.
- Lin Tian, Xiuzhen Zhang, and Jey Han Lau. 2022. [DUCK: Rumour detection on social media by modelling user and comment propagation networks](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4939–4949, Seattle, United States. Association for Computational Linguistics.
- Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.
- Lingwei Wei, Dou Hu, Wei Zhou, Zhaojuan Yue, and Songlin Hu. 2021. [Towards propagation uncertainty: Edge-enhanced Bayesian graph convolutional networks for rumor detection](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pages 3845–3854, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Rui Xia, Kaizhou Xuan, and Jianfei Yu. 2020. [A state-independent and time-evolving network for early rumor detection in social media](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9042–9051, Online. Association for Computational Linguistics.

Jianfei Yu, Jing Jiang, Ling Min Serena Khoo, Hai Leong Chieu, and Rui Xia. 2020. [Coupled hierarchical transformer for stance-aware rumor verification in social media conversations](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1392–1401, Online. Association for Computational Linguistics.

Fengzhu Zeng and Wei Gao. 2022. Early rumor detection using neural hawkes process with a new benchmark dataset. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4105–4117.

Zhixue Zhao, George Chrysostomou, Kalina Bontcheva, and Nikolaos Aletras. 2022. [On the impact of temporal concept drift on model explanations](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4039–4054, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kaimin Zhou, Chang Shu, Binyang Li, and Jey Han Lau. 2019. [Early rumour detection](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1614–1623, Minneapolis, Minnesota. Association for Computational Linguistics.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys*, 51(2):1–36.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989.

MUTANT: A Multi-sentential Code-mixed Hinglish Dataset

Rahul Gupta
IIT Gandhinagar
Gandhinagar, Gujarat, India
gupta.rahul@iitgn.ac.in

Vivek Srivastava
TCS Research
Pune, Maharashtra, India
srivastava.vivek2@tcs.com

Mayank Singh
IIT Gandhinagar
Gandhinagar, Gujarat, India
singh.mayank@iitgn.ac.in

Abstract

The multi-sentential long sequence textual data unfolds several interesting research directions pertaining to natural language processing and generation. Though we observe several high-quality long-sequence datasets for English and other monolingual languages, there is no significant effort in building such resources for code-mixed languages such as Hinglish (code-mixing of Hindi-English). In this paper, we propose a novel task of identifying multi-sentential code-mixed text (MCT) from multilingual articles. As a use case, we leverage multilingual articles from two different data sources and build a first-of-its-kind multi-sentential code-mixed Hinglish dataset i.e., MUTANT. We propose a token-level language-aware pipeline and extend the existing metrics measuring the degree of code-mixing to a multi-sentential framework and automatically identify MCT in the multilingual articles. The MUTANT dataset comprises 67k articles with 85k identified Hinglish MCTs. To facilitate future research directions, we will make the dataset and the code publicly available upon publication.

1 Introduction

Over the years, we have seen enormous downstream applications of multi-sentential datasets in the areas such as question-answering (Joshi et al., 2017; Tapaswi et al., 2016), summarization (Sharma et al., 2019; Cachola et al., 2020), machine translation (Bao et al., 2021), etc. The existing state-of-the-art methods prove challenging to scale effectively and efficiently on multi-sentential long sequence text (Ainslie et al., 2020), which unplugs several exciting research avenues. Unfortunately, to a large extent, the majority of the research on multi-sentential data is dominated by a few popular monolingual languages such as English, Chinese, and Spanish. Due to this, code-mixed languages (among other low-resource and under-explored languages) suffer from non-existent works in the aforementioned areas of interest.

(A) TITLE: रिलीज से 4 दिन पहले मुश्किल में Salman Khan की Bharat, दिल्ली हाईकोर्ट में PIL दाखिल
MCT: ...सलमान खान (Salman Khan) की फिल्म 'भारत' (Bharat) पांज जून को रिलीज होने वाली है, लेकिन उससे पहले फिल्म के सामने एक मुश्किल आ गई है। दरअसल, फिल्म के नाम को लेकर हाईकोर्ट में याचिका दाखिल की गई है। याचिकाकर्ता को फिल्म के नाम से आपत्ति है। उनका कहना है कि इस टाइटल से लोगों की भावनाएं आहत हो सकती हैं। दाखिल की गई याचिका में फिल्म का नाम बदलने की गुंजरिश की गई है।...

(B) TITLE: मन की बात : जनवरी 2016
MCT: ...मेरे प्यारे नौजवान साथियो, 15 अगस्त को लाल किले से मैंने 'Start-up India, Stand-up India' उसके संबंध में एक प्राथमिक चर्चा की थी। उसके बाद सरकार के सभी विभागों में ये बात चल पड़ी। क्या भारत 'Start-up Capital' बन सकता है? क्या हमारे राज्यों के बीच नौजवानों के लिए एक उत्तम अवसर के रूप में नये - नये Start-ups, अनेक with Start-ups, नये-नये Innovations! चाहे manufacturing में हो, चाहे Service Sector में हो, चाहे Agriculture में हो। हर चीज़ में नयापन, नया तरीका, नयी सोच, दुनिया Innovation के बिना आगे बढ़ती नहीं है।...

Figure 1: Example MCT and the corresponding article's title form two multilingual data sources: (A) Dainik Jagran news article and (B) Man-ki-baat speech transcript. We color code the tokens as: English, Hindi, and language independent.

We posit that due to several inherent challenges, the NLP community hold back on building multi-sentential datasets for the low-resource and code-mixed languages. One of the most significant bottlenecks in building such resources is the unavailability of MCT on traditional and widely popular data sources such as social media platforms where the short-length and noisy code-mixed text is available in abundance. It presents several challenges such as the difficulty in curating a large-scale multi-sentential dataset at ease. Another major challenge is the lack of metrics to measure the degree of code-mixing in the multi-sentential framework. The existing metrics such as code-mixing index (Das and Gambäck, 2014) and multilingual-index (Barnett et al., 2000) already suffers from major limitations (Srivastava and Singh, 2021a) in the short-length text format. In such a scenario, it gets mystifying

Dataset	Task(s)	Data Source(s)	# Instances	Avg Tokens	Avg Sentences	Retrieval
(Srivastava and Singh, 2020)	Machine Translation	Social media posts on Twitter & Facebook	13738	13	1.04	Automatic
(Khanuja et al., 2020)	Natural Language Inference	Hindi Bollywood movie transcripts	2240	87	7.15	Automatic
(Mehnaz et al., 2021)	Dialogue Summarization	Manual translation of dialogues and summaries from (Gliwa et al., 2019)	6830	31	7.85	-
(Srivastava and Singh, 2021b)	Generation & Evaluation	IIT-B En-Hi parallel corpus (Kunchukuttan et al., 2018)	1974	20	1.05	-
MUTANT	Summarization	Speech transcripts, press releases, and news articles	84937	159	10.23	Manual + Automatic

Table 1: Comparison of the MUTANT dataset with the currently available datasets in the Hinglish language.

to build a retrieval pipeline to identify MCT and we need to depend heavily on the expertise of human annotators which is a time and cost-demanding exercise. In this work, we address both of these challenges. As a representative use case, we base our work on Hinglish, a popular code-mixed language in the Indian subcontinent. But the insights from our exploration could be extended to other code-mixed language pairs.

To address the first challenge, we identify two non-traditional multilingual data sources¹ i.e., political speeches and press releases along with Hindi daily news articles (discussed in detail in Section 3). Figure 1 shows example Hinglish MCTs from two multilingual data sources. To address the second challenge, we propose a token-level language-aware pipeline and extend a widely popular metric (i.e., code-mixing index) measuring the degree of code-mixing in a multi-sentential framework. We demonstrate the effectiveness of the proposed pipeline with a minimal task-specific annotation which significantly reduces the overall human effort (discussed in detail in Section 4).

Eventually, we build a novel multi-sentential dataset for the Hinglish language with 85k MCTs identified from 67k articles. In Table 1, we compare MUTANT with four other Hinglish datasets (Srivastava and Singh, 2020; Khanuja et al., 2020; Mehnaz et al., 2021; Srivastava and Singh, 2021b) proposed for a variety of tasks such as machine translation, natural language inference, generation, and evaluation. The MUTANT dataset has a significantly higher average number of sentences along with longer MCT (high average number of tokens). Alongside, the dataset notably consists of a higher number of data instances which is a rarity for the code-mixed datasets (Srivastava and Singh, 2021a).

¹these data sources have not been actively employed in building datasets for the code-mixed languages

2 Multi-sentential Code-mixed Text Span (MCT)

Due to the absence of a formal definition of MCT in the literature, we propose and use the following definition of MCT throughout this work:

MCT: Consider a multilingual article $A = \{s_1, s_2, \dots, s_n\}$ consisting of n sentences denoted by s_i where $i \in [1, n]$. A unique non-overlapping MCT M_p in A is a chunk of $m > 1$ consecutive sentences i.e. $M_p = \{s_k, s_{k+1}, \dots, s_{k+m-1}\}$. M_p should satisfy the following two properties:

1. $P1$: At least one s_{k+j} in M_p should be code-mixed. Trivially, at most $m-1$ s_{k+j} in M_p could be monolingual. Here, $j \in [0, m-1]$.
2. $P2$: s_k in M_p is either the first sentence of the article or preceded by a line break. Likewise, s_{k+m-1} is either the last sentence of the article or succeeded by a line break.

It should be noted that an article A can have multiple non-overlapping unique MCTs i.e. $A = \{M_1, M_2, \dots, M_q\}$ where $q \geq 0$.

3 Multilingual and Multi-sentential Data Sources

Over the years, we observe several interesting and diverse code-mixed data sources such as Twitter, Facebook, movie transcripts, etc. Social media sites have acted as the cornerstone of the code-mixed data collection pipelines due to the ease of availability of large-scale data. Nonetheless, they present several challenges such as noisy data, short text, abusive, and multimodal data. Given the requirements of *MUTANT* (i.e. multi-sentential and high-quality data), we refrain from using social media sites in this work. Here, we focus on two major data sources:

3.1 Political speeches and press releases

Here, we scrape data from five different web sources. Collectively, we denote this data source as D_{speech} .

Aam Aadmi Party press releases (AAP): We scrape the press releases from the official website of Aam Aadmi Party². We have scraped 320 Hindi press releases from their website. The website contains all the press releases in the last five years starting from June 2017.

Indian National Congress speeches (INC): The official website of the INC stores some of the speeches by major INC political leaders. We have extracted 112 of these speeches from their official website³. The timeline for the scraped speeches is between August 2018 to March 2022.

Man-ki-baat (MKB): Man-ki-baat is a radio program hosted by the Indian prime minister Narendra Modi where he periodically addresses the people of the nation. The MKB website⁴ stores the official transcripts in Hindi and English languages. We have extracted the transcripts of 67 of these programs between December 2015 to December 2021.

Press Information Bureau (PIB): The Press Information Bureau houses the official press releases from all Indian government ministries including President’s office, the Prime Minister’s office, Election Commission, etc. We have extracted 30283 articles from the PIB website⁵. The timeline for these articles is from June 2017 to March 2022.

PM speech (PMS): Majority of the Indian Prime Minister speeches (different from MKB speeches) are stored digitally on the PM India website⁶. We have extracted 694 of these speeches that are recorded between November 2016 to October 2021.

3.2 Hindi news articles

Here, we scrape data from two major Hindi news daily websites. Collectively, we denote this data source as D_{news} .

Dainik Bhaskar (DB): Dainik Bhaskar is one of the most popular Hindi newspapers in India. It is ranked 4th in the world by circulation according to

²<https://aamaadmiparty.org/media/press-releases>

³<https://www.inc.in/media/speeches>

⁴<https://www.pmindia.gov.in/hi/mann-ki-baat/>

⁵<https://www.pib.gov.in>

⁶<https://www.pmindia.gov.in/hi/news-updates/>

World Press Trends 2016⁷. They have digitized the daily newspapers on their website⁸. Articles on DB website have been divided into many categories such as ‘Entertainment’ and ‘Sports’. We have extracted 115324 articles uploaded on the website between February 2019 to May 2022. In Table 2, we present the category-wise distribution of the articles scraped from the DB website.

Category	DB	DJ
Business	16012	4203
Entertainment	18498	52173
Featured	5536	19373
Lifestyle	12189	-
Miscellaneous	20221	-
National	18615	160005
Politics	-	33604
Sports	9950	-
World	14303	42478
Total	115324	311836

Table 2: Number of articles in various news categories in the DB and DJ datasets.

Dainik Jagran (DJ): Dainik Jagran is another popular Indian Hindi newspaper. According to World Press Trends 2016, DJ is ranked 5th in the world by circulation. Similar to the DB website, they have also created a repository of articles on their official website⁹. Here, we extract 311836 of these articles from the website that were uploaded between April 2013 to May 2022. In Table 2, we present the category-wise distribution of the articles scraped from the DJ website.

4 Experimental Setup

Problem definition: Given a multilingual article A comprising of q multi-sentential text spans (MST) i.e. $A = \{M_1, M_2, \dots, M_q\}$, we predict a binary outcome L_{CM} for each MST M_i i.e. $L(A) = \{L_{CM}^{M_1}, L_{CM}^{M_2}, \dots, L_{CM}^{M_q}\}$. $L_{CM}^{M_i} = 1$, if M_i is code-mixed, otherwise 0. In a nutshell, a code-mixed MST M_i is a MCT and it satisfies the properties $P1$ and $P2$ (ref. §2).

Figure 2 shows the architecture of the MCT identification pipeline. Next, we discuss the various components of this pipeline in detail.

4.1 Token-level language annotation (TLA)

We exploit the token-level language information to identify MCT given a multilingual article A .

⁷<https://web.archive.org/web/20170706110804/http://www.wptdatabase.org/world-press-trends-2016-facts-and-figures>

⁸<https://www.bhaskar.com>

⁹<https://www.jagran.com>

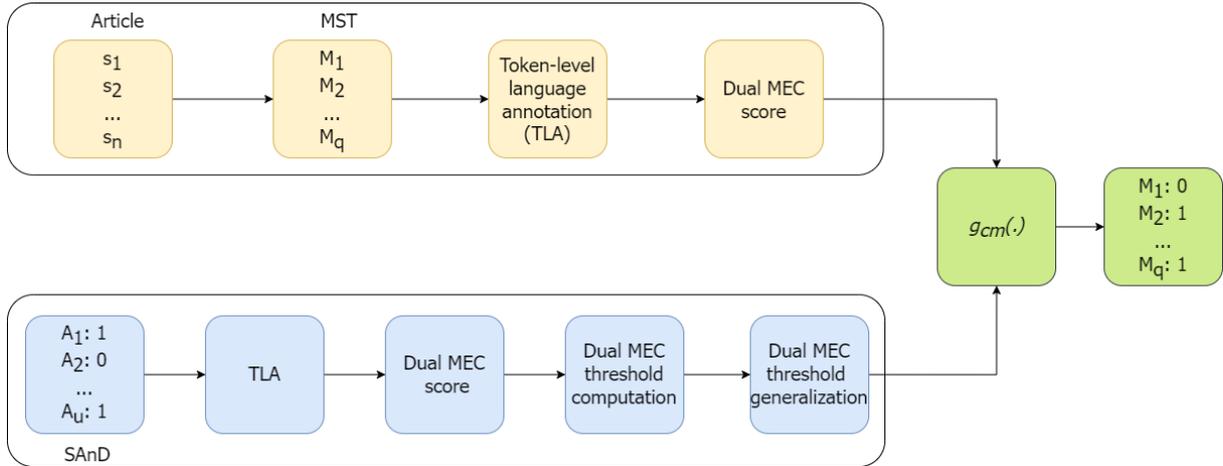


Figure 2: Architecture of MCT identification pipeline.

We annotate the words in A using a code-mixed language identification tool. Specifically, we use L3Cube-HingLID (Nayak and Joshi, 2022) for this task. A word $w_i \in A$ can take either of the three language tags from the set $\{English, Hindi, Other\}$. Given that L3Cube-HingLID works only on the Roman script text, we use a Devanagari to Roman script transliteration tool¹⁰ for the tokens written in Devanagari script. In Table 3, we report the percentage of *Hindi* and *English* tokens. With an exception of the AAP dataset, *Hindi* is the predominant language in all the data sources.

	Articles	AW	AC	%H	%E
AAP	320	1129	6033	53.97	45.09
INC	112	2312	10691	63.83	33.12
MKB	67	4151	20706	77.17	22.41
PIB	30283	525	3015	80.96	17.59
PMS	694	2591	13400	79.02	20.45
DB	115324	382	1977	80.22	18.25
DJ	311836	391	2037	79.28	19.60
D_{speech}	31476	590	3339	79.97	18.65
D_{news}	427160	388	2020	80.18	18.51
$D_{speech} + D_{news}$	458636	401	589	80.05	18.54

Table 3: Distribution of the scraped articles from various data sources. AW: average number of words. AC: average number of characters. %E: percentage of English tokens. %H: percentage of Hindi tokens.

4.2 Code-Mixing Index (CMI)

In the literature, we observe several metrics that has been proposed to measure the degree of code-mixing in text such as code-mixing index (CMI,

¹⁰<https://github.com/ritwikmishra/devanagari-to-roman-script-transliteration>

(Das and Gambäck, 2014)), multilingual-index (M-index, (Barnett et al., 2000)) and integration-index (I-index, (Guzmán et al., 2017)). Each of these metrics has its own merits and limitations (Srivastava and Singh, 2021a). In this work, we use the most widely used CMI metric due to the ease of interpretation and the suitability for the task. CMI, by definition, measures the degree of code-mixing in a text as:

$$CMI = \begin{cases} 100 * [1 - \frac{\max\{w_i\}}{n-u}] & n > u \\ 0 & n = u \end{cases} \quad (1)$$

Here, w_i is the number of words of the language i , $\max\{w_i\}$ represents the number of words of the most prominent language, n is the total number of tokens, u represents the number of language-independent tokens (such as named entities, abbreviations, mentions, and hashtags). The CMI score ranges from 0 to 100. A low CMI score suggests the prevalence of only one language in the text whereas a high CMI score indicates a high degree of code-mixing.

4.3 Small annotated dataset (SAnD)

We create a small manually annotated dataset comprising all seven data sources. The objective of the annotation is to assign a binary label to each MST such that we can identify if the MST is code-mixed or not from the assigned label.

More formally, $SAnD = \{A_1: l_1, A_2: l_2, \dots, A_u: l_u\}$, represents u manually annotated MST¹¹ where $l_i \in \{0, 1\} \forall i \in [1, u]$. Here, $l_i = 1$, if A_i is code-mixed, otherwise 0.

¹¹For distinctive representation, we denote MST in $SAnD$ with A instead of M .

	Articles		MST	
	Total	Hing	E/H	
AAP	5	6	2	4
INC	3	69	5	64
MKB	3	66	25	41
PIB	47	62	27	35
PMS	2	36	13	23
DB	30	207	48	159
DJ	30	122	28	94
D_{speech}	60	239	72	167
D_{news}	60	329	76	253
$D_{speech} + D_{news}$	120	568	148	420

Table 4: *SAnD* dataset statistics. Hing: Hinglish, E/H: English/Hindi.

For this annotation task, we have selected a small number of articles (60 each from D_{speech} and D_{news}) randomly from the scraped articles. We leave it to the judgment of the annotator to decide if a sentence (and subsequently the MST) is code-mixed or not. The annotator has expert-level proficiency in Hindi, English, and Hinglish languages. In Table 4, we show the distribution of the annotated articles for each data source. In total, we annotate 120 articles and 568 MST where we identify 121 MST (21.3%) as code-mixed.

4.4 Estimating multilinguality

Though CMI is widely used in numerous previous works, we couldn't find any discussion on the ideal CMI score thresholding criteria to identify a good code-mixed text. The problem becomes even more challenging when we use the CMI metric in a multi-sentential framework along with constraints $P1$ and $P2$ (ref §2). Various works (Khanuja et al., 2020) have used empirically identified CMI thresholds to measure the degree of code-mixing in the text. But, we couldn't find any experimental justification for their findings.

Dual MEC score: Here, we propose a novel adoption of the CMI metric in a constrained multi-sentential framework. For MST M_p with k sentences, we compute the scores for dual multilinguality estimation criteria (MEC) as:

1. Sentence-level CMI (CMI): We compute $CMI(s_i)$ for the sentence $s_i \in M_p$ using the language-information of all the words in s_i and the formulation given in 1.
2. Multilinguality ratio (MR): We compute C_{MR} for the MST M_p as:

$$MR(M_p) = \frac{N_{cm}}{k} \quad (2)$$

Here, N_{cm} and k are the number of code-mixed and total sentences in M_p respectively.

Figure 3 shows the mean and standard deviation of dual MEC scores on seven different data sources.

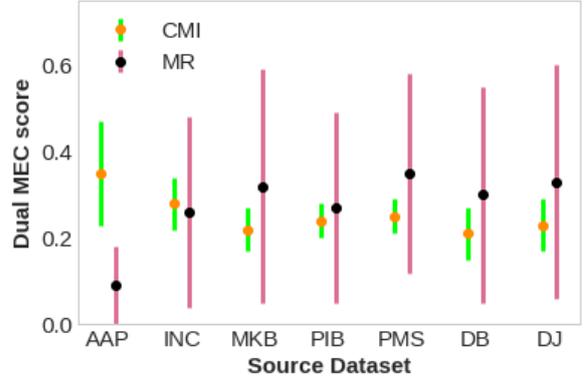


Figure 3: The mean and standard deviation of the dual MEC score for different data sources. The CMI score is scaled between 0 to 1.

Formulation: We identify if the sentence s_i is code-mixed or monolingual using $CMI(s_i)$ score as:

$$f_{cm}(s_i) = \begin{cases} 1, & CMI(s_i) > \alpha \\ 0, & otherwise \end{cases} \quad (3)$$

Here, $\alpha \in [0, 100]$ is the sentence-level CMI score threshold and $f_{cm}(\cdot)$ estimates the code-mixing status (1 being code-mixed and 0 being monolingual) of the sentence under consideration. Using 3, we compute N_{cm} as:

$$N_{cm} = \sum_{i=1}^k f_{cm}(s_i) \quad (4)$$

Using 2 and 4, we compute $MR(M_p)$ as:

$$MR(M_p) = \frac{\sum_{i=1}^k f_{cm}(s_i)}{k} \quad (5)$$

We formulate the following function to identify if MST M_p with k sentences is code-mixed:

$$g_{cm}(M_p) = \begin{cases} 1, & MR(M_p) > \beta \\ 0, & otherwise \end{cases} \quad (6)$$

Here, $\beta \in [0, 1]$ is the multilinguality ratio threshold and $g_{cm}(\cdot)$ estimates the code-mixing status (1 being code-mixed and 0 being monolingual) of the MST under consideration.

4.5 Dual MEC threshold computation

The dual MEC formulation helps us to identify the MCT in a constrained setting by jointly modeling the sentence-level and MST-level multilinguality information. As discussed in Section 4.4, the ideal thresholds α and β are a conundrum that needs further exploration. Here, we propose to use the *SAnD* dataset to identify the dual MEC thresholds (α and β). Algorithm 1 shows the procedure to compute the thresholds. The algorithm takes *SAnD* dataset D with u labeled MST. We represent the parameter search space for α and β with α_{cand} and β_{cand} respectively. α_{cand} ranges from α_{low} to α_{high} with a step-size of α_{step} whereas β_{cand} ranges from β_{low} to β_{high} with a step-size of β_{step} . Based on our empirical observation, we set $(\alpha_{low}, \alpha_{high}, \alpha_{step})$ with $(0, 50, 1)$ and $(\beta_{low}, \beta_{high}, \beta_{step})$ with $(0, 0.5, 0.025)$.

We perform the grid search on each threshold combination of (α_i, β_j) to identify the best combination. For each threshold combination, we identify the accuracy of identifying the MCT in D leveraging $f_{cm}(\cdot)$ and $g_{cm}(\cdot)$ formulations. We select the threshold combination with the highest accuracy as the final threshold (α and β). Table 5 shows the best-identified thresholds on various data sources of the *SAnD* dataset. Figure 4 shows the mean and standard deviation of the accuracy on various dual MEC threshold combinations for different data sources.

Algorithm 1 $compute_{\alpha,\beta}(D)$

Require: $D = \{A_1: l_1, A_2: l_2, \dots, A_u: l_u\}$ where $A_i = \{s_1, s_2, \dots, s_k\}$
Require: $\alpha_{cand} = [\alpha_{low}, \alpha_{low} + \alpha_{step}, \dots, \alpha_{high}]$
Require: $\beta_{cand} = [\beta_{low}, \beta_{low} + \beta_{step}, \dots, \beta_{high}]$
Require: $Accuracy = \{\}$
1: **for** α_i in α_{cand} **do**
2: **for** β_j in β_{cand} **do**
3: $hits = 0$
4: **for** $A_p \in D$ **do**
5: $F_{cm} = f_{cm}(s_q) \forall s_q \in A_p$
6: Compute $g_{cm}(A_p)$ using F_{cm}
7: **if** $g_{cm}(A_p) == l_p$ **then**
8: $hits = hits + 1$
9: **end if**
10: **end for**
11: $Accuracy[(\alpha_i, \beta_j)] = 100 * (hits/u)$
12: **end for**
13: **end for**
14: $\alpha = \max_{value}(Accuracy).key()[0]$
15: $\beta = \max_{value}(Accuracy).key()[1]$
16: **return** α, β

	α	β	Accuracy(%)
AAP	25	0.35	100
INC	28	0.30	89
MKB	22	0.35	64
PIB	26	0.15	68
PMS	21	0.45	89
DB	18	0.40	72
DJ	28	0.40	79
D_{speech}	24	0.35	72
D_{news}	29	0.475	78
$D_{speech} + D_{news}$	29	0.45	75

Table 5: Best identified thresholds (α and β) along with the accuracy of identifying MCT on various data sources in the *SAnD* dataset.

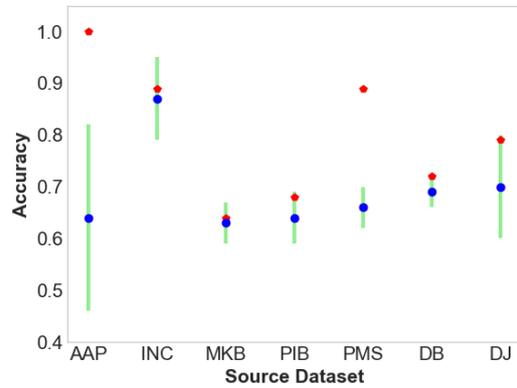


Figure 4: The mean and standard deviation of the accuracy on various dual MEC threshold combinations. The red dot corresponding to each data source indicates the accuracy against the best-identified thresholds.

4.6 Dual MEC threshold generalization

As evident from Table 5, the thresholds α and β vary across the data sources. So, it is important to identify which of these identified thresholds will result in a robust and stable performance across datasets. Here, we experiment with five dual MEC threshold generalisation techniques:

1. **Local Average (LA):** For the data source D_i , we take the mean sentence-level CMI score and mean MR score as the dual MEC thresholds.
2. **Global Average (GA):** For the data source D_i , we take the mean sentence-level CMI score and mean MR score of the corresponding category data-source (D_{speech} or D_{news}) as the dual MEC thresholds.
3. **Average of LA and GA (ALG):** For the data source D_i , we take the average of LA and GA identified thresholds as the dual MEC thresholds.
4. **Single data source generalization (SDG):** In

this approach, we generalize the dual MEC thresholds identified locally on a single data source D_i (using Algorithm 1) to identify MCT globally on other data sources.

- Multi data source generalization (MDG):** In this approach, we use the dual MEC threshold information from multiple sources and use the majority voting to identify the best thresholds. For the data source D_i , we use the thresholds identified on three data sources (using Algorithm 1), namely D_i , D_{speech} (if $D_i \in D_{speech}$, else $\check{a}D_{news}$), and $D_{speech} + D_{news}$. We then make an independent prediction on each of the three thresholds and take majority voting for the final classification of M_p .

5 MUTANT: A Multi-sentential Code-mixed Hinglish Dataset

We evaluate the performance of MCT identification pipeline and the five dual MEC threshold generalization techniques using the three subsets of the *SAnD* dataset: D_{speech} , D_{news} , and $D_{speech} + D_{news}$. We report the following metric scores on each of the seven data sources:

- Accuracy:** We compute accuracy as the ratio of the total correct prediction of MCT and non-MCT to the total number of MST. We multiply this ratio by 100 and report the accuracy percentage. A high accuracy % is preferred.
- False MCT Rate (FMR):** We define FMR as the ratio of incorrectly identified MCT to the total number of actual monolingual MST. We report the FMR% and a low FMR% is preferred.
- Diversity@10 (D@10):** We define D@10 as the percentage of articles in data source D_i having more than 10% correctly identified MCT. A high D@10 score is preferred.

We report the results in Tables 6, 7, 8. The mean-based threshold generalization techniques (LA, GA, and ALG) consistently show poor performance on all the metrics. Given the nature of the problem, we prefer a low rate of misidentification of mono-

	Accuracy					FMR					D@10				
	L	G	A	S	M	L	G	A	S	M	L	G	A	S	M
AAP	62	66	64	72	74	15	21	20	17	17	49	46	51	60	62
INC	63	66	64	73	74	17	21	20	16	12	49	46	51	59	59
MKB	61	66	62	69	72	28	21	26	22	18	51	46	48	68	70
PIB	62	66	64	67	72	24	21	24	30	17	53	46	55	73	74
PMS	67	66	64	71	74	17	21	23	20	16	51	46	53	67	69
DB	66	63	62	67	78	29	26	28	30	5	57	56	57	78	78
DJ	62	63	64	75	78	26	26	26	6	5	48	56	49	73	74

Table 6: Results on D_{speech} dataset. L: LA, G: GA, A: ALG, S: SDG, M: MDG.

	Accuracy					FMR					D@10				
	L	G	A	S	M	L	G	A	S	M	L	G	A	S	M
AAP	72	70	71	72	73	17	15	17	14	14	60	58	62	70	72
INC	69	70	71	73	73	14	15	15	9	7	58	58	58	65	66
MKB	66	70	68	70	72	25	15	21	21	15	73	58	71	79	80
PIB	68	70	68	70	73	23	15	22	29	14	73	58	71	79	80
PMS	61	70	69	74	73	14	15	18	14	12	63	58	63	71	69
DB	66	69	67	68	71	28	22	26	29	3	76	72	74	84	85
DJ	68	69	68	72	71	22	22	22	4	3	70	72	68	77	73

Table 7: Results on D_{news} dataset. L: LA, G: GA, A: ALG, S: SDG, M: MDG.

	Accuracy					FMR					D@10				
	L	G	A	S	M	L	G	A	S	M	L	G	A	S	M
AAP	69	70	69	73	74	12	15	15	13	13	55	60	57	65	66
INC	70	70	69	73	74	11	15	14	10	8	57	60	56	62	63
MKB	67	70	69	70	72	21	15	19	17	14	62	60	65	68	65
PIB	69	70	69	67	73	18	15	18	23	14	63	60	64	75	74
PMS	62	70	70	72	74	13	15	17	16	12	57	60	59	65	69
DB	67	68	67	67	75	23	19	22	24	4	64	62	62	76	75
DJ	68	68	69	74	75	19	19	19	5	4	57	62	62	71	74

Table 8: Results on $D_{speech}+D_{news}$ dataset. L: LA, G: GA, A: ALG, S: SDG, M: MDG.

lingual MST as the MCT and at the same time a high number of actual MCT should also be identified. MDG threshold generalization technique satisfies both conditions with low FMR and high accuracy on all the datasets. D@10 depicts if the threshold generalization technique is influenced by the presence of a few outliers in the dataset. SDG and MDG both show competitive results on the D@10 metric outperforming the mean-based threshold generalization techniques by a large margin. The constant poor performance of mean-based threshold generalization against SDG and MDG also shows the efficacy of the proposed threshold computation strategy (Algorithm 1).

Finally, to build the MUTANT dataset, we use the MCT identification pipeline with the MDG threshold generalization technique. Table 9 shows the statistics of the MUTANT dataset. To facilitate future work on this novel task of MCT identification, we will release the MUTANT dataset along with the initially scraped data from all the data sources and the annotated *SAnD* dataset. The MUTANT dataset can be used for various tasks including but not limited to question-answering, text summarization and machine translation for Hinglish texts. This dataset could be used as a pre-training dataset to train efficient NLU models for various tasks on Hinglish data.

6 Analysis and Discussion

In this section, we qualitatively evaluate the *MUTANT* dataset by employing two human evaluators, different from the one used for the *SAnD* to avoid any biases in the evaluation. Both evalua-

	A	M	M/A	Avg CMI			Avg Words			Avg Characters		
				A	M	H	A	M	H	A	M	H
AAP	30	32	1.07	33.0	35.2	21.1	1347	1263	16	6993	6556	63
INC	85	306	3.6	28.1	27.5	-	751	208	-	3368	935	-
MKB	58	243	4.19	20.1	22.4	-	1034	246	-	4843	1156	-
PIB	8473	8786	1.04	23.0	23.2	21.0	572	552	15	3139	3028	87
PMS	597	3909	6.55	25.8	24.7	26.4	952	145	13	4585	700	79
DB	12851	15433	1.20	21.0	21.2	20.2	107	89	24	528	440	123
DJ	44913	56228	1.25	22.2	22.3	21.6	146	117	16	734	586	82
D_{speech}	9243	13276	1.44	23.2	23.8	21.3	604	420	15	3258	2268	87
D_{news}	57764	71661	1.24	21.9	22.0	21.2	137	111	18	688	555	91
$D_{speech} + D_{news}$	67007	84937	1.27	22.0	22.3	21.2	201	159	17	1043	822	90

Table 9: MUTANT dataset statistics. A: Articles, M: MCT, and H: Headings. The INC and MKB datasets contain generic and very-low informative headlines and we do not include them in the final dataset.

	A	MST	CA		CKS	Acc	FMR	D@10
			Hing	E/H				
AAP	5	5	2	3	1.0	100	0	100
INC	5	82	10	67	0.76	88	10	80
MKB	5	119	23	80	0.67	75	25	80
PIB	5	5	2	3	1.0	80	0	50
PMS	5	141	13	110	0.52	84	12	100
DB	5	49	3	43	0.63	78	20	50
DJ	5	18	2	15	0.77	88	13	100
D_{speech}	25	352	50	263	0.65	82	14	71
D_{news}	10	67	5	58	0.69	80	18	75
$D_{speech} + D_{news}$	35	419	55	321	0.65	82	15	74

Table 10: Qualitative evaluation of the MUTANT dataset. A: Articles, CA: complete agreement between the annotators, Hing: Hinglish MST. E/H: English/Hindi MST, CKS: Cohen’s kappa score.

tors are proficient in English, Hindi, and Hinglish languages. We randomly sample five articles from each of the seven source datasets and share the originally scraped articles containing both identified MCT and monolingual MST with both evaluators. During the evaluation, we do not disclose which of the MSTs is identified as MCT and share the following guidelines:

1. Any MST containing only Hindi words or only English words is monolingual.
2. Any named entity, date, number, or word common in both English and Hindi languages should be considered a language-independent word.

In Table 10, we report our findings from the qualitative evaluation study. Out of a total of 419 MST, we observe the complete agreement on 321 monolingual MST and 55 code-mixed MST resulting in $\approx 90\%$ complete agreement. A complete agreement means that both annotators agree that any particular MST is code-mixed or not. On MST with CA, we further compute the three metric scores using MDG. The results strengthen our earlier findings from Section 5. In Figure 5, we report two example MCT incorrectly identified by our MCT identifica-

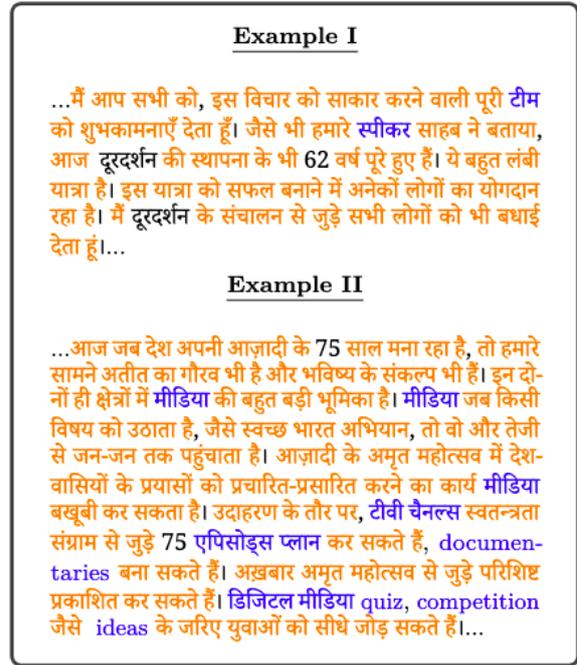


Figure 5: False positive MCT. We color code the tokens as: Hindi, English, and language independent.

tion pipeline. In the first example, both evaluators show complete agreement whereas in the second example there is a disagreement between the evaluators. We attribute this behavior to the poor state of the current code-mixed LID systems (Srivastava and Singh, 2021a) and since the CMI metric and our dual MEC formulation depend heavily on the code-mixed LID tools, the final results get affected. This limitation further provides an opportunity for future works to explore the problem from different perspectives such as a token-level language-independent MCT identification pipeline. It will also be interesting to see how this pipeline performs with other code-mixed languages, especially in a low-resource setting.

7 Conclusion

In this paper, we present a novel task of identifying MCT from multilingual documents. We propose an MCT identification pipeline by extending CMI to the multi-sentential framework and leveraging the pipeline we build a dataset for the Hinglish language. We highlight several challenges in building such resources and our insights will be useful to future works in code-mixed and low-resource languages.

8 Limitations

The limitations with the *MUTANT* dataset include but are not limited to:

- Contrary to the previous works, all the data sources comprises the non social media sites. This could potentially limit the diversity in the code-mixed text as observed on social media platforms.
- In the current form, the dataset is limited to only one code-mixed language. We believe the proposed technique to extract MCT could be expanded to other code-mixed languages in the future.
- The data sources could potentially have their own biases (topical, style of writing, etc). We expect future works to be cautious while generalizing the results obtained on this dataset.

References

- Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284.
- Guangsheng Bao, Yue Zhang, Zhiyang Teng, Boxing Chen, and Weihua Luo. 2021. G-transformer for document-level machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3442–3455.
- Ruthanna Barnett, Eva Codó, Eva Eppler, Montse Forcadell, Penelope Gardner-Chloros, Roeland Van Hout, Melissa Moyer, Maria Carme Torras, Maria Teresa Turell, Mark Sebba, et al. 2000. The lides coding manual: A document for preparing and analyzing language interaction data version 1.1–july 1999. *International Journal of Bilingualism*, 4(2):131–271.
- Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S Weld. 2020. Tldr: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777.
- Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387.
- Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsun corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79.
- Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. *Proc. Interspeech 2017*, pages 67–71.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.
- Simran Khanuja, Sandipan Dandapat, Sunayana Sitaram, and Monojit Choudhury. 2020. A new dataset for natural language inference from code-mixed conversations. In *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 9–16.
- Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhat-tacharyya. 2018. The iit bombay english-hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Laiba Mehnaz, Debanjan Mahata, Rakesh Gosangi, Uma Sushmitha Gunturi, Riya Jain, Gauri Gupta, Amardeep Kumar, Isabelle G Lee, Anish Acharya, and Rajiv Shah. 2021. Gupshup: Summarizing open-domain code-switched conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6177–6192.
- Ravindra Nayak and Raviraj Joshi. 2022. L3cubehingcorpus and hingbert: A code mixed hindi-english dataset and bert language models. *arXiv preprint arXiv:2204.08398*.
- Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213.
- Vivek Srivastava and Mayank Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49.

- Vivek Srivastava and Mayank Singh. 2021a. Challenges and limitations with the metrics measuring the complexity of code-mixed text. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 6–14.
- Vivek Srivastava and Mayank Singh. 2021b. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208.
- Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4631–4640. IEEE.

Bridging the Gap between Native Text and Translated Text through Adversarial Learning: A Case Study on Cross-Lingual Event Extraction

Pengfei Yu¹, Jonathan May², Heng Ji¹

¹University of Illinois Urbana-Champaign ²University of South California

{pengfei4, hengji}@illinois.edu

jonmay@isi.edu

Abstract

Recent research in cross-lingual learning has found that combining large-scale pretrained multilingual language models with machine translation can yield good performance (Phang et al., 2020; Fang et al., 2021). We explore this idea for cross-lingual event extraction with a new model architecture that jointly encodes a source language input sentence with its translation to the target language during training, and takes a target language sentence with its translation back to the source language as input during evaluation. However, we observe significant representational gap between the native texts and translated texts, both in the source language and the target language. This representational gap undermines the effectiveness of cross-lingual transfer learning for event extraction with machine-translated data. In order to mitigate this problem, we propose an adversarial training framework that encourages the language model to produce more similar representations for the translated text and the native text. To be specific, we train the language model such that its hidden representations are able to fool a jointly trained discriminator that distinguishes translated texts’ representations from native texts’ representations. We conduct experiments on cross-lingual event extraction across three languages. Results demonstrate that our proposed adversarial training can effectively incorporate machine translation to improve event extraction, while simply adding machine-translated data yields unstable performance due to the representational gap.¹

1 Introduction

There are over 6,000 living languages in the world, and for many of them, too little appropriate data exists to build natural language processing (NLP) models. Cross-lingual learning has been proposed to leverage resources in data-rich languages to train NLP models for data-scarce languages (Ruder

et al., 2019). There are two main strategies for building cross-lingual models: (1) train models with multilingual language models and language-universal features that are transferable to the target language (Huang et al., 2019; Hsu et al., 2019; Hu et al., 2020a; Luo et al., 2020; Wei et al., 2021; Ouyang et al., 2021; Liu et al., 2019; Subburathinam et al., 2019; M’hamdi et al., 2019; Ahmad et al., 2021); (2) use machine translation models in a pipeline, either by transforming annotated training data into the desired target language to build target-language models, or by translating data at inference time into the source language and applying source-language models (Cui et al., 2019; Hu et al., 2020a; Yarmohammadi et al., 2021). The first approach relies on the quality of the constructed multilingual semantic space; the discrepancy between source-language training data and target-language evaluation data may cause overfitting. The second approach does not require a perfect multilingual semantic space since models can be trained in a monolingual fashion, but it depends on the quality of machine translation.

A combination of both approaches showed good performance on a variety of tasks such as natural language inference and question answering (Phang et al., 2020; Fang et al., 2021), but is underexplored for event extraction. Compared with previous research in cross-lingual event extraction mainly adopting the first approach (Liu et al., 2019; Subburathinam et al., 2019; M’hamdi et al., 2019; Ahmad et al., 2021), we explore the idea of combining both machine translations and language-universal representations for cross-lingual event extraction in this work. We perform translation by extending the previous effort on cross-lingual reading comprehension (Hsu et al., 2019) and question answering (Hu et al., 2020a) by adding special tags around the trigger and entity spans to translate the annotations. We use a multilingual language model to simultaneously encode a sentence and its corresponding

¹Code at <https://github.com/Perfec-Yu/CrossIE>

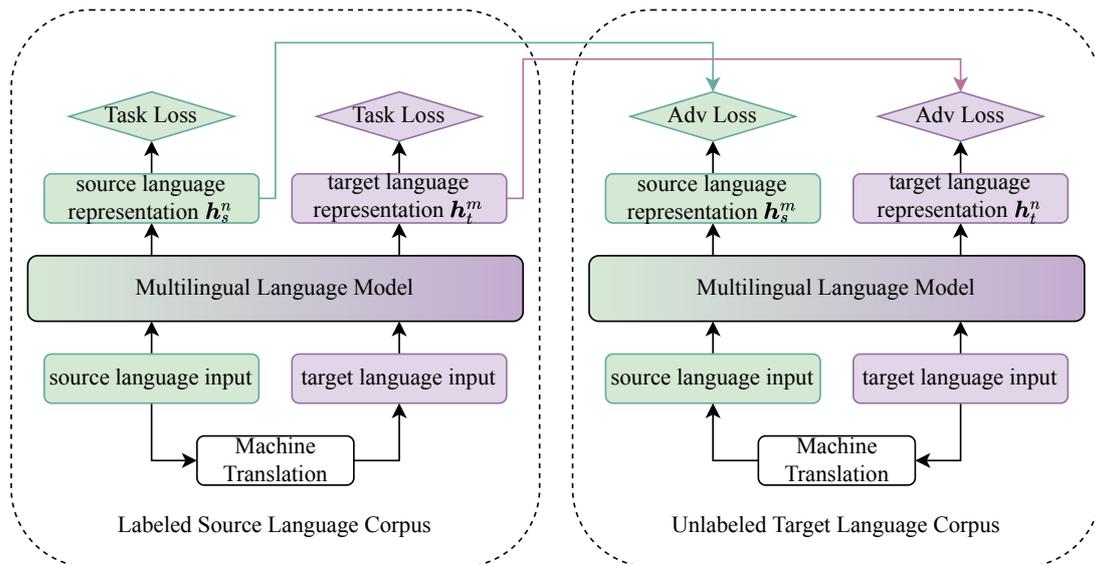


Figure 1: Overall cross-lingual information extraction framework

translation as shown on the left side of Figure 1. For example, in an English-to-Chinese cross-lingual learning setting, we would train a model with English sentences with their Chinese translations as training data, and evaluate our model with Chinese sentences and their English translations as inputs. Since our work includes both cross-lingual learning and machine translation, to avoid ambiguity, we will use “source” language as the one we perform cross-lingual learning from, and “target” language as the one we perform cross-lingual learning to. We will call texts before translation “native” text and text after translation “translated” text for the machine-translation-related descriptions.

We found that one challenge in cross-lingual event learning with machine translations is that the machine-translated text $\mathcal{M}_{\mathcal{K} \rightarrow \mathcal{L}}$ from one language \mathcal{K} into another language \mathcal{L} may be different from the native text in the target language $\mathcal{N}_{\mathcal{L}}$. This difference is also introduced and studied as the problem of “translationese” (translated text as a different language) in previous machine translation research (Pylypenko et al., 2021; Riley et al., 2020). In cross-lingual event extraction, we observe from a simple preliminary experiment that there indeed exists a distinguishable gap between representations of native texts $H(\mathcal{N}_{\mathcal{L}})$ and translated text $H(\mathcal{M}_{\mathcal{K} \rightarrow \mathcal{L}})$ in some multilingual language model H . The pretrained language models appear to be “unaccustomed” to the translated text. The representational gap will negatively impact the cross-lingual learning with machine-translated data. Since we, as introduced above, simultaneously encode a native

source language sentence $\mathcal{N}_{\mathcal{S}}$ and its translation into the target $\mathcal{M}_{\mathcal{S} \rightarrow \mathcal{T}}$ language during training, and a native target language sentence $\mathcal{N}_{\mathcal{T}}$ and its translation back to the source language $\mathcal{M}_{\mathcal{T} \rightarrow \mathcal{S}}$ during evaluation, the problem of representational gap between $\mathcal{N}_{\mathcal{S}}$ and $\mathcal{M}_{\mathcal{T} \rightarrow \mathcal{S}}$, as well as $\mathcal{N}_{\mathcal{T}}$ and $\mathcal{M}_{\mathcal{S} \rightarrow \mathcal{T}}$ need to be resolved. Here \mathcal{S} and \mathcal{T} refer to the source and the target language respectively.

In order to mitigate the representational gap problem between machine-translated text \mathcal{M} and native text \mathcal{N} in both source and target languages, we propose to take advantage of an unlabeled corpus in the target language and use adversarial training to make the encoder produce more similar representations for $\mathcal{N}_{\mathcal{S}}$ and $\mathcal{M}_{\mathcal{T} \rightarrow \mathcal{S}}$, as well as $\mathcal{N}_{\mathcal{T}}$ and $\mathcal{M}_{\mathcal{S} \rightarrow \mathcal{T}}$. The adversarial framework trains the language model H such that its hidden representations can fool a jointly trained discriminator that distinguishes translated texts’ representations $H(\mathcal{M})$ from native texts’ representations $H(\mathcal{N})$. Our complete cross-lingual IE framework is shown in Figure 1, which combines translation-based methods with transfer-based methods, and uses an unlabeled target language corpus to improve the representations in multilingual language models. Our method shows superior performance on event trigger labeling and argument role labeling, and through quantitative studies, we observe that adversarial training indeed makes the multilingual language model generate closer representations for the translated text and the native text. We believe our proposed adversarial training can also be helpful in other NLP tasks where machine

translation can boost performance.

To summarize, our contributions are two-fold:

- We observe the gap between representations of the machine-translated text and the native text in multilingual language models.
- We propose an adversarial training method to close the representational gap, which improves event extraction performance.

2 Approach

In this section, we will start with a simple preliminary experiment to validate the problem of the representational gap, and then introduce our approaches to cross-lingual event trigger and argument role labeling. For both tasks, we first design specific methods to use machine translation models to translate source language annotations into the target language. We then use XLM-RoBERTa (Conneau et al., 2020) to encode pairs of parallel sentences simultaneously into hidden representations. Task-specific losses are used on top of the hidden representations. In order to make the multilingual language model produce more similar representations for translated sentences and native sentences, we further use an unlabeled target language corpus for adversarial training.

2.1 Preliminary Experiment on Representational Gap

We translate Chinese sentences from the ACE 2005 Chinese corpus into English and encode the translated English sentences $\mathcal{M}_{ZH \rightarrow EN}$ and native English sentences \mathcal{N}_{EN} in the ACE 2005 English data using the multilingual language model XLM-RoBERTa (Conneau et al., 2020). We then train linear Support Vector Machines (SVMs) (Cortes and Vapnik, 1995) to classify the encoded representations of these two sets of sentences as $\mathcal{N}_{\text{native}}$ or $\mathcal{M}_{\text{machine-translated}}$. The model achieves 83.4% accuracy on a held-out test set classifying the translated English sentences $\mathcal{M}_{ZH \rightarrow EN}$ and native English sentences \mathcal{N}_{EN} . We also perform translation from English to Chinese and achieve 93.4% accuracy classifying native Chinese sentences \mathcal{N}_{ZH} and translated Chinese sentences $\mathcal{M}_{EN \rightarrow ZH}$. Both numbers are significantly higher than the random 50% accuracy, indicating that the translated text and the native text are almost linearly separable in the multilingual language models and hence validating the representational gap between the two types of texts.

2.2 Event Trigger Labeling

In monolingual event trigger labeling, the input to the model is a sequence of text tokens $\{w_0, w_1, \dots, w_l\}$. The model identifies consecutive text spans as event triggers and classifies the spans into event types. We first obtain the token representations using the text encoder as $\{h_0, h_1, \dots, h_l\}$. Then we apply a linear layer to classify each token into one of the event types.

For the cross-lingual setting, we first translate the monolingual training data in the source language into the target language together with the trigger annotations. We will explain the translation process in Section 2.4. We encode the source language text sequence $\{w_{s0}, w_{s1}, \dots, w_{sl}\}$ and its translation $\{w_{t0}, w_{t1}, \dots, w_{tk}\}$ using the XLM-RoBERTa (Conneau et al., 2020) model. We also adopt a special fusion strategy as introduced in the FILTER (Fang et al., 2021), which adds cross-lingual attention between the source language text and its translation in some hidden Transformer layers. We apply the classification step as in the monolingual setting for both w_s and w_t . The task loss is the summation of losses from w_s and w_t .

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_t. \quad (1)$$

In the training phase described above, the input sequences to the multilingual language model consist of a native source language sequence w_s^n and its translations w_t^m . In the evaluation phase, the input sequence becomes a native target language sequence w_t^n and a translated source language sequence w_s^m . Therefore, we need to bridge the representational gap in the multilingual LM between two pairs: (w_s^n, w_s^m) and (w_t^m, w_t^n) . In order to encourage the multilingual LM to generate closer representations for w_s^n and w_s^m , as well as for w_t^m and w_t^n , we further propose an adversarial loss using another unlabeled target language corpus. We first translate the unlabeled target language corpus, from which we sample w_t^n , into the source language (w_s^m) to construct an unlabeled parallel corpus. Then parallel sentence pairs (w_s^m, w_t^n) in the unlabeled corpus are encoded by the multilingual LM in the same way as the labeled training sentence pairs (w_s^n, w_t^m) . We train two additional two-layer discriminators, D_s and D_t . D_s attempts to distinguish native source language representations w_s^n from translated source language representations w_s^m . D_t attempts to distinguish translated target language representations w_t^m from the native

	Trigger Labeling	Argument Role Labeling
Source Language	Now that Enron has ceased to exist, Bechtel and GE are suing the Indian Government for 5.6 billion US dollars.	The electricity that Enron produced was so exorbitant that the government decided it was cheaper not to buy electricity and <a>pay Enron the mandatory fixed charges specified in the contract.
Target Language	现在安然已经不复存在，柏克德和通用电气正在起诉印度政府，要求赔偿56 亿美元	安然生产的电力如此昂贵，以至于政府决定不购买电力并<a>支付安然合同中规定的强制性固定费用更便宜

Table 1: Example of training data translation for trigger labeling and argument role labeling.

target language representations w_t^n . The adversarial loss is also illustrated in Figure 1. For adversarial training, we adopt W-GAN (Arjovsky et al., 2017) with gradient penalty (Gulrajani et al., 2017) in this work. Specifically, D_s and D_t are two-layer neural networks with one output unit, i.e., they output single scalars. Optimization targets of the two discriminators are

$$\begin{aligned}\mathcal{L}_{D_s} &= D_s(\mathbf{h}_s^m) - D_s(\mathbf{h}_s^n; \theta) \\ &\quad + \text{GP}(D_s; \mathbf{h}_s^m, \mathbf{h}_s^n), \\ \mathcal{L}_{D_t} &= D_t(\mathbf{h}_t^m) - D_t(\mathbf{h}_t^n) \\ &\quad + \text{GP}(D_t; \mathbf{h}_s^m, \mathbf{h}_s^n).\end{aligned}\quad (2)$$

Here GP refers to the gradient penalty loss in Gulrajani et al. (2017) to regularize the discriminators. D_s and D_t are both neural networks that output a single value. We use $D_s(\mathbf{w}_s^m; \theta)$ to denote the average output value of all token representations in the sequence \mathbf{w}_s^m , and D_t in an analogous way. We expect our multilingual LM to produce representations that confuse both discriminators. The optimization target for the encoder is,

$$\begin{aligned}\mathcal{L}_G &= D_s(\mathbf{h}_s^n) - D_s(\mathbf{h}_s^m) \\ &\quad + D_t(\mathbf{h}_t^n) - D_t(\mathbf{h}_t^m).\end{aligned}\quad (3)$$

The gradients of the loss in Equation (1) are back propagated to both the multilingual language model and the trigger classification layers. The gradients of the discriminator loss in Equation (2) are back propagated to D_s and D_t only. The gradients of the generator loss in Equation (3) are back propagated to the multilingual language model. In practice we find that it is beneficial to back propagate \mathcal{L}_G to only the last layer of the XLM-RoBERTa to match the capacity of the discriminators D_s and D_t .

2.3 Argument Role Labeling

Argument Role Labeling identifies the roles entities play in events. Assuming gold-standard entity spans are provided, the input is a sentence x with a trigger span and an entity span, and the model predicts the argument role of the entity in the event. We use an additional None label for the case where the entity does not participate in the event.

For monolingual prediction, we first insert into the sentence two pairs of anchors to specify spans for the trigger and the entity: (“<a>”, “”) around the trigger span and (“”, “”) around the entity span. We encode the modified sentence into hidden representation x by a pretrained language model. We consider the token representation for the CLS token inserted into the beginning of every sentence x_{CLS} as the summarization of the sentence and feed it to a linear layer for classification. For adversarial training, we use a similar loss as in Equations (2) and (3), but use the CLS token representation x_{CLS} as the input to the discriminators.

2.4 Annotation Translation

We show two examples in Table 1 for translating annotations for trigger labeling and argument role labeling respectively. For trigger labeling, we first enclose each trigger span in the source language sentences with special tokens (“”, “”) inspired by previous efforts on question answering (Hu et al., 2020b). The machine translation model is applied to the new sentence. If the paired special tokens (“”, “”) exist in the translated sentence, we label the text span inside the pair as the event trigger. Otherwise we consider the translation as invalid and discard the target language loss \mathcal{L}_t in Equation (1) when training. We still use the invalid translations for the adversarial training loss

in Equation (2) and Equation (3) since the computation of these losses doesn't require trigger spans.

For argument role labeling, we take advantage of the anchor tokens used for training and simply translate the sentences with trigger and entity spans enclosed by anchor tokens into the target language. Due to the imperfections in the machine translation model, there are corrupted translated samples missing “<a>” or “” tags. However, since the role labeling model architecture doesn't require the existence of these tags to be runnable, we still consider them as valid inputs and use the corrupted translated samples as training data for both the target language loss \mathcal{L}_t in Equation (1) and the adversarial losses in Equation (2) and Equation (3).

2.5 Evaluation

At inference time, the inputs to the framework are sentences in the target language. We first translate the target language sentence into the source language using the same machine translation model used for the unlabeled target language corpus during training and apply our framework to the sentence pairs. We make predictions using the hidden representations of the target language.

3 Experiments

3.1 Dataset and Machine Translation

We use the ACE² 2005 dataset for experiments. We study all six transfer learning settings among the three languages in the dataset: Arabic, Chinese and English. We follow previous work on event extraction (Lin et al., 2020) to split the ACE dataset for the trigger labeling task. For the argument role labeling task, previous work (Subburathinam et al., 2019; Ahmad et al., 2021) has adopted a different split from Lin et al. (2020). We therefore follow the split in (Subburathinam et al., 2019; Ahmad et al., 2021) in this task. However, since their processed version of ACE dataset is not available, we use our own processed version and re-train their models on our version for comparison. We provide basic data statistics in Table 2. We also provide more fine-grained data statistics in Appendix. There are some other competitive cross-lingual event extraction baselines that we are not able to compare due to limited availability of code or split information. We provide further discussion in

²<https://www ldc.upenn.edu/collaborations/past-projects/ace>

the related work section. We use Google Translate for all machine translation components.

		Trigger		Role	
		#Docs	#Events	#Cands	#Args
EN	Train	529	4,419	14,036	7,018
	Dev	28	468	1,754	719
	Test	40	424	1,756	878
ZH	Train	551	2,926	11,826	5,931
	Dev	40	217	1482	602
	Test	42	190	1484	578
AR	Train	303	1,751	7,918	3,959
	Dev	50	255	990	495
	Test	50	262	990	495

Table 2: Data statistics for ACE 2005 dataset. EN, ZH and AR refer to the English, the Chinese and the Arabic splits respectively. The trigger labeling task (Trigger) and the argument role labeling task (Role) use different splits to compare with previous methods. We present the number of documents and the number of event mentions for Trigger splits. For Role splits, we present the number of candidate trigger-entity pairs for prediction (#Cands) and the total number of pairs that hold some argument role relationship (#Args).

3.2 Experiment Settings

Methods in Comparison We compare the following approaches in evaluation:

Direct, which directly trains a model on the source language with a multilingual language model and evaluates it on the target language. We use XLM-RoBERTa as the multilingual LM to be comparable with our method;

GATE (Ahmad et al., 2021) is a state-of-the-art cross-lingual model for the argument role labeling task. Hence we only compare with GATE in the argument role labeling task;

Trans is a baseline that excludes our proposed adversarial loss but keeps all the remaining components;

Trans+Adv is our proposed framework;

Target Supervision is a mono-lingual IE model trained on the target language data.

Evaluation Settings Except for **Target Supervision**, all cross-lingual models are trained with the source language annotations. We use the target language training corpus without annotations to compute the adversarial loss in our proposed method. We report F1 scores in the following sections and include precision and recall scores in Appendix.

Event Trigger Labeling	AR - EN	ZH - EN	AR - ZH	EN - ZH	ZH - AR	EN - AR
Direct	39.8	44.4	33.4	46.9	36.7	39.0
Trans	39.4	46.3	38.8	47.3	36.6	39.3
Trans+Adv (ours)	41.5	54.6	40.1	49.3	38.4	42.3
Target Supervision	68.5		65.6		56.1	

(a) Event trigger labeling.

Argument Role Labeling	AR - EN	ZH - EN	AR - ZH	EN - ZH	ZH - AR	EN - AR
GATE	50.3	57.0	55.7	63.6	65.1	65.0
Direct	56.8	61.5	64.6	71.7	64.0	62.5
Trans	57.5	60.6	64.9	71.3	63.8	62.2
Trans+Adv (ours)	58.4	62.9	65.6	72.0	68.0	65.1
Target Supervision	77.2		82.0		77.8	

(b) Argument role labeling.

Table 3: F1(%) scores for the cross-lingual event extractions. GATE (Ahmad et al., 2021) is a state-of-the-art method for cross-lingual argument role labeling. Direct, Trans and Target Supervision are introduced in Section 3.2. AR, EN and ZH correspond to Arabic, English and Chinese respectively.

3.3 Experiment Results

We show the evaluation results for trigger labeling in Table 3a. We show results for the argument role labeling task in Table 3b. Our model shows superior performance compared with other cross-lingual baselines in both trigger labeling and role labeling tasks and across all six cross-lingual transfer settings. Our model outperforms the Trans baseline that is trained without the adversarial loss. This indicates that our proposed approach effectively narrows the gap between the translations and the original natural language to improve the performance. Moreover, we notice that the Trans that uses translated data for training cannot consistently outperform the Direct baseline which doesn’t use translated data. This shows that the representational gap can have a negative impact on the model performance than the positive impact brought by including the translated data. In the following sections, we provide further analysis on the representational gap, our model’s improvements and remaining errors.

3.4 Effect of Adversarial Training

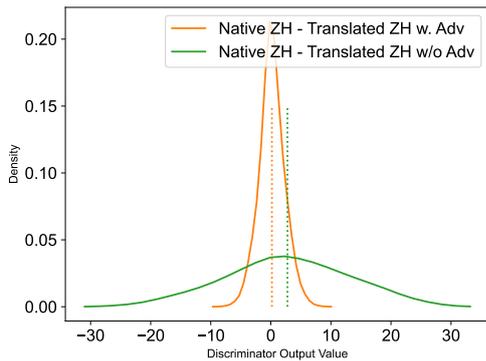
In this section we evaluate the effect of the adversarial training on reducing the representational gap. Hence we compare our model against the Trans baseline that doesn’t use the adversarial training loss. We take the English-to-Chinese transfer learning setting as a case study in this section.

Argument Role Labeling	EN-to-ZH		
	T-ZH	ZH	Diff
Trans	74.3	71.3	-3.0
Trans+Adv (ours)	74.5	72.0	-2.5

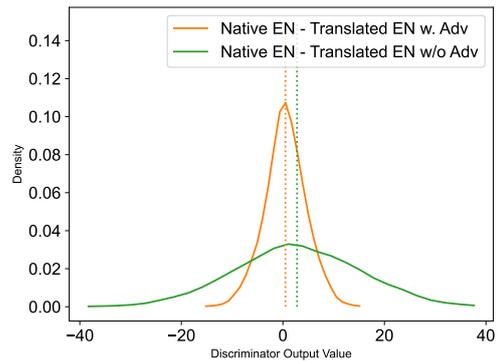
Table 4: F1 scores (in %) of the English-Chinese cross-lingual argument role labeling models on translated Chinese test corpus (from English test corpus), T-ZH and the native Chinese test corpus, ZH. Diff is the performance gap between two test corpora.

A straightforward way to examine the representational gap between the native text and the translated text inside a model is to compare its performance on these two types of texts on role labeling. In Table 4, we report the F1 scores on the native Chinese test set and translated Chinese text from English dataset respectively. The performance on translated Chinese is better than native Chinese since both models use the translated Chinese instead of native Chinese during training. Our adversarial training method shows a smaller performance gap compared with the non-adversarial baseline, indicating that our model indeed reduces the representational gap.

In addition to this evaluation, we further check whether the proposed generator loss helps the model to produce representations that confuse the discriminators. We compare the discriminator out-



(a) Native Chinese v.s. Translated Chinese



(b) Native English v.s. Translated English

Figure 2: Distribution of differences in discriminator outputs between native text and translated text. We compute the density with NumPy³ histogram function on original data points. *w. Adv* refers to our model with the adversarial training. *w/o Adv* is the output of the additional discriminators trained on the baseline *Trans* without adversarial training. (See Appendix for details on how the additional discriminators are trained)

Task	Sentence	Error
Trigger Labeling	...徐鹏航...支持参与亲属购买内部职工股... (...Penghang Xu... supported and participated in relatives' purchasing internal employee shares)	Baseline model makes a false positive prediction of “支持” (support) as a trigger for <i>Transfer-Money</i> event

Table 5: An Example error that the baseline approach fails but our proposed model succeeds.

puts for the native text representations and the translated text representations in Figure 2, for both English and Chinese. Since we use W-GAN (Arjovsky et al., 2017) for adversarial training, the discriminator output for an input sentence is a single scalar. For each language, we plot the distribution of difference in the output scalars $D_{s,t}(\mathbf{h}_{s,t}^n) - D(\mathbf{h}_{s,t}^m)$ between the native test corpus and the translated test corpus. These difference values are closer to 0 if the model fools the discriminators. For comparison we trained additional discriminators for the *Trans* baseline as the *w/o ADV* curves on the plot. The adversarial training makes the difference between the native text and the translated text much smaller for both English and Chinese.

Apart from the quantitative analysis, we show an example error from the baseline model that our proposed framework with adversarial training has managed to avoid in Table 5. The model makes the wrong prediction because in the English training data, “support”(支持) can trigger a *Transfer-Money* event with certain context which is uncommon in Chinese. By aligning the representation spaces with adversarial training, the model will align 支持 in translated text to represen-

tations of more common used Chinese words that trigger the *Transfer-Money* event.

3.5 Remaining Challenges

Chinese Sentences	Error
40年来, 日本皇室就没有再添男丁。(For 40 years, the Japanese royal family has not added any more males.)	Misses the trigger 添(add), Be-Born
德仁皇太子唯一的弟弟, 是[皇室]entity最后一名[出生]trigger的男性 (the only brother of Prince Naruhito was the last male [born]trigger in the [Royal Family]entity.)	False positive role prediction:Place.

Table 6: Remaining error examples of cross-lingual trigger and argument role labeling from our proposed model. We provide Chinese test sentences and English translations on the left and errors on the right.

Our experiments show cross-lingual trigger la-

being from English to Chinese is very challenging. In Table 6, the first two examples are from the trigger labeling task. In the first example, the Chinese trigger span has the meaning of “add,” which can only trigger a `Born` event under specific context such as “add children.” However, this is not a typical English expression, and it appears very rarely in the ACE 2005 English training data. Therefore cross-lingual learning fails on this case.

The second example is from the argument role labeling task. The model makes the wrong prediction because “室” in the entity span has the meaning of “room,” making the model to consider the entity as a location. Joint learning of entity typing and role labeling can be helpful for such cases.

4 Related Work

Multilingual Language Representations

Early work on multilingual representations learns aligned word or sentence embeddings from dictionaries (Mikolov et al., 2013; Faruqui and Dyer, 2014; Pan et al., 2019), parallel corpora (Gouws et al., 2015; Luong et al., 2015) or semi-supervised or unsupervised approaches (Artetxe et al., 2017; Zhang et al., 2017; Artetxe et al., 2018; Lample et al., 2018). Recent advances in pretrained language models have inspired research on cross-lingual language models such as mBERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019) and XLM-RoBERTa (Conneau et al., 2020).

Cross-Lingual Learning for NLP There is research in cross-lingual learning for many NLP tasks such as name tagging (Huang et al., 2019), reading comprehension (Cui et al., 2019; Hsu et al., 2019), summarization⁴ (Zhu et al., 2019; Cao et al., 2020). XGLUE (Liang et al., 2020), XTREME (Hu et al., 2020a) and XTREME-R (Ruder et al., 2021) present benchmarks covering a wide range of tasks including natural language inference, paraphrase detection, part-of-speech tagging, name tagging, question answering, sentence retrieval and generation, which are followed by (Phang et al., 2020; Fang et al., 2021; Luo et al., 2020; Wei et al., 2021; Ouyang et al., 2021). However these benchmarks don’t include event extraction as a subtask. For cross-lingual event extraction, early work utilizes multilingual embeddings and language universal parsing structures for cross-lingual transfer for trigger labeling (Liu et al., 2019) and argument role

labeling (Subburathinam et al., 2019). It is worth mentioning that Liu et al. (2019) focus on augmenting the existing supervision in the target language with cross-lingual learning that is different from the setting in this work, which requires no supervision in the target language. M’hamdi et al. (2019) explore using mBERT (Devlin et al., 2019) for direct cross-lingual trigger labeling and find it outperforms previous methods. Our `Direct` baseline can be considered as a re-implementation of their method with XLM-RoBERTa (Conneau et al., 2020). GATE (Ahmad et al., 2021) follows (Subburathinam et al., 2019) and uses a graph convolutional architecture and pretrained knowledge from language models to further improve the performance. Yarmohammadi et al. (2021) first translate the whole sentence and then uses token aligners to get a sub-sentential alignment, which has shown to be beneficial. We use a different translation strategy, and our proposed adversarial training approach may also be helpful with their translations. A more recent and parallel attempt (Guzman-Nateras et al., 2022) proposes to use adversarial training to close the gap between the source language and target language for event trigger labeling, which is different from our approach. (Fincke et al., 2022) uses priming methods to make the model understand the critical information for argument labeling. The performance of these two methods is not directly comparable due to different splits and limited code availability. We will add comparison once they release code. (Huang et al., 2022) proposes a generative approach to directly generate arguments for cross-lingual event argument extraction. However they don’t take entity spans as inputs for evaluation and results are not comparable.

5 Conclusions and Future Work

In this paper, we proposed a new cross-lingual event extraction framework and evaluated the framework on the ACE 2005 dataset. Our framework combines the multilingual language models with a machine-translation-based method. Meanwhile, we observe the representational gap between the translated text and the native text in multilingual language models that may affect the performance and propose an adversarial training approach to make the language model produce more similar representations for these two types of text.

One potential reason for remaining errors in cross-lingual transfer learning could be that the

⁴Cross-lingual summarization has a different task formulation than common cross-lingual learning, but it is still related.

source and the target languages may differ in the common expressions of an event type. It will be helpful to detect such differences from pretrained multilingual language models and incorporate them for training. Although we focus on cross-lingual event extraction in this work, our adversarial training approach could be extended to other cross-lingual language understanding tasks.

6 Limitations

Although we have demonstrated our framework’s performance in six cross lingual transfer learning directions for both the trigger labeling and argument role labeling, our experiments is mostly on the ACE 2005 dataset due to the availability of multilingual event extraction data. Since the ACE 2005 dataset only contains Arabic, Chinese and English, we were not able to test our framework on some languages with extremely limited resources, which are more common use cases for the cross lingual transfer learning .Besides, although our proposed adversarial loss is a general approach not specific to the event extraction task, we have not validate the effectiveness of it on other cross lingual NLP benchmarks or using other machine translation models. Moreover, our supervised models are trained in the multilingual language model (XLM-RoBERTa) for direct comparison. However, the performance is different from models trained with monolingual language models specific to the target language.

7 Acknowledgement

This research is based upon work supported in part by U.S. DARPA LORELEI Program No.HR0011-15-C-0115, U.S. DARPA AIDA Program No. FA8750-18-2-0014 and KAIROS Program No. FA8750-19-2-1004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. [GATE: graph attention transformer encoder for cross-lingual relation and event extraction](#). In

Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pages 12462–12470. AAAI Press.

Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. [Wasserstein GAN](#). *CoRR*, abs/1701.07875.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. [Learning bilingual word embeddings with \(almost\) no bilingual data](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. [A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Yue Cao, Hui Liu, and Xiaojun Wan. 2020. [Jointly learning to align and summarize for neural cross-lingual summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6220–6231, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. [Cross-lingual language model pretraining](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Corinna Cortes and Vladimir Vapnik. 1995. [Support-vector networks](#). *Mach. Learn.*, 20(3):273–297.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2019. [Cross-lingual machine reading comprehension](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1586–1595, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of](#)

- deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2021. **FILTER: an enhanced fusion method for cross-lingual language understanding**. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12776–12784. AAAI Press.
- Manaal Faruqui and Chris Dyer. 2014. **Improving vector space word representations using multilingual correlation**. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden. Association for Computational Linguistics.
- Steven Fincke, Shantanu Agarwal, Scott Miller, and Elizabeth Boschee. 2022. **Language model priming for cross-lingual event extraction**. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10627–10635. AAAI Press.
- Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. **Bilbowa: Fast bilingual distributed representations without word alignments**. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 748–756. JMLR.org.
- Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. **Improved training of wasserstein gans**. *CoRR*, abs/1704.00028.
- Luis F Guzman-Nateras, Minh Van Nguyen, and Thien Huu Nguyen. 2022. **Cross-lingual event detection via optimized adversarial training**. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, Seattle, Washington, USA. Association for Computational Linguistics.
- Tsung-Yuan Hsu, Chi-Liang Liu, and Hung-yi Lee. 2019. **Zero-shot reading comprehension by cross-lingual transfer learning with multi-lingual language representation model**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5933–5940, Hong Kong, China. Association for Computational Linguistics.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020a. **XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation**. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020b. **XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation**. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.
- Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. **Multilingual generative language models for zero-shot cross-lingual event argument extraction**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4633–4646, Dublin, Ireland. Association for Computational Linguistics.
- Lifu Huang, Heng Ji, and Jonathan May. 2019. **Cross-lingual multi-level adversarial transfer to enhance low-resource name tagging**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3823–3833, Minneapolis, Minnesota. Association for Computational Linguistics.
- Guillaume Lample, Alexis Conneau, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. **Word translation without parallel data**. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Bruce Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroong Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Rangan Majumder, and Ming Zhou. 2020. **XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation**. *CoRR*, abs/2004.01401.
- Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. **A joint neural model for information extraction with global features**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

- Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019. [Neural cross-lingual event detection with minimal parallel resources](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 738–748, Hong Kong, China. Association for Computational Linguistics.
- Fuli Luo, Wei Wang, Jiahao Liu, Yijia Liu, Bin Bi, Songfang Huang, Fei Huang, and Luo Si. 2020. [VECO: variable encoder-decoder pre-training for cross-lingual understanding and generation](#). *CoRR*, abs/2010.16046.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. [Bilingual word representations with monolingual quality in mind](#). In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado. Association for Computational Linguistics.
- Meryem M’hamdi, Marjorie Freedman, and Jonathan May. 2019. [Contextualized cross-lingual event trigger extraction with minimal resources](#). In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 656–665, Hong Kong, China. Association for Computational Linguistics.
- Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. [Exploiting similarities among languages for machine translation](#). *CoRR*, abs/1309.4168.
- Xuan Ouyang, Shuhuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. [ERNIE-M: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 27–38, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiaoman Pan, Thamme Gowda, Heng Ji, Jonathan May, and Scott Miller. 2019. [Cross-lingual joint entity and word embedding to improve entity linking and parallel sentence mining](#). In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 56–66, Hong Kong, China. Association for Computational Linguistics.
- Jason Phang, Iacer Calixto, Phu Mon Htut, Yada Puskaschatkun, Haokun Liu, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. [English intermediate-task training improves zero-shot cross-lingual transfer too](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 557–575, Suzhou, China. Association for Computational Linguistics.
- Daria Pylypenko, Kwabena Amponsah-Kaakyire, Koel Dutta Chowdhury, Josef van Genabith, and Cristina España-Bonet. 2021. [Comparing feature-engineering and feature-learning approaches for multilingual translationese classification](#). *CoRR*, abs/2109.07604.
- Parker Riley, Isaac Caswell, Markus Freitag, and David Grangier. 2020. [Translationese as a language in "multilingual" NMT](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7737–7746. Association for Computational Linguistics.
- Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. [XTREME-R: towards more challenging and nuanced multilingual evaluation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10215–10245. Association for Computational Linguistics.
- Sebastian Ruder, Ivan Vulic, and Anders Søgaard. 2019. [A survey of cross-lingual word embedding models](#). *J. Artif. Intell. Res.*, 65:569–631.
- Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. [Cross-lingual structure transfer for relation and event extraction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 313–325, Hong Kong, China. Association for Computational Linguistics.
- Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. 2021. [On learning universal representations across languages](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Mahsa Yarmohammadi, Shijie Wu, Marc Marone, Haoran Xu, Seth Ebner, Guanghui Qin, Yunmo Chen, Jialiang Guo, Craig Harman, Kenton Murray, Aaron Steven White, Mark Dredze, and Benjamin Van Durme. 2021. [Everything is all it takes: A multi-pronged strategy for zero-shot cross-lingual information extraction](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1950–1967, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. [Earth mover’s distance minimization for unsupervised bilingual lexicon induction](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1934–1945, Copenhagen, Denmark. Association for Computational Linguistics.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. *NCLS: neural cross-lingual summarization*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3052–3062. Association for Computational Linguistics.

A Appendix

A.1 Details for Model Training

For both the trigger labeling and role labeling task, we use batch size of 8 for training. We evaluate performance after each epoch and select the best model based on the development performance. We use early-stop strategy with a patience of 5 epochs. We conduct our experiments on a single Nvidia Tesla V100 GPU with 16GB memory.

The learning rate for both the trigger labeling and role labeling loss is $1e - 5$. In adversarial training, the learning rate for the discriminator loss is $1e - 5$. For the generator loss, we found in practice it is very likely to confuse the discriminators within a few steps if we finetune the whole XLM-RoBERTa architecture or the learning rate is set too large. Hence the generator learning rate for the generator loss is chosen between $\{1e - 5, 1e - 6, 1e - 7, 1e - 8, 1e - 9\}$ on the dev set for each cross lingual transfer learning task. We empirically found that the trigger labeling tasks usually take a smaller learning rate ($1e - 8, 1e - 9$) and the argument role labeling tasks usually take a larger one ($1e - 5, 1e - 6$). We also only finetune the last output layer of the XLM-RoBERTa model for the generator loss to match the capacity of the discriminators. The discriminator and the generator are trained alternatively. We train 5 discriminator steps per generator step.

For the simultaneous encoding of a sentence and its translation, we adopt the special fusion strategy in FILTER (Fang et al., 2021) for the role labeling task. FILTER will select some hidden layers of the XLM-RoBERTa model, for which it will concatenate the hidden representation of the original sentence and its translation together for self-attention computation. We follow FILTER to use the 21st layer for representation fusion. We found this strategy to be more helpful in role labeling task than trigger labeling task. In trigger labeling task, it suffices to simply encode the sentence pairs individually for prediction.

The approximate number of parameters is 3.5 million (mainly parameters of XLM-RoBERTa). We run our model on a single NVIDIA V100 with 16 GB memory. Training our framework takes approximately 20-40 minutes/epoch since 16GB memory can only take batch size of 1 for training. We need to accumulate the gradients over multiple runs for larger batch size. However, we notice that our model usually converges much faster than a simple XLM-RoBERTa baseline (Direct baseline). Usually we achieve our best model with 2-4 epochs. In total it usually takes around 4-5 hours to train a model. We implement the XLM-RoBERTa model using Transformers⁵ Library.

For the back propagation, note that the gradients of the loss in Equation (1) are back propagated to both the language model and the trigger classification layers, the gradients of the loss in Equation (2) are back propagated to D_s and D_t , and the gradients of the loss in Equation (3) are back propagated to the language model. In practice we found that it is beneficial to back propagate loss in Equation (3) to only the last layer of the FILTER model to match the capacity of the discriminators D_s and D_t .

A.2 Details for Machine Translation

We use Google Cloud API⁶ for machine translation. For trigger labeling, if a sentence contains multiple triggers, we enclose each of them with “” and “” for translation. After the sentence is translated, we retrieve all trigger spans in the target language one by one, and map them back to the triggers in the source language according the offset in the sentence. For example, the first trigger span in the source language will be mapped to the first trigger span in the target language. If we retrieve less triggers spans in the target language than the source language, we consider this translation invalid and discard this instance for the trigger labeling loss. We still use it for the adversarial training. For argument role labeling, we directly translate the sentence with inserted “<a>”, “”, “”, “” and always apply the role labeling loss on the translated sentence even if it may not contain paired special tokens.

For trigger labeling, our translation method retrieved⁷ 4,284 event triggers out of 4,419 triggers in

⁵<https://huggingface.co/docs/transformers/index>

⁶<https://cloud.google.com/translate>

⁷Here “retrieved” means that after the translation of a source language sentence of the format in Table 1, the trans-

the ACE 2005 English training data. For argument role labeling, there is no simple automatic metric to evaluate our translation method. Therefore, we sampled a small portion of the translation and conduct a small scale manual evaluation. 80.0% of the translations are considered reasonable by human assessors.

The reason behind this translation strategy is that the machine translation model trained on large-scale web-crawled data could have seen some HTML tags during training. “” are HTML tags for displaying bold characters, and “<a>” are tags for the content of reference links. Therefore we expect the model to translate properly if it can translate HTML formatted text.

A.3 ACE 2005 Dataset Details

This dataset is licensed by LDC.⁸ Membership is required for access. The dataset can be used for research purpose.

There are three languages in this dataset. For all the languages, we notice a significant long-tailed distribution among event types. We provide number of event mentions for all splits in Table 7. We also notice that the most frequent types for all languages are similar with minor differences.

A.4 Details of Additional Discriminators for Case Study

For fair comparison of the additional discriminators for the `Trans` baseline and the discriminators in our framework, we also jointly train the the discriminators with the `Trans` baseline in the same way as we conduct adversarial training in our framework. The training process can be seen as training our framework with the generator learning rate being 0. Note that the parameters of the discriminators are disjoint of that of the `Trans` baseline model. Therefore the joint training will not affect the learning the `Trans` baseline model.

A.5 Corruption Ratio of Translated Training Data

We provide corruption ratio for the argument role labeling task here for translation of the training data. Due to our strategy of inserting special tokens, a corrupted translation is defined as a translated

lated sentence include paired “” and “” tokens and the content between them are not empty. In this sense retrieved triggers are not guaranteed to be correct annotations. This is just a rough estimation of the performance of proposed translation method.

⁸<https://www ldc.upenn.edu>

sentence without either of the special tokens. In sentences translated into Arabic, we noticed that special tokens are sometimes translated as ‘<a >’ or ‘’ with additional spaces. We don’t consider them as corrupted and automatically cleaned up such errors. The corruption ratios are as below: EN-ZH, 10%; EN-AR: 22%; ZH-EN: 12%, ZH-AR: 27%; AR-EN: 26%; AR-ZH: 38%.

It is also worth mentioning that Google translate offers the option to respect HTML mark up. However, we didn’t adopt this option in our experiments. We believe enabling this function can further reduce the corruption ratio and potentially improve the performance.

A.6 Full Results

We present full results of all six cross-lingual transfer settings across two tasks, including the precision, recall and f1 scores. We include trigger labeling performance in Table 8a-8f. We include role labeling performance in Table 9a-9c.

Split	English			Chinese			Arabic		
	train	dev	test	train	dev	test	train	dev	test
Conflict:Attack	1,272	172	93	470	37	17	377	45	55
Movement:Transport	611	59	48	662	54	43	354	46	34
Life:Die	524	53	17	211	18	14	177	33	34
Contact:Meet	200	29	50	163	19	26	152	38	27
Personnel:Elect	162	4	16	28	1	9	31	6	4
Personnel:End-Position	159	19	22	71	5	11	37	14	7
Transaction:Transfer-Money	128	52	14	84	3	5	34	11	3
Life:Injure	127	9	1	149	7	7	92	14	21
Contact:Phone-Write	112	3	8	77	8	2	45	3	8
Justice:Trial-Hearing	103	1	5	79	4	8	58	1	6
Justice:Charge-Indict	96	2	8	50	0	2	45	2	5
Transaction:Transfer-Ownership	92	4	30	84	2	1	9	0	1
Personnel:Start-Position	92	12	13	95	5	2	36	10	0
Justice:Sentence	84	4	11	79	4	7	46	1	4
Justice:Arrest-Jail	78	4	6	115	11	6	82	13	14
Life:Marry	73	0	10	55	0	2	9	7	0
Conflict:Demonstrate	65	9	7	72	3	1	55	8	10
Justice:Convict	64	6	6	13	3	0	3	1	1
Justice:Sue	60	12	4	76	0	3	2	0	0
Life:Be-Born	47	0	3	22	0	6	6	0	0
Justice:Release-Parole	46	0	1	31	5	2	18	6	7
Business:Declare-Bankruptcy	40	1	2	15	0	4	1	0	0
Business:End-Org	31	1	5	16	0	2	6	1	1
Justice:Appeal	30	7	6	35	0	0	12	0	7
Business:Start-Org	29	0	18	77	2	5	12	0	2
Justice:Fine	22	0	6	7	4	2	33	0	0
Life:Divorce	20	0	9	11	0	0	3	2	0
Business:Merge-Org	14	0	0	36	16	1	1	0	0
Justice:Execute	14	5	2	5	0	1	0	0	0
Personnel:Nominate	11	0	1	24	0	1	4	0	3
Justice:Extradite	6	0	1	2	2	0	7	0	0
Justice:Acquit	5	0	1	3	0	0	3	0	0
Justice:Pardon	2	0	0	9	4	0	1	0	1

Table 7: Event type distribution for the event trigger labeling task

Trigger Labeling	P(%)	R(%)	F(%)
Direct	42.3	52.5	46.9
Trans	39.9	58.1	47.3
Trans+Adv (ours)	42.5	58.7	49.3
ZH Supervision	65.2	65.9	65.6

(a) English-to-Chinese.

Trigger Labeling	P(%)	R(%)	F(%)
Direct	50.6	39.6	44.4
Trans	56.0	39.4	46.3
Trans+Adv (ours)	63.2	48.1	54.6
EN Supervision	63.0	75.0	68.5

(c) Chinese-to-English.

Trigger Labeling	P(%)	R(%)	F(%)
Direct	30.3	37.3	33.4
Trans	34.5	44.3	38.8
Trans+Adv (ours)	36.1	45.1	40.1
AR Supervision	49.4	64.9	56.1

(e) Chinese-to-Arabic.

Trigger Labeling	P(%)	R(%)	F(%)
Direct	32.0	50.0	39.0
Trans	33.1	48.1	39.3
Trans+Adv (ours)	38.1	47.7	42.3
AR Supervision	49.4	64.9	56.1

(b) English-to-Arabic.

Trigger Labeling	P(%)	R(%)	F(%)
Direct	43.1	33.3	39.8
Trans	57.4	29.9	39.4
Trans+Adv (ours)	56.0	33.0	41.5
EN Supervision	63.0	75.0	68.5

(d) Arabic-to-English.

Trigger Labeling	P(%)	R(%)	F(%)
Direct	35.3	38.3	36.7
Trans	37.6	35.6	36.6
Trans+Adv (ours)	49.6	31.4	38.4
ZH Supervision	65.2	65.9	65.6

(f) Arabic-to-Chinese.

Table 8: Precision(P), recall(R) and f1(F) scores for the cross-lingual trigger labeling task. *Direct*, *Trans* and *Target Supervision* are introduced in Section 3.2.

Argument Role Labeling	Chinese-to-English			Chinese-to-Arabic		
	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)
GATE	48.0	70.0	57.0	64.1	66.1	65.1
Direct	59.7	63.4	61.5	68.2	60.3	64.0
Trans	56.6	65.0	60.6	67.9	60.1	63.8
Trans+Adv (ours)	59.1	67.3	62.9	72.4	64.4	68.0
Target Supervision	75.1	79.5	77.2	77.5	78.1	77.8
(a) Chinese as the source language.						
Argument Role Labeling	English-to-Chinese			English-to-Arabic		
	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)
GATE	60.7	66.8	63.6	72.5	58.9	65.0
Direct	72.6	70.8	71.7	81.5	50.7	62.5
Trans	73.0	69.7	71.3	76.3	52.5	62.2
Trans+Adv (ours)	72.2	71.8	72.0	76.0	57.0	65.1
Target Supervision	79.7	84.4	82.0	77.5	78.1	77.8
(b) English as the source language.						
Argument Role Labeling	Arabic-to-English			Arabic-to-Chinese		
	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)
GATE	40.4	70.5	50.3	44.7	74.1	55.7
Direct	50.5	64.8	56.8	60.7	69.0	64.6
Trans	50.6	66.6	57.5	62.2	67.8	64.9
Trans+Adv (ours)	54.1	63.4	58.4	64.1	67.1	65.6
Target Supervision	75.1	79.5	77.2	79.7	84.4	82.0
(c) Arabic as the source language.						

Table 9: Precision(P), recall(R) and f1(F) scores for the cross-lingual argument role labeling task. *GATE* (Ahmad et al., 2021) is a state-of-the-art method for cross-lingual argument role labeling. *Direct*, *Trans* and *Target Supervision* are introduced in Section 3.2.

Scalable Prompt Generation for Semi-supervised Learning with Language Models

Yuhang Zhou*
University of Maryland
College Park, MD
tonyzhou@umd.edu

Suraj Maharjan*
Amazon
Seattle, WA
mhjsuraj@amazon.com

Beiye Liu
Amazon
New York, NY
beiyeliu@amazon.com

Abstract

Prompt-based learning methods in semi-supervised learning (SSL) settings have been shown to be effective on multiple natural language understanding (NLU) datasets and tasks in the literature. However, manually designing multiple prompts and verbalizers requires domain knowledge and human effort, making it difficult and expensive to scale across different datasets. In this paper, we propose two methods to automatically design multiple prompts and integrate automatic verbalizer in SSL settings without sacrificing performance. The first method uses various demonstration examples with learnable continuous prompt tokens to create diverse prompt models. The second method uses a varying number of soft prompt tokens to encourage language models to learn different prompts. For the verbalizer, we use the prototypical verbalizer to replace the manual one. In summary, we obtained the best average accuracy of 73.2% (a relative improvement of 2.52% over even the previous state-of-the-art SSL method with manual prompts and verbalizers) in different few-shot learning settings.

1 Introduction

Pre-training large language models with huge amounts of text corpora in masked language modeling tasks and then fine-tuning the pre-trained language model (PLM) on downstream tasks have shown superior performance in many natural language processing tasks. However, the discrepancy between the pretraining task (masked language modeling objective) and the downstream fine-tuning task (task without MASK token) could lead to unexpected behaviors. Recently, there has been growing research interest in the area of prompt-tuning, where any NLU task is transformed into a cloze task to mimic the pre-training objective of a large masked language model (Kumar et al.,

2016; McCann et al., 2018; Radford et al., 2018). Prompt-based learning transforms an input x into x' using a prompt function. It makes use of the vast amount of acquired knowledge of PLMs to predict a distribution of tokens at the masked position. The verbalizer then maps the predicted tokens to classes. The main advantage of this approach is that this method works well in a few-shot learning environment (Schick and Schütze, 2021). However, the main disadvantage of this method is the limitation posed by the prompt and verbalizer functions, which require human knowledge to carefully craft them. Such handcrafting work is expensive and not scalable with the increase in the variety of tasks and datasets. For example, in Alexa, there are thousands of domains and manually designing prompts and verbalizer for intent classification for each of them according to the dataset content demand human expertise, which is time consuming and not applicable. It is essential to reduce the human efforts in the process of prompt generation. Prompt-based learning requires finding the right tokens in the prompts that align with the task requirement and dataset content. However, since the objective of these prompt tokens is only for the language models to perform the task at hand, it is not necessary for them to be a sequence of words that humans can understand.

Continuous prompt-based learning alleviates the need for human intervention to determine prompt tokens. Instead, it automates the prompt design process. In the literature, there are mainly two methods: i) automatically search for discrete prompt text tokens (Shin et al., 2020a) ii) automatically learn numerical prompt embeddings (Lester et al., 2021; Li and Liang, 2021; Liu et al., 2021c,b; Hambarzumyan et al., 2021). The main difference between these two approaches is that the first searches for actual discrete tokens from the language model vocabulary, whereas the second method directly learns the embeddings for prompt tokens, which

*Equal contribution. This work was done during Yuhang's internship at Amazon, Alexa AI.

may not be human comprehensible. Similarly, automatic selection of label words (Shin et al., 2020a; Schick et al., 2020a; Gao et al., 2021), soft verbalizer (Hambardzumyan et al., 2021; Liu et al., 2021b), and prototypical verbalizer (Cui et al., 2022) are the methods proposed to eliminate the tedious process of manually defining verbalizer mapping functions.

Most of these continuous prompt and automatic verbalizer methods focus on supervised learning (SL) settings but ignore their generalization under semi-supervised learning (SSL) settings. The previous state-of-the-art (SoTA) SSL method with various manual prompts and verbalizers has shown superiority over SL language models with a single manual prompt (Schick and Schütze, 2021). In this SSL pipeline, we normally train several labeler models with different manual prompts to capture diverse information from the limited training data and make use of them to annotate a huge amount of unlabeled data. Having to design several manual prompts and verbalizer models for SSL settings and applying them across multiple datasets and tasks will exacerbate the scalability and cost problem. In this paper, we tackle the problem posed by manual prompt and verbalizer design and propose automatic methods to fully automate the design of diverse prompts and verbalizers in SSL settings. Our main contributions are as follows.

- We propose methods to generate various prompts by adding multiple demonstration examples with continuous prompt tokens for use in SSL settings.
- To the best of our knowledge, we are the first to completely eliminate human involvement in designing multiple prompts and verbalizers in SSL settings and obtain similar and even better performance than the SoTA methods with manual prompts and verbalizers.
- We empirically show that using the automatic verbalizer with manual prompts can achieve a similar performance to manual verbalizers’ performance in the SSL pipeline.

2 Methodology

Our overall prompt-based SSL workflow follows Pattern-exploiting Training (PET) semi-supervised learning setting (Schick and Schütze, 2021). PET first transforms the input sequence x to a cloze

question containing a single MASK token. Next, it uses PLM to fill in the value of the MASK token and applies verbalizers to map the output tokens to the class labels $y \in Y$. They devise a semi-supervised framework to produce soft labels on a large amount of unlabeled data, which are later used to train a final supervised classifier F . They report strong performance over other supervised prompt-tuning methods and other semi-supervised approaches without prompts across multiple NLU tasks. Before this paper, the PET approach was the state-of-the-art (SoTA) framework that integrates the prompt-tuning method into the SSL pipeline.

The PET method fine-tunes multiple PLMs with different prompts. It introduces diversity in the prompts by manually designing several prompts using domain and task knowledge. Similarly, it uses human expertise to design verbalizer mappings for each of the datasets based on the knowledge of the tasks. Here, we use continuous and automatic prompts and verbalizers, thus eliminating the need for human involvement in designing manual prompts and verbalizers.

2.1 Overall Pipeline

Figure 1 shows the overall pipeline of our proposed methods. Unlike the original PET pipeline with manual prompts and verbalizers, we use a prompt generation function to generate multiple automatic prompts. Each PLM with automatic prompts serves as a labeler model. We train each of these prompts + automatic verbalizer models with a labeled dataset \mathcal{T} in few-shot settings. With an input sequence $x_t \in \mathcal{T}$ and the given label y_t , we first use the prompt function P to transform x_t into a sequence $P(x_t)$ with a MASK token. The verbalizer then maps the predicted word probability at the masked position to the label probability. For each PLM m , the predicted probability $p_m(y_t|x_t)$ is defined as

$$p_m(y_t|x_t) = \frac{\exp m(y_t|x_t)}{\sum_{y' \in Y} \exp m(y'|x_t)} \quad (1)$$

where $m(y|x)$ is the raw score of PLM m in the masked position. After obtaining the probability, we minimize the cross-entropy loss \mathcal{L}_c between $p_m(y|x)$ and y .

We apply trained labeler models to each sentence $x_d \in \mathcal{D}$ in the unlabeled dataset \mathcal{D} and get the probability $p_m(y_d|x_d)$ for each trained model. We then take the average of these probabilities from each

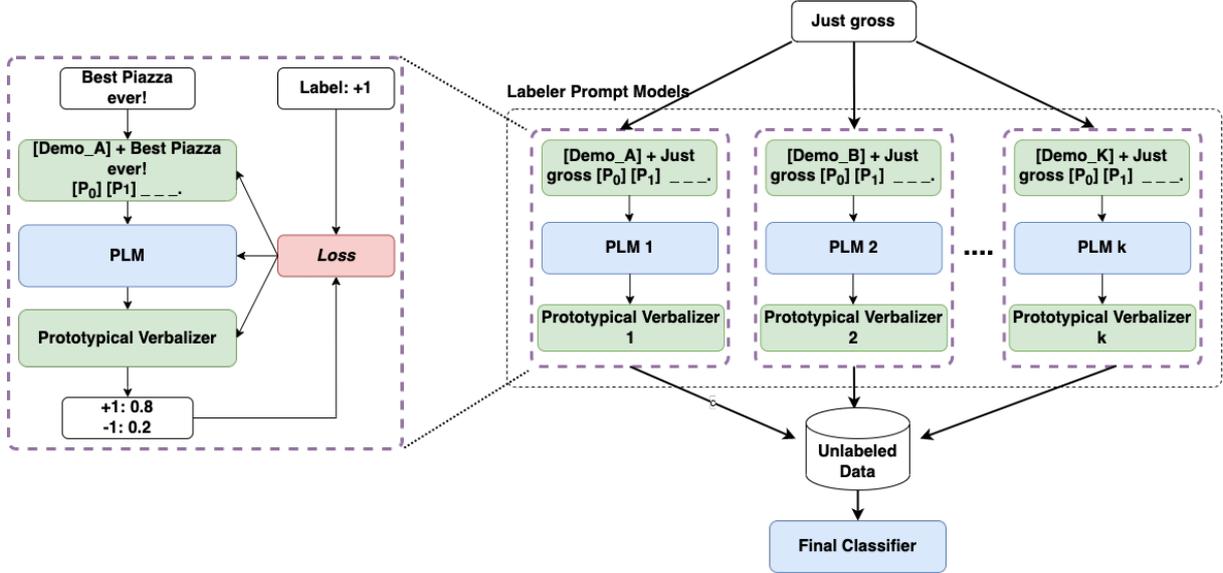


Figure 1: Semi-Supervised Learning (SSL) Training. Multiple diverse prompt-based learning models are trained on labeled data to soft label huge amounts of unlabeled data. The soft labels serve as ground truth to train the final classifier. P_0, P_1, \dots are continuous prompt tokens and $Demo_A, Demo_B, \dots$ are demonstration examples randomly sampled from the training data.

trained model m as the ground-truth probability,

$$p_t(y_d|x_d) = \frac{1}{Z} \sum_{m \in M} p_m(y_d|x_d)$$

where Z is the total number of trained PLMs with different automatic prompts. Eventually, we fine-tune a final pre-trained language model \mathbf{F} with a standard sequence classification head. We use the Kullback-Leibler (KL) divergence as our loss function. Given $p_t(y_d|x_d)$ and the predicted probability $\hat{p}(y_d|x_d)$ of the final classifier \mathbf{F} , the divergence loss \mathcal{L}_{div} for this input is:

$$\mathcal{L}_{div}(x_d) = \sum_{y' \in Y} p_t(y'|x_d) \log \left(\frac{p_t(y'|x_d)}{\hat{p}(y'|x_d)} \right) \quad (2)$$

The final classifier \mathbf{F} is then applied to the test set to obtain the results.

Schick and Schütze (2021) introduce diversity in their SSL pipeline by training several models with different manual prompts and applying them to softly label a large number of unlabeled datasets. The diversity between manual prompts brings consistent improvements. We observe that diverse knowledge learned by the language model is mostly introduced by the prompts rather than manual verbalizers, since in most datasets, they prepare only one manual verbalizer but multiple prompts for experimentation. Thus, we propose replacing manual

prompts with multiple automatic prompts and using the same automatic verbalizer for all labeler models.

2.2 Continuous Prompt Design

Several researchers have proposed methods to automate the prompt design process (Liu et al., 2021c; Li and Liang, 2021; Lester et al., 2021). In most of these methods, they insert the continuous trainable prompt tokens into the input sentence and learn the token embeddings during the training process. However, existing continuous prompt-based learning methods do not consider their application in the PET pipeline, which requires training several labeler models (Schick and Schütze, 2021), in order to learn diverse knowledge from the datasets. Therefore, most methods do not define strategies to compose multiple continuous prompts. We propose two scalable solutions to introduce different variables in the design of continuous prompt labeler models (various demonstration examples or varying numbers of continuous prompt tokens). We expect that with these diverse continuous prompts, trained language models can fully learn different aspects of knowledge from the training dataset.

2.2.1 Scalable Prompt Generation

Inspired by the P-tuning (Liu et al., 2021c) method, we insert multiple continuous prompt tokens p_n into the input sentence x , transforming it into

$[\mathbf{x}][p_0, p_1, \dots, p_n][\text{MASK}]$. Different from the original P-tuning method, we invent two scalable designs to make it suitable for the prompt-based SSL pipeline.

Add Demonstration Examples: In this method, we add different demonstration examples to construct diverse prompts. This is similar to the prompt augmentation method, in which one chooses to add additional answered prompts to demonstrate what kind of answer the language model should produce for the MASK token (Liu et al., 2021a). These additional answered prompts are called the demonstration example $[demo]$. To reduce the discrepancy between the demonstration examples and the input sentences, we also add a fixed number of continuous prompt tokens p between the demonstration sentence and its true label. Thus, given the labeled input \mathbf{x}_d and its corresponding ground-truth label \mathbf{y}_d from the labeled training dataset, we construct the demonstration example as $[demo] = [\mathbf{x}_d][p_0, p_1, \dots, p_n][\mathbf{y}_d]$, where p_0, p_1, \dots, p_n are continuous prompt tokens.

After composing the demonstration examples $[demo]$, given a training input from the labeled dataset $x_t = (s_i, s_2, \dots, s_k) \in \mathcal{T}$ and label y_t , where s_i, s_2, \dots, s_k are input tokens for the PLM m , the prompt template function $P_1(x_t)$ is formally defined as

$$\begin{aligned} P_1(x_t)_1 &= [demo_1][\mathbf{x}_t][p_0, \dots, p_n][\text{MASK}] \\ \dots & \\ P_1(x_t)_k &= [demo_k][\mathbf{x}_t][p_0, \dots, p_n][\text{MASK}] \end{aligned} \quad (3)$$

We create multiple prompts by adding different demonstration examples with exactly n continuous soft tokens with the input sentence. Demonstration examples are randomly sampled from the labeled datasets. For longer input sentences, we first truncate the length of $[demo]$ to fit the PLM requirement. Our intuition is that different demonstration examples will introduce the diversity necessary for SSL experimentation.

Vary Soft Token Numbers: In this method, we vary the number of continuous prompt tokens between different labeler models. In other words, this prompt function $P_2(x_t)$ with input sentence x_t is defined as

$$\begin{aligned} P_2(x_t)_1 &= [\mathbf{x}_t][p_0, p_1, \dots, p_{n_1}][\text{MASK}] \\ \dots & \\ P_2(x_t)_k &= [\mathbf{x}_t][p_0, p_1, \dots, p_{n_k}][\text{MASK}] \end{aligned} \quad (4)$$

and each of the labeler models uses different n_1 to n_k number(s) of continuous prompt tokens p . Here, we do not prepend the demonstration example. Our intuition is that given different numbers of continuous prompt tokens, the optimized learned continuous prompts may also be different. For example, for AG’s News dataset (Zhang et al., 2015a) about news topics, the optimized prompts with two continuous prompt tokens could be: $[[\mathbf{x}][\text{News : }][\text{MASK}]]$, while optimized prompts with three continuous prompt tokens could be: $[[\mathbf{x}][\text{the category is}][\text{MASK}]]$. We expect that varying the number of continuous prompt tokens will have a similar impact to manually constructing different prompts.

2.2.2 Reparameterization Block

Li and Liang (2021) and Liu et al. (2021c) empirically show that directly updating the parameters in continuous prompts leads to unstable optimization. Hence, we first feed prompt embeddings through a reparameterization block rather than directly feeding them into the PLM. Our reparameterization block uses a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) network with a two-layer *ReLU* activated multilayer perceptron (MLP) (Liu et al., 2021c; Li and Liang, 2021).

We denote the random initialized tokens as p'_i and the real input embeddings, which are fed into the PLM, as p_i . The p_i are the output of the bidirectional LSTM network and the MLP as,

$$p_i = \text{MLP}([\text{LSTM}(p'_{0:i}), \text{LSTM}(p'_{i:n})])$$

where p_i is also the soft token used in Equations 3 and 4. We learn the optimized continuous prompt tokens $\hat{p}_{0:n}$ during the training process. With the downstream cross-entropy loss \mathcal{L}_c , we can differentially optimize the continuous prompts by:

$$\hat{p}_{0:n} = \underset{p}{\text{argmin}} \mathcal{L}_c(p_m(x|y), y) \quad (5)$$

2.3 Automatic Verbalizers

There are several automatic verbalizer methods that eliminate the need for human intervention and expertise to build mapping functions. We experiment with three types of automatic verbalizers: i) soft verbalizer (Hambarzumyan et al., 2021), ii) prototypical verbalizer (Cui et al., 2022), and iii) search-based verbalizer (Schick et al., 2020b).

Cui et al. (2022) experimentally show the superiority of the prototypical verbalizer in a supervised learning environment. However, they did not

conduct such experiments for SSL settings. Our experiment with the SSL PET method (details in Section 3.5) with different automatic verbalizers showed that the prototypical verbalizer performed better than the soft verbalizer and the search-based verbalizer on multiple datasets. Thus, we choose to use the prototypical verbalizer as a replacement for the manual verbalizer.

With the optimized embedding of the MASK token from PLM m and the ground-truth labels y , the prototypical verbalizer learns the prototype vectors for each class using contrastive learning (Oord et al., 2018). The prototypical verbalizer first initializes a prototype embedding for each class label and then uses the embedding of the MASK token as the instance embedding. It uses instance-instance loss \mathcal{L}_{ins} to maximize intra-class similarity and minimize inter-class similarity. Similarly, it uses instance-prototype loss \mathcal{L}_{proto} to maximize the similarity between the prototype and instances belonging to the same class and minimize the similarity of instances belonging to other classes. The probability distribution of the MASK token for each class is calculated by the cosine similarity between the instance embedding and each optimized prototype embedding. For inference, it assigns the class of the prototype vector to the instance with the highest probability score, which is computed by taking the similarity scores of the instance vector with the prototype vectors and normalizing them.

2.4 Training and Inference Strategy

All model parameters to be optimized are randomly initialized. As mentioned in Section 2.2.2 and 2.3, we update the parameters in the continuous prompts and PLMs with the loss \mathcal{L}_c and optimize the parameters in the verbalizers with the loss \mathcal{L}_{ins} and \mathcal{L}_{proto} . Instead of summing all losses together, our training strategy is to first freeze the parameters in the prototypical verbalizer and then train the parameters in the reparameterization block and the PLM together with the cross-entropy loss \mathcal{L}_c . Then we freeze the learned parameters and train the parameters in the prototypical verbalizers with instance-instance loss \mathcal{L}_{ins} and instance-prototype loss \mathcal{L}_{proto} . After training all labeler models and obtaining the class probability on the unlabeled dataset, we use \mathcal{L}_{div} to fine-tune the final language model classifier. During inference, we do not rely on any prompt-based labeler models and directly use the final fine-tuned language model \mathbf{F} to predict

on the test dataset.

3 Experiments

To verify the effectiveness of our framework, we conduct multiple semi-supervised learning experiments with several strong baseline frameworks on the commonly-used NLU benchmarks.

3.1 Dataset Collection

We experiment with five different datasets¹: AG’s News (Zhang et al., 2015a), Yahoo Answers (Zhang et al., 2015b), MNLI (MultiNLI, Multi-Genre Natural Language Inference, Williams et al. (2018)), RTE (Recognizing Textual Entailment, Dagan et al. (2006)) and CB (Commitment-Bank, de Marneffe et al. (2019)). AG’s News and Yahoo answers are topic classification (TC) datasets, while MNLI, RTE, and CB are natural language inference (NLI) datasets. In Table 1, we provide the number of distinct classes, the unlabeled dataset size used for SSL, and the test size for all five datasets. Details about the design of prompts and verbalizers can be found in Appendix A.

Dataset	Task	#Class	#Unlabeled	#Test
AG’s News	TC	4	40,000	7,600
Yahoo	TC	10	100,000	60,000
CB	NLI	3	30,000	56
RTE	NLI	2	20,000	277
MNLI	NLI	3	30,000	9,815

Table 1: Data statistics. TC= Topic Classification, NLI= Natural Language Inference

We perform multiple experiments in few-shot settings for all datasets. For few-shot experiments, we use 1, 5, 10, 20 examples per class for all datasets except for CB and RTE, where we experiment with 32 examples to align with earlier research work (Schick and Schütze, 2021). We report the average accuracy for the evaluation across three runs of each experiment with three different random seeds.

3.2 Proposed Models

Demo+Soft Tokens PET: The first method is to replace the manual verbalizer with the prototypical verbalizer and manual prompts with demonstration examples and continuous prompt tokens.

¹We downloaded these datasets using the script provided by OpenPrompt <https://github.com/thunlp/OpenPrompt>

Vary Soft Tokens PET: The second method is to introduce diversity by varying the number of continuous prompt tokens, and we use the prototypical verbalizer across multiple labeler models.

3.3 Models for Comparison

We design several strong baseline experiments in addition to our proposed models and also perform an ablation study to show the superiority of our proposed models in multiple NLU tasks.

3.3.1 Baseline Models

Fine-tune: This is a supervised method, where we directly fine-tune the RoBERTa-large PLM with training examples in different few-shot settings. In this method, we do not leverage the unlabeled data.

Prototypical Verbalizer PET: This is a semi-supervised learning method similar to Schick and Schütze (2021), but we replace the manual verbalizer with the prototypical verbalizer and keep the manual prompts. Experiments with this setup will show the benefits of applying automatic verbalizer in the PET framework.

Manual PET: This is a semi-supervised learning method from Schick and Schütze (2021). Our main goal is to show that, with our proposed method, we can achieve similar or better results than this manual method.

There are other SSL methods that rely on data augmentation without prompt tuning, such as UDA (Xie et al., 2020) and MixText (Chen et al., 2020). Since their performance is consistently worse than the Manual PET model across multiple datasets (Schick and Schütze, 2021), we do not choose these models for comparison in this work.

3.3.2 Model Intervention for Ablation Study

Fixed Soft Tokens PET: This semi-supervised learning method is similar to our second proposed method, where we vary the number of continuous tokens to create multiple prompts. However, here we keep the number of continuous tokens fixed and do not add demonstration examples as well. This experiment will help us to understand the importance of diversity introduced by varying continuous tokens in prompt design.

Demo+Soft in SL: This is a supervised method, where we use a prompt template to transform the input by adding a randomly selected demonstration example from the training data and a fixed number of continuous prompt tokens to the input,

and we use the prototypical verbalizer for classification. We use RoBERTa-large for PLM. With this experiment, we try to understand the power of semi-supervised learning methods with multiple prompts over supervised training.

3.4 Implementation Details

We use the RoBERTa-Large model (Liu et al., 2019) as our PLM for all of our experiments. We use AdamW as our optimizer with a learning rate of $1e-5$ and a weight decay of 0.01 with linear scheduler, batch size of 2, and trained for 5 epochs. The reparameterization block contains 2-layer bidirectional LSTM and 2 linear layers with ReLU activation function. The hidden dimension of the linear layer and LSTM layer is 768, as well as the hidden dimension of Roberta-Large. We train the parameters in the reparameterization block and the PLM together. For the prototypical verbalizer, we base our implementation on the Pytorch², Huggingface transformer³, and OpenPrompt⁴ frameworks (Ding et al., 2021). The number of continuous prompt tokens is consistent 5. For our Vary Soft Tokens PET, we prepare 5 prompts for each dataset and the number of soft tokens in each prompt ranges from 1 to 5.

3.5 Results of Multiple Automatic Verbalizers

Datasets	# instances	SSL PET		
		SoftVerb	SearchVerb	ProtoVerb
AG’s News	10	49.4	80.5	77.2
Yahoo	10	11.8	34.0	51.9
CB	32	88.7	73.2	85.7
RTE	32	48.2	50.2	52.8
MNLI	10	39.0	37.0	50.0

Table 2: Average accuracy on different datasets by replacing manual verbalizers with automatic verbalizers in the PET SSL setup. For CB and RTE, we use 32 training examples, whereas for other datasets, we use 10 training examples to train labeler models. The best performance is marked in bold.

To understand which automatic verbalizer is a better replacement for manual verbalizer, we first experiment with three automatic verbalizers: soft verbalizer (Hambardzumyan et al., 2021; Liu et al., 2021c,b), search verbalizer (Gao et al., 2021; Shin et al., 2020a; Schick et al., 2020a), and prototypical verbalizer (Cui et al., 2022). For all of these

²<https://pytorch.org/>

³<https://huggingface.co/>

⁴<https://github.com/thunlp/OpenPrompt>

experiments, we apply experimental setups similar to PET paper, but only replace the manual verbalizer with the automatic verbalizer (Schick and Schütze, 2021). Table 2 shows the average accuracy over three runs with three different seeds on different datasets with these verbalizers. From Table 2, the prototypical verbalizer shows better performance than other verbalizers for three (Yahoo, RTE, and MNLI) out of five datasets. The search verbalizer and soft verbalizer models perform better than the prototypical verbalizer model only on one dataset each. Since the prototypical verbalizer performs better than other verbalizers in majority of the datasets, we decided to use this as our automatic verbalizer.

3.6 Comparison with Manual PET

With the prototypical verbalizer as our automatic verbalizer, we then experiment with our proposed methods for automatic prompt design. Table 3 shows our results on different datasets and tasks in the few-shot setting. Table 3 shows that by only replacing the manual verbalizer with the prototypical verbalizer (column **Protoverb**) and keeping other aspects of the experiment the same as the PET method, we can achieve slightly lower performance (70.1 average accuracy) compared to Manual PET (71.4 average accuracy) (Schick and Schütze, 2021). This shows that to eliminate human involvement in designing verbalizers, we can simply replace the manual verbalizer with the prototypical verbalizer with only a little performance sacrifice.

For our next set of experiments, we replace manual prompts with our proposed method, automatically creating multiple prompts. The first method (Demo+Soft Tokens PET), which adds randomly sampled demonstration examples from training data with a fixed number of trainable continuous prompt tokens with input, achieves better performance than Manual PET method. The next method (Vary Soft PET), in which we vary the number of trainable tokens, also achieves better performance than Manual PET method. For topic classification tasks, under multiple few-shot settings, the average accuracy of Demo+Soft and Vary Soft PET are 77.0 and 77.3, respectively, while the average accuracy of Manual PET method is 77.1. Similarly, for NLI datasets under different few-shot settings, the average accuracy of our Vary Soft PET method is 69.6 and Demo+Soft Tokens

PET method is 70.7. Both of these results are better than Manual PET method (67.7). Furthermore, across all these datasets, Demo+Soft Tokens PET and Vary Soft PET achieve an average performance of 73.2 and 72.6, respectively. These results are better than Manual PET (71.4) method. This experiment shows that it is possible to completely eliminate human involvement and expertise in designing prompts and verbalizers for the SSL pipeline with even better performance.

We also observe that for the case of one-shot experiments with MNLI dataset, Demo + Soft PET method obtains an accuracy of 36.1, which is much worse than other prompt baseline models. This may be due to randomly sampled [*demo*] examples, as previous studies have shown that the choice of examples in the few-shot setting can result in high-variance performance (Lu et al., 2021). In future work, we can utilize sentence embeddings to make intelligent decisions while selecting demonstration examples.

3.7 Ablation Study

3.7.1 Impact of Semi-supervised Learning

We compare our proposed methods with supervised learning methods: fine-tuning and prompt-based tuning methods (Demo+Soft in SL). All semi-supervised learning methods perform significantly better than supervised learning methods. Traditional fine-tuning methods perform the worst (45.1 average accuracy) on different datasets and tasks. Demo+Soft in SL method is similar to our proposed Demo+Soft Tokens PET method but does not make use of unlabeled data. Demo+Soft in SL performs better than the fine-tuning method and achieves an average accuracy of 68.7 on multiple datasets and tasks in different few-shot settings. Both of the supervised learning methods perform worse than any SSL prompting model, indicating the necessity of the SSL pipeline in NLU tasks.

3.7.2 Impact of Diversity in the Prompts

In order to understand the effect of introducing diversity through multiple prompts in SSL, we devise another experiment, where we use the SSL setup but use only **one** prompt labeler model (not adding a demonstration example but using trainable soft tokens) to label unlabeled data. We name this method as Fixed Soft Tokens PET. Table 3 shows that in most comparisons (13/14), our proposed Vary Soft PET or Demo+Soft PET method achieves better performance. When comparing with the Fixed Soft

		Semi Supervised Learning PET					Supervised	
Dataset	# Training	Demo+Soft	Vary Soft	Fixed Soft	Proverb	Manual	Fine-Tune	Demo+Soft
Topic Classification								
AG’s News	1	83.5	81.3	82.8	80.0	80.7	25.7	62.2
AG’s News	5	87.6	88.0	87.3	87.3	87.8	32.6	84.9
AG’s News	10	88.3	88.3	86.5	88.7	88.8	58.3	87.2
AG’s News	20	88.8	89.3	88.9	89.2	89.2	86.1	88.0
Yahoo	1	61.1	62.9	59.6	62.0	62.3	10.7	55.6
Yahoo	5	67.4	67.9	67.1	67.8	68.0	12.1	65.2
Yahoo	10	68.9	69.5	69.1	70.0	69.5	37.8	67.0
Yahoo	20	70.7	71.0	70.4	70.9	70.7	66.7	66.5
TC Avg	-	77.0	77.3	76.5	77.0	77.1	41.2	72.1
Natural Language Inference								
MNLI	1	36.1	51.7	52.7	44.2	44.8	34.3	35.1
MNLI	5	51.2	58.1	57.7	55.3	55.2	33.5	46.9
MNLI	10	60.4	57.8	58.4	62.3	60.5	34.3	54.4
MNLI	20	64.0	64.7	60.5	69.6	68.6	35.0	41.9
CB	32	88.7	88.1	88.7	85.7	86.9	60.7	87.6
RTE	32	70.4	62.5	62.6	52.8	58.8	48.1	67.4
NLI Avg	-	70.7	69.6	69.5	65.5	67.7	47.7	66.5
Overall Avg	-	73.2	72.6	72.3	70.1	71.4	45.1	68.7

Table 3: Few-shot experiment results (average accuracy) on different datasets with our proposed methods in PET SSL setup. For CB and RTE, we use 32 training examples, whereas for other datasets we use {1, 5, 10, 20} randomly selected examples per class for few-shot learning experiments. The best performance is marked in bold. Note that to report the average results for NLI task, we first average over the MNLI results under different few-shot settings, and then average over the three NLI datasets to give each task equal weight. The overall average results are computed following a similar approach, giving each dataset an equal weight.

PET, our proposed Demo+Soft PET shows an improvement of average accuracy from 72.3 to 73.2 ($p < 0.05$ by paired t test) (Hsu and Lachenbruch, 2014). Moreover, both Demo+Soft and Vary Soft PET methods obtain better average performance than the Fixed Soft Tokens PET in NLI and topic classification tasks. These results show the importance of diversity introduced by multiple prompt labeler models.

4 Related Work

4.1 Language Model Prompting

Cui et al. (2021) authors fine-tuned the pre-trained generative language model, BART, with a predefined template (*candidate_span* is a *entity_type* entity) for NER classification. Wang et al. (2021) proposed Entailment as Few-shot Learner (EFT) method, which transforms classification tasks into natural language textual entailment tasks and then fine-tunes the LM. The transformation also makes it easy to leverage unsupervised contrastive data augmentation methods to add pairwise examples to the limited annotated data. This setting further showed an average 2.7% improvement in 15 different NLP tasks. In addition to using the prompts

for supervised learning, PET is the SoTA method to adapt the manual prompts along with semi-supervised learning to obtain strong performance across multiple NLU tasks. (Schick and Schütze, 2021).

4.2 Automatic Prompts and Verbalizers

Shin et al. (2020a) used a gradient-guided search to find the discrete tokens for prompts based on task accuracy, initialize tokens, and then fine-tune the LM. For automatic label token selection, they first train a logistic regression classifier from the contextualized embedding of the MASK token and then predict the score from MLM’s output word embeddings. They select the top-k highest scoring words for each label. They showed better performance over manual prompting methods for sentiment classification and textual entailment tasks. Similarly, instead of using a gradient-guided search for prompt tokens, Li and Liang (2021) and Lester et al. (2021) attached prefix vectors and learned the embeddings for prefix vectors by keeping the LM model parameters frozen. Liu et al. (2021c) proposed P-tuning, which replaces the input embeddings of pre-trained language models with its differentiable output embeddings, using the pat-

tern based on human design. Liu et al. (2021b) optimized and adapted the Prefix Tuning model for NLU. Vu et al. (2021) proposed to learn soft prompt embeddings from one or more source tasks and then transfer them to initialize the prompts for the target task. In addition, they also proposed an efficient retrieval approach to find task embeddings and predict the most transferable source tasks for a given novel target task.

Several automatic verbalizers, such as search-based verbalizers, soft verbalizers, and prototypical verbalizers, have been proposed to automate the design of the verbalizer mapping function. Search-based verbalizers aim to find the appropriate tokens to replace human selection (Schick et al., 2020a; Shin et al., 2020b; Gao et al., 2020). Both soft verbalizers and prototypical verbalizers learn trainable class or prototype embeddings during the training process (Cui et al., 2022; Zhang et al., 2021; Hambardzumyan et al., 2021).

Mahabadi et al. (2022) proposed a prompt-free method (PERFECT) to train the language model, which does not rely on manual commands and verbalizers. PERFECT reported performance similar to that of PET (Schick and Schütze, 2021) in a few-shot setting. However, they used a supervised learning setup and compared their results with the single labeler model with one prompt rather than the results from the final classifier. Here, we use a similar SSL setting to Schick and Schütze (2021) and report the results of the final classifier.

5 Conclusions

In this paper, we are able to successfully use automatic prompts and verbalizers in semi-supervised learning settings. We show that our proposed automatic prompt generation methods with prototypical verbalizer can eliminate human engineering in prompt-based SSL setup and achieve similar or better performance than the SoTA Manual PET method. Our methods have the added advantage of being scalable with multiple tasks and datasets. We also empirically verify the power of semi-supervised learning methods, which take advantage of large amounts of unlabeled data, over supervised methods.

In the next steps, we plan to investigate whether we would be able to achieve similar performance by freezing PLMs’ parameters and only tuning verbalizer and prompt parameters. This setup will save a tremendous amount of space by making it easy

to share and reuse PLMs. Moreover, we plan to explore ways to combine the two proposed methods Demo+Soft PET and Vary Soft PET, which would take advantage of both methods.

6 Limitations

Although we experiment with multiple NLU tasks and datasets, these datasets are only in the English language. Prompt-based learning relies on large language models, which have acquired knowledge through pre-training on huge corpora. With low-resource languages, it might be difficult to get PLMs trained on a huge corpus, which might make it hard to reproduce performance similar to the English corpus. The fine-tuning and inference of PLM requires multiple large GPUs, which might not be accessible to everyone.

Acknowledgments

We would like to thank the anonymous reviewers as well as Wei Ai, Paiheng Xu, Akram Almatarky, Jangwon Kim, Morteza Ziyadi, and Giannis Karanoulakis for reviewing the paper and for providing helpful comments and suggestions.

References

- Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239*.
- Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. [Prototypical verbalizer for prompt-based few-shot tuning](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7014–7024, Dublin, Ireland. Association for Computational Linguistics.
- Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. [Template-based named entity recognition using BART](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. [Openprompt: An open-source framework for prompt-learning](#). *arXiv preprint arXiv:2111.01998*.

- Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.
- Karen Hambardzumyan, Hrant Khachatryan, and Jonathan May. 2021. **WARP: Word-level Adversarial ReProgramming**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Henry Hsu and Peter A Lachenbruch. 2014. Paired t test. *Wiley StatsRef: statistics reference online*.
- Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. **The power of scale for parameter-efficient prompt tuning**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. **Prefix-tuning: Optimizing continuous prompts for generation**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.
- Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. **P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks**. *CoRR*, abs/2110.07602.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. Gpt understands, too. *arXiv:2103.10385*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. **Roberta: A robustly optimized bert pretraining approach**.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.
- Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Marzieh Saeidi, Lambert Mathias, Veselin Stoyanov, and Majid Yazdani. 2022. Perfect: Prompt-free and efficient few-shot learning with language models. *arXiv preprint arXiv:2204.01172*.
- Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. **The commitmentbank: Investigating projection in naturally occurring discourse**. *Proceedings of Sinn und Bedeutung*, 23(2):107–124.
- Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020a. **Automatically identifying words that can serve as labels for few-shot text classification**. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020b. Automatically identifying words that can serve as labels for few-shot text classification. *arXiv preprint arXiv:2010.13641*.
- Timo Schick and Hinrich Schütze. 2020. It’s not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.
- Timo Schick and Hinrich Schütze. 2021. **Exploiting cloze-questions for few-shot text classification and natural language inference**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020a. **AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts**. In *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing (EMNLP), pages 4222–4235, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020b. Auto-prompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2021. Spot: Better frozen model adaptation through soft prompt transfer. *arXiv preprint arXiv:2110.07904*.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner. *arXiv preprint arXiv:2104.14690*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Ningyu Zhang, Luoqi Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. *arXiv preprint arXiv:2108.13161*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015a. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Prompts and Verbalizers

A.1 Manual Prompts and Manual Verbalizers

We use the same manual prompts and manual verbalizers for our baseline experiment as used by Schick and Schütze (2021, 2020).

AG’s News is a news topic classification dataset with four classes. We use the manual verbalizer that maps class 1-4 to “World”, “Sports”, “Business” and “Technology”. For the input sentence $x = (a, b)$, where a is the news headline and b is the

body of the news text, we use the manual prompts below:

$$\mathcal{P}_1(x) = [\text{MASK}] : [a] [b]$$

$$\mathcal{P}_2(x) = [\text{MASK}] - [a] [b]$$

$$\mathcal{P}_3(x) = [a] ([\text{MASK}]) [b]$$

$$\mathcal{P}_4(x) = [a] [b] ([\text{MASK}])$$

$$\mathcal{P}_5(x) = [\text{MASK}] \text{ News: } [a] [b]$$

$$\mathcal{P}_6(x) = \text{Category : } [\text{MASK}] [a] [b]$$

Yahoo Questions is another dataset for topic classification with ten classes. We use the same manual prompts as AG’s News, but define the manual verbalizer for the Yahoo dataset, which maps the classes 1-10 to “Society”, “Science”, “Health”, “Education”, “Computer”, “Sports”, “Business”, “Entertainment”, “Relationship” and “Politics”.

MNLI is the dataset for textual entailment tasks, consisting of text pairs $x = (a, b)$. We define two manual verbalizer pairs v_1 and v_2 . v_1 verbalizer maps class 0-2 to “Wrong”, “Right” and “Maybe”. v_2 verbalizer maps class 0-2 to “No”, “Yes”, “Maybe”. We use the following manual prompts:

$$\mathcal{P}_1(x) = “[a]” ? || [\text{MASK}], “[b]”$$

$$\mathcal{P}_2(x) = [a] ? || [\text{MASK}], [b]$$

RTE and **CB** are datasets for textual entailment tasks. We use v_1 as the manual verbalizer similar to MNLI task. We use the following manual prompts:

$$\mathcal{P}_1(x) = “[a]” ? || [\text{MASK}], “[b]”$$

$$\mathcal{P}_2(x) = [a] ? || [\text{MASK}], [b]$$

$$\mathcal{P}_3(x) = [a] ? || [\text{MASK}]. [b]$$

$$\mathcal{P}_4(x) = “[a]” ? || [\text{MASK}]. “[b]”$$

A.2 Continuous Prompts

For our proposed models: **Demo+Soft** and **Vary Soft** models, we apply continuous prompts and automatic verbalizers to ensure that the prompt-tuning SSL method can be scaled across multiple datasets. From previous works, we find that few anchor tokens help to improve the performance of NLU tasks (Liu et al., 2021c), so we design two different continuous prompts dependant on the nature of NLU tasks. For the continuous prompt for AG’s News and Yahoo Questions (text classification task), our design is:

$$\mathcal{P}(x) = [a] [b] \text{ Category: } [p_0, p_1, \dots, p_n] [\text{MASK}]$$

For continuous prompt for MNLI, CB and RTE (NLI tasks), our design is:

$\mathcal{P}(x) = [a] [b] ? [p_0, p_1, \dots, p_n]$ answer : [MASK]

The construction of continuous prompts also follow the design of the P-tuning paper (Liu et al., 2021c). Rather than designing multiple manual prompts for different datasets, we can use our proposed methods to automate this process. This reduces human efforts and costs when we scale across multiple datasets and tasks.

Novel Feature Discovery for Task-Oriented Dialog Systems

Vinh Thinh Ho
Amazon Alexa AI
Berlin, Germany
hovnh@amazon.com

Mohamed Soliman
Amazon Alexa AI
Aachen, Germany
mohsol@amazon.com

Abdalghani Abujabal
Amazon Alexa AI
Aachen, Germany
abujabaa@amazon.com

Abstract

A novel feature represents a cluster of semantically equivalent novel user requests e.g., requests to play a song on a service or reading user’s messages. Detecting and supporting novel features is crucial towards wider adoption of dialog systems by end users. Intuitively, features are represented by a combination of intents, slot types and/or their values. For example, while playing a song is a feature represented by a single intent (`PlayMusic`) only, playing a song on a service is another feature represented by the combination of `PlayMusic` intent and `ServiceName` slot type. Prior work on novelty detection limits the scope of features to those represented by novel single intents, leading to (1) giant clusters spanning several user-perceived fine-grained features belonging to the same intent, (2) incoherent interpretation of clusters from users’ perspective (no direct connection to some user-perceived feature), and (3) missing those features spanning several intents. In this work, we introduce *feature discovery* as opposed to single intent discovery, which aims at discovering novel features spanning a combination of intents and slots, and present a technique for discovering novel features from user utterances. Experiments on two datasets demonstrate the effectiveness of our approach and consistently show its ability to detect novel features.

1 Introduction

Advances in Natural Language Understanding (NLU) have led to accelerated adoption of dialog systems such as Apple Siri and Amazon Alexa by end users. Standing at the core of a dialog system is an NLU model for parsing and understanding user utterances. Two of the key tasks of an NLU model are (1) Intent Classification, which classifies an utterance into a fixed set of intent labels, and (2) Slot Labeling, which classifies slot values into a predefined set of slot types (Weld et al., 2023). Determining the intent guides the dialog system

to perform the proper actions as response to user’s utterance. For example, user utterances “*play some music*” and “*play despacito*” express the intent `PlayMusic`, while “*how is the weather?*” and “*is it raining today*” express the intent `GetWeather`. Intents can be further grouped into domains, for instance, `PlayMusic` and `RateSong` intents belong to `Music` domain. Detecting slots and their corresponding values within an utterance gives information about objects upon which the actions should be performed. For example, ‘*despacito*’ in “*play despacito*” is of type `SongName`.

A feature represents a user *experience* with the dialog system, for example, playing a song on a service or reading user’s messages. Over time, users build up expectations about the features/experiences that the dialog system offers. Unsupported features cause friction and degrade user’s experience. In terms of NLU, a novel feature could be mapped to a new combination of domain(s), intent(s), slot(s) and/or their values, where each is not necessarily novel. For example, while `PlayMusic` intent was seen by the dialog system, the combination of `PlayMusic` and `ServiceName` is never seen before, causing friction with the NLU model when parsing utterances like “*play despacito on spotify*”. Consequently, it becomes crucial to discover such features that are frequently requested by the users but are still unsupported by the NLU model, which we address in this work.

The task of novel intent discovery has been intensively studied in prior work, by harnessing unsupervised techniques (e.g., Liu et al., 2021) or semi-supervised methods for incorporating existing knowledge from labeled data (e.g., Vedula et al., 2020a; Lin et al., 2020; Zhang et al., 2021). Existing work limits the scope of novel features to those represented by novel single intents. However, this does not cover all types of features that naturally span several domains, intents, slots and their values. Consider the following user utterances, “*play*

despacito”, “*play despacito on spotify*” and “*play despacito in 30 minutes*”. While the utterances belong to the same intent `PlayMusic`, handling each of them is different. The first requires the dialog system to play a song on the device itself, the second asks for playing the same song on a specific service, namely Spotify, while the third requires playing the song after a certain amount of time is elapsed. Moreover, each request corresponds to a different user experience and requires a different underlying implementation for responding. Assuming `PlayMusic` intent is novel, applying standard novel intent discovery will group the three utterances in a single cluster, which creates two issues: (1) having many different user-perceived experiences within the same intent (play song on device, on specific service or after some time), results in a giant cluster that needs manual inspection to be decomposed into smaller sub-groups to make meaningful business decisions, and (2) while the cluster represents a novel intent, it might not correspond to a user perceived experience – what the end users are looking for. On the other hand, assuming `PlayMusic` intent is not novel, intent discovery might miss those features at more fine-grained levels. For instance, playing a song on a service might not be detected as novel. Additionally, intent discovery cannot handle those features spanning several intents.

To allow for general feature discovery, and close the gap between user requests and the underlying models, we define a *feature* to be any combination of domain(s), intent(s) and slot(s) and/or their value(s) and move towards discovering clusters of utterances with novel feature definitions rather than only focusing on novel single intents. Novel intent discovery can be seen as a special case of feature discovery, where new features correspond to new intents unseen before.

In this paper, we present DNF (Discovery of Novel Features), a semi-supervised approach for discovering novel features from a given set of user utterances with respect to an underlying NLU model. Our method consists of a cascaded system with two steps: feature clustering and novelty detection. First, we employ state-of-the-art language model BERT (Devlin et al., 2019) with multi-stage fine-tuning to produce feature/experience-aware representations of user utterances. Then, user utterances are clustered into features. Second, we classify each resulting feature cluster as either novel or

already supported by the NLU model. The salient contributions of our paper are:

- We introduce *Feature Discovery*, where, given a set of user utterances and a trained natural language understanding model, we extract clusters of novel features.
- We present DNF, an approach for discovering novel features from user utterances.
- We conducted extensive experiments on two datasets, the SNIPS dataset augmented with feature labels, and our internal real-world dataset. Experimental results demonstrate the effectiveness of our method across the two datasets.

2 Related Work

Intent classification and slot labeling are two fundamental tasks in spoken language understanding, dating back to early 90’s (Price, 1990). With the rise of task-oriented dialog systems, the two tasks have seen more attention, and progress has been made by applying various deep learning approaches (e.g., Abujabal and Gaspers, 2019; Abujabal et al., 2021; Goo et al., 2018; Jolly et al., 2020; Mesnil et al., 2013).

Discovering novel domains and intents from a pool of user utterances has been well addressed in earlier works, with fully unsupervised and semi-supervised settings. These include clustering user utterances with novel domains or intents individually, using various techniques such as constrained deep adaptive clustering (Lin et al., 2020), deep aligned clustering (Zhang et al., 2021), contrastive learning (Gao et al., 2021), capsule network (Liu et al., 2019; Xia et al., 2018), open intent extraction (Vedula et al., 2020b), and others (e.g., Lin and Xu, 2019; Shivakumar et al., 2019; Yan et al., 2020; Kim and Kim, 2018). Alternatively, Vedula et al., 2020a explore the joint discovery of domains and intents, using hierarchical linking to form an intent-domain taxonomy. The task is also performed jointly with slot filling in recent works (Wang et al., 2018; Goo et al., 2018; Kim et al., 2017; Castellucci et al., 2019; Gangadharaiah and Narayanaswamy, 2019; Liu and Lane, 2016). All of the above works require a small amount of labeled data as prior knowledge to guide the discovery process.

The most prominent work for detecting novel intents without any prior knowledge was proposed by Liu et al. (2021), which employs a pre-trained network for generating sentence embeddings, and

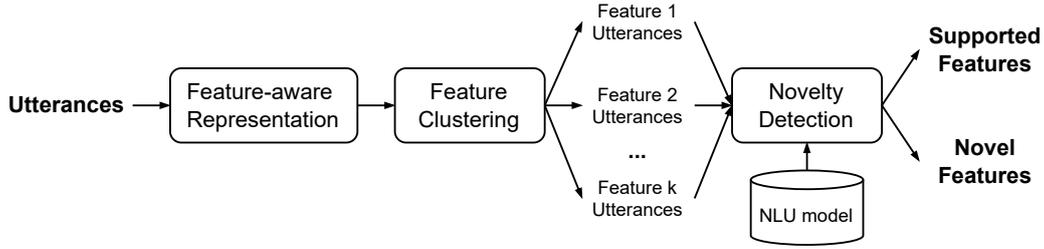


Figure 1: DNF overview with feature-aware fine-tuning, feature clustering and novelty detection.

K-Means for intent clustering. However, due to the limited supervision, the method is shown to perform poorly on novel domains.

All of the above methods are limited to a discovery objective at intent level, and thus fail to operate on more fine-grained levels (e.g., slot type and/or value) within the same intent. Moreover, they fail to detect features composed of multiple domains, intents and/or slots.

3 Methodology

Given a set of user utterances $\{u_1, u_2, \dots, u_n\}$, we aim to detect a set of clusters $\{C_1, \dots, C_m\}$ where each C_j is a cluster of utterances pertaining to a novel feature and m is the total number of novel features. As depicted in Figure 1, our method consists of two components, feature clustering and novelty detection. First, we assign each utterance u_i a feature label and eventually produce a set of feature-labeled utterances $(u_1, f_1), \dots, (u_n, f_k)$, where f_1, f_2, \dots, f_k is the list of k unique features. This is performed by employing an utterance representation model specifically trained with both feature-labeled and -unlabeled data to project the input utterances into a feature-aware vector space that helps clustering the input utterances into k features. Second, a feature novelty detection model is used to classify each of the k feature clusters as either novel or already supported by the dialog system. We measure the novelty of a feature by exploiting signals from the NLU model. With DNF being a semi-supervised technique, we distinguish between two types of training data supervision: *feature-labeled* and *feature-unlabeled* utterances:

- Feature-labeled training data ($Train_L$): each utterance is annotated with its feature label f along with its intent label I and slot labels $S = \{s_1, s_2, \dots\}$ where each s_i is a pair of slot type and its value in the utterance such as `SongName:despacito` and `ServiceName:spotify`.

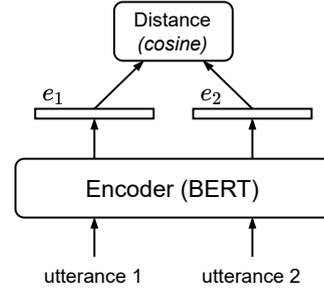


Figure 2: Fine-tuning with utterance similarity.

- Feature-unlabeled training data ($Train_U$): each utterance is only annotated with its intent and slot labels, however, with no feature label. In comparison to feature-labeled data, such data can be obtained in larger quantities and helps the feature discovery process as we exploit information about utterances’ intent and slots.

3.1 Feature-Aware Utterance Representation

To first encode utterances as high-dimensional vectors separable in the feature space, we use a representation model specifically adapted for feature awareness. To this end, we employ the state-of-the-art language model SBERT – Sentence BERT (Reimers and Gurevych, 2019), which is a BERT model pretrained for generating sentence embeddings. Specifically, by feeding an utterance u into SBERT, we get a list of token embeddings $[CLS, t_1, t_2, \dots, t_m]$, where CLS is the classification token. By applying mean-pooling, we obtain the representation vector of u :

$$e_u = \text{mean-pooling}([CLS, t_1, t_2, \dots, t_m])$$

We transfer feature knowledge into the utterance representation model through a multi-stage fine-tuning process as described below.

3.1.1 Utterance Similarity

As depicted in Figure 2, we use a Siamese Neural Network (SNN) training paradigm for transferring feature knowledge into the model. The intuition

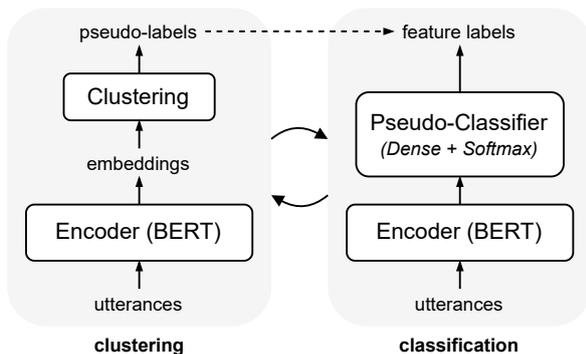


Figure 3: Fine-tuning with pseudo-classification.

behind this fine-tuning step is to directly optimize the utterance representations based on their feature similarity. In particular, a pair of utterances (u_i, u_j) is encoded into its respective embedding vectors e_i and e_j . The utterance feature similarity is computed as the cosine distance between the two vectors, and is optimized towards minimizing the distance between utterances belonging to the same feature. As our training data are both feature-labeled ($Train_L$) and feature-unlabeled ($Train_U$), we consider three kinds of training samples for this fine-tuning step:

1. Both u_i and u_j are sampled from the same $Train_L$ feature: (u_i, u_j) is a positive sample.
2. Both u_i and u_j are sampled from $Train_L$, but are from different features: (u_i, u_j) is a negative sample.
3. u_i is sampled from $Train_L$ and u_j is sampled from $Train_U$: (u_i, u_j) is a negative sample.

For case 3, it is possible that the training sample is a false-negative, since the actual feature-label of u_j could be the same as of u_i . However, we hypothesize that the number of such false-negative utterance pairs is much lower than the true-negative pairs given the large size of this training set.

As the number of utterance pairs is quadratic, we use a simple way to build a random dataset for each training epoch. For each utterance u_i in $Train_L$, we randomly sample an utterance u_j that belongs to the same feature as a positive sample, and k negative samples from both $Train_L$ and $Train_U$; hence maintaining the ratio of $1 : k$ between the number of positive and negative pairs. Empirically, we found that setting $k = 3$ provides a decent balance between positive and negative pairs.

3.1.2 Pseudo Classification

Utterance similarity considers pairwise distances between utterances, without setting global constraints across all utterances. Moreover, it optimizes the distances according to the feature-labeled clusters in $Train_L$ but does not consider the unknown feature clusters in $Train_U$. The second fine-tuning step, shown in Figure 3, aims to overcome the above issues. Inspired by the DeepCluster work (Caron et al., 2018), pseudo classification is a semi-supervised iterative training process, alternating between *clustering* and *classification*.

In the *clustering* step, we first encode the training utterances into their representation vectors. Then a clustering algorithm is used to group the representation vectors into clusters while assigning a pseudo-label to each cluster. We use COP-K-Means (Wagstaff et al., 2001) as the clustering algorithm – an extension of K-Means that allows putting constraints on the clustering process. For example, which utterances must be grouped in the same clusters, and which must not. In our case, utterances belong to the same feature cluster inside $Train_L$ must be grouped in the same candidate cluster given by COP-K-Means.

In the *classification* step, we fine-tune the BERT encoder by employing a feature classification task using the pseudo-labels generated from the clustering step as the ground truth feature labels. The pseudo-classifier consists of a dense layer followed by a softmax on top of the encoder. In the DeepCluster approach, the pseudo-classifier is reinitialized after each iteration, since the indices of the pseudo-labels are permuted randomly after each clustering step. This makes training slow as the parameters cannot be reused. To alleviate this issue, we adopt the cluster centroid alignment technique proposed by Zhang et al. (2021), where we re-assign the pseudo-label indices from the clustering step, and thus, aligning them with the pseudo-classifier trained from the previous iteration. This allows the parameters to be reusable across iterations, hence speeding up training.

Compared to utterance similarity, pseudo classification clusters $Train_U$ utterances, either into $Train_L$ clusters or into totally new ones. Figure 4 compares the expected effect of the two steps.

3.1.3 Slot Classification

Information about slots in user utterances is a good source for feature awareness. For instance, while “play *despacito*” and “play *despacito* on *spotify*” are

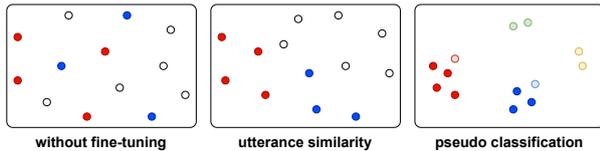


Figure 4: Utterance similarity vs. pseudo classification. Solid red and blue dots are $Train_L$ utterances, while green and yellow dots are $Train_U$ utterances grouped in new clusters.

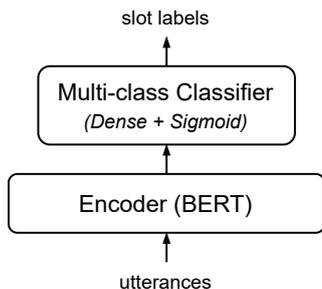


Figure 5: Fine-tuning with slot classification.

semantically similar utterances, the existence of `ServiceName:spotify` slot hints at the existence of a fine-grained user feature of playing a song on a service. In contrast to utterance similarity and pseudo classification, slot classification does not directly pull/push the utterances close to or far away from each other in the embedding space. Instead, we aim to make the representation model aware of the presence of slots in the input utterances, and we hypothesize that such information guides the model towards producing utterance representation with better feature separability.

This is modeled as a multi-class classification task (Figure 5), where the model is trained to detect which slots appear in the input utterances (e.g., `ServiceName`). Concretely, we add on top of the encoder a multi-class classifier, comprising a dense layer followed by sigmoid activation, and fine-tune the model using binary-cross-entropy loss. Note that we do not consider the exact position of the slots in the utterances, but rather their presence.

3.2 Feature Clustering

At inference time, we encode user unlabeled utterances into their representation vectors and use K-Means to cluster them into k feature clusters. To automatically choose the optimal value of k , we employ two techniques, namely the Elbow method (Thorndike, 1953) and the Silhouette score (Rousseeuw, 1987). Since utterances are a mix of novel and supported features, we classify whether

each candidate feature cluster is novel using a feature novelty detection model, described next.

3.3 Feature Novelty Detection

While the trained representation model helps at projecting the utterances into a feature-separable space, it lacks information regarding feature’s novelty w.r.t the NLU model. We detect novel feature clusters by exploiting signals from the NLU and utterance representation models. The NLU model is trained to jointly recognize intent and slot labels. The intent classifier (IC) and slot tagger (ST) heads are plugged on top of a BERT encoder. Slot labels follow the BIO schema (Ramshaw and Marcus, 1995). Note that our approach is agnostic to the choice of the NLU model. For novelty detection, we define the novelty confidence of each candidate feature cluster $C = \{u_1, u_2, \dots\}$ as the average novelty of its utterances:

$$feat_{novel}(C) = \frac{1}{|C|} \sum_{u \in C} utt_{novel}(u)$$

where utt_{novel} is computed as follows:

$$utt_{novel}(u) = mean(st(u), ic(u), pc(u))$$

where *slot tagging confidence* (st) is the confidence produced from the slot tagger. We tested with two variants: average over tokens (st_{avg}), and minimum over tokens (st_{min}). *Intent classification confidence* (ic) is the confidence of the most probable intent label produced by the intent classifier. *Pseudo classification confidence* (pc) is the confidence of the most probable pseudo cluster, produced by the pseudo-classifier from the utterance representation model. Feature clusters with novelty score greater than a pre-defined threshold are labeled as novel.

4 Experimental Setup

We evaluate the performance of each component of DNF independently as well as the overall discovery system using two datasets: SNIPS and a large French internal dataset.

4.1 Datasets

The SNIPS dataset (Coucke et al., 2018) consists of 14K English utterances spanning 7 intents with 72 unique slot labels. Since the dataset does not contain any feature labels, we augment it with feature labels using hand-crafted rules. We end up with a total of 43 features, each of which forms a cluster of utterances that semantically map to a

user-perceived feature. Features cover three combinations:

- **Slot value features:** utterances in these features share the same intent, slots and at least one slot value. For example, “*play a song on spotify*” and “*play music on spotify*” belong to the same feature of *playing a song on Spotify service*.
- **Slot features:** utterances in these features share the same intent and same slots. For example, “*what is the weather tomorrow in new york*” and “*weather tonight in brooklyn*” belong to the same feature of *asking about weather at a specific time in a specific city*.
- **Intent features:** utterances in these features share the same intent, with potentially, several slots and/or slot values. For example, “*book me a restaurant tomorrow at 9pm*” belong to the feature of *booking a restaurant*.

Given the nature of the dataset, it was not possible to create cross-intent features that are semantically sensible. Out of the 43 features, 11 are used as a test set, while 32 as a training set, where 19 out of the 32 features are feature-labeled $Train_L$, and 13 features are feature-unlabeled training set $Train_U$. These 32 features are deemed supported by the NLU model, i.e., not novel. Out of the 43 features, 26 are slot value features, 15 are slot features, while 2 are intent features. We randomly sample utterances out of 32 features and add them to the test set so that our test set contains a mix of supported and novel features. The final dataset contains 32 features as training set and 43 features in the test set (11 of which are novel). On average, we have 210 utterances per feature cluster.

Our internal dataset has a total of 273K utterances comprising 41 features with different combinations. 15 out of the 41 features span single intent while the other features span two or more intents. 31 features are included in the feature-labeled training set $Train_L$. The number of utterances without feature labels in $Train_U$ set is in the order of millions. The remaining 10 features are part of the test set. We also sample utterances out of 31 features and add them to the test set. The final test set contains 41 features (10 of which are novel). The dataset covers 14 domains, 89 intents, 128 slot labels and has, on average, 6.6K utterances per feature cluster. All utterances were pre-processed such that users are not identified.

Table 1: Feature-aware utterance representation performance on SNIPS and internal datasets.

Training Strategy	SNIPS			Internal Dataset		
	NMI	ARI	ACC	NMI	ARI	ACC
<i>No fine-tuning</i>	0.626	0.309	0.396	==== baseline ====		
<i>US only</i>	0.737	0.451	0.512	+0.096	+0.147	+0.149
<i>PC only</i>	0.728	0.374	0.471	+0.116	+0.161	+0.130
<i>US→PC</i>	0.749	0.475	0.537	+0.116	+0.178	+0.166
<i>SC→US→PC</i>	0.766	0.474	0.557	+0.104	+0.166	+0.129
<i>(SC+US)→PC</i>	0.782	0.531	0.605	+0.137	+0.216	+0.140

To assess the ability of our model to accurately cluster cross-domain features, we split the internal dataset into:

- **Single-domain features**, where utterances belong to the same domain. 32 out of the 41 features are single domain features (25 train and 7 test), and
- **Cross-domain features**, where utterances belong to multiple domains (e.g., `Music` and `SmartHome`). 9 out of the 41 features are cross-domain features (6 train and 3 test). Features in these two splits cover different combinations.

4.2 Feature-Aware Utterance Representation

We use SBERT as a baseline utterance representation model, and compare different variants fine-tuned on different feature-related tasks. We consider the following fine-tuning strategies:

- *No fine-tuning*: We directly use the SBERT base model for utterance representation.
- *Utterance Similarity (US)*: The model is fine-tuned with only utterance similarity.
- *Pseudo Classification (PC)*: The model is fine-tuned with only pseudo classification.
- *US→PC*: The model is fine-tuned with both utterance similarity and pseudo classification sequentially.
- *Slot Classification (SC)→US→PC*: The model is fine-tuned with slot classification, utterance similarity and pseudo classification sequentially.
- *(SC+US)→PC*: Slot classification and utterance similarity are jointly trained to prevent overfitting, and then pseudo classification.

We use `paraphrase-mpnet-base-v2`, a pre-trained SBERT model, where we unfreeze all the layers during fine-tuning. We use AdamW as our optimizer (Loshchilov and Hutter, 2017), with an initial learning rate of $5e^{-5}$ and a weight decay

Table 2: Performance across different feature types.

Feature Type		SNIPS			Internal Dataset		
		NMI	ARI	ACC	NMI	ARI	ACC
By Novelty	<i>supported</i>	0.836	0.698	0.746	0.949	0.870	0.849
	<i>novel</i>	0.741	0.566	0.637	0.772	0.640	0.662
By Combination	<i>slot</i>	0.753	0.610	0.626	–	–	–
	<i>slot value</i>	0.811	0.590	0.646	–	–	–
	<i>intent</i>	0.547	0.512	0.804	–	–	–

Table 3: Intent-agnostic vs. intent-targeted discovery.

Intent	Intent-agnostic			Intent-targeted		
	NMI	ARI	ACC	NMI	ARI	ACC
<i>AddToPlaylist</i>	0.544	0.402	0.595	0.671	0.573	0.789
<i>PlayMusic</i>	0.700	0.429	0.527	0.747	0.625	0.672
<i>SearchCreat.Work</i>	0.631	0.431	0.598	0.766	0.602	0.707
<i>SearchScrn.Event</i>	0.639	0.474	0.634	0.740	0.541	0.678

of 0.01. We set the batch size to 16 and apply early stopping whenever we observe no improvement on a development set. On average, all fine-tuning strategies converge after 4 epochs.

Evaluation Metrics. To evaluate clustering quality against ground-truth, we follow previous work and report on the following metrics: Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and clustering Accuracy (ACC). All metrics range from 0 to 1. The higher the score, the better the clustering quality. We report the relative gains/losses over the baseline for the internal dataset. Since we are only interested in evaluating the quality of the fine-tuned representations, we use the reference number of clusters k in subsequent experiments and run a separate experiment to find the optimal k when evaluating the end-to-end system.

5 Experimental Results

We evaluate (1) the effect of our fine-tuning strategies to produce feature-aware representation on feature clustering, and (2) our feature novelty detection model.

5.1 Fine-tuning Results

Table 1 shows clustering performance using different fine-tuning strategies. Across datasets, our fine-tuning strategies outperform the vanilla SBERT baseline across all metrics. Stacking different fine-tuning tasks consistently results in better models. Fine-tuning with slot classification individually either before or after the other tasks (e.g., $SC \rightarrow US \rightarrow PC$) yields inferior performance compared to jointly running slot classification with other tasks. The best performing strategy is fine-tuning with slot classification and utterance similar-

Table 4: Ablation study results.

Model	SNIPS			Internal Dataset		
	NMI	ARI	ACC	NMI	ARI	ACC
<i>Standard</i>	0.782	0.531	0.605	====	<i>baseline</i>	====
<i>Ablation</i>	0.769	0.444	0.523	-0.081	-0.132	-0.087
<i>Upperbound</i>	0.812	0.530	0.672	–	–	–

Table 5: Choosing the number of clusters k .

Method	k	SNIPS			k	Internal Dataset		
		NMI	ARI	ACC		NMI	ARI	ACC
<i>Gold k</i>	43	0.782	0.531	0.605	41	====	<i>baseline</i>	====
<i>Silhouet.</i>	24	0.761	0.511	0.593	35	-0.030	-0.063	-0.013
<i>Elbow</i>	30	0.790	0.573	0.632	39	+0.002	+0.003	+0.020

ity jointly and then running pseudo classification. This shows that slot classification is able to improve utterance representations in the feature space. In all subsequent experiments, we use the best observed model ($SC+US$) \rightarrow PC .

Performance breakdown across feature types.

In Table 2, we report the performance of the best representation model on different feature types. First, we split the features in the test set by their novelty (whether the feature has been seen during training). Across the two datasets, our model clusters utterances from supported features better than from novel ones, which is expected.

In the second half of the table, we show a breakdown per combination. On SNIPS dataset, we achieve better clustering results for slot and slot value features than for intent features, which is reasonable as our training data contains only 2 features at intent level. On the internal dataset, clustering single-domain features is slightly better than cross-domain ones. For example, the model achieves an NMI of 0.867 for single domain features, and 0.823 NMI for cross-domain features.

Intent-agnostic vs. Intent-targeted discovery.

For deeper analysis, we consider another discovery setup in an intent-targeted way, in which we only train and test with features from the same intent. This setup is particularly useful in cases where we focus on fine-grained discovery where the intent is assumed to be known. In Table 3, we report the results for this study on four SNIPS intents separately. The models trained with features belonging to the same intent perform generally better than when being trained with cross-intent features.

Ablation study. During fine-tuning, we leverage both feature-labeled ($Train_L$) and feature-unlabeled ($Train_U$) data. To understand their im-

Table 6: Performance of the feature novelty detection.

Signal	SNIPS			Internal Dataset		
	Prec.	Rec.	F1	Prec.	Rec.	F1
st_{avg}	0.407	1.000	0.579	==== baseline ====		
st_{min}	0.750	0.545	0.632	+0.076	0.000	+0.043
ic	0.500	0.636	0.560	-0.083	-0.100	-0.091
pc	0.615	0.727	0.667	+0.167	-0.200	-0.020
$feat_{novel}$	0.750	0.545	0.632	+0.167	0.000	+0.091

impact on model performance, we compare our model, trained with both kinds of data (standard model), against a model trained with only $Train_L$ data (ablation model). Moreover, we compare both models to an *upperbound* model, in which we also include the true feature labels of $Train_U$. Hence, the upperbound model is also trained with feature-labeled data only, similar to the ablation model, but with more features. We did not build an upperbound model on the internal dataset since all utterances in $Train_U$ are not annotated with feature labels.

As shown in Table 4, the standard model outperforms the ablation model on all metrics across both datasets, with NMI and ACC gains reaching 8-9%, and ARI gains of 13% on the internal dataset. This shows that our model benefits from feature-unlabeled data during training. Naturally, an abundance of feature-labeled data, although impractical, provides better feature-aware representations.

Choosing the number of feature clusters. As we are interested in evaluating the quality of the fine-tuned representations, we use the number of ground-truth features k from test data. However, this number is unknown in practice. We experimented with two popular techniques for predicting k : the Elbow method (Thorndike, 1953) and the Silhouette score (Rousseeuw, 1987). As shown in Table 5, on both datasets, Elbow method works slightly better than Silhouette score, with the predicted k closer to the ground-truth value. In terms of other metrics, Elbow method even produces better feature clusters than the baseline with gold k .

5.2 Results of Feature Novelty Detection

The base NLU model is a pre-trained BERT with 12 hidden layers, each with a size of 768. On top of the CLS classification token, we plug a linear projection head followed by softmax to perform intent classification, and a similar head on top of each token to perform slot tagging. To train the NLU model, we unfreeze all 12 hidden layers and fine-tune the two heads jointly.

To evaluate novelty detection in isolation from

clustering, we use the ground-truth feature clusters and run novelty detection on top, i.e., to decide whether a ground-truth feature cluster is *novel* or *supported*. We harness different signals from the NLU model: st_{avg} , st_{min} , ic , pc and the combined signal $feat_{novel}$. As shown in Table 6, the combined signal $feat_{novel}$ performs the best across all metrics on the internal dataset. On SNIPS, $feat_{novel}$ excels against other signals in terms of precision, and places the second best in F1.

5.3 End-to-end DNF Evaluation

In this experiment, we first ran the clustering step with Elbow method to generate feature clusters. This resulted in 30 predicted clusters on SNIPS, and 39 clusters on the internal dataset (see Table 5). Then, we perform novelty detection on the predicted clusters.

To generate ground-truth labels for the predicted clusters, we assign a feature label l_C for each predicted cluster C by taking the majority vote of the labels of the utterances within the cluster. Table 7 shows the number of novel clusters predicted using each signal and their quality metrics. *all* is the baseline, where we assume all predicted feature clusters as novel. $feat_{novel}$ signal performs the best in terms of precision on SNIPS, together with st_{min} , which shows that the NLU slot confidence is a strong indicator of the novelty of an utterance. On the internal dataset, $feat_{novel}$ also achieves highest precision and F1, while discovering almost all novel features in the data (9 out of 10).

6 Conclusion

We introduced *feature discovery*, where, given a set of user utterances and a trained NLU model, we extract clusters of novel features composed of a combination of domains, intents, slots and/or their values. To this end, we presented DNF, a semi-supervised approach for extracting novel user features from a set of raw utterances, utilizing minimal feature knowledge from labeled data combined with feature-unlabeled data. DNF supports several fine-tuning strategies to improve utterance representation and make them separable in the feature space. We evaluated DNF on two datasets and observed significant improvements over baselines, showing the effectiveness of our method. In the future, we plan to explore various fine-tuning strategies for better utterance representations, as well as extending DNF to support different languages.

Table 7: End-to-end DNF evaluation.

Method	SNIPS				Internal Dataset			
	<i>#novel features</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>	<i>#novel features</i>	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
<i>all</i>	30	0.233	0.636	0.341	39	==== <i>baseline</i> ====		
<i>st_{avg}</i>	23	0.238	0.455	0.312	16	+0.295	0.000	+0.288
<i>st_{min}</i>	5	0.600	0.273	0.375	12	+0.462	0.000	+0.400
<i>ic</i>	21	0.238	0.455	0.312	8	+0.545	-0.200	+0.340
<i>pc</i>	10	0.400	0.364	0.381	9	+0.462	-0.200	+0.305
<i>feat_{novel}</i>	5	0.600	0.273	0.375	9	+0.573	-0.100	+0.410

7 Limitations

While we empirically showed that our approach performs well for novel feature discovery and generalizes across different datasets, we can identify avenues for improvement in terms of efficiency and model training. With DNF relying on good feature-aware sentence representations, obtaining such representations requires expensive fine-tuning steps. For example, using a single GPU, our cascaded fine-tuning strategy takes on average 7 hours on our internal dataset to reach convergence. Moreover, in workflows where NLU model refresh is frequent, the model’s intent classification and slot tagging confidence distribution can shift over time. With DNF relying on observing confidence signals from the NLU model to determine feature novelty, a retraining/tuning of the novelty detection component would have to be performed frequently. Furthermore, our approach requires a small manually annotated feature-labeled dataset (feature labels in addition to intent and slot labels). These additional annotations require expertise and time, which poses a challenge during the data collection phase.

References

- Abdalghani Abujabal, Claudio Delli Bovi, Sungho Ryu, Turan Gojayev, Fabian Triefenbach, and Yannick Versley. 2021. *Continuous model improvement for language understanding with machine translation*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 56–62. Association for Computational Linguistics.
- Abdalghani Abujabal and Judith Gaspers. 2019. Neural named entity recognition from subword units. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*.
- Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*.
- Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual intent detection and slot filling in a joint bert-based model. *CoRR*.
- Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*.
- Rashmi Gangadharaiyah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*.
- Xibin Gao, Radhika Arava, Qian Hu, Thahir Mohamed, Wei Xiao, Zheng Gao, and Mohamed AbdelHady. 2021. Graphire: Novel intent discovery with pretraining on prior knowledge using contrastive learning. In *KDD 2021 Workshop on Pretraining: Algorithms, Architectures, and Applications*.
- Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

- Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*.
- Shailza Jolly, Tobias Falke, Caglar Tirkaz, and Daniil Sorokin. 2020. Data-efficient paraphrase generation to bootstrap intent classification and slot labeling for new features in task-oriented dialog systems. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020 - Industry Track, Online, December 12, 2020*.
- Joo-Kyung Kim and Young-Bum Kim. 2018. Joint learning of domain classification and out-of-domain detection with dynamic class weighting for satisfying false acceptance rates. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*.
- Young-Bum Kim, Sungjin Lee, and Karl Stratos. 2017. ONENET: joint domain, intent, slot prediction for spoken language understanding. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*.
- Ting-En Lin and Hua Xu. 2019. Deep unknown intent detection with margin loss. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*.
- Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*.
- Bing Liu and Ian R. Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*.
- Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert Y. S. Lam. 2019. Reconstructing capsule networks for zero-shot intent classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*.
- Pengfei Liu, Youzhang Ning, King Keung Wu, Kun Li, and Helen Meng. 2021. Open intent discovery through unsupervised semantic clustering and dependency parsing. *CoRR*.
- Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*.
- Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*.
- Patti J. Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, USA, June 24-27, 1990*.
- Lance A. Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora, VLC@ACL 1995, Cambridge, Massachusetts, USA, June 30, 1995*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*.
- Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*.
- Prashanth Gurunath Shivakumar, Mu Yang, and Panayiotis G. Georgiou. 2019. Spoken language intent detection using confusion2vec. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*.
- Robert L Thorndike. 1953. Who belongs in the family? *Psychometrika*.
- Nikhita Vedula, Rahul Gupta, Aman Alok, and Mukund Sridhar. 2020a. Automatic discovery of novel intents & domains from text utterances. *CoRR*.
- Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. 2020b. Open intent extraction from natural language interactions. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*.
- Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*.
- Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*.

- Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2023. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Comput. Surv.*, 55(8):156:1–156:38.
- Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*.
- Guangfeng Yan, Lu Fan, Qimai Li, Han Liu, Xiaotong Zhang, Xiao-Ming Wu, and Albert Y. S. Lam. 2020. Unknown intent detection using gaussian mixture model with an application to zero-shot intent classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*.
- Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021. Discovering new intents with deep aligned clustering. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*.

Context Generation Improves Open Domain Question Answering

Dan Su^{‡,*}, Mostofa Patwary[§], Shrimai Prabhunoye[§], Peng Xu[§], Ryan Prenger[§],
Mohammad Shoeybi[§], Pascale Fung[‡], Anima Anandkumar[§], Bryan Catanzaro[§]

[‡]The Hong Kong University of Science and Technology, [§]NVIDIA

dsu@connect.ust.hk, mpatwary@nvidia.com

Abstract

Closed-book question answering (QA) requires a model to directly answer an open-domain question without access to any external knowledge. Prior work on *closed-book* QA either directly finetunes or prompts a pretrained language model (LM) to leverage the stored knowledge. However, they do not fully exploit the parameterized knowledge. To address this inefficiency, we propose a two-stage, *closed-book* QA framework which employs a *coarse-to-fine* approach to extract the relevant knowledge and answer a question. We first generate a related context for a given question by prompting a pretrained LM. We then prompt the same LM to generate an answer using the generated context and the question. Additionally, we marginalize over the generated contexts to improve the accuracies and reduce context uncertainty. Experimental results on three QA benchmarks show that our method significantly outperforms previous *closed-book* QA methods. For example on TriviaQA, our method improves exact match accuracy from 55.3% to 68.6%, and is on par with *open-book* QA methods (68.6% vs. 68.0%). Our results show that our new methodology is able to better exploit the stored knowledge in pretrained LMs without adding extra learnable parameters or needing finetuning, and paves the way for hybrid models that integrate pretrained LMs with external knowledge.

1 Introduction

Open-domain question answering (ODQA) produces an answer to a given question in the form of natural language, and the task has been extensively studied in recent years. Significant progress on ODQA has been made by developing the *open-book* QA methods (Chen et al., 2017; Lewis et al., 2020b; Guu et al., 2020; Izacard and Grave, 2021;

*This work was done when the first author was an intern at NVIDIA. Corresponding authors: Dan Su, Mostofa Patwary.



Figure 1: An example illustrating our two-stage, **CGAP** framework. CGAP generates more accurate answer (e.g. *Richard Marx*) compared to standard few-shot prompting (e.g. *Ross Bagdasarian*).

Lazaridou et al., 2022) that explicitly exploit external knowledge corpus via dense retrieval techniques like DPR (Karpukhin et al., 2020). However, learning a good retriever requires substantial resources, such as a large number of domain-specific pairs of question and contexts in the knowledge corpus (Karpukhin et al., 2020), or intensive compute resources (Lee et al., 2019). In addition, as the size of the knowledge corpus increases, it becomes harder to retrieve accurate contexts due to the high dimensionality of the search space (Reimers and Gurevych, 2021).

Another class of models, known as *closed-book* question answering (CBQA), were recently proposed (Roberts et al., 2020). CBQA tries to directly answer the open-domain questions without accessing any external knowledge sources, and instead leverages the parametric knowledge stored in the pretrained language models (LMs) (Raffel et al., 2020; Brown et al., 2020; Ye et al., 2020).

However, even with larger LMs, the *closed-book* methods are not competitive with the *open-book* methods in term of accuracy (Lewis et al., 2021).

While it has been shown that large pretrained LMs store an abundant amount of knowledge (Petroni et al., 2019; Roberts et al., 2020), we hypothesize the accuracy gaps are largely because the way of exploiting the parameterized knowledge are not sophisticated enough. Prior works on CBQA either finetune pretrained LM models on the entire QA datasets (Roberts et al., 2020; Ye et al., 2020), or they directly prompt those models using several few-shot QA pairs (Brown et al., 2020; Radford et al., 2019). On the contrary, *open-book* models use a two-stage pipeline. They first retrieve relevant contexts from external corpus, then they extract the answer based on the retrieved contexts.

Therefore, to better exploit the parameterized knowledge in pretrained LMs and bridge the large accuracy gaps between the *closed-book* and *open-book* methods, we propose a *coarse-to-fine*, two-stage method for CBQA task. The main idea is to leverage generated contexts as an intermediate bridge between the huge amount of parameterized knowledge stored in the LM and the answer that lies within this knowledge. To the best of our knowledge, no previous work has been conducted on generating context from large pretrained LMs for CBQA and leveraging them to predict answer.

Our proposed framework **CGAP** consists of two stages. It first performs **Context Generation** relevant to a given question by prompting a pretrained LM. It then prompts the same LM for **Answer Prediction** using the generated context and the question. In order to improve the accuracies and to reduce context uncertainties, we generate multiple contexts for each question and predict the final answer by majority voting. This step does not increase the inference cost as we generate the contexts in parallel by batching in a single inference call. Figure 1 illustrates how our two stage prompting and majority voting works. For the input question, CGAP generates 3 contexts and 3 predicted answers at the two stages respectively, and choose the most voted answer as the final answer. Note that we do not finetune the large pretrained LMs for context generation or answer prediction. This facilitates our approach to take advantage of large LMs such as GPT-3 (Brown et al., 2020), PALM (Chowdhery et al., 2022) or Megatron-Turing NLG 530B (Smith et al., 2022),

which are only available through APIs.

We conduct in-depth experimental studies on three open-domain QA benchmarks, Natural Questions (Kwiatkowski et al., 2019), WebQuestions (Berant et al., 2013), and TriviaQA (Joshi et al., 2017), and demonstrate significant improvements by our two stage prompting method. Our contributions are summarized as follows:

- We propose a simple yet effective few-shot prompting approach for ODQA that does not rely on any external knowledge sources or fine-tuning, but performs significantly better than existing *closed-book* approaches (e.g. exact matching 68.6% vs. 55.3%), and is on par with *open-book* methods (e.g. 68.6% vs. 68.0%).
- We show that the generated context can improve standard few-shot prompting based *closed-book* QA accuracy at various model scales (e.g. from 11.7% to 28.5%), and demonstrate that scaling up the context generation model further enlarges their accuracy gaps (e.g. 357M 28.5% vs. 530B 68.6%). To the best of our knowledge, we are the first to leverage generated context from large pretrained LMs for open-domain question answering.
- We show that generating multiple contexts without increasing the inference cost by batching can mitigate errors in answer prediction caused by variability in the unknown context (e.g. from 36.3% to 45.7%).

2 Methodology

Our proposed **Context Generation and Answer Prediction (CGAP)** framework is illustrated in Figure 2. CGAP consists of two stages. First, it generates relevant context to a given question by prompting a large pretrained LM. In the second stage, it predicts an answer using the generated context and the question by prompting the same LM. To accurately predict the answer, we generate multiple contexts. We run each of the two stages multiple times in parallel in batch for the same question, generating different contexts for each, and use majority voting to select the final answer.

Formally, for our task we have a question Q to be answered, and a support repository $\mathcal{D} = \{(c_1, q_1, a_1), \dots, (c_n, q_n, a_n)\}$ that consists of tuples of question q_i and answer a_i pairs with mapping to the context c_i . In our experiments, we use

the training sets of the corresponding datasets as \mathcal{D} .

2.1 Context Generation

As shown in Figure 2, in the first stage, given question Q , we select the m context generation prompts $S = \{(q_1, c_1), \dots, (q_m, c_m)\}$ from the support repository \mathcal{D} . We then use S with Q to prompt pretrained LM to generate k contexts, which are denoted by $C_{gen} = \{c_{gen}^1, c_{gen}^2, \dots, c_{gen}^k\}$.

Sample Selection Selecting appropriate samples for the prompts is the key to generate high-quality context relevant to a given question. Previous work has shown that leveraging relevant samples helps the LM to generate contextually relevant and factually correct context (Liu et al., 2021, 2022). We therefore use a similarity-based retriever to search relevant samples S from the corresponding supporting repository, \mathcal{D} . We use DPR (Karpukhin et al., 2020) in our framework. In our DPR setup, we represent the question and the samples in \mathcal{D} as 768-dimensional dense vector representations, computed via the BERT-based bi-encoder networks. We rank the documents according to their similarity score, calculated as:

$$Score(Q, (q_j, c_j)) = \text{BERT}(Q)^T \cdot \text{BERT}(q_j; c_j) \quad (1)$$

where $;$ denotes concatenation of the tokens of the question q_j and the context c_j . Finally, we get $S = \{(q_1, c_1), \dots, (q_m, c_m)\}$ which are the top- m retrieved samples for question Q .

We would like to emphasize that the selected samples from \mathcal{D} are used as examples in the few-shot prompting to the pretrained LM to generate context, not as the source of external knowledge containing the answer.

Prompts Construction Given the question Q and the set of question-context pair samples S selected, we use few-shot prompting to condition pretrained LMs on the samples. We use similar few-shot prompting technique for *closed-book* QA as in (Brown et al., 2020), that considers multiple \langle question, answer \rangle pairs. The template we used to construct prompts is: $Q: \dots A: \dots$. Thus the constructed prompt $Prompt(Q)$ for a given question Q becomes:

$$Prompt(Q) = Q: q_m \setminus n A: c_m \setminus n \dots \\ Q: q_1 \setminus n A: c_1 \setminus n Q: Q \setminus n$$

We use ' $\setminus n$ ' to separate the question, context and the samples. We investigated the order of samples to optimize the prompt and find that using the retrieved samples in reversed order of similarity yields better accuracies across all datasets¹. We now pass $Prompt(Q)$ through a pretrained LM to generate the context as follows:

$$c_{gen} = \mathcal{LM}(Prompt(Q))$$

To generate a set of k contexts, $\{c_{gen}^1, \dots, c_{gen}^k\}$, we increase the inference batch size to k and generate all the k contexts in parallel in one inference call to the LM. Thus, the overall latency remains the same as using a single context.

2.2 Answer Prediction

In the second stage, we select m answer prediction prompts $S' = \{(q_1, a_1, c_1), \dots, (q_m, a_m, c_m)\}$ from \mathcal{D} and then we prompt the same LM using the generated context C_{gen} from the first stage, along with the question Q and S' . The LM predicts a set of k answers $A_p = \{a_p^1, a_p^2, \dots, a_p^k\}$ each corresponding to the k contexts in C_{gen} . The final answer A is selected by majority voting on A_p .

Sample Selection Constrained by the maximum sequence length of the LM, we can feed the LM only a few (c, q, a) samples. Thus, it could be difficult for the LM to learn how to predict the answer for the given question conditioned on the context, unless similar examples have been provided. For example, if we were asking the question '*who is the current director of the us mint?*', the example that answering the question '*who is the fbi director of the united states?*' from the provided context will be more helpful, than the example that is answering '*how many episodes are there in 'Dragon Ball Z'?*' from the given context. We therefore use the same criteria for answer prediction as has been used for context generation. We use the same set of samples as selected in the first stage as described in Equation 1 and denote as $S' = \{(q_1, c_1, a_1), \dots, (q_m, c_m, a_m)\}$.

Prompt Construction We are prompting LMs with few-shot examples to predict answer for the question conditioned on the generated context. To equip the LM with this capability, we constructed intuitive prompts for the selected examples and feed them into the LM. Specifically, the template

¹We show an concrete example of $Prompt(Q)$ in Appendix Table 12

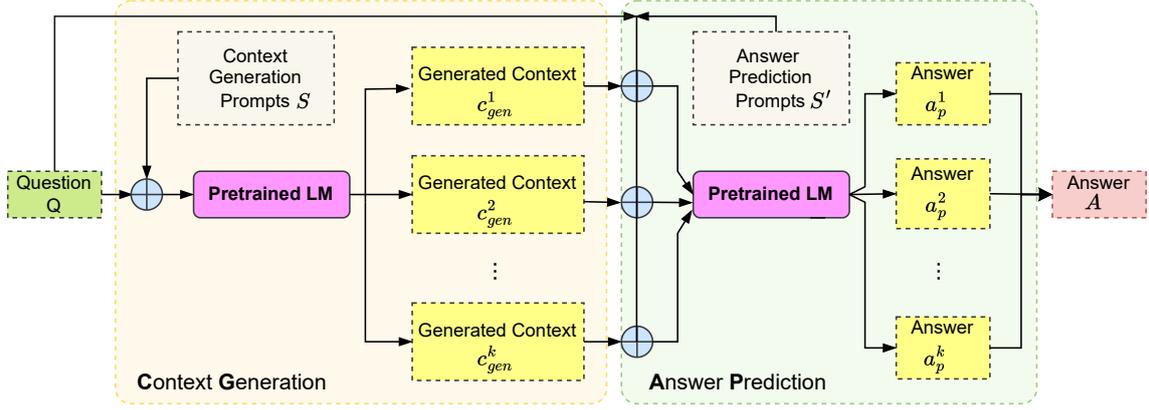


Figure 2: Overview architecture of our **CGAP** framework. It first does **Context Generation** by prompting large pretrained LMs, then it further prompts the LMs for **Answer Prediction** by feeding the generated context to the LM models alongside the question. k contexts are generated and the final answer A is chosen by majority voting. (If computation capability allows, it could prompt multiple (k) LMs in parallel at both two stages to speed up.)

we used to construct answer prediction prompts is: $C : \dots Q : \dots A : \dots$. Thus, the constructed prompt for a given question Q and the i -th generated context c_{gen}^i is:

$$\begin{aligned} Prompt(c_{gen}^i, Q) &= C : c_m \setminus n Q : q_m \setminus n A : a_m \setminus n \\ &\dots \\ &C : c_1 \setminus n Q : q_1 \setminus n A : a_1 \setminus n \\ &C : c_{gen}^i \setminus n Q : Q \setminus n \end{aligned} \quad (2)$$

We then feed $Prompt(c_{gen}^i, Q)$ into the pretrained LM to predict the answer:

$$a_p^i = \mathcal{LM}(Prompt(c_{gen}^i, Q)) \quad (3)$$

where we use a_p^i to denote the i -th answer predicted by the LM. The k generated contexts in c_{gen} will yield a set of answers $A_p = \{a_p^1, \dots, a_p^k\}$.

2.3 Context Marginalization

The large pretrained LM can generate impressively fluent and relevant context given input, it also has a tendency to generate factually incorrect statements, ranging from subtle inaccuracies to wild hallucinations (Shuster et al., 2021; Krishna et al., 2021; Su et al., 2022). Answers conditioned solely on hallucinated or erroneous statements are likely to be incorrect (Equation 3). Thus, we would like to remove the variability in the answer due to any particular generated context.

Ideally, we could marginalize over this unknown context by producing an answer for every possible context, weighting each answer by the probability of the context. Here we approximate this by

generating a set of contexts, and selecting the final answer based on majority voting. Suppose there are T unique answers $\{A_p^1, \dots, A_p^T\}$ from the k predicted answer from Equation 3 where $T \leq k$, then we select the J -th answer that receives the highest number of votes from the T different answers via:

$$J = \operatorname{argmax}_{j \in \{1, 2, \dots, T\}} \sum_{i=1}^k (\mathbb{1}(a_p^i = A_p^j)) \quad (4)$$

as the final answer A . As k gets larger, the final answer A will converge to the answer that would be produced marginalizing over all possible contexts. We refer to this majority vote over multiple generated contexts as context marginalization.

3 Experimental Setup

3.1 Datasets

We evaluated our experiments on three open-domain QA benchmark datasets: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (TQA) (Joshi et al., 2017), and WebQuestions (WQ) (Berant et al., 2013), using the same data splits for train, validation and test as in Lee et al. (2019); Izcard and Grave (2021).

NQ contains questions from Google search queries; TQA contains a collection of questions from trivia and quiz-league websites, and we use their unfiltered set; while questions of WQ were from Google Suggest API. For NQ and TQA, we use the processed data provided by Izcard and Grave (2021), in which each question-answer pair is accompanied by a 100-words Wikipedia passage

Model Type	Model	Method	NQ	TQA	WQ
Open-book	RAG (Lewis et al., 2020c)	<i>Finetuned</i>	44.5	68.0	45.5
	Fusion-in-Decoder (large) (Izcard and Grave, 2021)	<i>Finetuned</i>	51.4	67.6	-
	OB_{Google}^{PoE} (Lazaridou et al., 2022)	<i>Few-shot</i>	38.4	-	-
Closed-book	T5-11B (Roberts et al., 2020)	<i>Finetuned</i>	32.6	42.3	37.2
	T5-11B+SSM (Roberts et al., 2020)	<i>Finetuned</i>	34.8	51.0	40.8
	BART-large, pre-finetuned on PAQ (Lewis et al., 2021)	<i>Finetuned</i>	32.7	33.2	-
	LM-530B (API)	<i>Few-shot</i>	23.0	55.3	23.6
	CGAP (ours, 530B)	<i>Few-shot</i>	<u>42.0</u>	68.6	<u>41.8</u>

Table 1: Exact Match score for **CGAP** (highest accuracy configurations) in comparison to recent state-of-the-art *open-book* and *closed-book* based systems. Highest score indicated in **bold**, highest *closed-book* model underlined.

containing the answer. For WQ, we retrieved the corresponding context passage for each question from 2019/08/01 Wikipedia dump, using the DPR-based retriever that is trained jointly on the union of knowledge-intensive training data in KILT benchmark (Petroni et al., 2021).

3.2 Baselines

We compare our **CGAP** framework with the following baseline methods for *closed-book* QA.

Standard Few-shot Prompting We use the standard few-shot prompting technique similar to GPT-3 (Brown et al., 2020) in our evaluation on the *closed-book* QA datasets as described in Section 3.1. We consider this technique as the few-shot baseline in all our experiments. The baseline that is experimented using 530 billion (530B) parameterized LM is referred as **LM-530B**.

LM Fine-tuning Roberts et al. (2020) first proposed the *closed-book* QA task for open domain QA, and they directly fine-tuned T5 (Raffel et al., 2019) using the entire QA pairs in the training data, without access to any external knowledge corpus (referred as **T5-11B**). They also experimented with using ‘Salient Span-Masking’ (SSM) to continue pretraining the T5 checkpoints before fine-tuning for QA (referred as **T5-11B+SSM**). Lewis et al. (2021) pre-finetuned BART-large (Lewis et al., 2020a) on *Probably Asked Questions* (PAQ), a very large resource of 65M automatically generated QA-pairs, then further finetuned the model on corresponding training data (referred as **BART-large, pre-finetuned on PAQ**).

Open-book Few-shot Prompting Lazaridou et al. (2022) used few-shot prompting for open domain QA task, but they generate the answer via conditioning on retrieved documents from Google

Search API. (referred as OB_{Google}^{PoE})

3.3 State-of-the-art Open-book QA Models

We compare the state-of-the-art *open-book* QA models with **CGAP**. **Fusion-in-Decoder** (FiD) (Izcard and Grave, 2021) uses DPR (Karpukhin et al., 2020) to retrieve 100 passages from Wikipedia. Then they encode each passage independently and combine all outputs from the T5 encoder before passing them to the T5 decoder to generate a final answer. **RAG** (Lewis et al., 2020b) is an end-to-end retrieval-augmented generation model.

3.4 Implementation Details

To test how different model scales affect the performance of our approach, we train and experiment on a collection of decoder-only LMs using the Megatron-LM framework (Shoeybi et al., 2019), with 357 million (357m), 1.3 billion (1.3b), and 530 billion (530b) (Smith et al., 2022) parameters, at both context generator and answer prediction stage. We use top- p sampling with a value of 0.9 to generate diversified contexts. However, to handle the deterministic generation (e.g. short answer), we use greedy decoding at the answer prediction stage, similar to (Chowdhery et al., 2022; Wang et al., 2022).

For the prompt configuration at both stages, we choose 10 samples, constrained by the maximum sequence length of the LMs. We use DPR checkpoint from Huggingface² to select samples from the supporting repository.

3.5 Evaluation

For evaluating the open-domain QA task, we followed the recent works (Rajpurkar et al., 2016; Lee

²https://huggingface.co/facebook/dpr-ctx_encoder-multiset-base

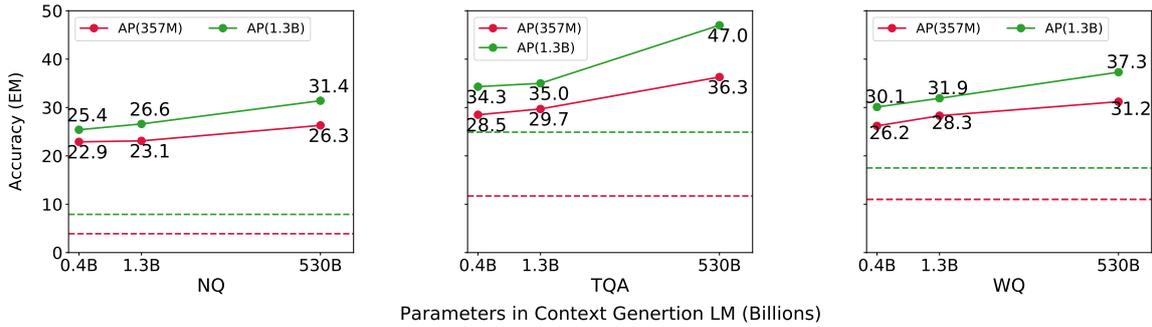


Figure 3: Ablation on context generation LM size. The dash lines represent standard few-shot prompting baselines.

et al., 2019; Izcard and Grave, 2021) that use Exact Match (EM) as the evaluation metric. Each predicted answer is compared to the ground-truth after both are lowercased and stripped of articles, punctuation, and duplicate whitespace.

4 Results and Ablation Studies

We now show our main results as well as ablations to further analyze the effectiveness of our approach.

4.1 Main Results

Table 1 shows the EM score comparison between our CGAP-based method with existing *closed-book* baseline approaches³. We also compare with state-of-the-art *open-book* models at the upper section of the table.

As we can see, our CGAP based method outperforms other existing *closed-book* methods by large margin, especially on NQ and TQA datasets. The CGAP also outperforms the standard few-shot prompting baseline LM-530B on all three datasets (at least by 13.3 EM point).

Furthermore, CGAP obtains highest score on TriviaQA. The scores are also very close to the state-of-the-art *open-book* method RAG on NQ and WebQuestions, but only lose few points on NQ to FiD. While FiD uses 100 retrieved passages for answer prediction, CGAP only uses 8 generated contexts for approximate context marginalization.

4.2 Ablation Studies

We conducted a systematic ablation study to further investigate the contribution of the context generation model and the effect of context marginalization.

³GPT-3 API shows different results than reported in the paper (Brown et al., 2020). We therefore did not compare to it. Details are shown in Appendix A

4.2.1 Context Generation

While previous work (Roberts et al., 2020; Brown et al., 2020) demonstrated that the scale of the model sizes improves the answer accuracy of *closed-book* QA, there are also other findings showing that simply increasing the model size does not lead to substantive accuracy gains (Rae et al., 2021). Thus, we intend to investigate **how will the context generation LM affect the answer accuracy**.

We experimented by varying the LM sizes for context generation, and fix the answer generation LM. We used context generation LM sizes of 357m, 1.3B and 530B, and answer generation LM with 357m and 1.3B parameters. We also compare with standard few-shot prompting which has no context generation.

We plot the results in Figure 3. As we can see, there are huge accuracy gains from standard prompting, to CGAP method that has context generation. The accuracy increases by absolute 19.00% for NQ, 16.87% for TQA and 15.26% for WQ, when using 357M model for both standard prompting and CGAP approach. The answer accuracy continues to increase when we increase the LM size for context generation. Furthermore, we notice that the slopes of the accuracy gain curve using larger answer prediction model is steeper than using smaller one on all three datasets. This suggests the use of larger answer prediction LM to fully exploit the knowledge in generated context.

4.2.2 Context Marginalization

Since there will be some hallucinated content or erroneous statements in the generated context, we approximate context marginalization by sampling multiple contexts and selecting the final answer based on majority voting, as introduces in Section 2.3. Here, we investigate the performance gains brought in by context marginalization, and

AP LM Size	CG LM Size	Margin-alization	NQ	TQA	WQ
357M	357M	✗	22.9	28.5	26.2
		✓	25.7 (+2.8)	33.4 (+4.9)	29.6 (+3.4)
	1.3B	✗	23.1	29.7	28.3
		✓	26.1 (+3.0)	34.8 (+5.1)	31.3 (+3.0)
530B	530B	✗	26.3	36.3	31.2
		✓	28.9 (+2.7)	45.7 (+9.4)	34.0 (+2.8)
	530B	✗	29.5	56.3	28.3
		✓	42.0 (+12.5)	68.6 (+12.4)	41.8 (+13.5)

Table 2: Ablation on context marginalization. (AP and GP represent Answer Prediction and Context Generation, respectively.)

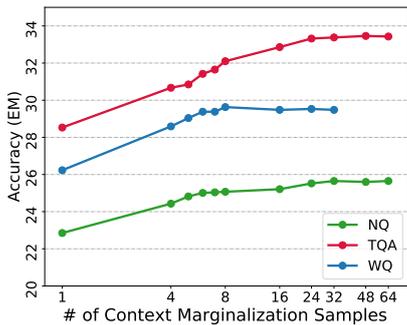


Figure 4: Ablation on k , the number of contexts for marginalization.

also the accuracy curves with varied number of sampled contexts used in the approximate marginalization k .

In Table 2, we show the accuracy comparisons w/ and w/o using marginalization ($k=8$), with different LM sizes. As we can see, **context marginalization improves the answer accuracy consistently on the three datasets**⁴, under all settings. Notably, there is much larger performance gains using marginalization when we scale up the model sizes to 530 billion parameters (i.e. increase EM score by 12.8% averaged on three datasets).

The larger the number of context samples k , the more accurately the majority vote reflects the true marginalization over all possible contexts. Therefore, we perform further ablation by changing the value k for 357M LM for both context generation and answer prediction. We plot the accuracy curves in Figure 4. We see that there are accuracy improvements when we use more context samples. As expected and curves plateau for larger values of k as the approximation approaches the true marginalization over all possible contexts.

⁴We show a concrete example in Appendix B Table 11

5 Analysis

Considering that it is the first time leveraging context generated by large pretrained LMs for ODQA, we also conducted further analysis.

We compare generated context with retrieved context in the two-stage, few-shot prompting based CBQA framework. It is a dominant paradigm to use retrieved context from external corpus together with the question for answer prediction for *open-book* QA (Chen et al., 2017; Lewis et al., 2020c; Izacard and Grave, 2021; Lazaridou et al., 2022).

5.1 Retrieved vs. Generated Context

In CBQA setting, we are not allowed to retrieve context from external knowledge sources. However, we can retrieve the contexts from the supporting repository based on their relevance to the given question. We use $c_r = \{c_r^1, c_r^2, \dots, c_r^m\}$ to represent the top- m relevant context for question Q . It can be obtained via Equation 1.

Let the top-1 retrieved context be $c_r^{\text{top-1}}$ for question Q . We use $c_r^{\text{top-1}}$ to compare with the generated context, c_{gen} . We use the same top- m prompts S' for answer prediction as introduced in Section 2.2. The answer a_p^r for the $c_r^{\text{top-1}}$ will be:

$$a_p^r = \mathcal{LM}(\text{Prompt}(c_r^{\text{top-1}}, Q)) \quad (5)$$

where $\text{Prompt}(c_r^{\text{top-1}}, Q)$ can be obtained via Equation 2.

The comparison between $c_r^{\text{top-1}}$ and c_{gen} is shown in Table 4. From the upper part of the table, we see that using $c_r^{\text{top-1}}$ gives slightly higher EM score than using c_{gen} generated by 357M and 1.3B LMs. However, c_{gen} gives higher EM scores than $c_r^{\text{top-1}}$ on all three datasets when we scale up the context generation LM size to 530B. This suggests the use of large pretrained LM for a better generated context.

Question: Which sitcom star appeared on the big screening 'The Object of My Affection'?	
Golden Answer: [Jennifer Anniston, Jen Aniston, ...]	
Predicted Answer (w/o c_{gen}):	Ross Hatley
Predicted Answer ($c_r^{\text{top-1}}$):	Laurie Metcalfe
Predicted Answer (c_{gen}):	Jennifer Anniston / Paul Rudd / Christine Baranski / Lisa Kudrow
Predicted Answer ($(c_{gen}^1, \dots, c_{gen}^k)$):	Jennifer Anniston

Table 3: Comparison of answers predicted w/o and w/ different context. Example from TriviaQA (Joshi et al., 2017) test set. Red and green colors denote in-correct and correct answer, respectively.

AP	Context	NQ	TQA	WQ
357M	$c_r^{\text{top-1}}$	25.1	32.2	28.3
	c_{gen} (357M LM)	22.9	28.5	26.2
	c_{gen} (1.3B LM)	23.1	29.7	28.3
	c_{gen} (530B LM)	26.3	36.3	31.2
530B	$c_r^{\text{top-1}}$	30.8	58.1	29.5
	c_{gen} (530B LM)	29.5	56.3	28.3

Table 4: Comparison of using retrieved top-1 context $c_r^{\text{top-1}}$, with few-shot generated context c_{gen} on *closed-book* QA task.

5.2 Multiple Retrievals vs. Context Marginalization

We notice that in Table 4, $c_r^{\text{top-1}}$ performs slightly better than c_{gen} when using 530B LM for answer prediction. We argue that this might be caused by the hallucination in c_{gen} . While we have shown in Section 4.2.2 that context marginalization could mitigate the problem and improve answer accuracy, we further facilitate c_{gen} (530B) with context marginalization and compare with retrieved context.

For fair comparison, we perform majority voting using the top- k retrieved context c_r , since Karpukhin et al. (2020) showed that the quality of the retrieved documents will also affect the final answer accuracy. Specifically, we replace $c_r^{\text{top-1}}$ with each retrieved context c_r^i in Equation 5 to predict answer $a_p^{r(i)}$ ($i = 1, \dots, k$), and use Equation 4 to select the most frequent answer as the final answer.

Furthermore, we replace $c_r^{\text{top-1}}$ with golden context c_{golden} in Equation 5. This will be the upper-bound of using retrieved/generated context in the two-stage, few-shot prompting CBQA task.

We show the results in Table 5. As we can see, using marginalization over c_{gen} consistently outperforms $c_r^{\text{top-1}}$, and also better than majority voting over multiple retrieved contexts c_r for answer prediction on all three datasets. Notably, marginal-

AP	Context	NQ	TQA	WQ
530B	c_{golden}	36.0	61.3	30.2
	$c_r^{\text{top-1}}$	30.8	58.1	29.5
	(c_r^1, \dots, c_r^k)	29.5	56.3	28.3
	c_{gen}	23.0	55.3	23.6
	$(c_{gen}^1, \dots, c_{gen}^k)$	42.0	68.6	41.8

Table 5: Comparison of using context marginalization ($c_{gen}^1, \dots, c_{gen}^k$), multiple retrievals (c_r^1, \dots, c_r^k), and golden context c_{golden} on *closed-book* QA task.

ization over c_{gen} yields higher EM score than using c_{golden} when using 530B LM for answer prediction. We observed similar trends when experimented on 357M and 1.3B parameter models. In Table 3, we show a concrete example that compares using different context for answer generation for better understanding⁵.

6 Related Works

Open-domain QA is the task of answering general-domain questions (Chen et al., 2017), in which the evidence is usually not given. Models that explicitly exploit an external corpus are referred as *open-book* models (Roberts et al., 2020). They typically index the corpus and then *retrieve-and-read* to extract the answer span from documents (Chen et al., 2017; Lee et al., 2019; Izacard and Grave, 2021; Lewis et al., 2020b; Lazaridou et al., 2022). Another recently proposed class of methods is *closed-book* QA models. Ye et al. (2020); Roberts et al. (2020) finetune pre-trained LMs such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020a) with QA pairs without access to any external knowledge or context.

Few-shot LM Prompting Radford et al. (2019); Brown et al. (2020) prompt GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) conditioned

⁵More concrete comparison examples are shown in Appendix B Table 9 and Table 10.

on several few-shot examples to predict the answer for ODQA. Most recent work by Lazaridou et al. (2022) further empower LM’s few-shot prompting abilities with information returned from the web using Google-Search API, and experimented on QA task. While Wei et al. (2022); Wang et al. (2022) use *chain of thought* few-shot prompting of LM to generate a coherent chain of short sentences that mimic the reasoning process of human might employ to solve reasoning tasks.

7 Conclusion

We propose a simple yet effective framework named **CGAP** for open-domain QA. CGAP performs **C**ontext **G**eneration followed by **A**nswer **P**rediction via two-stage prompting using large pre-trained LMs. It does not rely on external knowledge sources, and does not need finetuning or add extra learnable parameters. To the best of our knowledge, we are the first to leverage generated context from large pretrained LMs for open-domain QA. Experimental results on three QA benchmarks show that our method significantly outperforms previous *closed-book* QA methods and is par with *open-book* methods. We demonstrate our method up to 530B parameter models and showcase that larger models boost the accuracy by huge margins.

8 Limitations

As we show in the paper, **CGAP** has obtained satisfactory results on open-domain QA task. However, the method have limitations. The accuracy of CGAP will be affected by the size of LMs it uses, as we shown in Figure 3. In Section 4.1, our highest accuracy results reported in Table 1 used a large 530B pretrained LM, which is only accessible via API. Also, the generated context may contain hallucinated content.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4940–4957.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020c. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.
- Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.
- Zihan Liu, Mostofa Patwary, Ryan Prenger, Shrimai Prabhumoye, Wei Ping, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Multi-stage prompting for knowledgeable dialogue generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1317–1337.
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2021. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog 1.8 (2019)*: 9.
- Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Mari-beth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsim-poukelli, Nikolai Grigorev, Doug Fritz, Thibault Sotiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d’Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorraine Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. [Scaling language models: Methods, analysis & insights from training gopher](#).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Nils Reimers and Iryna Gurevych. 2021. [The curse of dense low-dimensional information retrieval for large index sizes](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 605–611, Online. Association for Computational Linguistics.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deep-speed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 744–756.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Qinyuan Ye, Belinda Z Li, Sinong Wang, Benjamin Bolte, Hao Ma, Wen-tau Yih, Xiang Ren, and Madian Khabsa. 2020. Studying strategically: Learning to mask for closed-book qa. *arXiv preprint arXiv:2012.15856*.

A Standard Few-shot Prompting of GPT-3

Brown et al. (2020) adopted the standard few-shot prompting on GPT-3, and evaluated on the three open-domain QA datasets NQ (Kwiatkowski et al., 2019), WQ (Berant et al., 2013) and TQA (Joshi et al., 2017), for *closed-book* QA task. In order to compare with their reported results, we re-implement their method using the same few-shot configuration as described in the paper and query the OpenAI API.

Experimental Setups As OpenAI hasn’t officially release information about their API model sizes, we deduce the sizes of OpenAI API models based on their performances from EleutherAI’s blog⁶. Specifically, we query Ada and Babbage

models’ API, trying to reproduce the reported results for GPT-3 Medium (350M) and GPT-3 XL (1.3B) models, respectively.

We use two prompt formats to query the OpenAI API. The first prompt format is the one described in the paper (Brown et al., 2020) (referred as *GPT-3 format*): randomly draw 64 question-answer pairs from the corresponding supporting repository, and use ‘Q: ’ and ‘A: ’ respectively as prefix before each question and answer, to build the conditioning prompts. We also use the prompt format from EleutherAI’s language model evaluation harness github⁷ (referred as *EleutherAI*). Furthermore, we experiment using the same prompting format as we used in our standard prompting baseline (LM-530B) in Section 3.2 (referred as *Our format*), and prompting the LM of size 357M and 1.3B to compare.

Results We show the results of prompting GPT-3 under zero-shot, one-shot and few-shot settings in Table 6, Table 7 and Table 8 respectively. As we can see, no matter what prompting formats we use, the results reported in the GPT-3 paper (Brown et al., 2020) are almost always higher than our reproduced ones on all three datasets, over the two different LM sizes. The gaps become even larger at few-shot setting. Thus we conjecture that we are not able to reproduce the results reported by Brown et al. (2020) using GPT-3 (175B) on the three QA datasets. So we did not include their reported results to compare with our CGAP method in Table 1.

Furthermore, we notice that the results based on our baseline’s prompting configuration are always on par with the results from querying OpenAI API. Thus we believe that the **LM-530B** is a reliable and fair standard few-shot prompting baseline to compare with.

B Examples

We show three examples from NQ, TQA and WQ test set in Table 9, Table ?? and Table 10 respectively. In each table, we show the predicted answers from (1) standard prompting, (2) two-stage prompting using top-1 retrieved context $c_r^{\text{top-1}}$, (3) CGAP w/o marginalization, and (4) CGAP. All those predicted answers are based on LMs of size 530B.

We also show an example illustrate CGAP with 8 generated context and their corresponding pre-

⁶<https://blog.eleuther.ai/gpt3-model-sizes/>

⁷<https://github.com/EleutherAI/lm-evaluation-harness>

dicted answer in Table 11. As we can see, the contexts that contains lot of factually inaccurate or irrelevant content (e.g. generated context 1, 2, 4, 5, 8), thus the corresponding answer is wrong/inaccurate. However, the context generation LM also generates contexts that are more relevant and factual (e.g. generated context 3, 6, 7), and they help the answer prediction LM generate a correct answer. Therefore, CGAP can predict the final answer correctly based on marginalization over generated contexts.

Model Sizes	Model sources	Prompting format	zero-shot		
			NaturalQuestion	TriviaQA	WebQuestion
350M	GPT-3 Medium	GPT-3 paper (Brown et al., 2020)	1.75	7.61	3.20
	OpenAI API (Ada)	GPT-3 format	1.36	5.45	1.92
		EleutherAI	1.39	5.54	2.46
	LM-357M	Our format	1.41	5.04	2.12
1.3B	GPT-3 XL	GPT-3 paper (Brown et al., 2020)	4.40	19.70	4.63
	OpenAI API (Babbage)	GPT-3 format	2.27	9.84	2.12
		EleutherAI	2.47	12.77	5.22
	LM-1.3B	Our format	3.88	14.13	5.61

Table 6: Standard zero-shot prompting of GPT-3 for open-domain QA.

Model Sizes	Model sources	Prompting format	one-shot(k=1)		
			NaturalQuestion	TriviaQA	WebQuestion
350M	GPT-3 Medium	GPT-3 paper (Brown et al., 2020)	3.07	12.90	6.20
	OpenAI API (Ada)	GPT-3 format	1.83	10.26	5.07
		EleutherAI	1.77	10.02	5.61
	LM-357M	Our format	2.24	9.75	5.12
1.3B	GPT-3 XL	GPT-3 paper (Brown et al., 2020)	5.43	26.50	9.15
	OpenAI API (Babbage)	GPT-3 format	3.55	20.56	8.27
		EleutherAI	3.55	21.45	9.45
	LM-1.3B	Our format	4.71	21.21	8.76

Table 7: Standard one-shot prompting of GPT-3 for open-domain QA.

Model Sizes	Model sources	Prompting format	few-shot(k=64)		
			NaturalQuestion	TriviaQA	WebQuestion
350M	GPT-3 Medium	GPT-3 paper (Brown et al., 2020)	4.46	16.30	12.60
	OpenAI API (Ada)	GPT-3 format	3.43	12.46	10.73
		EleutherAI	3.71	12.46	10.29
	LM-357M	Our format	3.85	11.66	10.97
1.3B	GPT-3 XL	GPT-3 paper (Brown et al., 2020)	9.72	32.10	19.60
	OpenAI API (Babbage)	GPT-3 format	8.28	24.70	18.95
		EleutherAI	7.81	24.93	18.16
	LM-1.3B	Our format	7.87	24.88	17.52

Table 8: Standard few-shot (k=64) prompting of GPT-3 for open-domain QA.

Question: When is the next deadpool movie being released?	
Golden Answer:	[May 18, 2018]
Predicted Answer (standard prompting):	Prime availability TBD
Predicted Answer ($c_r^{\text{top-1}}$):	May 18, 2018
Predicted Answer (CGAP w/o marginalization):	May 18, 2018 / date21-May-2018 / May 29, 2019 /16th May 2018
Predicted Answer (CGAP):	May 18, 2018

Table 9: Example from NQ (Kwiatkowski et al., 2019) test set. Red and green colors denote in-correct and correct answer, respectively.

Question: Where George Lopez was born?	
Golden Answer:	[Mission Hills]
Predicted Answer (standard prompting):	San Fernando, California
Predicted Answer ($c_r^{\text{top-1}}$):	Washington Heights
Predicted Answer (CGAP w/o marginalization):	Mission Hills / Los Angeles/ San Fernando /Castle Hill /canada
Predicted Answer (CGAP):	Mission Hills

Table 10: Example from WebQuestions (Berant et al., 2013) test set.

Question: Where George Lopez was born? **Golden Answer:** [Mission Hills]

Generated Context 1: George Lopez They are American citizens, born in Los Angeles, to Gregorio Lopez (October 26, 1934 – January 25, 2011) and Maria Torres (October 12, 1938 – October 25, 2014), Mexican immigrants who were born in Reynosa, Tamaulipas. Lopez's paternal grandparents were Mexican, while his maternal grandparents were of Mexican and Spanish descent. Lopez's father was a worker for the Los Angeles Metropolitan Transit Authority. He married Maria in Los Angeles when he was 16 years old, and she was 15. Maria filed for divorce from Gregorio in the mid-1970s.

Predicted Answer: Los Angeles

Generated Context 2: George Lopez born April 23, 1961. The family eventually moved from Marin to Los Angeles in the mid-1960s. After Lopez graduated from San Fernando High School in 1979, he attended the California State University, Los Angeles. It was there where Lopez discovered his comedic talent after a video of him doing a stand-up routine made it into the hands of "Tonight Show" booker, Fred De Cordova, in 1985.

Predicted Answer: canada

Generated Context 3: George Lopez at 20th Century Fox Television until 2010, when he began appearing in his self-produced sitcom, George Lopez. Lopez was born on April 23, 1961, in Mission Hills, Los Angeles, California. He was raised by his maternal grandmother, Benita Gutierrez, a factory worker, and his maternal grandfather, Refugio Gutierrez, who worked as a janitor and metal fabricator at an ice skating rink.

Predicted Answer: Mission Hills

Generated Context 4: George Lopez in San Fernando, California. He is of Mestizo Mexican descent. His father, Saul Lopez, was from Sonora, and his mother, Elvira, was a native of Arizona. They were migrant workers who picked crops and moved often, sometimes taking their seven children with them. While growing up, Lopez did not speak English until he was seven. Lopez attended Our Lady of Lourdes, a Roman Catholic elementary school in Pico Rivera.

Predicted Answer: San Fernando

Generated Context 5: George Lopez A theater called "The Tower", which now houses a branch of Bank of America, stands on the former site of his childhood home. Lopez was born at Daniel Freeman Memorial Hospital in Inglewood, California and was raised by his maternal grandmother, Benita Gutierrez, a factory worker who took him to the movies every week. He grew up in the Echo Park neighborhood of Los Angeles, California and attended many different schools in the Los Angeles Unified School District.

Predicted Answer: Los Angeles

Generated Context 6: George Lopez López was born in Mission Hills, Los Angeles, California. He is the son of Frieda and Anatasio López, Mexican immigrants from Tepic, Nayarit, Mexico. Lopez's father was a farm worker, and his mother worked at a canning factory. Lopez has stated that he had a difficult childhood; his parents divorced when he was five, and his mother struggled to raise him and his two sisters.

Predicted Answer: Mission Hills

Generated Context 7: George Edward Lopez. Lopez was born in Mission Hills, Los Angeles. His parents, Frieda and Anatasio Lopez, were both born and raised in Texas. His mother was of Mexican descent and his father, also called "Charlie", was of Sicilian descent. Lopez grew up with two older sisters. He attended San Fernando High School in the Los Angeles neighborhood of San Fernando.

Predicted Answer: Mission Hills

Generated Context 8: George Lopez had in his mouth. He was raised by his mother, Frieda, a telephone company employee, and his grandmother, Benita Gutierrez, a factory worker. Lopez graduated from the Roman Catholic Saint Monica's High School in 1974, where he met his future wife, Ann Serrano. He moved to California and graduated from the California State University, Los Angeles, in 1986 with a degree in communications.

Predicted Answer: Castle Hill

Predicted Answer (CGAP): Mission Hills

Table 11: Example illustrates CGAP, in which we marginalize over 8 generated contexts.

Q: Who was the first U.S. president to be born in the United States of America?

A: with the issue of 1863 (displayed in First appearances) above). At that time, as aforesaid, he had already appeared on two different Confederate 200a2 stamps. Andrew Jackson appears on the following commemorative issues: Martin Van Buren (December 5, 1782 - 2013 July 24, 1862) was the eighth President of the United States from 1837 to 1841. Before his presidency, he served as the eighth Vice President (1833 2013 1837) and the 10th Secretary of State under Andrew Jackson. Van Buren was the first U.S. President to be born an American citizen. William Henry Harrison (February 9, 1773 2013 April 4, 1841) was the

Q: Which US President who was in office 1923-29, was the only one to be born on 4th July?

A: Calvin Coolidge John Calvin Coolidge Jr. (; July 4, 1872 2013 January 5, 1933) was an American politician and the 30th President of the United States from 1923 to 1929. A Republican lawyer from New England, born in Vermont, Coolidge worked his way up the ladder of Massachusetts state politics, eventually becoming governor. His response to the Boston Police Strike of 1919 thrust him into the national spotlight and gave him a reputation as a man of decisive action. Soon after, he was elected Vice President of the United States in 1920, and succeeded to the presidency upon the sudden

Q: What long running comic strip, introduced on Oct 2, 1950, was drawn by the same cartoonist, Charles M. Schultz, during its' entire run?

A: Peanuts Peanuts is a syndicated daily and Sunday American comic strip written and illustrated by Charles M. Schulz that ran from October 2, 1950, to February 13, 2000, continuing in reruns afterward. "Peanuts" is among the most popular and influential in the history of comic strips, with 17,897 strips published in all, making it "arguably the longest story ever told by one human being. At its peak in the mid-to late 1960s, "Peanuts" ran in over 2,600 newspapers, with a readership of around 355 million in 75 countries, and was translated into 21 languages. It helped to cement the"

Q: In the comic strip "Peanuts", whose father is a barber?

A: Franklin (Peanuts) Franklin Armstrong is a character in the long-running comic strip "Peanuts", created by Charles M. Schulz. Introduced on July 31, 1968, Franklin was the first African American character in the strip. He goes to school with Peppermint Patty and Marcie. In his first story arc, he met Charlie Brown when they were both at the beach. Franklin's father was a soldier fighting in Vietnam, to which Charlie Brown replied "My dad's a barber...he was in a war too, but I don't know which one." Franklin later paid Charlie Brown a visit and found some of Charlie Brown's other

Q: Which US President was given the nickname 'The Trust Buster'?

A: Howard Taft. A Progressive reformer, Roosevelt earned a reputation as a "trust buster" through his regulatory reforms and anti-trust prosecutions. His presidency saw the passage of the Pure Food and Drug Act, which established the Food and Drug Administration to regulate food safety, and the Hepburn Act, which increased the regulatory power of the Interstate Commerce Commission. Roosevelt took care, however, to show that he did not disagree with trusts and capitalism in principle, but was only against monopolistic practices. His "Square Deal" included regulation of railroad rates and pure foods and drugs; he saw it as a fair deal

Q: American cartoonist Thomas Nast is credited with creating the traditional image of which popular Christmas character?

A: Thomas Nast Thomas Nast (September 27, 1840 December 7, 1902) was a German-born American caricaturist and editorial cartoonist considered to be the "Father of the American Cartoon". He was the scourge of Democratic Representative "Boss" Tweed and the Tammany Hall Democratic party political machine. Among his notable works were the creation of the modern version of Santa Claus (based on the traditional German figures of Sankt Nikolaus and Weihnachtsmann) and the political symbol of the elephant for the Republican Party (GOP). Contrary to popular belief, Nast did not create Uncle Sam (the male personification of the United

Q: October 2, 1950 saw the debut of what comic strip which ran until creator Charles M. Schulz's death on Feb 12, 2000?

A: from a wide range of acclaimed contemporary artists and designers who have been inspired by the cartoon. Peanuts Peanuts is a syndicated daily and Sunday American comic strip written and illustrated by Charles M. Schulz that ran from October 2, 1950, to February 13, 2000, continuing in reruns afterward. "Peanuts" is among the most popular and influential in the history of comic strips, with 17,897 strips published in all, making it "arguably the longest story ever told by one human being". At its peak in the mid- to late 1960s, "Peanuts" ran in over 2,600 newspapers, with a readership of

Q: In which state was Charles Schulz born?

A: Charles M. Schulz Charles Monroe "Sparky" Schulz (November 26, 1922 February 12, 2000), nicknamed Sparky, was an American cartoonist. Schulz is known for the comic strip "Peanuts" (which featured the characters Charlie Brown and Snoopy, among others). He is widely regarded as one of the most influential cartoonists of all time, cited by cartoonists including Jim Davis, Bill Watterson, and Matt Groening. Born in Minneapolis, Minnesota, Schulz grew up in Saint Paul. He was the only child of Carl Schulz, who was born in Germany, and Dena Halverson, who had Norwegian heritage. His uncle called him "Sparky" after

Q: **Who was President when the first Peanuts cartoon was published?**

Table 12: $Prompt(Q)$ Example. For the question "**Who was President when the first Peanuts cartoon was published?**" from TQA (Joshi et al., 2017), we selected 8 $\langle q_i, c_i \rangle$ samples from the supporting repository \mathcal{D} , and construct the $Prompt(Q)$ as above. to prompt LMs for c_{gen} generation.

RedHOT: A Corpus of Annotated Medical Questions, Experiences, and Claims on Social Media

Somin Wadhwa[†] Vivek Khetan[◇] Silvio Amir[†] Byron C. Wallace[†]
Northeastern University[†] Accenture AI Labs[◇]
{wadhwa.s,s.amir,b.wallace}@northeastern.edu
vivek.a.khetan@accenture.com

Abstract

We present **Reddit Health Online Talk (RedHOT)**, a corpus of 22,000 richly annotated social media posts from Reddit spanning 24 health conditions. Annotations include demarcations of spans corresponding to medical claims, personal experiences, and questions. We collect additional granular annotations on identified claims. Specifically, we mark snippets that describe patient **Populations**, **Interventions**, and **Outcomes** (PIO elements) within these. Using this corpus, we introduce the task of retrieving trustworthy evidence relevant to a given claim made on social media. We propose a new method to automatically derive (noisy) supervision for this task which we use to train a dense retrieval model; this outperforms baseline models. Manual evaluation of retrieval results performed by medical doctors indicate that while our system performance is promising, there is considerable room for improvement. We release all annotations collected (and scripts to assemble the dataset), and all code necessary to reproduce the results in this paper at: <https://sominw.com/redhot>.

1 Introduction

Social media platforms such as Reddit provide individuals places to discuss (potentially rare) medical conditions that affect them. This allows people to communicate with others who share in their condition, exchanging information about symptom trajectories, personal experiences, and treatment options. Such communities can provide support (Biyani et al., 2014) and access to information about rare conditions which may otherwise be difficult to find (Glenn, 2015).

However, the largely unvetted nature of social media platforms make them vulnerable to *mis* and *disinformation* (Swire-Thompson and Lazer, 2019). An illustrative and timely example is the idea that consuming bleach might be a viable treatment for

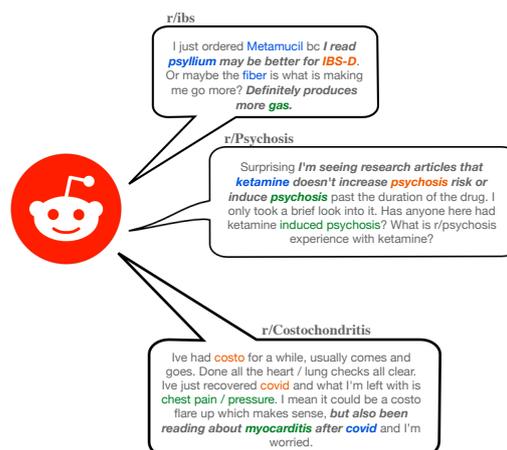


Figure 1: Examples of health-related Reddit posts annotated for populations, interventions, and outcomes.

COVID-19,¹ which quickly gained traction on social media. All misinformation can be dangerous, but *medical* misinformation poses unique risks to public health, especially as individuals increasingly turn to social media to inform personal health decisions (Nobles et al., 2018; Barua et al., 2020).

In this paper, we introduce **RedHOT**: an annotated dataset of health-related claims, questions, and personal experiences posted to Reddit. This dataset can support development of a wide range of models for processing health-related posts from social media. Unlike existing health-related social media corpora, **RedHOT**: (a) Covers a broad range of health topics (e.g., not just COVID-19), and, (b) Comprises “natural” claims collected from real health-related fora (along with annotated questions and personal experiences). Furthermore, we have collected granular annotations on claims, demarcating descriptions of the **Population** (e.g., **diabetics**), **Interventions**, and **Outcomes**, i.e., the *PIO* elements (Richardson et al., 1995). Such annotations may permit useful downstream processing: For exam-

¹<https://www.theguardian.com/world/2020/sep/19/bleach-miracle-cure-amazon-covid>

ple, in this work we use them to facilitate retrieval of evidence relevant to a claim.

Specifically, we develop and evaluate a pipeline to automatically identify and contextualize health-related claims on social media, as we anticipate that such a tool might be useful for moderators keen to keep their communities free of potentially harmful misinformation. With this use-case in mind, we propose methods for automatically retrieving *trustworthy* published scientific evidence relevant to a given claim made on social media, which may in aggregate support or debunk a particular claim.

The contributions of this work are summarized as follows. First, we introduce **RedHOT**: A new dataset comprising 22,000 health-related Reddit posts across 24 medical conditions annotated for claims, questions, and personal experiences. Claims are additionally annotated with PIO elements. Second, we introduce the task of identifying health-related claims on social media, extracting the associated PIO elements, and then retrieving relevant and trustworthy evidence to support or refute such claims. Third, we propose **RedHOT-DER**, a Dense Evidence Retriever trained with heuristically derived supervision to retrieve medical literature relevant to health-related claims made on social media. We evaluate baseline models for the first two steps on the **RedHOT** dataset and assess the retrieval step with relevance judgments collected from domain experts (medical doctors).

The Reddit posts we have collected are public and typically made under anonymous pseudonyms, but nonetheless these are health-related comments and so inherently sensitive. To respect this, we (a) notified all users in the dataset of their (potential) inclusion in this corpus, and provided opportunity to opt-out, and, (b) we do not release the data directly, but rather a script to download annotated comments, so that individuals may choose to remove their comments in the future. Furthermore, we consulted with our Institutional Review Board (IRB) and confirmed that the initial collection and annotation of such data does not constitute human subjects research. However, EAACL reviewers rightly pointed out that certain *uses* of this data may be sensitive. Therefore, to access the collected dataset we require researchers to self-attest that they have obtained prior *approval* from their own IRB regarding their intended use of the corpus.

2 The **RedHOT** Dataset

We have collected and manually annotated health related posts from Reddit to support development of language technologies which might, e.g., flag potentially problematic claims for moderation. Reddit is a social media platform that allows users to create their own communities (*subreddits*) focused on specific topics. Subreddits are often about niche topics, and this permits in-depth discussion catering to a long tail of interests and experiences. Notably, subreddits exist for most common (and many rare) medical conditions; we can therefore sample posts from such communities for annotation.

2.1 Data Annotation

We decomposed data annotation into two stages, performed in sequence. In the first, workers are asked to demarcate spans of text corresponding to a Claim, Personal Experience, or Question. We characterize these classes as follows (we provide detailed annotation instructions in Appendix A):

Claim suggests (explicitly or implicitly) a causal relationship between an **Intervention** and an **Outcome** (e.g., “*I* completely cured my *O*”). Operationally, we are interested in identifying statements that might reasonably be interpreted by the reader as implying a causal link between an intervention and outcome, as this may in turn influence their perception regarding the efficacy of an intervention for a particular condition and/or outcome (i.e., relationship between an *I* and *O*).

Question poses a direct question, e.g., “Is this normal?”; “Should I increase my dosage?”.

Personal Experience describes an individual’s experience, for instance the trajectory of their condition, or experiences with specific interventions.

This is a *multi-label* scheme: Spans can (and often do) belong to more than one of the above categories. For example, personal experiences can often be read as implying a causal relationship. Consider this example: “My doctor put me on *I* for my *P*, and I am no longer experiencing *O*”. This describes an individual treatment history, but could also be read as implying that *I* is a viable treatment for *P* (and specifically for the outcome *O*). Therefore, we would mark this as both a Claim and a Personal Experience. By contrast, a general statement asserting a causal relationship outside of any personal context like “*I* can cure *O*” is what

Reddit post	Span labels	PIO elements from claims
<i>I've seen a bunch of posts on here from people who say that glycopyrrolate suddenly isn't working anymore for hyperhidrosis. I'm one of those person who has been facing this for a while now. Just wondering if anyone fixed it? Can't really ask my GP about it since he didn't even know the meds existed. He just prescribed them for me when I asked for it</i>	Claim: I've seen a bunch of posts on here from people who say that glycopyrrolate suddenly isn't working anymore for Hyperhidrosis Question: Just wondering if anyone fixed it?	P hyperhidrosis I glycopyrrolate
<i>so i recently read that adderall can trigger a psychotic break & i was prescribed adderall years ago for my adhd but now i just have constant hallucination episodes. anyone else experience adderall induced psychosis?</i>	Claim: so i recently read that adderall can trigger a psychotic break Personal Experience: i was prescribed adderall years ago for my adhd but now i just have constant hallucination episodes Question: anyone else experience adderall induced psychosis?	P adhd I adderall O hallucinations
<i>I've had costochondritis for a while, usually comes and goes. Done all the heart/lung checks all clear. I've just recovered covid and what I'm left with is chest pain/pressure. I mean it could be a costo flare up which makes sense, but also <i>been reading about myocarditis after covid</i> and I'm worried, how can I tell which is which?</i>	Claim: been reading about myocarditis after covid Personal Experience: I'm left with is chest pain/pressure Question: how can I tell which is which?	P costochondritis I covid O myocarditis, chest-pain

Table 1: Example annotations, which include: extracted spans (phase 1), and spans describing **Populations**, **Interventions**, and **Outcomes** — PIO elements — within them (phase 2). We collect the latter only for claims.

we will refer to as a “pure claim”, meaning it exclusively belongs to the Claim category.

In the second stage, workers are asked to further annotate “pure claim” instances by marking spans within them that correspond to the Populations, Interventions/Comparators,² Outcomes (the PIO elements) associated with the claim.

2.2 Crowdsourcing Annotations

We hired crowdworkers to perform the above annotation tasks on Amazon Mechanical Turk (AMT).³ To estimate required annotation time and determine fair pay rates, we ran an internal pilot with two PhD students (both broadly familiar with this research area) on 100 samples.⁴ To gauge quality and recruit workers from AMT, we ran two pilot experiments in which we collected sentence-level annotations on posts sampled from three medical populations (i.e., subreddits), comprising ~6,000 posts in all.

We required all workers have an overall job approval rate of $\geq 90\%$. Based on an initial set of AMT annotations we re-hired only workers who

²This is the standard PICO framework, but we collapse Interventions and Comparators into the Intervention category, as the distinction is arbitrary.

³We consulted with an Institutional Review Board (IRB) to confirm that this annotation work did not constitute human subjects research.

⁴Based on the estimate from our pilot experiments, payrate for AMT workers was fixed to US \$9 per hour for stage-1 annotations and US \$11 per hour for stage-2 annotations, irrespective of geographic location.

	Fliess κ	P	R	F1
Questions	0.86	0.85	0.82	0.84
Claims	0.69	0.63	0.53	0.58
Experiences	0.71	0.78	0.69	0.73
POP	0.92	0.94	0.91	0.92
INT	0.74	0.76	0.70	0.73
OUT	0.78	0.73	0.68	0.70

Table 2: Token-wise label agreement among experts measured by Fleiss κ on a subset of data. We further compute precision, recall, and F1 scores for “aggregated” labels by evaluating them against unioned “in-house” expert labels.

reliably followed annotation instructions (details in Appendix A), and we actively recruited the top workers to continue on with increased pay. We obtained annotations from at least three workers for each post, allowing for robust inference of reference labels. Recruited workers were also paid periodic bonuses (equivalent to two hours of pay) based on the quality of their annotated samples.

2.3 Quality Validation

To evaluate annotation quality we calculate token-wise label agreement between annotators, and amongst ourselves. We emphasize here that token-level κ for sequences is quite strict and disagreements often reflect *where* annotators decide to mark

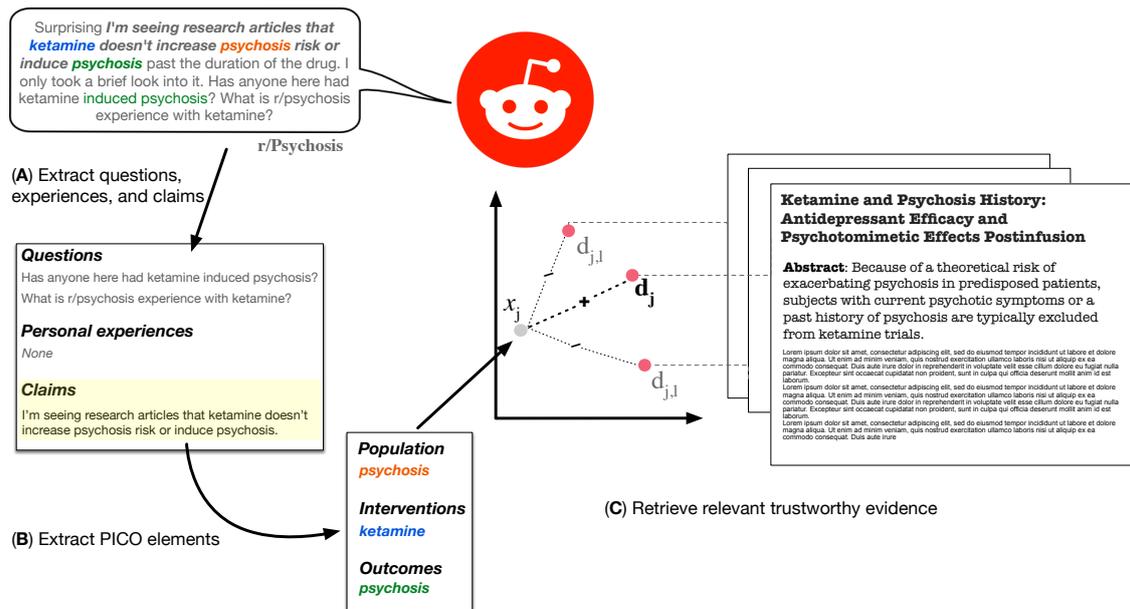


Figure 2: Examples portraying potential use cases of our corpus. We showcase three distinct tasks, to be performed in sequence. The first (A) entails extracting spans corresponding to claims (highlighted in **bold**) from a given Reddit post. The second step (B) is to identify the PICO elements associated with each claim. In the final step (C), we use the outputs of the first two models with the original post to obtain a dense representation, enabling us to retrieve relevant evidence from a large dataset of trusted medical evidence (e.g., PubMed).

span boundaries. Despite this, for the first stage agreement (Fleiss κ) on labeled questions, experiences, and claims was 0.62, and for the second stage 0.55. We consider this *moderately strong* agreement, in line with agreement reported for related annotation tasks in the literature (Nye et al., 2018; Deléger et al., 2012). To quantify this and further gauge the quality of collected annotations, we run a few additional analyses.

As previously stated, prior to collecting annotations on Amazon MTurk, we (the authors) annotated a subset of data (100 samples/stage) internally to assess task difficulty and to estimate the time required for annotation. As an additional quality check, we use these annotations to calculate token-wise label agreement. Table 2 reports the results; while there remains some discrepancy owing to the inherent complexity of the task, there is higher agreement between the us than between workers.

Each of these samples was also annotated by three workers. We aggregate these labels using majority-vote and compute token-wise precision-recall of these aggregated labels against the reference “in-house” labels (Table 2). We report the same metrics per annotator evaluated against aggregated MTurk labels in Table 9 (Appendix B). Despite moderate agreement between annotators, aggregated labels agree comparatively well with

the “expert” consensus, indicating that while individual worker annotations are somewhat noisy, aggregated annotations are reasonably robust.

2.4 Dataset Details

Table 1 provides illustrative samples from RedHOT and Table 8 provides some descriptive statistics along with examples of included health populations. We broadly characterize populations (conditions) as *Very Common*, *Common* or *Rare*, and sought a mix of these. This was not the only attribute that informed which conditions we selected for inclusion in our dataset, however. For example, we wanted a mix of populations with respect to volume of online activity (e.g., the Diabetes subreddit has over 60k active visitors; Lupus has 8k). We also wanted to include both chronic and treatable conditions (e.g., Narcolepsy is a rare and chronic condition, while Gout is common and treatable), and mental and physical disorders (e.g., ADHD, Rheumatoid Arthritis). Another consideration was whether a condition can be self-diagnosed or requires professional assessment (e.g., Bulimia is usually self-diagnosable but can potentially be life-threatening; Gastroparesis is chronic but requires a professional medical diagnosis).

The number of *claims* across different categories of health populations are far outnumbered by *ques-*

tions ($\sim 10x$) and experiences ($\sim 13x$). The average post length is ~ 117 tokens while the average length of a claim within a post is ~ 20 tokens. Questions and experiences have average lengths of ~ 11 and ~ 27 tokens, respectively. We provide per condition statistics in Appendix B.

3 Tasks and Evaluation

RedHOT may support a range of tasks related to processing health-related social media posts. Here we focus on an important, timely task: Identifying medical claims on social media, and then retrieving relevant and trustworthy evidence that may support or refute them. Methods for this task could aid content moderation on health-related forums, by providing an efficient means to (in)validate claims. More generally, such methods may permit meaningful “fact checking” of health-related claims by providing relevant contextualizing evidence.

We outline a three-step approach for this task. (1) Identify spans/sentences corresponding to *pure* claims. (2) Extract from these specific PICO elements. (3) Retrieve clinical literature — specifically, reports of RCTs — relevant to the claim, i.e., the extracted PIO elements. We limit our focus to the problem of evidence retrieval here; future work might consider the subsequent step of automated claim validation on the basis on this.

Below we assess components for each of these steps. For the span and PIO extraction steps (1 and 2), we evaluate models retrospectively under standard classification metrics (i.e. precision, recall, and F1 scores) using fixed train, development, and test sets which we will distribute with **RedHOT**. The final step (3) requires relevance judgments to evaluate model performance; for this we enlisted medical doctors (Section 3.4).

3.1 Identifying Claims, Experiences, Questions, and PIO Elements

We treat the first two steps as sequence tagging tasks for which we evaluate two types of models: A simple linear-chain Conditional Random Field (CRF; Lafferty et al. 2001), and Transformer-based models (Vaswani et al., 2017) — specifically BERT variants (Devlin et al., 2019; Liu et al., 2019).⁵ The features for the CRF we use are: Indicators of next, previous, and current words; Part-of-speech tags,⁶

⁵We also explored t5 (Raffel et al., 2020) with middling results, which we report in the Appendix.

⁶Extracted with SciSpacy (<https://allenai.github.io/scispacy/>).

and; Indicators encoding if sentences contain digits, uppercase letters, and/or measurement units. BERT variants yield contextualized representations of input tokens, which we then use to predict labels (i.e., Claims, Experiences and Questions) by adding a linear layer on top of the encoder outputs. PIO elements are extracted using a concatenated input of the original Reddit post and an identified claim.

3.2 Evidence Retrieval

For the retrieval task we assume the model is given: (i) The original Reddit post and a claim; (ii) PIO elements associated with that claim, and; (iii) A large set of candidate articles featuring trustworthy evidence to rank. We use $\sim 800,000$ abstracts from *Trialstreamer*⁷ (Marshall et al., 2020), a continuously updated database of reports of randomized controlled trials (RCTs). RCTs are appropriate here because of our focus on causal claims — results from randomized trials are the most reliable means of evaluating such assertions (Meldrum, 2000).

3.2.1 Task Formulation

Formally, we represent a single input instance as $(p, c_j, \text{pop}_j, \text{int}_j, \text{out}_j)$ where p is a post comprising n sentences, c_j is the j th claim, and $\text{pop}_j, \text{int}_j, \text{out}_j$ are the sets of populations, interventions, outcomes associated with claim j .

The model is tasked with finding relevant abstracts from the candidate set \mathcal{A} , which comprises abstracts from published clinical trial reports. This is particularly challenging because a large number of candidates can mention the same set of PIO entities (i.e., investigate the same interventions and/or outcomes), but in a context unrelated to the claim being made in the social media post. This may be especially problematic for retrieval methods based primarily on string overlap measures. We therefore propose a learning based approach. This requires supervision; we next describe our approach to deriving this automatically.

3.2.2 Pseudo Training Data

Supervised neural retrieval models require annotations indicating the relevance of instances (here, published evidence) to inputs (claims on social media). We do not have such judgments, and so instead derive “pseudo” training data automatically.

We started with $\sim 800,000$ abstracts of medical RCTs in *Trialstreamer*. We then used Reddit

⁷<https://trialstreamer.ieai.robotreviewer.net/>

	P	R	F1	F1 _{POP}	F1 _{INT}	F1 _{OUT}
BERT (Devlin et al., 2019)	43.88	36.13	39.62	41.77	44.68	33.05
BioRedditBERT (Basaldella et al., 2020)	44.44	36.55	40.12	41.92	44.31	34.61
biomedRoBERTa (Gururangan et al., 2020)	38.80	21.48	27.66	30.54	28.13	24.54
RoBERTa (Liu et al., 2019)	47.45	39.27	42.97	46.09	45.99	36.38
t5-small (Raffel et al., 2020)	41.49	38.55	39.97	39.61	45.02	32.41

Table 3: Results on the test set for the token-level PICO tagging task.

	Claims			Experiences			Questions		
	F1	P	R	F1	P	R	F1	P	R
CRF (Lafferty et al., 2001)	33.87	35.61	32.29	40.08	40.52	39.64	86.89	85.55	88.27
BERT (Devlin et al., 2019)	52.63	58.82	47.61	56.68	59.46	54.33	92.39	88.76	96.34
RoBERTa (Liu et al., 2019)	47.05	61.53	38.09	56.81	57.11	56.52	93.06	89.01	98.34
BioRedditBERT (Basaldella et al., 2020)	45.16	70.92	33.29	59.51	62.49	58.92	93.61	89.29	98.37

Table 4: Results on the test-set of span-classification to identify pure claims, questions, and experiences.

posts containing pure claims as *templates* to create pseudo matches between medical claims and abstracts. Specifically, we substituted annotated PIO elements in claims made within Reddit posts with PIO elements sampled from Trialstreamer abstracts. (Trialstreamer includes PICO elements automatically extracted from all articles that it indexes.) This yields pairs of (a) naturally occurring claims (with their PIO spans replaced) and (b) RCT abstracts that are relevant to said claims by construction. We provide examples of this pseudo matching in Appendix D. We generated a total of 85,000 examples of (pseudo claims, evidence abstract) one-to-many pairs to be used to train a neural retrieval model (described below). The generated examples may be noisy, but hopefully sufficient to train a model to retrieve medical abstracts relevant to health-related claims made on social media.

3.2.3 RedHOT Dense Evidence Retriever

We train a neural retrieval model on the RedHOT corpus, using a setup similar to DPR (Karpukhin et al., 2020). We first assemble a collection of m RCT abstracts to create an evidence corpus, $\mathcal{A} = \{d_1, d_2, \dots, d_m\}$. There are hundreds of thousands of RCTs, so we need an *efficient* retriever that can select a small set of relevant abstracts. Formally, a retrieval operation $\{R: (x_j, \mathcal{A}) \rightarrow \mathcal{A}_{\mathcal{F}}\}$ accepts an input contextualizing string x_j and a corpus of evidence \mathcal{A} , and returns a *much smaller* filtered set $\mathcal{A}_{\mathcal{F}} \subset \mathcal{A}$, where $|\mathcal{A}_{\mathcal{F}}| = k$.

We form an input context string x_j for a claim j made within a post p by concatenating the post, claim, and PIO elements extracted from the claim:

$x_j = [p \oplus c_j \oplus \text{pop}_j \oplus \text{int}_j \oplus \text{out}_j]$, where \oplus denotes concatenation with [SEP] tokens. We define two dense neural encoders (E_C, E_D ; both initialized with RoBERTa-base) to project the context string x_j , and evidence (abstracts) from \mathcal{A} to fixed 768 dimensional vectors. Similarity between the context string and evidence abstract is defined using the dot product of their vectors, $\phi(x_j, d_l) = E_C(x_j)^T E_D(d_l)$.

We train the model to minimize the negative log-likelihood of the positive evidence such that it pushes the context string vector x_j close to the representation of relevant evidence d_j^+ , and away from b irrelevant abstracts ($d_{j1}^-, d_{j2}^-, \dots, d_{jb}^-$) in the same mini-batch⁸ (“in-batch negative sampling”):

$$\mathcal{L} = \frac{\exp \phi(x_j, d_j^+)}{\exp \phi(x_j, d_j^+) + \sum_{l=1}^b \exp \phi(x_j, d_{jl}^-)}$$

In-batch negative sampling has been shown to be effective for dual-encoder training (Henderson et al., 2017; Gillick et al., 2019). Here, all samples in a minibatch are taken from the same *population* (condition) set, e.g., a mini-batch with a sample containing a claim about diabetes will have negative evidence abstracts that are also related to diabetes.

For test examples, we rank all evidence (abstracts in Trialstreamer) according to their similarity to the context string. To do this efficiently, we induce representations of all the abstracts in the Trialstreamer database using the evidence encoder and index these using the Facebook AI Similarity Search library (Johnson et al., 2021).⁹

⁸We set the size of the mini-batch to 100.

⁹FAISS: Open-source library for efficient similarity search

k	MRR @k					Precision @k				
	1	5	10	50	100	1	5	10	50	100
random	0.00	0.003	0.02	0.02	0.02	0.00	0.02	0.00	1.10	2.80
BM25	5.34	7.98	9.86	14.36	16.70	5.34	10.40	14.45	26.20	33.14
DPR (Karpukhin et al., 2020)	8.07	10.96	11.89	12.20	13.77	8.07	16.50	23.58	31.98	36.87
<i>(trained on the RedHOT pseudo training set)</i>										
RedHOT-DER (BERT-based)	39.14	47.99	49.3	50.28	50.35	39.14	62.55	72.64	83.73	91.74
RedHOT-DER (RoBERTa-based)	45.93	54.60	55.90	56.73	56.78	45.93	69.90	78.81	94.73	98.06

Table 5: Results of evidence retrieval baselines evaluated on pseudo test data.

3.2.4 Baseline Models

BM25 A standard Bag-of-Words method for IR (Robertson et al., 1995). We form queries by concatenating the Reddit post with a single claim and its corresponding PIO frames. We used a publicly available BM25 implementation from the Rank-BM25 library.¹⁰

Dense Passage Retrieval (DPR) is a dense retrieval model trained to retrieve *relevant* context spans (“paragraphs”) in an open domain question-answering setting (Karpukhin et al., 2020). In general, such models map **queries** and **candidates** to embeddings, and then rank candidates with respect to a similarity measure (e.g., dot product) taken between these. While originally designed for open-domain question answering, use of DPR-inspired models has been extended to general retrieval tasks (Thai et al., 2022a). We use a DPR context encoder trained on Natural Questions (Kwiatkowski et al., 2019) with dot product similarity.¹¹

3.3 Results

We evaluate models for the tasks of identifying claims, experiences, and questions and extracting PIO elements using precision, recall, and F1 scores. We report results per class for the first task in Table 4. BioRedditBERT (Basaldella et al., 2020) — a BERT model initialized from BioBERT (Lee et al., 2019) and further pre-trained on health-related Reddit posts — fares best here. We report results for the second task (PIO tagging) in Table 3.¹² Here RoBERTa (Liu et al., 2019) modestly outperforms BioRedditBERT (Basaldella et al., 2020).

and clustering of dense vectors; <https://ai.facebook.com/tools/faiss/>.

¹⁰https://github.com/dorianbrown/rank_bm25

¹¹<https://huggingface.co/facebook/dpr-ctx-encoder-single-nq-base>

¹²Results from additional experiments using other model variants are reported in Appendix C.

Models for the retrieval task rank evidence candidates for each input (post, claim, PIO frame). We therefore use standard ranking metrics for evaluation, including mean reciprocal rank, and precision@ k (for $k = 1, 5, 10, 50, 100$). Baseline results are reported in Table 5. We emphasize that these results are with respect to pseudo annotated data, effectively providing an unfair advantage to RedHOT-DER, given that this was optimized on data from this distribution. We report results with respect to manual relevance judgments provided by experts in Section 3.4.

As we might expect, the pre-trained neural DPR model outperforms the naive string matching BM25 method. Furthermore, as anticipated, explicitly training for evidence retrieval confers pronounced advantages: RedHOT-DER fares $\sim 8x$ better than BM25 and $\sim 5x$ better than “off-the-shelf” pre-trained DPR (Karpukhin et al., 2020) with respect to retrieving relevant evidence (precision@1) corresponding to medical claims. Again, this is not particularly surprising given that we are evaluating models with respect to the pseudo annotations with which RedHOT-DER was trained (because we do not otherwise have access to explicit relevance judgments). Therefore, we next present results from more meaningful manual relevance evaluations performed by domain experts.

3.4 Expert Manual Relevance Judgments

We evaluated models in terms of retrieving evidence relevant to *naturally occurring* medical claims, as opposed to the *pseudo* data derived for training. We hired three domain experts (medical doctors) on the Upwork platform.¹³ Providing hundreds of retrieved medical abstracts per claim to a human evaluator for assessment is infeasible, so

¹³Upwork (<https://www.upwork.com/>) allows clients to interview, hire and work with freelancers. All of our evaluators had medical degrees and were hired at wages ranging from \$15 to \$20 per hour for a minimum of 15 hours.

Cumulative # of relevant abstracts @ k				
k	1	3	5	10
<i>Pre-trained DPR (Karpukhin et al., 2020)</i>				
Relevant	6	16	29	58
Somewhat relevant	14	39	66	135
Irrelevant	80	245	405	807
<i>RedHOT-DER trained on pseudo data</i>				
Relevant	18	62	101	201
Somewhat relevant	17	49	87	193
Irrelevant	65	189	312	606

Table 6: Results from manual (domain expert) evaluations for DPR and our pseudo-supervised DER model.

we instead provided evaluators with 10 retrieved abstracts each for 100 individual claims, retrieved using the pretrained DPR (Karpukhin et al., 2020) model and our **RedHOT**-DER trained on pseudo data. (We compared the proposed distantly supervised model to DPR because it is the strongest baseline we evaluated in preliminary experiments.)

We asked evaluators to categorize each retrieved abstract as: (1) Relevant; (2) Somewhat Relevant, or; (3) Irrelevant to the corresponding claim. An abstract was to be considered Relevant if and only if it (1) contained to the same **P**, **I**, and **O** elements mentioned in the original Reddit post, **and** (2) provided information to support or refute the claim in question. An abstract might be deemed Somewhat Relevant if it contains a **P**, **I**, and **O** set in line with the given claim, but does not provide any information relating these elements. We provide examples in the Appendix D.

Human evaluators achieve strong agreement: All three evaluators chose the same relevance label 71.33% of the time, while they all chose a different label only in 1.29% of the total instances. They also show substantial agreement in terms of Fleiss κ (0.71). We derive final relevance labels by majority vote. Comparing results from Table 5 and Table 6, at $k = 1$ we see similar values of precision in the manually annotated data and pseudo test data. However, for higher values of k large differences emerge, indicating considerable room for improvement. Compared to the pre-trained DPR model, at $k = 1$ **RedHOT**-DER retrieves a substantially larger fraction of relevant evidence abstracts (3x). At higher k , we also observe a large reduction in the number of *irrelevant* abstracts retrieved (e.g., at $k = 10$, the number of irrelevant abstracts de-

creases by $\sim 30\%$). We believe this highlights the value of our proposed distant supervision scheme.

4 Related Work

Claim validation via evidence retrieval Past work has typically treated (open domain) claim validation as a two-step process in which one retrieves evidence relevant to a given claim, and then makes a prediction regarding claim validity on the basis of this. Information retrieval (IR) models are usually used in the first step to rank order documents based on relevance to a given claim (Thorne et al., 2018; Wadden et al., 2020; Thai et al., 2022b; Hanselowski et al., 2018; Samarin et al., 2021; Saeed et al., 2021). The next step is usually to characterize *retrieved* evidence as supporting, refuting, or not providing enough information (although this latter category is not always included). Evidence might be individually characterized (Pradeep et al., 2021), or aggregated to make a single prediction about the veracity of the claim (Sarroufi et al., 2021).

Scientific claim verification Beyond “general domain” verification, there have been efforts focused specifically on vetting *scientific* claims. SciFact (Wadden et al., 2020) largely follows the typical fact verification setup outlined above (but for scientific claims). Subsequent efforts have focused specifically on verifying claims related to COVID-19 (Saakyan et al., 2021). The evidence inference task (Lehman et al., 2019; DeYoung et al., 2020) entails inferring whether a given trial report supports a significant effect concerning a specific intervention, comparator, and outcome.

Crowd-sourcing annotation of scientific and medical texts We have relied on crowdworkers to annotate the instances comprising **RedHOT**. This is in keeping with a body of work that has shown crowdworkers capable of annotating health-related texts, even when these are technical (Drutsa et al., 2021). For example, several past efforts have crowdsourced annotation of texts drawn from PubMed, e.g. for mentions of diseases (Nye et al., 2018; Good et al., 2014). More recently, Bogensperger et al. (2021) crowdsourced a dataset of drug mentions (a type of *intervention*) on the darknet. Khetan et al. (2022) crowdsourced annotations of electronic health records to identify causal relations between medical entities. Similarly, there is a body of work relying on crowdsourcing to accom-

plish a diverse set of domain-specific non-medical NLP tasks (Sukhareva et al., 2016; Fromreide et al., 2014; Bhardwaj et al., 2019; Gardner et al., 2020).

Health-related Reddit corpora Past work has also built corpora of health-related Reddit posts. For example, Cohan et al. (2018) assembled a dataset of Reddit posts made by individuals who self-reported one of nine mental health diagnoses of interest. Building on this work, Jiang et al. (2020) introduced a dataset of Reddit posts to evaluate models for automatically detecting psychiatric disorders.

5 Conclusions

We presented **RedHOT**: a new, publicly available dataset comprising of about 22,000 richly annotated Reddit posts extracted from 24 medical condition-based communities (“subreddits”). This dataset meets a need for corpora that can facilitate development of language technologies for processing health-related social media posts.

We evaluated baseline models for categorizing posts as containing claims, personal experiences, and/or questions. Focusing on claims, we then proposed and evaluated models for extracting descriptions of populations, interventions, and outcomes, and then using such snippets to inform retrieval of trustworthy (published) evidence relevant to a given claim. To this end, we introduced a heuristic supervision strategy, and found that this outperformed pre-trained retrieval models.

Limitations

We have introduced a new annotated dataset of medical questions, experiences, and claims across a range of health populations from social media. We showed that this data can be used to train models potentially useful for downstream applications, e.g., by facilitating content moderation. However, there are important limitations to this work, specifically with respect to the raw data we sampled and the annotations on this that we have collected.

First, the dataset we have annotated is inherently limited. While we have tried to select a diverse set of health populations (i.e., subreddits), these nonetheless constitute a small sample of the diverse set of existing health conditions. Moreover, our selection has led to a corpus comprising nearly entirely of English-language posts, which is a clear limitation.

We relied on non-expert (layperson) workers from Amazon Mechanical Turk (AMT) to carry out

the bulk of annotation work. While we took steps to try and ensure annotation quality (described in Section 2), we nonetheless acknowledge that these annotations will contain noise. This is especially true given that AMT workers are not medical-experts and ultimately do not have (nor are they expected to have) sufficient knowledge of different kinds of medical terms appearing in the dataset (e.g., SSRIs stand for *selective serotonin reuptake inhibitor* and is a common form of intervention which may lead to outcomes like dizziness, anxiety, and/or insomnia, but many laypeople might simply be unaware of ordinary meaning of complicated medical terms leading them to *not* matching all or part of such terms to their respective labels).

In Section 3.2.2, we describe how we obtained *pseudo* training labels to build a supervised dense retriever. To generate this data, several natural language claims get reused with substitute set of populations/interventions/outcomes. This heuristic may induce certain biases (as evident from Table 6 and Table 5). An ideal way to train a dense retriever here would be to collect positive annotation labels for *every* claim in our dataset. Collecting such supervision at scale sufficient for model training would be expensive, given that one would strongly prefer expert (medical doctor) annotations concerning the factual accuracy of claims.

Ethics Statement

This work has the potential to contribute to human well-being by supporting development of language technologies for processing health-related social media posts. Such models might in turn provide insights about patient experiences and viewpoints in general, and more specifically may help community moderators identify and remove posts containing medical misinformation.

Realizing these potentially positive contributions requires annotated data with which to train relevant models; such data is the main contribution on offer in this work. However, releasing an annotated corpus of health-related social media posts raises concerns regarding individual privacy. The Reddit posts we have assembled and collected annotations were posted publicly on the Internet (almost always under pseudonyms), but nonetheless we have taken steps to ensure that individuals can choose not to be represented in this dataset.

Specifically, we sent a message to every user in the **RedHOT** explaining our intent to construct and

release this dataset and offering the option to “opt out”. In addition, although this is not required by Reddit, we have decided not to release the collected *posts* directly. Instead we release a script that will download the posts comprising our data on-demand and align these with the collected annotations. This means that if a user chooses to delete their post(s) from Reddit, they will also effectively be removed from our dataset. Further, we require anyone accessing this data to self-certify that they have obtained prior approval from their own IRB concerning the use-cases of their research.

Acknowledgements

This work was supported in part by the National Science Foundation (NSF) CAREER award 1750978.

References

- Zapan Barua, Sajib Barua, Salma Aktar, Najma Kabir, and Mingze Li. 2020. [Effects of misinformation on covid-19 individual responses and recommendations for resilience of disastrous consequences of misinformation](#). *Progress in Disaster Science*, 8:100119.
- Marco Basaldella, Fangyu Liu, Ehsan Shareghi, and Nigel Collier. 2020. [COMETA: A corpus for medical entity linking in the social media](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3122–3137, Online. Association for Computational Linguistics.
- Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. [CaRB: A crowdsourced benchmark for open IE](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6262–6267, Hong Kong, China. Association for Computational Linguistics.
- Prakhar Biyani, Cornelia Caragea, Prasenjit Mitra, and John Yen. 2014. Identifying emotional and informational support in online health communities. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 827–836.
- Johannes Bogenasperger, Sven Schlarb, Allan Hanbury, and Gábor Recski. 2021. [DreamDrug - a crowdsourced NER dataset for detecting drugs in darknet markets](#). In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 137–157, Online. Association for Computational Linguistics.
- Arman Cohan, Bart Desmet, Andrew Yates, Luca Soldaini, Sean MacAvaney, and Nazli Goharian. 2018. [SMHD: a large-scale resource for exploring online language usage for multiple mental health conditions](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1485–1497, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Louise Deléger, Qi Li, Todd Lingren, Megan Kaiser, Katalin Molnár, Laura Stoutenborough, Michal Kouril, Keith A. Marsolo, and Imre Solti. 2012. Building gold standard corpora for medical natural language processing tasks. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2012:144–53.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jay DeYoung, Eric Lehman, Ben Nye, Iain J Marshall, and Byron C Wallace. 2020. Evidence inference 2.0: More data, better models. *arXiv preprint arXiv:2005.04177*.
- Alexey Drutsa, Dmitry Ustalov, Valentina Fedorova, Olga Megorskaya, and Daria Baidakova. 2021. [Crowdsourcing natural language data at scale: A hands-on tutorial](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 25–30, Online. Association for Computational Linguistics.
- Hege Fromreide, Dirk Hovy, and Anders Søgaard. 2014. [Crowdsourcing and annotating NER for Twitter #drift](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 2544–2547, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Rachel Gardner, Maya Varma, Clare Zhu, and Ranjay Krishna. 2020. [Determining question-answer plausibility in crowdsourced datasets using multi-task learning](#). In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 22–27, Online. Association for Computational Linguistics.
- Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldrige, Eugene Ie, and Diego Garcia-Olano. 2019. [Learning dense representations for entity retrieval](#). In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.
- Adriana D Glenn. 2015. Using online health communication to manage chronic sorrow: mothers of children with rare diseases speak. *Journal of Pediatric Nursing*, 30(1):17–24.

- Benjamin M Good, Max Nanis, Chunlei Wu, and Andrew I Su. 2014. Microtask crowdsourcing for disease mention annotation in pubmed abstracts. In *Pacific Symposium on Biocomputing Co-Chairs*, pages 282–293. World Scientific.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.
- Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium. Association for Computational Linguistics.
- Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.
- Zhengping Jiang, Sarah Ita Levitan, Jonathan Zomick, and Julia Hirschberg. 2020. Detection of mental health from Reddit via deep contextualized representations. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 147–156, Online. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Vivek Khetan, Md Imbesat Rizvi, Jessica Huber, Paige Bartusiak, Bogdan Sacaleanu, and Andrew Fano. 2022. MIMICause: Representation and automatic extraction of causal relation types from clinical notes. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 764–773, Dublin, Ireland. Association for Computational Linguistics.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.
- Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3705–3717, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.
- Iain J Marshall, Benjamin Nye, Joël Kuiper, Anna Noel-Storr, Rachel Marshall, Rory Maclean, Frank Soboczenski, Ani Nenkova, James Thomas, and Byron C Wallace. 2020. Trialstreamer: A living, automatically updated database of clinical trial reports. *Journal of the American Medical Informatics Association*, 27(12):1903–1912.
- Marcia L. Meldrum. 2000. A brief history of the randomized controlled trial: From oranges and lemons to the gold standard. *Hematology/Oncology Clinics of North America*, 14(4):745–760.
- Alicia L Nobles, Caitlin N Dreisbach, Jessica Keim-Malpass, and Laura E Barnes. 2018. "is this an std? please help!": Online information seeking for sexually transmitted diseases on reddit. In *Twelfth International AAAI Conference on Web and Social Media*.
- Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain Marshall, Ani Nenkova, and Byron Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 197–207, Melbourne, Australia. Association for Computational Linguistics.
- Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, online. Association for Computational Linguistics.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- W Scott Richardson, Mark C Wilson, Jim Nishikawa, Robert S Hayward, et al. 1995. The well-built clinical question: a key to evidence-based decisions. *Acp j club*, 123(3):A12–A13.
- Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. [Okapi at trec-3](#). In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Gaithersburg, MD: NIST.
- Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. Covid-fact: Fact extraction and verification of real-world claims on covid-19 pandemic. *arXiv preprint arXiv:2106.03794*.
- Mohammed Saeed, Giulio Alfarano, Khai Nguyen, Duc Pham, Raphael Troncy, and Paolo Papotti. 2021. [Neural re-rankers for evidence retrieval in the FEVEROUS task](#). In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 108–112, Dominican Republic. Association for Computational Linguistics.
- Chris Samarinas, Wynne Hsu, and Mong Li Lee. 2021. [Improving evidence retrieval for automated explainable fact-checking](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 84–91, Online. Association for Computational Linguistics.
- Mourad Sarrouiti, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. 2021. [Evidence-based fact-checking of health-related claims](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3499–3512, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Maria Sukhareva, Judith Eckle-Kohler, Ivan Habernal, and Iryna Gurevych. 2016. [Crowdsourcing a large dataset of domain-specific context-sensitive semantic verb relations](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2131–2137, Portorož, Slovenia. European Language Resources Association (ELRA).
- Briony Swire-Thompson and David Lazer. 2019. Public health and online misinformation: challenges and recommendations. *Annual review of public health*, 41:433–451.
- Katherine Thai, Yapei Chang, Kalpesh Krishna, and Mohit Iyyer. 2022a. [Relic: Retrieving evidence for literary claims](#). *arXiv preprint arXiv:2203.10053*.
- Katherine Thai, Yapei Chang, Kalpesh Krishna, and Mohit Iyyer. 2022b. [RELiC: Retrieving evidence for literary claims](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7500–7518, Dublin, Ireland. Association for Computational Linguistics.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. [FEVER: a large-scale dataset for fact extraction and VERification](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. [Fact or fiction: Verifying scientific claims](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Appendix for “RedHOT: A Corpus of Annotated Medical Questions, Experiences, and Claims on Social Media”

A Data Collection

A.1 Sampling from Reddit

We retrieved the *newest* 1,000 posts from the respective subreddits using the Reddit PRAW¹⁴ API. While we could have relied on alternative sampling strategies — e.g., ranking posts according to “hot” or “best” under Reddit’s metrics — retrieving the newest posts yields an unfiltered snapshot of the full variety of posts made to social media. We also considered performing completely uniform sampling over all posts ever made to a given forum, but the Reddit API limits callers to retrieving 1000 posts for any search criteria; this practically precludes uniform sampling across all time periods.

Preprocessing We identified and removed all non-English text post extraction.¹⁵ Reddit allows its users to post media content (images/videos) in addition to text, and such imagery can be explicit or disturbing. Therefore, we only retained posts that did not contain any media content.

A.2 Annotations on Amazon Mechanical Turk

Amazon Mechanical Turk (AMT) is a popular platform for recruiting non-expert workers to perform “micro-tasks” (here, annotation). We initially recruited workers by collecting annotations on relatively simple examples for which we already had ground truth labels. We provided AMT workers with a comprehensive set of instructions including (templated) examples of the respective categories. For instance:

- **Questions:** Does this work?; Are X, Y, Z symptoms normal for Condition A?; Will increasing my dosage of X help Y in any way?
- **Personal Experiences:** I was diagnosed with X and have since experienced symptoms Y,Z.; I took X and it seemed to help.; My mother took Y and it helped improved her Z
- **Claim:** My doctor told me that X should help with Y; Since increasing dosage of Z, my X levels have normalized (also an example of personal-experience); I heard from multiple

¹⁴<https://github.com/praw-dev/praw>

¹⁵Langid is a python tool that allows filtering data by language: <https://github.com/saffsd/langid.py>.

people that A helps with C; I read online that X & Y are directly causing Z; I heard from my cousin that X helps control Z

For additional context we provided workers with the “Topic”, i.e., the subreddit from which the post being annotated was sampled. For example, if the topic was “Diabetes”, the piece of text will (presumably) be about diabetes, its treatments, individual experiences with the condition, and so on. We highlight the stage-1 annotation interface in Figure 3. The complete set of instructions we provided to AMT workers are available at https://anonymous.4open.science/r/med_val-64C2/stg1_instructions.pdf.

We retained all qualified AMT workers from stage-1 to carry out additional annotations for us in stage-2, with a higher pay rate. The objective here was to recruit people who had established a working understanding of the data, and would presumably be proficient as a result. Similar to stage-1, we provided workers with a comprehensive set of instructions containing (templated) examples to give a sense of what might be qualify as PIO elements:

- **Populations** coronavirus, asthma, narcoleptic, diabetic, children, young, women etc.
- **Interventions** diet, aspirin, allopurinol, insulin, exercise, botox etc
- **Outcomes** depression, sweating, anxiety, pain, flares, covid etc

Interface used for stage-2 annotations is provided in Figure 4. Complete set of stage-2 instructions provided to AMT workers are available at https://anonymous.4open.science/r/med_val-64C2/stg2_instructions.pdf.

B Dataset Summary

Table 7 provides descriptive statistics for all patient populations (that is, subreddits) included in our dataset. Dysthymia has the highest number of posts included in our corpus while Ankylosing Spondylitis has the lowest (due to data filtering described above). There is substantial variation in the length of the posts written under different subreddits (e.g., in r/ADHD the average post is ~222 tokens, while in r/Lupus it’s only ~93 tokens long). Similarly, there are variations in the number of questions, claims, and experiences across populations. We used subscriber count as a proxy for

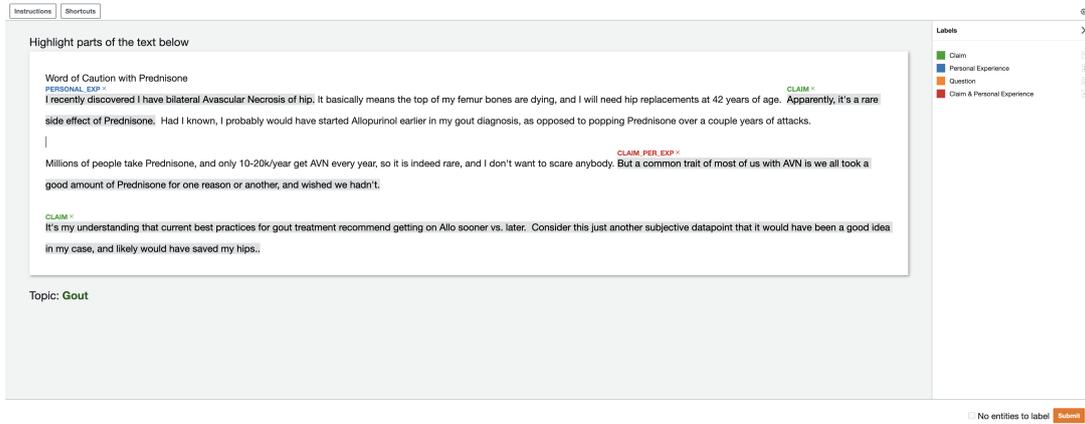


Figure 3: Stage-1 annotations interface for demarcation of spans associated with *questions*, *experiences*, and *claims*.

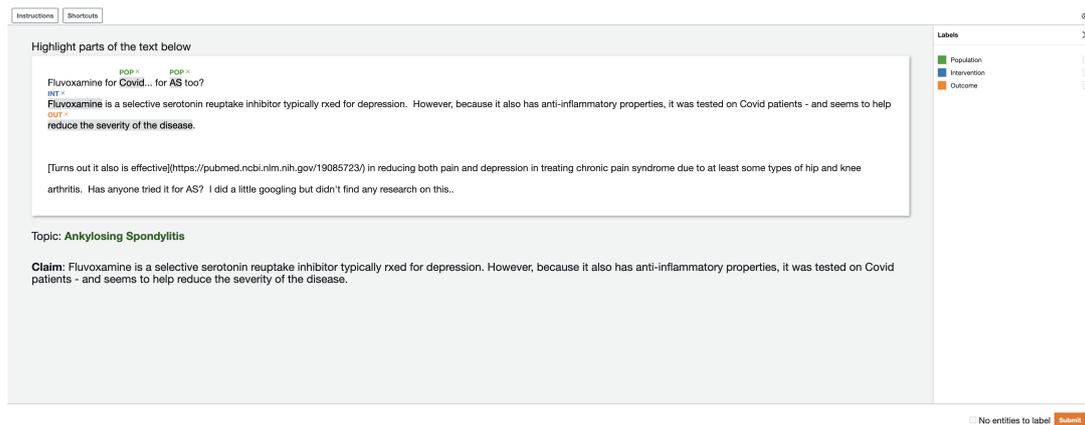


Figure 4: Stage-2 annotations interface for demarcation of PICO frames associated with a given Reddit post and a claim.

Population Type (subreddit)	# of posts in RedHOT	Avg. length of post (# tokens)	# claims	# questions	# experiences	# subscribers on Reddit
Dysthymia	999	175.42	102	989	1387	6.8k
Chronic Fatigue Syndrome	998	139.50	162	1034	1292	31.1k
IBS	998	118.70	71	987	1337	77.1k
Narcolepsy	997	148.65	121	1311	1547	18.9k
Bulimia	996	122.99	46	761	1316	32.8k
Hypothyroidism	995	125.91	111	1585	2088	35.2k
Costochondritis	995	116.97	98	1136	1488	8.8k
Hyperhidrosis	994	97.21	184	1076	1245	25k
Sinusitis	991	135.45	136	1242	1979	5.9k
Psychosis	984	122.91	53	932	933	39.8k
Thyroid Cancer	976	121.80	143	1157	1405	3.2k
Cystic Fibrosis	970	96.11	77	1001	882	7.1k
POTS	963	111.03	77	1155	1274	21.8k
Multiple Sclerosis	958	152.47	129	1081	1309	31.6k
Gout	933	128.87	154	1251	1730	14.2k
ADHD	899	222.41	141	875	1222	1.4M
Gastroparesis	861	134.91	52	909	1319	8k
Diabetes (Type I & II)	748	113.85	40	667	620	90.4k
Crohn's Disease	791	99.79	92	1026	995	43.7k
Lupus	784	93.13	96	978	972	18.2k
Rheumatoid Arthritis	759	103.08	105	1033	1010	6.4k
Epilepsy	670	165.77	37	634	1170	27.8k
GERD	650	164.12	45	669	1518	44.2k
Ankylosing Spondylitis	644	170.83	32	649	1139	12.6k

Table 7: Population-wise descriptive statistics.

Population type	# Posts	Average # per population			Average # per claim		
		Questions	Experiences	Claims	Populations	Interventions	Outcomes
Very Common (Dysthymia, Hypothyroidism, Gout, etc)	5467	1101.82	1654.00	114.83	0.82	2.66	3.57
Common (Chronic Fatigue Syndrome, Bulimia, Psychosis, etc)	9539	847.01	1141.72	74.27	1.05	2.95	3.22
Rare (Narcolepsy, Hyperhidrosis, Thyroid Cancer, etc)	7295	1028.50	1166.25	104.75	0.97	2.79	3.81

Table 8: For descriptive purposes we categorize conditions into: Very Common (>3 million US cases per year), Common (>200k US cases per year), and Rare (<200k US cases per year). We only include posts that do *not* contain any media (photos/videos). Number of experiences here *include* claims based on personal experiences. Diabetes is included as both Common (Type II) and a Rare (Type I) type.

	P	R	F1
Questions	0.73	0.68	0.70
Claims	0.47	0.40	0.43
Experiences	0.33	0.29	0.31
POP	0.85	0.81	0.83
INT	0.56	0.50	0.53
OUT	0.48	0.37	0.42

Table 9: Individual annotator labels evaluated against their own “aggregated” labels.

gauging how active a community on Reddit is. For instance, *r/ADHD* has 1.4M subscribers and so can be considered substantially more active than, say, *r/Psychosis*, which has 39.8K subscribers.

C Additional Results and Experimental Details

We provide results from additional BERT (Devlin et al., 2019) variants for the first task of identifying claims, questions, and experiences in Table 10. Unsurprisingly, pre-trained neural models consistently outperform linear-chain Bag-of-Words CRFs. Similarly, Table 11 provides results from BERT variants and t5-small (Raffel et al., 2020) for the second task of extracting PICO elements conditioned on the post and a given claim. For the t5 model, the target was to produce `<entity token>` followed by `<entity label>` in the same order as they appear in the input sentence (sequential linearization scheme). We evaluated the generated entities against the true sets of PICO elements for each output. While it may be possible to come up with a more optimal linearization scheme for sequence labelling, we posit that to be beyond the scope of our work.

To use dense retrieval models to rank evidence (abstracts) with respect to their relevance to a given claim we need an efficient means to index vectors for $\sim 800k$ abstracts of RCTs in the Trialstreamer database. We did so using FAISS (Johnson et al., 2021) on an Intel Xeon E5-2650 V3 CPU @2.3GHz with 512GB memory. Building an index of dense embeddings for hundreds of thousands passages is highly resource intensive and required roughly 9 hours on two NVIDIA GeForce GTX 1080Ti GPUs.

To train the dense retriever, we used standard split of train, development, and test sets (80%-10%-10%). We trained the two encoders for 40 epochs with a learning rate of 10^{-5} using the Adam optimizer, linear scheduling with warm up, and a dropout rate of 0.1. We parallelized training over

multiple-GPUs; it took roughly 40 hours to train the retriever. Our best-performing retrieval model was initialized with RoBERTa-base (250M parameters). In addition to the results provided in section 3.3, we provide additional results for the retrieval task (evaluated on pseudo test set) in Table 12.

D Deriving Pseudo Training Data: Examples

Generating pseudo training data — i.e., matching reddit annotated reddit posts to “relevant” abstracts of RCTs — is an important component of our dense retrieval pipeline. In Table 13 we provide several examples of the pseudo data we generated from annotated claims. For each row we have inserted intervention and outcome elements from abstracts indexed in Trialstreamer, which makes them “relevant” by construction (while still featuring natural language as it used on social media). We showcase how stage-2 annotated (post, claim) pairs serve as templates to create pseudo claims by substituting PICO elements from an existing corpus.

In Section 3.4 we emphasize the need to evaluate retrieved evidence relevant to *naturally occurring* medical claims, as opposed to the *pseudo* data we derived for training. To this end, we hired domain experts (medical doctors) to look at the evidence abstracts from our retrieval model and assign a relevance score to each abstract (3: relevant, 2: somewhat relevant, 1: irrelevant). We provide some examples of retrieved evidence in Table 14 annotated by our experts as *relevant* (score: 3). Due to space constraints, we provide a link to the full article instead of the full abstract text.

	Claims			Experiences			Questions		
	F1	P	R	F1	P	R	F1	P	R
CRF (Lafferty et al., 2001)	33.87	35.61	32.29	40.08	40.52	39.64	86.89	85.55	88.27
BERT (Devlin et al., 2019)	52.63	58.82	47.61	56.68	59.46	54.33	92.39	88.76	96.34
BioRedditBERT (Basaldella et al., 2020)	45.16	70.92	33.29	59.51	62.49	58.92	93.61	89.29	98.37
RoBERTa (Liu et al., 2019)	47.05	61.53	38.09	56.81	57.11	56.52	93.06	89.01	98.34

Table 10: Additional results from the test set for the task of identifying spans of Claims, Experiences, and Questions.

	P	R	F1	F1 _{POP}	F1 _{INT}	F1 _{OUT}
BERT (Devlin et al., 2019)	43.88	36.13	39.62	41.77	44.68	33.05
RoBERTa (Liu et al., 2019)	47.45	39.27	42.97	46.09	45.99	36.38
BioRedditBERT (Basaldella et al., 2020)	44.44	36.55	40.12	41.92	44.31	34.61
biomedRoBERTa (Gururangan et al., 2020)	38.80	21.48	27.66	30.54	28.13	24.54
t5-small (Raffel et al., 2020)	41.49	38.55	39.97	39.61	45.02	32.41

Table 11: Additional results from the test set for the token-level PIO labelling task.

k	MRR @k					Precision @k				
	1	5	10	50	100	1	5	10	50	100
random	0.00	0.003	0.02	0.02	0.02	0.00	0.02	0.00	1.10	2.80
BM25	5.34	7.98	9.86	14.36	16.70	5.34	10.40	14.45	26.20	33.14
DPR (Karpukhin et al., 2020)	8.07	10.96	11.89	12.20	13.77	8.07	16.50	23.58	31.98	36.87
<i>(trained on the RedHOT pseudo training set)</i>										
RedHOT-DER (BERT-based)	39.14	47.99	49.3	50.28	50.35	39.14	62.55	72.64	83.73	91.74
RedHOT-DER (RoBERTa-based)	45.93	54.60	55.90	56.73	56.78	45.93	69.90	78.81	94.73	98.06

Table 12: Additional results from the retrieval task (tested on the pseudo test set).

	Original w/ PIO placeholders (Template)	w/ Substituted PIO elements (Pseudo)	Population
Claim	Global spread of [OUT] blamed on [INT]	Global spread of Gradual deterioration of renal function blamed on cyclophosphamide	Lupus
Post	Global spread of [OUT] blamed on [INT]	Global spread of Gradual deterioration of renal function blamed on cyclophosphamide	
Claim	I'll be starting [INT] soon and have heard/been told it can cause some serious side effects when first starting to take it.	I'll be starting solriamfetol treatment soon and have heard/been told it can cause some serious side effects when first starting to take it. Because of this, I let my employer know I may have to be out for a day or two during busiest time of the year, and I'm worried I overshared.	Narcolepsy
Post	I'll be starting [INT] soon and have heard/been told it can cause some serious side effects when first starting to take it. Because of this, I let my employer know I may have to be out for a day or two during busiest time of the year, and I'm worried I overshared.	I'll be starting solriamfetol treatment soon and have heard/been told it can cause some serious side effects when first starting to take it. Because of this, I let my employer know I may have to be out for a day or two during busiest time of the year, and I'm worried I overshared.	
Claim	I read that [OUT] could be due to [POP].	I read that hip and lumbar bone mineral density differences could be due to Ankylosing Spondylitis.	Ankylosing Spondylitis
Post	I'm 40M with [POP] and UC and my annual blood work just came back [OUT] (around 2.5). However, my other blood levels are all fine, I eat well, am relatively thin (BMI 24), exercise a lot. I read that [OUT] could be due to [POP].	I'm 40M with Ankylosing Spondylitis and UC and my annual blood work just came back hip and lumbar bone mineral density differences (around 2.5). However, my other blood levels are all fine, I eat well, am relatively thin (BMI 24), exercise a lot. I read that hip and lumbar bone mineral density differences could be due to Ankylosing Spondylitis.	
Claim	Surprising I'm seeing research articles that [INT] causes [OUT] past the duration of the drug	<ul style="list-style-type: none"> * Surprising I'm seeing research articles that quetiapine versus aripiprazole causes psychopathology, cognition, health-related quality of life, and adverse events past the duration of the drug. ◇ Surprising I'm seeing research articles that IPS causes levels of stress past the duration of the drug ● Surprising I'm seeing research articles that olanzapine causes discontinuation rate past the duration of the drug * Surprising I'm seeing research articles that quetiapine versus aripiprazole causes psychopathology, cognition, health-related quality of life, and adverse events past the duration of the drug. I only took a brief look into it. Has anyone here had quetiapine versus aripiprazole induced psychopathology, cognition, health-related quality of life, and adverse events? What is r/psychosis experience with quetiapine versus aripiprazole? ◇ Surprising I'm seeing research articles that IPS causes levels of stress past the duration of the drug. I only took a brief look into it. Has anyone here had IPS induced levels of stress? What is r/psychosis experience with IPS? ● Surprising I'm seeing research articles that olanzapine causes discontinuation rate past the duration of the drug. I only took a brief look into it. Has anyone here had olanzapine induced discontinuation rate? What is r/psychosis experience with olanzapine? 	Psychosis
Post	Surprising I'm seeing research articles that [INT] causes [OUT] past the duration of the drug. I only took a brief look into it. Has anyone here had [INT] induced [OUT]? What is r/psychosis experience with [INT]?	<ul style="list-style-type: none"> * Surprising I'm seeing research articles that quetiapine versus aripiprazole causes psychopathology, cognition, health-related quality of life, and adverse events past the duration of the drug. I only took a brief look into it. Has anyone here had quetiapine versus aripiprazole induced psychopathology, cognition, health-related quality of life, and adverse events? What is r/psychosis experience with quetiapine versus aripiprazole? ◇ Surprising I'm seeing research articles that IPS causes levels of stress past the duration of the drug. I only took a brief look into it. Has anyone here had IPS induced levels of stress? What is r/psychosis experience with IPS? ● Surprising I'm seeing research articles that olanzapine causes discontinuation rate past the duration of the drug. I only took a brief look into it. Has anyone here had olanzapine induced discontinuation rate? What is r/psychosis experience with olanzapine? 	

Table 13: Examples of template claims used for the creation of pseudo training labels for training a supervised evidence retrieval model.

Title of trial paper	Link to abstract/trial
<p>Claim: Vitamin D may prevent autoimmune disease</p> <p>Post: Okay so... the only bloodwork for me that was pretty abnormal was vitamin D. My neurologist did bloodwork for it a year ago and it was in the 20s. He said it should be 50+ and that <i>Vitamin D may prevent autoimmune disease</i>. Are there any long term problems I should be aware about if I can, how get it to go up?</p>	<p>Vitamin D and marine omega 3 fatty acid supplementation and incident autoimmune disease: VITAL randomized controlled trial.</p> <p>https://dx.doi.org/10.1136/bmj-2021-066452</p>
<p>Claim: been researching few weeks now and I recently came across POTS Syndrome. I found that it affects your heart rate so I decided to test mine while resting and then standing to see if maybe thats what it could be.</p> <p>Post: I just joined this group, so I apologize if this is not allowed. I have been researching what I feel to be abnormal symptoms I've been dealing with the majority of my life (dizziness, nausea when standing, etc)... Anyways, I've been researching few weeks now and I recently came across POTS Syndrome. I found that it affects your heart rate so I decided to test mine while resting and then standing to see if maybe thats what it could be. I took my heart rate three times while laying in bed. at 1:37am, by heart rate was 73bpm. at 1:39am, my heart rate was 74bpm. at 1:40am, my heart rate was 73 bpm again. I then stood up (right next to my bed) and proceeded to take my heart rate again. Immediately it shot up to more than double my resting heart rate at 1:41am my heart rate was 156bpm i took it again a minute later and at 1:42am my heart rate was 153bpm. Even if its not pots, just from standing up, I feel like this is not a normal bodily response for the majority of the population. Dont know how to go about getting this checked out. By the way, not sure if it matters, but I am a 19 year-old girl.</p>	<p>Cardiovascular exercise as a treatment of postural orthostatic tachycardia syndrome: A pragmatic treatment trial.</p> <p>https://dx.doi.org/10.1016/j.hrthm.2021.01.017</p>
<p>Claim: did some research and apparently smoking can effect bowel movements (bloating,cramping) which is what i struggle with exactly</p> <p>Post: i have been a smoker for only 3 years, and i recently had the realization that my IBS(like) symptoms correlated to the same period of time i started smoking. i then did some research and apparently smoking can effect bowel movements (bloating,cramping) which is what i struggle with exactly, so i dont know if anyone has a similar story or if quitting smoking helped with their IBS ?</p>	<p>The effect of alpha-tocopherol and beta-carotene supplementation on colorectal adenomas in middle-aged male smokers.</p> <p>https://www.ncbi.nlm.nih.gov/pubmed/10385137</p>
<p>Claim: I read of the issues it can cause the body but so much out there has it.</p> <p>Post: sugar alcohol vs sugar Just wondering what your thoughts are of sugar alcohol. I noticed a lot of sugar free foods have sugar alcohol inplace of sugar. I read of the issues it can cause the body but so much out there has it. Do you avoid sugar alcohol products or do you embrace it as a sugar alternative?</p>	<p>Glycemic Effects of Rebaudioside A and Erythritol in People with Glucose Intolerance.</p> <p>https://dx.doi.org/10.4093/dmj.2016.40.4.283</p>
<p>Claim: I cant help thinking it may be related to my meds</p> <p>Post: I stopped taking Levothyroxin for about a month. Ever since I started taking it again I feel like crying after taking it in the mornings. It could be that I really dont want to go to work, but I cant help thinking it may be related to my meds. Does this happen to anyone else?</p>	<p>Clinical Observation of Levothyroxine Sodium Combined with Selenium in the Treatment of Patients with Chronic Lymphocytic Thyroiditis and Hypothyroidism and the Effects on Thyroid Function, Mood, and Inflammatory Factors.</p> <p>https://dx.doi.org/10.1155/2021/5471281</p>

Table 14: Examples of evidence abstracts (marked relevant by domain experts) retrieved by the RoBERTa-based RedHOT-DER model trained on pseudo data.

Paparazzi: A Deep Dive into the Capabilities of Language and Vision Models for Grounding Viewpoint Descriptions

Henrik Voigt¹, Jan Hombeck¹, Monique Meuschke³, Kai Lawonn¹ and Sina Zarriß²

¹University of Jena ²University of Bielefeld ³University of Magdeburg

¹first.last@uni-jena.de

²first.last@uni-bielefeld.de

³last@isg.cs.uni-magdeburg.de

Abstract

Existing language and vision models achieve impressive performance in image-text understanding. Yet, it is an open question to what extent they can be used for language understanding in 3D environments and whether they implicitly acquire 3D object knowledge, e.g. about different views of an object. In this paper, we investigate whether a state-of-the-art language and vision model, CLIP, is able to ground perspective descriptions of a 3D object and identify canonical views of common objects based on text queries. We present an evaluation framework that uses a circling camera around a 3D object to generate images from different viewpoints and evaluate them in terms of their similarity to natural language descriptions. We find that a pre-trained CLIP model performs poorly on most canonical views and that fine-tuning using hard negative sampling and random contrasting yields good results even under conditions with little available training data.

1 Introduction

Recent advancements in pre-training large-scale language and vision (L&V) models, such as CLIP (Radford et al., 2021), have led to exceptional performance on benchmarks and leaderboards in 2D image-text retrieval (Shen et al., 2021; Fang et al., 2021; Baldrati et al., 2022). However, the image-text data in these benchmarks have specific properties and biases (Thomason et al., 2022) that may limit the language grounding capabilities of existing L&V models and their robustness in real-world scenarios (Khandelwal et al., 2022; Gadre et al., 2022). A fundamental bias in existing L&V data comes from the fact that images generally show single, human-centric views of *different objects*. This raises a simple but intriguing question: to what extent can a model acquire knowledge about the concept of viewpoints and identify *different views on the same object*? Figure 1 illustrates this challenge, showing the top-3 images

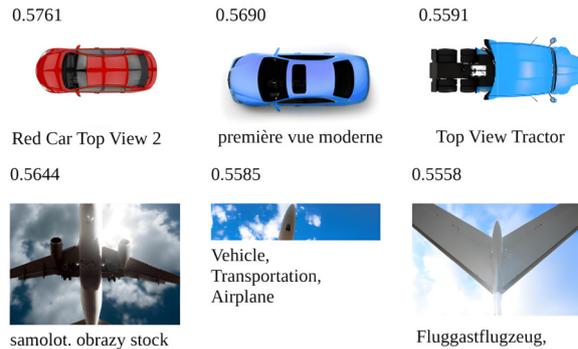


Figure 1: Top-3 retrieval results for *car/airplane from the bottom* using CLIP on the LAION-5B dataset.²

retrieved by CLIP for two basic viewpoint descriptions, *car/airplane from the bottom*, in the LAION-5B (Schuhmann et al., 2021) data set: the *airplane* images mostly correspond to the correct view, but none of the *car* images shows a bottom view. It suggests that the model does not generalize the meaning of viewpoint descriptions across different objects,¹ and may fail to acquire visual-linguistic knowledge that would be needed in more realistic 3D scenarios, such as when instructing a drone to take a picture of an object from a specific viewpoint (Thomason et al., 2020; Fan et al., 2022). This opens the door for a systematic examination of the capabilities of L&V models for grounding viewpoint descriptions, delving into the question of why, despite their excellent zero-shot capabilities, a model like CLIP struggles when it comes to representing perspectives of the same object.

In this paper, we investigate whether language understanding in pre-trained L&V models generalizes to simple text-viewpoint descriptions of common objects. We propose a new task – text-viewpoint retrieval – and a framework for analyzing and scaling image-text models with 3D data.

¹When searching the LAION-5B dataset via image embeddings of cars from the bottom, dozens of relevant results can be provided, which shows that these views exist in the data.

²<https://rom1504.github.io/clip-retrieval/>

We implement a **Paparazzi** agent that circles a spherical camera around a 3D object, samples images, and scores pairs of image-viewpoint descriptions using a pre-trained image-text matching model. In this framework, we evaluate and analyze whether CLIP, as a representative image-text-matching model with excellent zero-shot capabilities, systematically retrieves images of views of 3D shapes, regardless of potential reporting biases in 2D L&V data sets.

To successfully interpret viewpoint descriptions like *car from the bottom*, models need to connect concepts in natural language to visual representations and basic knowledge of object geometry. To investigate this, our approach is deliberately simple: we use 3D shapes from five categories of common objects in ShapeNet that have visually distinct canonical views (*front, back, left, right, top, bottom*). Based on Goldberg polyhedrons (Goldberg, 1937), that divide a sphere into hexagonal shapes, we analyze whether CLIP provides an adequate embedding for the viewpoint space around an object. Our analysis suggests that basic viewpoint understanding is indeed a systematic gap in the pre-trained CLIP model, as it achieves very poor performance in scoring view-description pairs and even retrieves nonsensical, non-human-centric views. Furthermore, we find that this problem is not fixed by standard fine-tuning. Thus, we propose a procedure for fine-tuning CLIP that extends the contrastive learning approach to viewpoints and descriptions generated from 3D visualizations. We find that a small amount of training data and extended fine-tuning is successful in scaling CLIP to basic viewpoint understanding in 3D.

2 Related Work

Vision, View, and Language. To date, research on grounding language in vision focuses on connecting language to visual representations of 2D human-centric views of scenes and objects based on, e.g., large image-caption data sets (Thomee et al., 2016; Schuhmann et al., 2021). Retrieval models in L&V usually rank a fixed set of images showing single views of different objects and scenes given a textual query or vice versa (Li et al., 2020a,b; Baldrati et al., 2022). Common understanding models process pairs of texts or questions and single-view images and predict labels for them, typical generation models process single-view images and generate descriptions for them (Mokady

et al., 2021; Yu et al., 2022). In this paper, we propose a new L&V retrieval task where the model needs to search for a specific view, represented as an image, of a 3D object given a textual query. In our task, the space of possible view-images is not restricted to a human-centric view.

Language Grounding in 3D. Achlioptas et al. (2019) present pioneering work in this area, with a referring expression data set designed for learning the language of shape for *chair* objects in ShapeNet, the most well-known resource for 3D object models (Chang et al., 2015). They build a neural resolution model that predicts which chair is referred to by a given shape description. Their encoder combines an autoencoder for point clouds of 3D shapes and a pre-trained image encoder for a single view of the object. As Achlioptas et al. (2019) collected descriptions of the 3D objects in a static environment with a fixed camera perspective, their approach does not account for dynamic viewpoints in 3D. Thomason et al. (2022) present a larger data set for expressions referring to ShapeNet objects and build a model that relies on image-text matching via the CLIP architecture, similar to ours. Their model takes images of eight fixed viewpoints of the object as input and integrates a component that estimates the viewing angle of an image. They evaluate on resolution accuracy and do not explicitly test viewpoint understanding in the CLIP model. In contrast to these existing works, the input to our model does not specify a fixed set of camera positions, and the output is an explicit, specific viewpoint of an object represented as an image.

Camera Position Estimation. Viewpoint selection in a 3D environment is a well-known problem in other areas (Kamada and Kawai, 1988; Roberts and Marshall, 1998; Arbel and Ferrie, 1999; Vázquez et al., 2001; Plemenos and Sokolov, 2006; Podolak et al., 2006; Mühler et al., 2007). Work in photogrammetry investigates camera position estimation minimizing the error in 3D measurements and reconstruction (Olague and Mohr, 2002). Systems in visualization aim to find an optimized viewpoint with the least possible occlusion and maximum information content for polygonal data (Vázquez et al., 2001; Neugebauer et al., 2013; Meuschke et al., 2017), volumetric data (Bordoloi and Shen, 2005) and vector fields (Lee et al., 2011; Tao et al., 2012). A key challenge in these areas is the definition of what actually constitutes a

good viewpoint (Bonaventura Brugués et al., 2018). Most algorithms aim to find a viewpoint that is of high interest to the user (Leifman et al., 2016; Neugebauer et al., 2013), but do not yet incorporate textual descriptions of viewpoints. In addition, most of these algorithms require expensive annotated mesh representations of 3D objects. L&V models pre-trained on raw image-text data constitute an extremely promising direction here, provided that they are capable of viewpoint understanding.

3 Text-Viewpoint Retrieval Task

We study viewpoint understanding from descriptions and describe a framework for text-viewpoint retrieval. We present a task definition, the set-up of the 3D environment and the camera, and our approach to evaluation and analysis.

3.1 Task Definition

We define the input of our viewpoint retrieval task to consist of a 3D scene with a single object O , a search query describing a viewpoint q , and an orbital camera C circling the object. The camera returns single views of the object v that are represented as RGB images. The retrieval model’s task is to find a viewpoint v that matches the query q . In this work, we implement retrieval via a scoring function S that passes pairs of images v (taken by the camera) and queries q to a pre-trained text-image matching model. The parameterization of the orbiting camera C determines the space of possible viewpoints V that the retrieval model has to search. The parameter setup we used in this work is explained in detail below.

This setting leverages the well-understood image-text matching in 2D for language grounding in 3D. Our retrieval model does not have a symbolic or explicit representation of the object’s geometry but can perceive it by taking images from various perspectives. This framework is independent of different types of 3D data and only requires an engine that renders images of 3D environments.

3.2 Camera Set-up

For the purpose of this study, we restrict the viewpoint space V to views that contain the object of interest. We use a spherical camera system where the center of the object defines its center, as shown in Figure 2. The camera in orbit can be navigated around the desired object using polar coordinates.

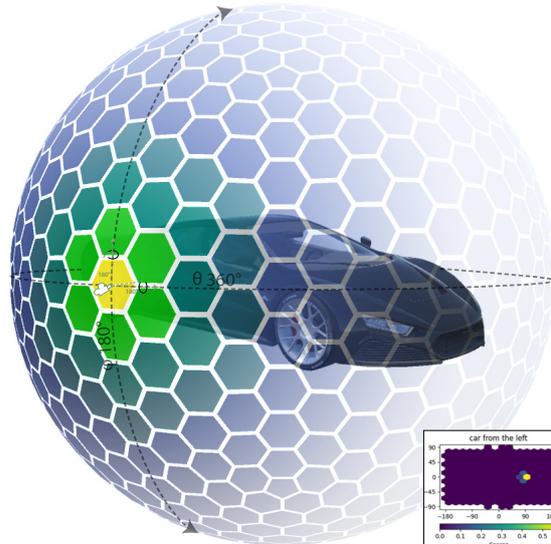


Figure 2: The camera setup: the viewing angles θ and φ describe the azimuthal and polar angle of the camera on the orbital sphere. The parameters x and y describe the camera’s orientation at the given location.

The position of the camera towards the object is defined by (r, θ, φ) for the radial distance, the azimuthal angle, and the polar angle. The center of the object is defined by the center of its bounding box. The camera’s local x and y axes are used to adjust the camera’s viewing angles. Rotation around the local z -axis of the camera is disabled in this work, as the results would be the same, only with a rotated output image. In summary, the exact camera position and rotation along the sphere can be described by five parameters: $(r, \theta, \varphi, x, y)$.

To create equidistant sample points for camera positions along the sphere, we use a Goldberg polyhedron (Goldberg, 1937). It divides a sphere into mostly hexagonal shapes, including a small finite number of pentagons, and creates a nearly equidistant sample space (see Figure 2). The centers of the hexagons give us a discrete number of sample points, which reduce the possible configurations of our camera setup to a finite number. The hexagon centers can be approached for different radii r . The polyhedron used in this work initially yields 1002 sample points per radius. This discretization of the sample space is fine enough to allow benchmarking and analysis of viewpoint retrieval models.

The object O lies at the origin of the Cartesian space $(0, 0, 0)$, which is also the center of the surrounding hypersphere. The radius r is clipped relatively to the size of the object. We estimate the extent of the object based on its bounding box. We

determine the extent of the bounding box based on the minimum r_{min} and maximum radius r_{max} of the surrounding orbital spheres. In our experiments, we set r_{min} to two times the edge length of the bounding box and r_{max} to ten times the edge length of the bounding box.

3.3 Evaluation and Analysis

Common Objects and Canonical Views. To systematically evaluate language-view understanding in CLIP, we limit the set of viewpoint descriptions Q in our experiments to the six **canonical views** *front*, *back*, *right*, *left*, *top*, *bottom* defined by Chang et al. (2015). We choose 3D models of common object categories in ShapeNet (Chang et al., 2015). From the available 55 categories, we selected five categories where all canonical views are visually distinct: *cars*, *airplanes*, *motor-bikes*, *mugs* and *benches*.³ As ShapeNet provides an aligned representation of all 3D models, these restrictions yield a fully controllable experimental setup where training and test data with pairs of queries and views can be generated automatically. The experimental setup is general enough to be transferable to arbitrary object domains and various forms of textual viewpoint descriptions.

Viewpoint Quality Evaluation. To assess the quality of text-viewpoint retrieval, we use the KL divergence (Kullback and Leibler, 1951) of a model’s scoring function against a gold standard scoring distribution as well as the classical retrieval metrics *precision@k* and *retrieval@k*. We use KL divergence in addition since retrieval metrics only reflect performance on gold standard viewpoints and do not allow us to infer the global performance needed to find out why models fail on certain queries, as discussed in Section 5. We define the gold standard score distribution with respect to a particular viewpoint as a discrete normal distribution around the gold standard viewpoint, which is the mean of the distribution. The three polygonal rings around the mean are assigned the normalized score value at one, two, or three times the standard deviation of the normal distribution. The scores for all these viewpoints sum to 1. The scores for all other viewpoints around the sphere are set to zero. The setup is illustrated in Figure 2. To visually

³Many object categories like *bottle*, *ball*, *table*, etc. do not have this property. For instance, the *front* and *back* views of a bottle are not or much less distinct than the *front* and *back* views of a car.

analyze the goodness of a scoring function over a sphere, we unfold the polyhedron and upsample it, as shown in the small map at the bottom right of Figure 2. In this way, we can visualize the difference between the gold standard and the predicted score distribution for an object.

Search Performance Evaluation. When searching a 3D scene, there are many possible viewpoints to consider. A scoring function that works well on a subset of pre-selected viewpoints may yield a good result in retrieval metrics, but in practical usage, it may lead the search algorithm to an unexpected or nonsensical viewpoint. Therefore, to evaluate the performance of a model, we need to consider not only how well it performs on the gold standard viewpoint images, but also how well it can guide a search algorithm to find the right viewpoint in the scene. We compare the performance of different search algorithms under different configurations of the scoring function to **understand the impact of the shape of the scoring function on search performance**. We compute search performance as follows: a search is considered successfully completed if the found viewpoint is within a certain radius of the respective gold standard viewpoint. We define the radius discretely based on the hexagonal rings around a gold standard viewpoint on the Goldberg polyhedron. In our experiments, we consider a search to be solved if a viewpoint is found within the first two rings around the gold standard viewpoint (see Figure 2). We compare performance in terms of the number c of calls to the scoring function required by the search algorithm to solve the search problem described above. We restrict the search length to a maximum number c_{max} of 300 viewpoints to visit. To obtain a robust comparison, we run the procedure n times at randomly selected starting positions on the hypersphere around the object. In our experiments, we set n to ten. Then, the number of calls $\frac{c}{n}$ is averaged.

4 Model

4.1 Scoring Function

The heart of our retrieval model is a function S that outputs matching scores for pairs of images and queries (v, q) . Pre-trained L&V models like CLIP (Radford et al., 2021) embed (v, q) pairs into a common subspace, resulting in latent vector representations \mathbf{z}_v and \mathbf{z}_q , e.g., of size 512 in the original CLIP. The output of the scoring function S

is the cosine similarity of the latent representations of the viewpoint image and the search query:

$$S(v, q) = \cos(z_v, z_q) = \frac{\mathbf{z}_v \cdot \mathbf{z}_q}{\|\mathbf{z}_v\| \|\mathbf{z}_q\|} = \frac{\sum_{i=1}^N z_{v_i} z_{q_i}}{\sqrt{\sum_{i=1}^N z_{v_i}^2} \sqrt{\sum_{i=1}^N z_{q_i}^2}} \quad (1)$$

To evaluate a given viewpoint with respect to a query, both are encoded into their latent representations \mathbf{z}_v and \mathbf{z}_q , and the cosine similarity of their latent representations is used as a **score** for how well the view matches the query.

4.2 Objective Functions

To achieve high similarity between associated texts and images, Radford et al. (2021) apply a contrastive learning paradigm. In a training batch of N image-text pairs, a cosine similarity score is computed for each possible text-image combination. This leads to $N \times N$ scores over which a cross-entropy loss is calculated across the rows and columns. For corresponding text-image pairs, the maximum class score is expected, while for all other pairs, a minimum score is targeted.

We extend this contrastive learning paradigm for fine-tuning CLIP with 3D data by minimizing the combination of three different loss objectives: a) for negative examples, b) for random examples, and c) for hard negative examples.

Cross-Entropy Loss on Negative Examples is calculated and summed for both queries \mathbf{q} and viewpoints \mathbf{v} as $L_{v,q}$. The parameter τ is a learnable parameter for scaling the logits:

$$L_{v,q} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\cos(z_{v_i}, z_{q_i})/\tau)}{\sum_{j=1}^N \exp(\cos(z_{v_i}, z_{q_j})/\tau)} \quad (2)$$

Cross-Entropy Loss on Random Examples is denoted as L_r and computed between annotated viewpoints and randomly generated viewpoints of the 3D scene. L_r is computed exactly as in equation (2), but the contrastive examples are random images from the scene in this case.

Cross-Entropy Loss on Hard Negative Examples referenced as L_h uses images that have a different annotation but appear to be similar in latent space (Li et al., 2021). Robinson et al. (2020) present a sampling method that rescales the loss of negative examples based on their similarity to the gold standard sample. Following this, the loss L_h is calculated as the weighted contrastive loss

between the positive samples x^+ and the hard negative samples x^- drawn from the modified negative sampling distribution q :

$$L_h = Ex^+ \sim p_x^+ x \sim p \left[-\log \frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + G E_{x^- \sim q} [e^{f(x)^T f(x^-)}]} \right] \quad (3)$$

In notation, p^+ is the marginal distribution of positive examples in the overall distribution of samples p . q is the distribution of negative samples. x is a single sample, x^+ and x^- are the respective positive and negative samples. f is a similarity measure, in our case it is cosine similarity. G is a weighting parameter that can be used to adjust the hardness of the negative sampling.

The total loss is parameterized as the weighted sum of the three objectives:

$$L_{total} = \alpha L_{v,q} + \beta L_r + \gamma L_h \quad (4)$$

The ablations resulting from the different combinations presented above are evaluated in Section 6. The parameters α , β , and γ are chosen based on the respective experiment.

4.3 Search Algorithms

At inference time, our retrieval model requires a search algorithm A , a function that optimizes the output of the scoring function S given the space of viewpoints V and a query q . We compare the performance of two search algorithms. **Greedy search** starts with a grid-based approach on the Goldberg polyhedron and tries to find the optimum by moving greedily in the direction of the neighboring region with the highest score in each iteration. **Bayesian search** samples positions on the hypersphere based on incrementally obtained function values, attempting to sample with higher probability in regions that contain optima (Mockus, 1994). See appendix A for implementation details.

5 Experiments

5.1 Experimental Setup

Training. For each of the six canonical view query types and five object categories, we generate 1,000 training images in a Unity scene on randomly selected objects from the ShapeNet training set. This results in 6,000 image and text pairs per object category, which is tiny as compared to the 15 million images in the YFCC100M (Thomee et al., 2016) data set for training the original CLIP.

Model	front	back	left	right	top	bottom	
PRE-TR	4.12	4.09	4.12	4.12	4.09	4.15	car
FT	3.91	3.90	3.91	3.89	3.97	3.92	
RC-HNS	2.85	2.88	3.26	2.99	3.43	3.24	
PRE-TR	4.12	4.10	4.13	4.15	4.08	4.08	airpln
FT	3.92	3.97	4.03	3.95	4.02	4.02	
RC-HNS	3.43	3.73	3.43	3.58	3.52	3.63	
PRE-TR	4.08	4.09	4.12	4.12	4.21	4.20	mbike
FT	3.98	3.89	3.94	3.94	4.04	3.85	
RC-HNS	2.81	2.60	2.84	2.81	3.46	3.47	
PRE-TR	4.15	4.14	4.07	4.05	4.21	4.21	mug
FT	3.96	3.98	3.98	3.94	3.91	3.90	
RC-HNS	3.34	3.10	3.19	2.52	2.52	2.11	
PRE-TR	4.08	4.09	4.17	4.17	4.15	4.13	bench
FT	3.94	3.90	4.00	4.04	3.98	3.93	
RC-HNS	1.88	1.98	2.62	2.18	3.25	3.19	

Table 1: KL-Divergence between gold and predicted viewpoint distribution for the models *PRE-TR*, *FT*, *RC-HNS* on the objects *car*, *airplane*, *motorbike*, *mug*, *bench* for *front*, *back*, *left*, *right*, *top*, *bottom* viewpoints on synthetic images. Lower values are better.

Test Set. For evaluating the retrieval quality for each object category we randomly select three 3D shapes from the ShapeNet test set. Then we compute the normalized score distribution on synthetic images around the sphere with radius five for all selected objects of a category, compute the KL-Divergence and average the results per viewpoint query (see Table 1). To assess the performance on real-world data, we carefully curated a data set of 600 images (5 categories \times 6 viewpoints \times 20 images) by retrieving visually similar images for a seed image using image similarity on LAION-5B. Synthetic gold standard views are obtained from the sampled spheres (see Table 2).

Models. From the official CLIP repository (OpenAI), we select ResNet-101 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) as image encoder and pre-trained BERT model (Devlin et al., 2018) as query encoder. We compare the following models: (i) **PRE-TR**ained CLIP, without further fine-tuning, (ii) CLIP-**FT**, a version of CLIP fine-tuned on the training data with standard cross-entropy loss, (iii) CLIP-**RC-HNS**, fine-tuned with extended loss objectives explained in Section 4.

5.2 Viewpoint Quality Results

Table 1 shows the results for the quality of viewpoint retrieval with different models, objects, and viewpoints. We find that a pre-trained CLIP model shows a high divergence from the gold standard

Model	P@1	P@5	P@10	R@1	R@5	R@10	
PRE-TR	0.044	0.044	0.031	0.007	0.032	0.043	synth
FT	0.622	0.442	0.401	0.090	0.267	0.412	
RC-HNS	0.811	0.607	0.541	0.117	0.355	0.524	
PRE-TR	0.300	0.307	0.290	0.015	0.077	0.145	real
FT	0.867	0.787	0.710	0.043	0.197	0.356	
RC-HNS	0.733	0.673	0.633	0.036	0.168	0.317	

Table 2: Precision@K and Recall@K per model ablation split by synthetic data and real data measured across all object categories.

distribution for all object categories under investigation. The fine-tuned model performs slightly better, but still shows large differences from the gold standard. The use of random contrasting and hard negative sampling brings the score distribution closer to the gold standard distribution. This shows that standard CLIP pre-training and fine-tuning on human-centered 2D images do not produce a suitable scoring function for the viewpoint space around a 3D object.

Evaluating performance on real data using KL divergence is not possible in a similar way as on synthetic data because we do not have access to images from arbitrary viewpoints. Therefore, we compare precision@k and recall@k between synthetic images from ShapeNet and real images at the gold standard viewpoints in Table 2. The results show that pre-trained CLIP performs poorly in grounding viewpoints on both synthetic data and real data. Fine-tuning the model on synthetic data greatly improves the retrieval metrics for both synthetic and real data. RC-HNS performs well on synthetic data that is within the distribution, however, it yields slightly lower scores on real-world data in comparison to FT. This may result from the fact that RC-HNS forces the model to generally score out-of-distribution data lower, thereby making the scoring function more sensitive to differences between synthetic and real-world images. In traditional 2D benchmarks, this may seem like a disadvantage compared to FT, but it proves to be advantageous in 3D viewpoint search, as demonstrated in the following section. Here, the FT model loses performance due to unpredictable scoring behavior in regions far from the gold standard viewpoints.

5.3 Search Performance Results

We test search performance in 3D as described in Section 3.3 for all six queries. Table 3 illustrates the

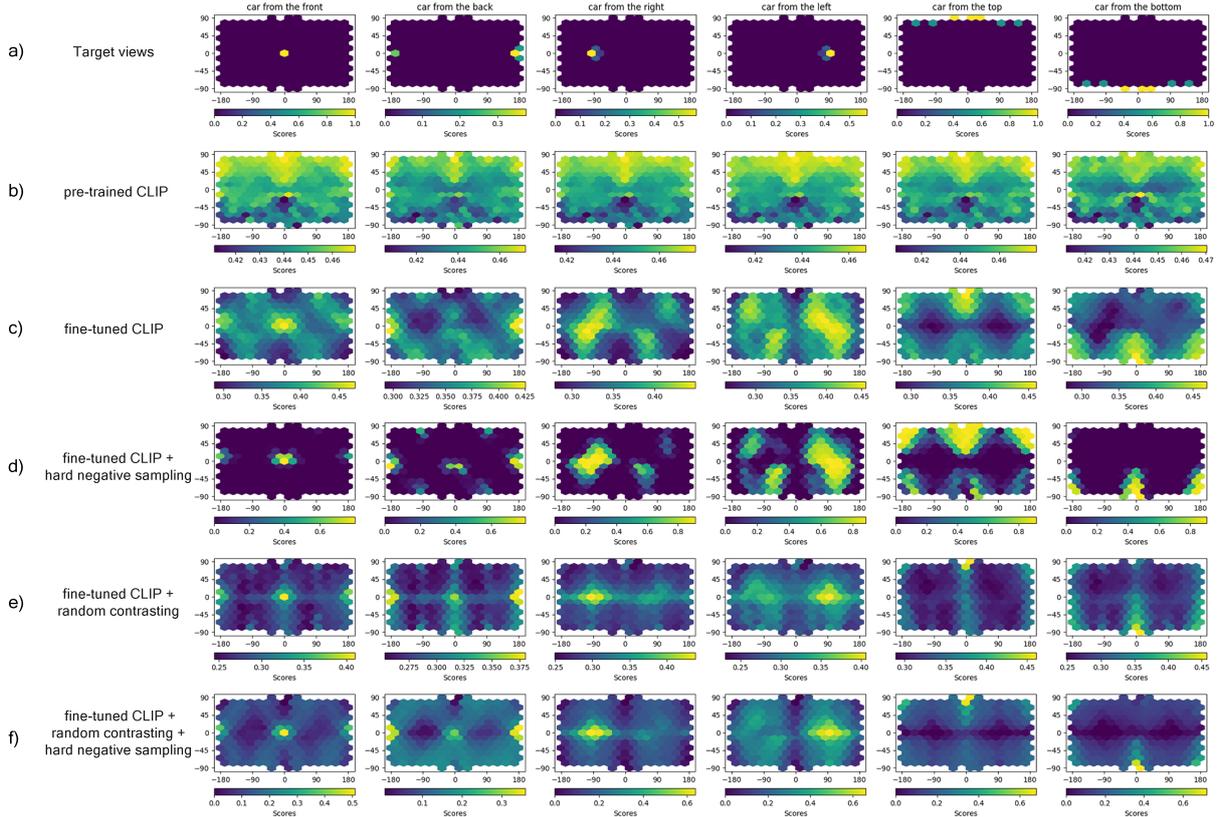


Figure 3: Score distribution on the six viewpoints per loss function combination on a car object. In a) gold-standard viewpoints expected to have high scores are shown, b) pre-trained CLIP, c) fine-tuned CLIP, d) hard negative sampling, e) random contrasting, and f) random contrasting + hard negative sampling. For more, see appendix A.

Model	front	back	left	right	top	bottom	
PRE-TR	171.6	168.3	165.7	159.8	174.1	165.0	Greedy
FT	135.1	137.1	189.1	130.1	142.2	127.4	
RC-HNS	130.5	134.5	182.7	115.9	140.3	144.4	
PRE-TR	259.4	223.2	294.0	264.8	198.4	261.6	Bayes
FT	82.4	79.1	133.0	101.1	29.7	21.5	
RC-HNS	73.5	62.7	62.6	49.4	22.0	22.9	

Table 3: Average number of calls to the scoring function per search algorithm and viewpoint query.

performance for Greedy and Bayes search. Both algorithms perform significantly better than an exhaustive search on the Goldberg polyhedron (= 1002 sample points, fixed radius). Bayesian search is much faster than greedy search, when using a finetuned scoring function (FT, RC-HNS), and it is more affected by the shape of the scoring function since it samples it strategically: it is fastest with the smoothest scoring function RC-HNS and very slow with pretrained CLIP. This is in line with the viewpoint quality results in Section 5.2, showing that pretrained CLIP has a poor representation of

the viewpoint space around an object.

6 Analysis

This section takes a closer look at how well the text-viewpoint embeddings capture understanding of different viewpoints. Specifically, we will explore whether the scoring functions correctly identify viewpoints that align with the linguistic description, while providing lower scores for those that do not.

6.1 Exhaustive Viewpoint Space Analysis

Based on the polyhedron that defines the viewpoint space of the camera, we carry out an exhaustive analysis of the scoring function over this space for specific objects and queries. We select a car from the test set of the ShapeNet data set and plot the scores of the evenly distributed samples from the surface of the Goldberg polyhedron at a radius of five for the six canonical viewpoint queries. We examine five different configurations of the loss objective shown in Equation (4). Figure 3a) illustrates the target region on the hexagon diagram, which contains the optimal viewpoint for a given query.

It can be seen in Figure 3b) that a pre-trained CLIP model even if trained on a large data set, is not able to discriminate between different viewpoints and that the scoring function has multiple optima. Fine-tuning the CLIP model (3c) on synthetic images improves viewpoint discriminability. Nevertheless, apart from the absolute gold standard regions, the function shows problematic local optima and in particular the left and right side views of the car are difficult to distinguish. In (d), we fine-tune the CLIP model by applying the hard negative sampling strategy proposed by Robinson et al. (2020). The results show that the gold standard viewpoints can be distinguished much more effectively when compared to previous experiments. However, the transition between viewpoints is quite sudden, making it challenging for a search algorithm to reach the optimum. In (e), a combination of negative contrastive loss $L_{v,q}$ and random contrastive loss L_r is applied. The results show that the additional objective makes the scoring function much more stable in regions farther away from known canonical viewpoints. In experiment (f), we combine hard negative sampling L_h with the idea of random contrasting. The plot of the scoring function shows that for each canonical viewpoint, the function increases steadily toward the optimal view.

6.2 Nonsensical Viewpoints

A further problem we noticed is that CLIP predicts high scores for nonsensical views that do not relate to the query, but rather seem to activate certain features to drive up the score, similar to adversarial examples (Goodfellow et al., 2014). Such behavior of models on unseen images has also been described by Du et al. (2022) and should be considered when using CLIP representations in continuous 3D environments, especially for vision-and-language navigation tasks, as in Khandelwal et al. (2022). Figure 4 shows retrieved nonsensical viewpoint images among the top-5 for *car from the front*.



Figure 4: Retrieved nonsensical viewpoints in the top-5 scored images on CLIP for the query *a picture of a car from the front*.

6.3 Data Set Size Ablations

To test how the scoring function is affected when only a small amount of training data is available, we gradually reduce the number of training samples from 1,000 to 1 for the best-performing model CLIP-RC-HNS. Access to 1,000 training examples per viewpoint, as shown in 5a), leads to a smooth function. Reducing the training data by 90 percent to 100 examples per viewpoint keeps good performance for the target viewpoints. Compared to the full data set, smoothness suffers slightly. Reducing the training data by 99 percent to ten samples per viewpoint still allows good results in the target regions. However, the surrounding regions become less smooth and drop more abruptly. Surprisingly, when breaking down the training data to one example per viewpoint, the target viewpoint areas still lead to global optima in all search queries. However, the transitions are no longer smooth but rather abrupt, especially for the front and back.

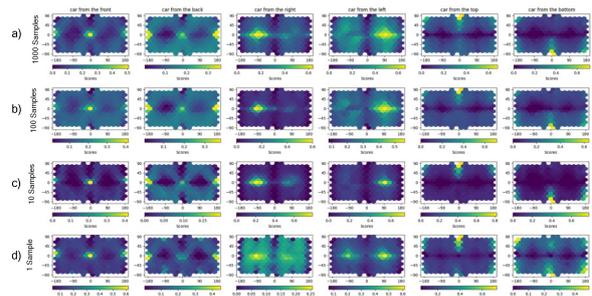


Figure 5: Overview of the effects of gradually reducing the number of training images per view from a) 1000 to b) 100 to c) 10 to d) 1 on CLIP-RC-HNS.

7 Conclusion

We developed a new framework to assess the capabilities of L&V models to ground viewpoint descriptions. Through our research, we discovered that a standard CLIP model struggles to distinguish between different viewpoints. To address this, we explored a combination of different loss objectives on synthetic data to make it easier to retrieve viewpoints from language descriptions. Our experiments revealed that incorporating random contrasting leads to a more accurate and seamless scoring function, as compared to using only text and human-centric images. Our framework thus offers a promising approach to scale L&V models trained on large-scale image-text datasets for applications that involve interaction in the 3D world.

Limitations

We deliberately opted for a simple controllable setup in order to gain a precise understanding of viewpoint representation in CLIP. Our experiments are restricted to canonical views and canned descriptions since they are easy to generate and evaluate automatically. Extending the data to other views and to human-like descriptions is the obvious avenue for future research. In particular, with the advent of NERF models in computer vision, we look forward to integrating these types of models into our framework, as this would allow the generation of near-realistic images in a controlled 3D setup, which would allow for even better evaluation of scoring functions in text viewpoint retrieval. Varying the level of detail of the 3D shapes, especially in complex 3D scenes where large objects consist of smaller parts is another interesting direction. Another restriction of our set-up is the fact that we consider context-free retrieval of viewpoints, whereas in many human-like descriptions such as the *right front tire of a car*, the viewpoint may not be visually unique and depend on the context of the scene, such as the relative position of the viewpoint to other viewpoints. The same applies to views that need to be delivered to a user in a task-oriented interaction, and are likely to be more complex and diverse than the canonical and synthetic ones used in this work. In conclusion, we believe that our framework has the potential to provide a more comprehensive understanding of reporting biases in image-text data used for pre-training LV models. By conducting a 360-degree analysis of the scoring function, our framework allows for a more thorough examination of these biases, as everything is visible and nothing can be hidden from the investigator, unlike when evaluating against a set of gold-standard viewpoints.

Ethics Statement

3D models from the ShapeNet dataset are available for research and non-commercial purposes as well as the LAION-5B data set. We did not collect any personal information from any annotators. We clearly state the intended use of our models, which is to support human-centric interaction with AI models in the 3D world.

Acknowledgments

We thank the Michael Stifel Center Jena for funding this work, which is part of the Carl Zeiss

Foundation-funded project 'A Virtual Workshop for Digitization in the Sciences' (062017-02).

References

- Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. 2019. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8938–8947.
- Tal Arbel and Frank P Ferrie. 1999. Viewpoint selection by navigation through entropy maps. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 248–254. IEEE.
- Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. 2022. Effective conditioned and composed image retrieval combining clip-based features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21466–21474.
- Xavier Bonaventura Brugués, Miquel Feixas Feixas, Mateu Sbert, Lewis Chuang, and Christian Wallraven. 2018. A survey of viewpoint selection methods for polygonal models. *Entropy*, 2018, vol. 20, núm. 5, p. 370.
- Udepta D Bordoloi and H-W Shen. 2005. View selection for volume rendering. In *VIS 05. IEEE Visualization, 2005.*, pages 487–494. IEEE.
- Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. 2022. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*.
- Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Eric Wang. 2022. Aerial vision-and-dialog navigation. *arXiv preprint arXiv:2205.12219*.
- Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. 2021. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*.

- Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. 2022. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*.
- Michael Goldberg. 1937. A class of multi-symmetric polyhedra. *Tohoku Mathematical Journal, First Series*, 43:104–108.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Tim Head, Manoj Kumar, Holger Nahrstaedt, Gilles Louppe, and Iaroslav Shcherbatyi. 2021. [scikit-optimize/scikit-optimize](#).
- Tomihisa Kamada and Satoru Kawai. 1988. A simple method for computing general position in displaying three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 41(1):43–56.
- Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. 2022. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14829–14838.
- Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.
- Teng-Yok Lee, Oleg Mishchenko, Han-Wei Shen, and Roger Crawfis. 2011. View point evaluation and streamline filtering for flow visualization. In *2011 IEEE Pacific Visualization Symposium*, pages 83–90. IEEE.
- George Leifman, Elizabeth Shtrom, and Ayellet Tal. 2016. Surface regions of interest for viewpoint selection. *IEEE transactions on pattern analysis and machine intelligence*, 38(12):2544–2556.
- Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344.
- Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34.
- Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020b. Oscar: Object-semantic aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.
- Monique Meuschke, Wito Engelke, Oliver Beuing, Bernhard Preim, and Kai Lawonn. 2017. Automatic viewpoint selection for exploration of time-dependent cerebral aneurysm data. In *Bildverarbeitung fuer die Medizin 2017*, pages 352–357. Springer.
- Jonas Mockus. 1994. Application of bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365.
- Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.
- Konrad Mühler, Mathias Neugebauer, Christian Tietjen, and Bernhard Preim. 2007. Viewpoint selection for intervention planning. In *EuroVis*, pages 267–274.
- Mathias Neugebauer, Kai Lawonn, Oliver Beuing, Philipp Berg, Gabor Janiga, and Bernhard Preim. 2013. Amnervis—a system for qualitative exploration of near-wall hemodynamics in cerebral aneurysms. In *Computer Graphics Forum*, volume 32, pages 251–260. Wiley Online Library.
- Gustavo Olague and Roger Mohr. 2002. Optimal camera placement for accurate reconstruction. *Pattern recognition*, 35(4):927–944.
- OpenAI. [Openai/clip: Contrastive language-image pre-training](#).
- Dimitri Plemenos and Dmitry Sokolov. 2006. Viewpoint quality and scene understanding. In *Eurographics Symposium on Virtual Reality*, pages 67–73. VAST’2005.
- Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. 2006. A planar-reflective symmetry transform for 3d shapes. In *ACM SIGGRAPH 2006 Papers*, pages 549–559.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.
- DR Roberts and A David Marshall. 1998. Viewpoint selection for complete surface coverage of three dimensional objects. In *BMVC*, pages 1–11. Citeseer.
- Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*.

- Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.
- Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*.
- Jun Tao, Jun Ma, Chaoli Wang, and Ching-Kuang Shene. 2012. A unified approach to streamline selection and viewpoint selection for 3d flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):393–406.
- Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.
- Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. 2022. Language grounding with 3d objects. In *Conference on Robot Learning*, pages 1691–1701. PMLR.
- Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73.
- Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Wolfgang Heidrich. 2001. Viewpoint selection using viewpoint entropy. In *VMV*, volume 1, pages 273–280. Citeseer.
- Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

A Experiment Details

This section provides additional details on our experimental setup. Section A.1 contains further visualizations of the experiments discussed in section 5. Section A.2 provides details about the implementation of the search algorithms used in our benchmark.

A.1 Scoring Function Analysis

The following plots illustrate the score distributions obtained with the different model ablations CLIP-**PRE-TR**, CLIP-**FT**, and CLIP-**RC-HNS**.

Scoring Function PRETR. Figure 6a shows the score distribution of the PRE-TRained CLIP model over 3D objects from the test set of the ShapeNet dataset.

Scoring Function FT. Figure 6b depicts the scoring distribution of the CLIP-FT model over 3D objects from the test set of the ShapeNet dataset.

Scoring Function RC-HNS. Figure 7a illustrates the score distribution of the CLIP-RC-HNS model over 3D objects from the test set of the ShapeNet dataset.

Comparison of Score Distributions for Object Only Queries. To understand which viewpoints CLIP scores best on an object-only query such as *a picture of a car*, we compare these object-only queries for all object categories tested on respective 3D objects from the test set. This tells us which viewpoints CLIP associates most with a given object category. Figure 8a indicates that a PRE-TRained CLIP model is not able to distinguish specific viewpoint queries from pure object queries.

Comparison of Optimal Viewpoints. Figure 8b shows the viewpoint images obtained from the optima of the scoring distributions generated by a CLIP model and a CLIP-RC-HNS model. The images illustrate that descriptions of viewpoints are indeed a bias in CLIP.

Figure 7b illustrates the viewpoints resulting from the global optima of the scoring functions obtained from the CLIP-RC-HNS model.

A.2 Search Algorithm Analysis

In our work, we are particularly interested in the impact of the shape of the scoring function on the performance of various search algorithms. Section A.2.1 provides details on the implementation

of greedy search. Section A.2.3 illustrates how the search algorithms listed above perform their task on a sphere.

A.2.1 Greedy Search Implementation Details

We implement a greedy search algorithm as a representative for gradient-based approaches. The greedy search starts with a grid-based approach on the Goldberg polyhedron and always follows the region with the highest score. It tries to find the optimum by greedily selecting the highest scoring regions at each iteration and searching in their neighboring regions at the next iteration. The search is initialized with k randomly selected starting points (here $k = 6$) from the Goldberg polyhedron. In addition, a cutoff value c must be chosen to determine how many grid points will be considered in the next iteration of the search. The cutoff value can be described as a relative percentage or as an absolute cutoff value. After evaluating all viewpoints with respect to the given query, the next iteration is started by selecting the locations with the highest scores considering the selected cutoff. All obtained scores and their neighboring sample points from the Goldberg polyhedron are added to the list of investigated viewpoints. After that, the next iteration is started. The neighborhood range n , which specifies the number of neighborhood grid points to be examined, can be adjusted. The search can be terminated after i iterations or when no new items have been added to the list of investigated viewpoints. In summary, the greedy search is parameterized by: (k, c, n, i) . We chose greedy search as a test algorithm for our benchmark to see how much gradient-based methods as candidate algorithms for the text-viewpoint retrieval task in a 3D environment depend on a smooth structure of the scoring function in their performance. We use a greedy nearest-neighbour heuristic, since the function is only defined at a fixed number of points due to the discretization of the search space.

A.2.2 Bayesian Search Implementation Details

Bayesian optimization (Mockus, 1994) is used to estimate the optimum of a black-box function that is costly to evaluate. The algorithm updates its Bayesian prior based on the stepwise function values obtained, increasing the certainty that the regions are likely to be optima and therefore more likely to be explored than other regions of the black box function. Then, the number of samples from

Model	P@1	P@5	P@10	R@1	R@5	R@10	
PRE-TR	0.111	0.056	0.050	0.017	0.040	0.066	car
FT	0.778	0.567	0.500	0.113	0.330	0.485	
RC-HNS	0.944	0.778	0.644	0.136	0.432	0.592	
PRE-TR	0.056	0.078	0.050	0.008	0.057	0.073	airpln
FT	0.778	0.500	0.433	0.112	0.297	0.424	
RC-HNS	0.833	0.522	0.439	0.119	0.310	0.441	
PRE-TR	0.000	0.045	0.033	0.000	0.030	0.042	mbike
FT	0.500	0.322	0.339	0.074	0.217	0.400	
RC-HNS	0.667	0.500	0.450	0.098	0.312	0.462	
PRE-TR	0.056	0.033	0.017	0.008	0.024	0.024	mug
FT	0.389	0.311	0.294	0.056	0.179	0.286	
RC-HNS	0.667	0.489	0.483	0.097	0.312	0.532	
PRE-TR	0.000	0.011	0.006	0.000	0.008	0.008	bench
FT	0.667	0.511	0.439	0.097	0.312	0.465	
RC-HNS	0.944	0.744	0.689	0.136	0.411	0.592	

Table 4: Precision and recall metrics on synthetic data for the models *PRE-TR*, *FT*, *RC-HNS* on the objects *car*, *airplane*, *motorbike*, *mug*, *benchs* for *front*, *back*, *left*, *right*, *top*, *bottom* viewpoints.

the regions of interest is increased accordingly. We construct the search problem as a Bayesian optimization as follows: The input of the search algorithm is a vector of size five describing the camera position on the hypersphere around the target object: r, θ, φ, x, y . In this parameterization, θ and φ are spherical coordinates, r is the distance to the center of the 3D object, and x and y are the orientations of the camera along the horizontal and vertical axes. The location of the optimum of the scoring function with respect to a query \mathbf{q} depends on the rotation of the 3D object, which we only know is centered around $(0, 0, 0)$. Therefore, Bayesian search tries to find the optimum of the scoring function with respect to the properties of the 3D object at hand given the search query \mathbf{q} . For our benchmarks, we use the implementation of the Bayesian optimization algorithm in [Head et al. \(2021\)](#).

A.2.3 Search Algorithm Behavior on Sphere

The experiments in Section 5 have shown that a smooth scoring function is advantageous for search algorithms in text-viewpoint retrieval. This section visually analyzes why this is the case by examining how the algorithms perform on a sphere around a target object.

Figure 8c illustrates how the different algorithms approach the regions with higher scores differently. The greedy search with a low cutoff spreads across the sphere in waves, starting from the initial points. Once it touches a high point, it remains attached to it. In this respect, a good initialization is impor-

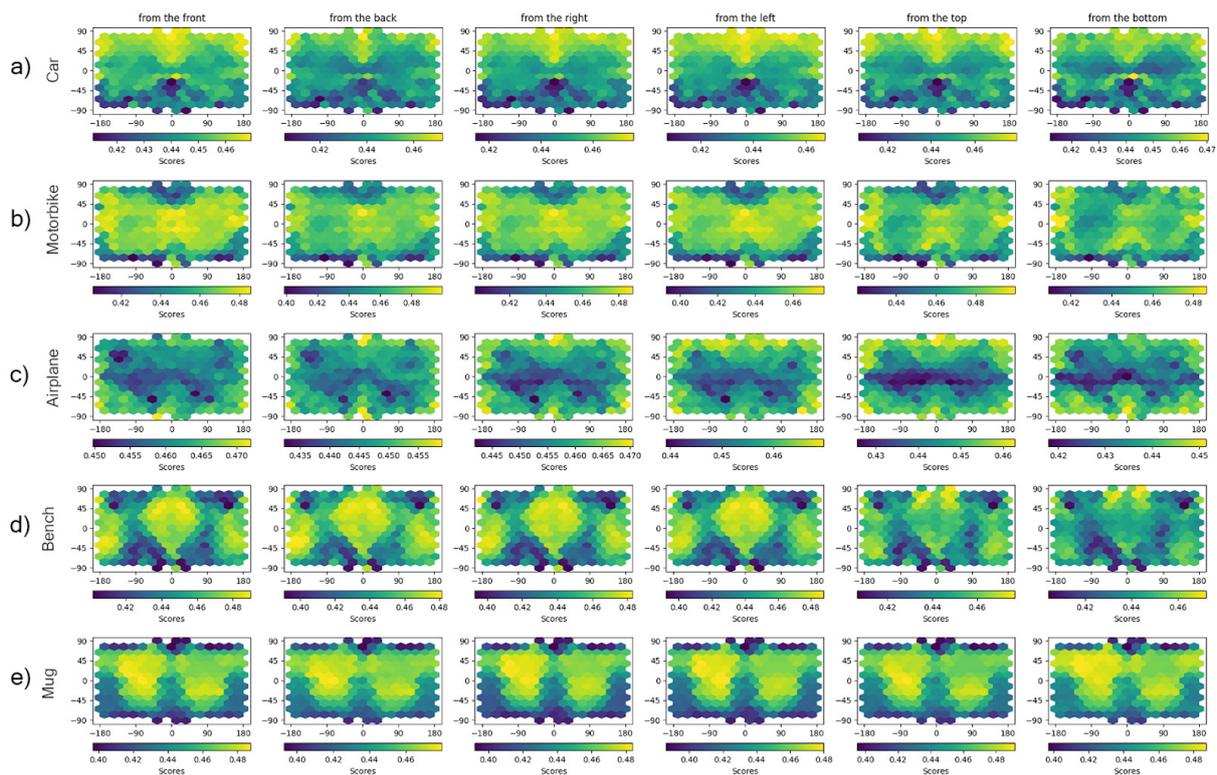
Model	P@1	P@5	P@10	R@1	R@5	R@10	
PRE-TR	0.500	0.500	0.467	0.025	0.125	0.233	car
FT	1.000	1.000	0.967	0.050	0.250	0.483	
RC-HNS	1.000	0.933	0.950	0.050	0.233	0.475	
PRE-TR	0.333	0.367	0.350	0.017	0.092	0.175	airpln
FT	1.000	1.000	0.917	0.050	0.250	0.458	
RC-HNS	1.000	0.833	0.750	0.050	0.208	0.375	
PRE-TR	0.167	0.300	0.300	0.008	0.075	0.150	mbike
FT	0.667	0.633	0.650	0.033	0.159	0.325	
RC-HNS	0.833	0.733	0.783	0.0417	0.183	0.392	
PRE-TR	0.167	0.167	0.167	0.008	0.042	0.08	mug
FT	1.000	1.000	0.967	0.050	0.250	0.483	
RC-HNS	0.833	0.933	0.933	0.042	0.233	0.467	
PRE-TR	0.333	0.200	0.167	0.0167	0.050	0.083	bench
FT	1.000	0.733	0.583	0.050	0.183	0.292	
RC-HNS	0.667	0.500	0.500	0.033	0.125	0.250	

Table 5: Precision and recall metrics on real data for the models *PRE-TR*, *FT*, *RC-HNS* on the objects *car*, *airplane*, *motorbike*, *mug*, *benchs* for *front*, *back*, *left*, *right*, *top*, *bottom* viewpoints.

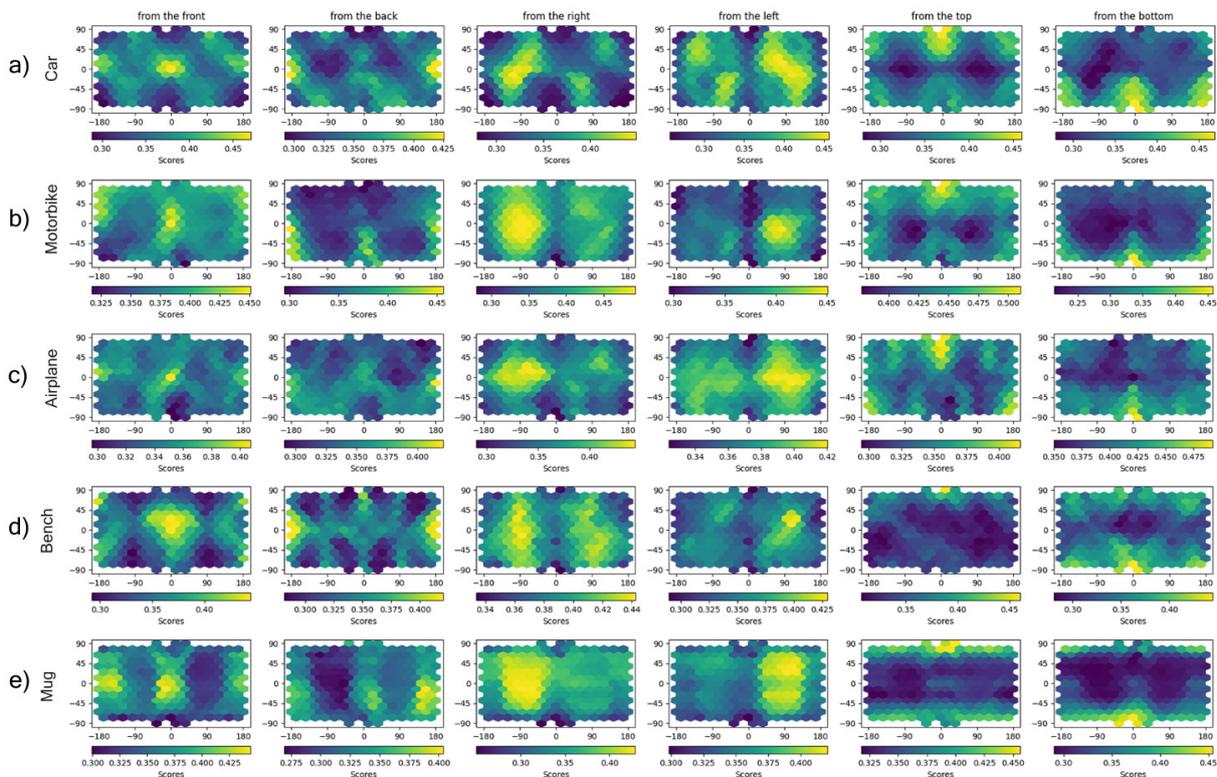
tant, e.g., through a high number of random starting points. Bayesian search also starts from randomly initialized starting points around the hypersphere. Compared to greedy search, it reaches the optimum much faster and more purposefully, since sampling is not bound to any local constraints, such as neighboring regions. Another advantage over greedy search is that random starting points have much lower cost than in greedy search, since they do not cause additional computations in the following iteration. The figure shows that the focus of sampling from random starting points across the sphere leads to small, concentrated regions with high scores. In terms of success rate, Bayesian search is less prone to confounding optima, since a certain number of samples are drawn randomly from different regions anyway. Therefore, the approach is more robust to cases with multiple optima, as is the case with the CLIP-FT model. Despite these obstacles, a solution is reached relatively quickly. However, if the scoring function has a ragged structure like the CLIP-PRETR model, even a sampling-based approach has difficulty identifying the optimal regions due to the raggedness and non-uniformity of the function.

A.3 Retrieval Metrics Analysis

Table 4 shows the precision and recall metrics on **synthetic data** broken down by object category. Table 5 shows the precision and recall metrics on **real data** obtained from the LAION-5B data set.

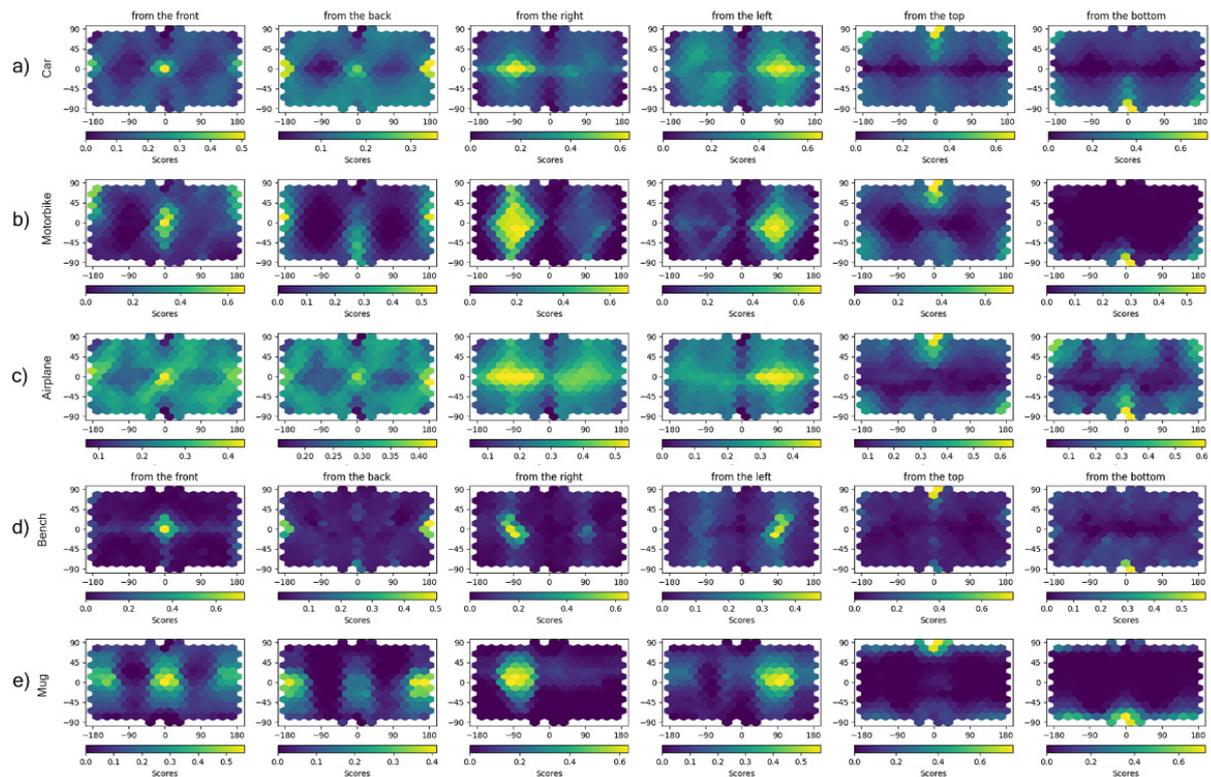


(a) Scoring Function Distribution of CLIP PRE-TR model on cars, motorbikes, airplanes, benches, and mugs for the six canonical viewpoint queries.

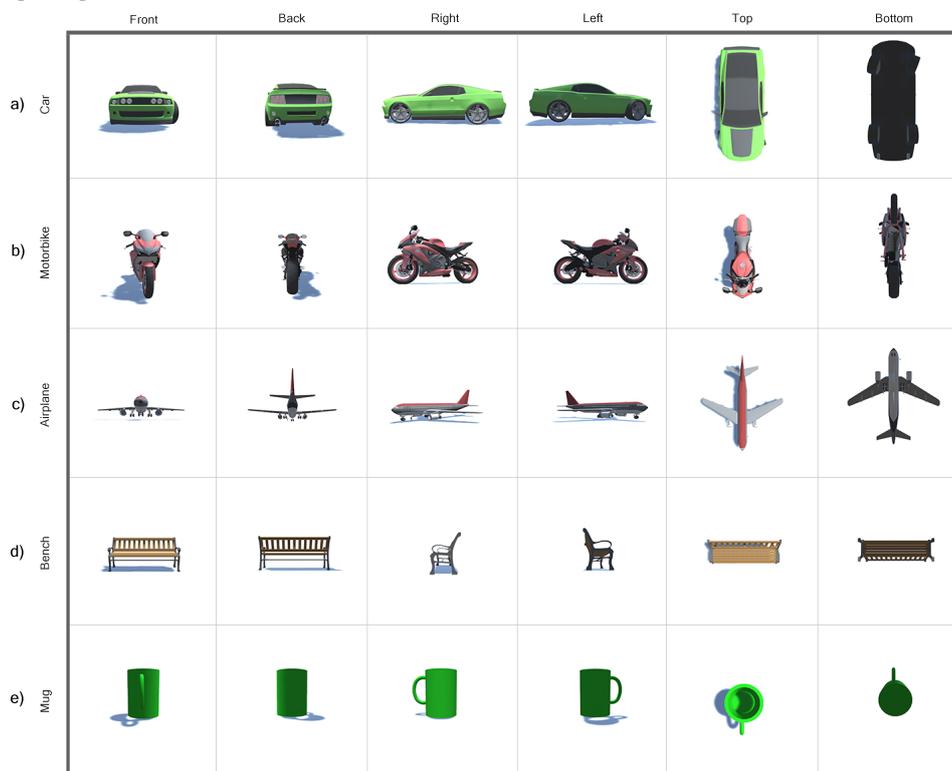


(b) Scoring Function Distribution of the CLIP-FT model on cars, motorbikes, airplanes, benches, and mugs for the six canonical viewpoint queries.

Figure 6: Scoring Function Distributions on CLIP PRE-TR and CLIP-FT.

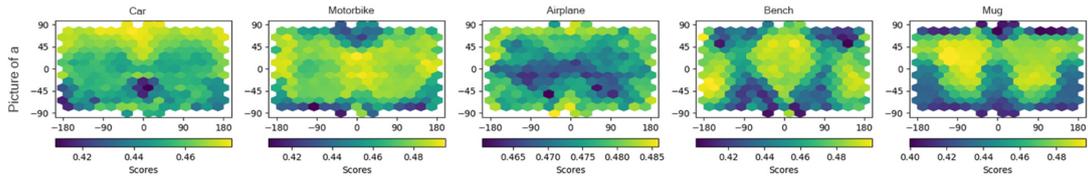


(a) Scoring Function Distribution of the CLIP-RC-HNS model on cars, motorbikes, airplanes, benches, and mugs for the six canonical viewpoint queries.

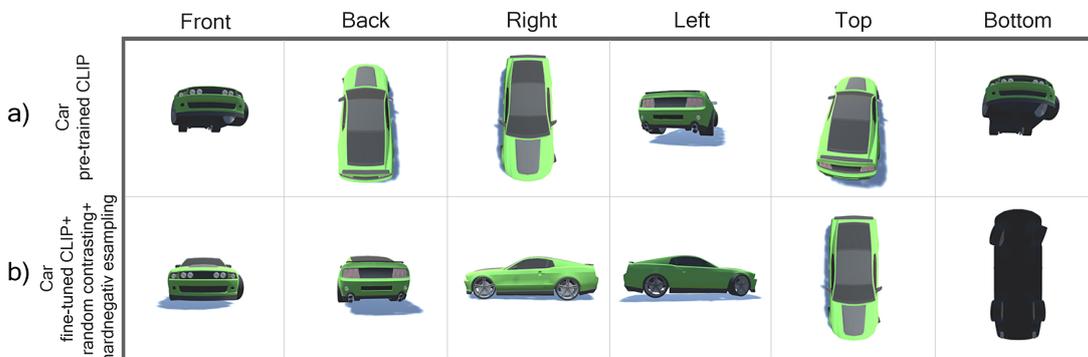


(b) Optimal viewpoints of the six canonical views for a) cars, b) motorbikes, c) airplanes, d) benches, and e) mugs of the ShapeNet data set (Chang et al., 2015) retrieved from the optima of the CLIP-RC-HNS scoring function.

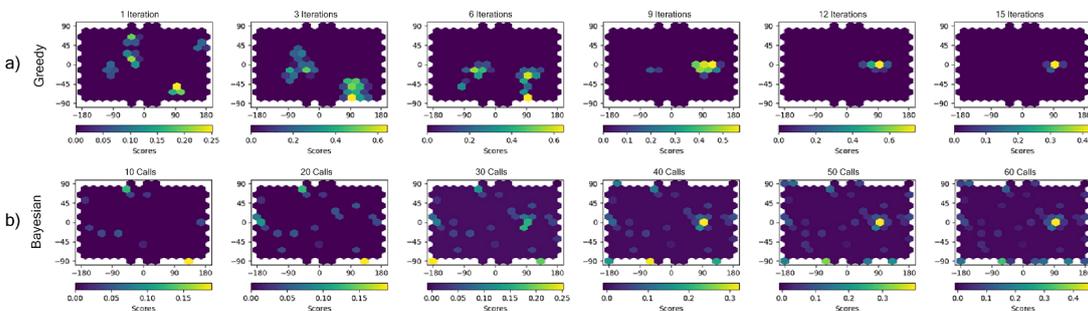
Figure 7: Scoring Function Distributions on CLIP-RC-HNS and retrieved viewpoint images.



(a) Scoring function distribution on cars, motorbikes, airplanes, benches, and mugs given the query *a picture of an X*, where X stands as a variable for *car/motorbike/airplane/bench/mug*



(b) Comparison of optimal viewpoints of the six canonical views between a) PRE-TRained CLIP and b) CLIP-RC-HNS.



(c) A single run of the search for the respective search algorithms a) greedy, b) Bayesian, on a randomly selected car object from the ShapeNet data set (Chang et al., 2015) given the search query *a picture of a car from the left*.

Figure 8: *top*: Distribution on object-only queries, *center*: retrieved optimal viewpoints on CLIP PRE-TR and RC-HNS, *bottom*: Execution of search algorithms.

PLACES: Prompting Language Models for Social Conversation Synthesis

Maximillian Chen^{1*}, Alexandros Papangelis², Chenyang Tao², Seokhwan Kim²,
Andy Rosenbaum², Yang Liu², Zhou Yu¹, Dilek Hakkani-Tur²

¹Columbia University, ²Amazon Alexa AI

maxchen@cs.columbia.edu, zy2461@columbia.edu

{papangea,chenyt,seokhwk,andros,yangliud,hakkanit}@amazon.com

Abstract

Collecting high quality conversational data can be very expensive for most applications and infeasible for others due to privacy, ethical, or similar concerns. A promising direction to tackle this problem is to generate synthetic dialogues by prompting large language models. In this work, we use a small set of expert-written conversations as in-context examples to synthesize a social conversation dataset using prompting. We perform several thorough evaluations of our synthetic conversations compared to human-collected conversations. This includes various dimensions of conversation quality with human evaluation directly on the synthesized conversations, and interactive human evaluation of chatbots fine-tuned on the synthetically generated dataset. We additionally demonstrate that this prompting approach is generalizable to multi-party conversations, providing potential to create new synthetic data for multi-party tasks. Our synthetic multi-party conversations were rated more favorably across all measured dimensions compared to conversation excerpts sampled from a human-collected multi-party dataset.

1 Introduction

Training dialogue models typically requires an abundance of data, as with any machine learning task. However, collecting high quality data is difficult and expensive, especially for dialogue tasks where there often is no “right answer” when developing the trajectory of a conversation. Typically dialogue data are sourced from crowdworkers and the quality of annotations, evaluations, and conversations can vary considerably (Zhao and Zhu, 2014), often necessitating guardrails such as credential-based worker selection or defensive task design for quality control (Allahbakhsh et al., 2013).

To accommodate data scarcity in training dialogue tasks, low resource methods have become

*Work done during internship at Amazon Alexa AI

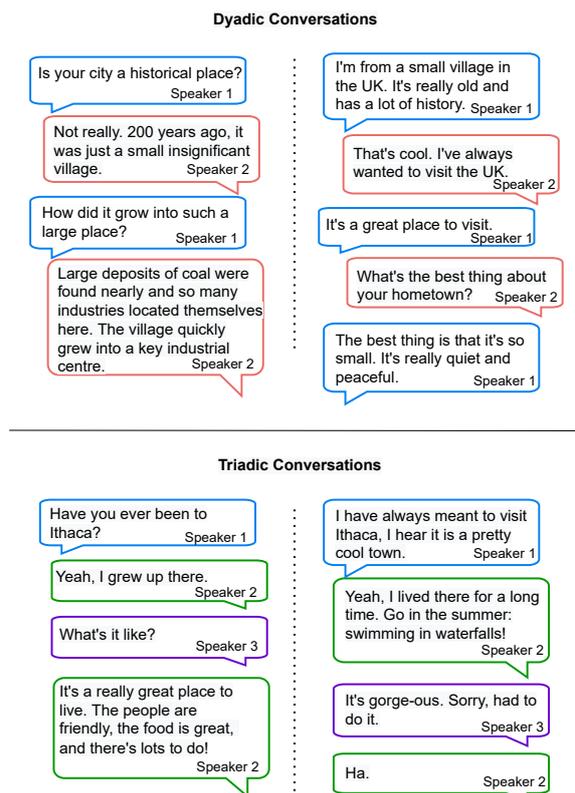


Figure 1: Pair of dyadic conversation excerpts about hometowns (upper) and pair of triadic conversation excerpts about Ithaca, NY (lower). In both pairings, one conversation is synthetically generated and the other is collected from humans. The answer is in Section 4.

a topic of growing interest and importance (Zhao et al., 2019; Mi et al., 2019; Qian and Yu, 2019; Li et al., 2019). One idea that has gained particular attention is transfer learning — specifically, finding ways to leverage knowledge learned by pre-trained large language models (PLMs) for new tasks. PLMs have demonstrated impressive emerging conversational capabilities, enabling big performance improvements in various dialogue tasks (Brown et al., 2020; Shuster et al., 2022; Peng et al., 2022; Kulhánek et al., 2021). Particularly, PLMs have been prompted to augment existing conversational data (Chen et al., 2022; Mehri et al.,

2022; Sahu et al., 2022).

Given some in-distribution seed examples, augmentation techniques attempt to generate data that are faithful to some task distribution (Kim et al., 2021b). Albeit powerful, one caveat common to all augmentation techniques is that the quality of synthetic data heavily relies on seed examples. But, what if crowdworkers do not possess the necessary background or skill set to complete a task en masse? How can we still get adequate high-quality synthetic data to learn a task?

In this work, we explore a novel application of **Prompting L**anguage models for social **Conv**ersation **S**ynthesis (PLACES). Synthesizing conversational datasets allows for the construction of training instances in nonexistent tasks. We specifically conduct open-domain, topic-conditioned conversation generation using few-shot in-context learning with expert-written synthetic conversations. We conjecture that expert end-users know exactly the types of conversations that they need. Rather than using existing datasets, they can simply write a small set of high quality conversation examples according to the structure of their desired conversational outputs. We reason that given structure through high-quality in-context demonstrations, large PLMs are able to utilize their expansive pre-training data (e.g. Gao et al. (2020)) to synthesize realistic social conversations, implicitly creating personalities and backgrounds for hypothetical speakers. The process of conversation writing would otherwise require human creativity and effort.

Our paper makes four core contributions.

(1) PLACES involves synthesizing an entire conversational dataset from a few targeted expert-written examples. These conversations match the quality of two widely adopted social dialogue datasets, Daily-Dialog (Li et al., 2017) and Topical Chat (Gopalakrishnan et al., 2019), in terms of human evaluation and automatic metrics. (2) We demonstrate that our synthetic conversations can be used as a fine-tuning dataset which matches the performance of its human-curated counterparts as measured by an interactive human evaluation and automatic metrics. (3) We apply PLACES to synthesize data for an under-studied subfield of dialogue research: multi-party conversations. We evaluate a set of synthetic triadic conversations in comparison to two human-collected multi-party conversational datasets (Shaikh et al., 2010; Poria et al., 2019).

To our knowledge, our work is the first to synthesize multi-party conversations, adding to the still-growing body of work on multi-party social dialogue. (4) Lastly, we conduct an error analysis on both dyadic and triadic synthetic conversations. We discuss the implications of our findings, as well as potential solutions to address the generation “errors.”

2 Related Work

Recently, the zero- and few-shot learning capabilities of large pre-trained language models have overtaken state-of-the-art performance on many classical natural language processing tasks, including dialogue (Brown et al., 2020). Many PLMs such as T5 (Raffel et al., 2020), GPT-J (Wang and Komatsuzaki, 2021), GPT-3 (Brown et al., 2020), and OPT (Zhang et al., 2022) have become the backbone of several dialogue-specific models (e.g., Peng et al. (2022); Madotto et al. (2021); Shuster et al. (2022)).

In particular, in-context learning, where few-shot examples are provided in the input prompt of a PLM, has been found to provide valuable information in guiding generation output (Min et al., 2022; Brown et al., 2020; Min et al., 2021; Lu et al., 2021b). As a result, many recent efforts in prompting PLMs have sought to augment various natural language processing datasets (Chen et al., 2022; Wang et al., 2022; Sahu et al., 2022; Mehri et al., 2022; Rosenbaum et al., 2022a). Prompting has become a viable “solution” for augmentation in dialogue tasks, which have traditionally been considered challenging due to the difficulty of augmenting dialogue context (Chen et al., 2022).

However, prompt-based augmentation strategies are uncontrolled forms of generation, which may result in generation mistakes for labeled datasets (Sahu et al., 2022; Chen et al., 2022; Meng et al., 2022). In contrast, other recent studies have instead proposed language augmentation strategies that use complex, highly-controlled frameworks that often involve fine-tuning generators (Papangelis et al., 2021; Zhang et al., 2020b; Kulhánek et al., 2021; Zhang et al., 2020a). Such complex augmentation frameworks require larger amounts of seed data to maintain a ground-truth language distribution (Rosenbaum et al., 2022b; Kim et al., 2021b), and are more costly than prompting PLMs (Chen et al., 2022). However, in the context of dataset synthesis, seed data and label correctness are less

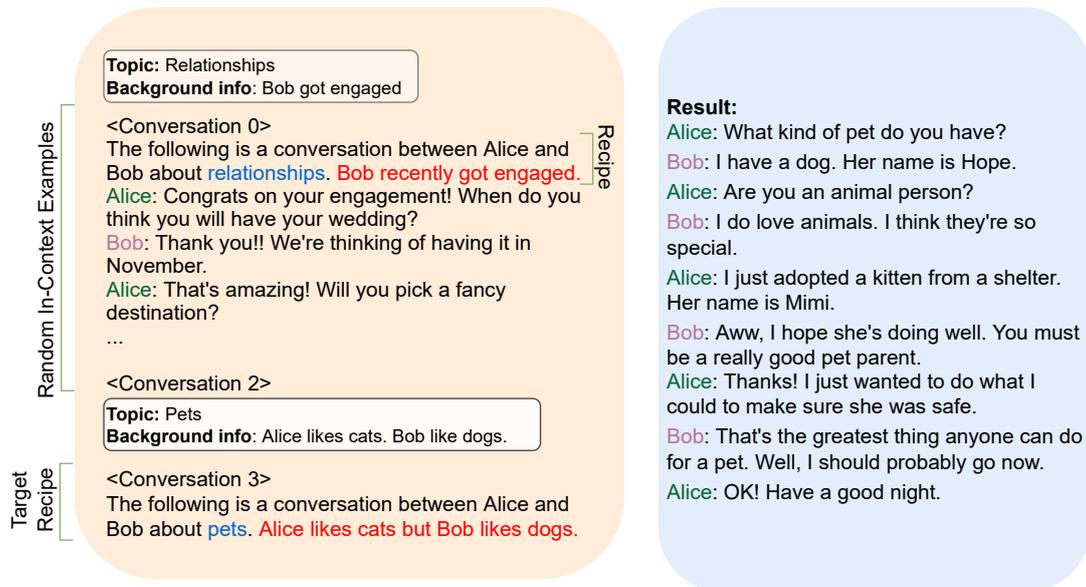


Figure 2: Example of the components of a prompt (left) used by OPT 30B to generate a synthetic conversation about pets (right). Conversations in the prompt are prefixed by recipes. Blue text: topic labels. Red text: seed background information metadata.

important considerations. There is no task distribution from which seed data is drawn that PLMs must remain faithful to, and similarly, invariant ground-truth knowledge for language models is dependent on the desired task being synthesized.

Our work differs from existing applications of prompting for conversations along several dimensions. Many studies examine utterance-level generation (Chen et al., 2022; Sahu et al., 2022; Aher et al., 2022; Rosenbaum et al., 2022b), whereas our work concerns the synthesis of full conversations. Bae et al. (2022) generated conversations for a narrow task and provided evaluations between their synthesis conditions. Recent concurrent work by Kim et al. (2022) sought to distill conversations from InstructGPT 175B using a common-sense knowledge graph. In our work, we synthesize conversations using an open-source PLM and demonstrate that they are comparable to human-collected datasets, in terms of both conversation quality and usability as a dataset. Moreover, all of these studies only concern dyadic conversations, because the vast majority of conversational tasks are dyadic. Our work is the first study to synthesize multi-party conversations.

3 Conversation Generation

In this section, we discuss our methods for conversation generation. We first detail the construction of our example conversations, then describe their application to prompting PLMs.

3.1 Writing Conversation Examples

We simply wrote a pool of ten conversations between two speakers representing everyday dialogue using proper grammar. Along with each conversation, we wrote a brief conversation “recipe” which includes a topic, as well as *background information* for the two speakers¹.

The *background information* represents some more fine-grained information about the two speakers, relevant to that particular topic. For example, Figure 2 depicts an example prompt with three in-context conversation demonstrations. Each conversation is prefixed by a recipe and is structured in the same manner: “The following is a conversation between Alice and Bob about *topic*” (e.g., “pets”) followed by detailed background information (e.g., “Alice love cats. Bob is more of a dog person.”).

3.2 Creating Conversations via Prompting

Each prompt consists of three randomly sampled conversations from the aforementioned pool, along with their accompanying recipe. After experimenting with PLMs of three different sizes (GPT-J 6B, GPT-NeoX 20B, OPT 30B), we primarily use OPT-30B and generate with nucleus sampling with $p = 0.92$. Inspired by the format of DailyDialog, our handwritten and synthetically generated conversations fall into three categories: start-to-finish conversations, excerpts from the start to the middle

¹The first-author spent approximately 45 minutes on this writing process.

Source	Words/Turn	Turns/Conv.
DailyDialog	11.58	7.84
Topical Chat	13.38	21.83
HW Examples	11.00	8.10
Synthetic	10.70	9.29

Table 1: Number of words per turn and number of turns per conversation for all conversations. HW Examples represents the ten handwritten conversation examples, and Synthetic represents synthetic conversations generated using OPT 30B.

of a conversation, and excerpts from the middle of a conversation. Several examples are given in the Appendix.

In this paper, we generate a dataset using a list of topics and tasks (i.e., subtopics) from the training set of the Feedback for Interactive Talk & Search Dataset (FITS; Xu et al. (2022)), a human-chatbot dataset designed to determine desirable human-chatbot tasks/conversations. FITS contains 5592 conversations which span 52 conversational topics (e.g., “nutrition,” “philosophy”) with 315 subtopics (e.g., “Italian food,” “Soren Kierkegaard”). We wrote background information for each of the 315 subtopics in the form given in Figure 2.

Using the product of this process once results in a new synthetic dataset with 5592 conversations using the same topic, subtopic pairings from FITS. The average length of each conversation is 9.29 turns, with 12.84 words per turn. This is comparable to the dataset statistics of DailyDialog and Topical Chat, as per Table 1. In the Appendix, we have included the 315 prompt headers (Tables S22, S23) and the pool of in-context examples (Tables S24, S25, S26).

4 Synthetic Conversation Evaluation

In Figure 1, the top-left is taken from DailyDialog, whereas the top-right is generated synthetically. The bottom-left is generated synthetically and the bottom-right is taken from MPC.

4.1 Evaluation of Conversation Quality

Table 2 provides a crowdworker evaluation of our synthetic dataset compared against DailyDialog and Topical Chat. We expect Topical Chat to be rated as the most interesting, due to the knowledge-grounding process utilized during the dialogue collection process. We randomly sampled 200 conversations for each conversation source and asked a pre-qualified pool of 28 crowdworkers on Amazon Mechanical Turk (AMT) to rate each conversation.

Source	Interesting	Coherent	Natural	Consistent
DailyDialog	3.44	4.51	4.85	4.57
Topical Chat	4.55	4.39	4.92	4.87
GPT-J 6B	3.96*	4.49	4.86	4.36
GPT-NeoX 20B	3.81*	4.40	4.63	4.35
OPT 30B	4.13*	4.61 *†	4.82	4.63

Table 2: Evaluation of conversations randomly sampled from DailyDialog, Topical Chat, and three synthetic datasets generated by prompting GPT-J 6B, GPT-NeoX 20B, and OPT 30B. * indicates statistical significance over DailyDialog. † indicates statistical significance over Topical Chat. Significance computed at $\alpha = 0.05$.

The instructions and details of our human evaluation setup are explained in Appendix A.

As these conversations are generated using prompting, we first checked whether each conversation followed the prescribed prompt. Crowdworkers identified 95% of the conversations generated by OPT 30B as matching the topic stated in the prompt², indicating this prompting strategy’s effectiveness for topic-grounded conversation generation. Overall, Table 2 indicates that synthetic conversations generated by OPT 30B are rated as the most coherent, and more interesting and consistent than DailyDialog. The synthetic conversations are almost as natural as DailyDialog, but are rated as less interesting and natural than Topical Chat. Given our results, we also hypothesize that larger models likely produce higher quality conversations. We provide several examples of conversations generated by OPT 175B using an online web interface³ in the Appendix.

A concern one might have is that since in-context examples heavily influence prompting (Min et al., 2022; Lu et al., 2021b), our small in-context example size may limit the lexical diversity of our synthetic conversations. Following earlier work evaluating text generation, we use Distinct-N to measure lexical diversity (Wu et al., 2021; Li et al., 2016). Figure 3 shows that our synthetically generated conversations are slightly more diverse than both DailyDialog and Topical Chat in terms of distinct bigrams and trigrams, and slightly less diverse than Topical Chat in terms of 4-grams.

We then sought to examine the impact of using expert handwritten examples by comparing against synthetic conversations generated using conversations from DailyDialog and Topical Chat as in-

²91% and 92% for GPT-J 6B and GPT-NeoX 20B.

³<https://opt.alpa.ai/>

Dimension	DD-IC	TC-IC	HW-IC
Interesting	3.82	4.35	4.27*
Coherent	4.48	4.56	4.77 **+
Natural	4.54	4.69	4.69 *
Consistent	4.76	4.87	4.86*
On-Topic	0.91	0.88	0.96 **+

Table 3: Human evaluation of conversations generated using OPT-30B with in-context examples randomly sampled from DailyDialog (DD-IC), Topical Chat (TC-IC), and handwritten examples (HW-IC). * indicates statistical significance over DD-IC and + indicates statistical significance over TC-IC.

context examples. We set the number of conversation examples such that the number of in-context dialogue turns are approximately equal across all conditions. Table 3 shows that synthetic conversations generated conditioned on handwritten in-context examples are the most coherent, natural, and on-topic. In terms of interestingness and consistency, the ratings of these conversations slightly trail the ratings of the conversations generated conditioned on Topical Chat.

4.2 Fine-Tuning with Synthetic Conversations

After establishing that our synthetic conversations are of rather high quality on their own, we attempted to use the synthetic dataset as training data for dialogue models. We fine-tuned distilled BlenderBot 400M (Roller et al., 2021) on DailyDialog, Topical Chat, and our synthetic conversations⁴.

Rather than directly prompting OPT as a response generator, we select BlenderBot as a lightweight, effective dialogue model. This allows for comparisons between the three data sources as training sets, because fine-tuning OPT is prohibitively expensive. Moreover, while prompting with larger PLMs can yield coherent responses, it is generally impractical as an end-to-end dialogue system if hosted on typically available hardware. For long inputs (e.g. with multiple dialogues in-context), generation time typically takes several minutes using OPT 30B⁵.

We first performed an interactive human evaluation of the three dialogue models as end-to-end social chatbots using the LegoEval platform (Li et al., 2021). Details can be found in Appendix A.

Table 4 shows that dialogue models fine-tuned on our synthetic conversations are rated compara-

⁴For fair comparison, we fine-tune on the same number of training instances via downsampling.

⁵All experiments are conducted using one p3dn.24xlarge AWS EC2 instance.

Dimension	DD	TC	Syn
Interesting	3.35	3.86	3.30
Coherent	3.52	3.71	3.68
Natural	3.52	3.57	3.68
Consistent	3.35	3.65	3.32
Engaging	3.73	3.88	3.65
Intelligent	3.41	3.55	3.24
Non-repetitive	3.37	3.37	3.40

Table 4: Interactive human evaluation yields comparable ratings for chatbots fine-tuned on conversations from DailyDialog (DD), Topical Chat (TC), and our Synthetic Data (Syn).

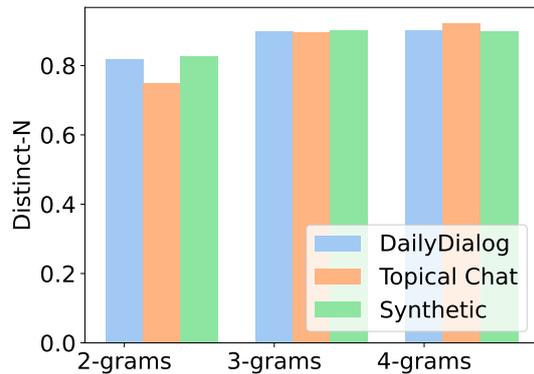


Figure 3: Distinct-N with $N = 2, 3, 4$ for conversations in DailyDialog, Topical Chat, and our synthetic conversations. Our synthetic conversations have the highest most unique bi-grams and tri-grams, and the second-most unique 4-grams.

bly to dialogue models fine-tuned on real human-human data — the chatbot fine-tuned on synthetic data appeared to be the most natural and non-repetitive, and was rated as the second-most coherent. It was rated as the least intelligent, engaging, consistent, and interesting. However, two-sided t-tests at $\alpha = 0.05$ revealed that there was not a statistically significant difference in ratings between the models fine-tuned on all three datasets across all dimensions except for interestingness. The Topical Chat model was rated as significantly more interesting, as expected.

In terms of automatic evaluation, we applied these dialogue models on out-of-distribution test sets to prevent an unfair comparison. We evaluated models fine-tuned on DailyDialog and our synthetic data on Topical Chat, and models fine-tuned on Topical Chat and our synthetic data on DailyDialog. Table 5 indicates that in terms of perplexity and ROUGE, models fine-tuned on our synthetic data generalize to out-of-distribution conversational data as well as models trained on real human-

Metric (Test Set)	DD-BB	TC-BB	Syn-BB
Perplexity (DD)	—	120.2	87.05
ROUGE-1 (DD)	—	12.34	12.90
ROUGE-2 (DD)	—	1.66	1.52
ROUGE-L (DD)	—	10.60	10.94
Perplexity (TC)	43.3	—	37.1
ROUGE-1 (TC)	16.63	—	15.13
ROUGE-2 (TC)	2.36	—	1.77
ROUGE-L (TC)	13.61	—	12.41

Table 5: Out-of-distribution automatic evaluation of perplexity and ROUGE is comparable for BlenderBot fine-tuned on DailyDialog (DD-BB), Topical Chat (TC-BB), and synthetic data generated using our handwritten examples in-context (Syn-BB), respectively.

human datasets. On the DailyDialog test set, the synthetic dataset model outperforms the Topical Chat model on all metrics except ROUGE-2, and on the Topical Chat test set, the synthetic dataset model underperforms the DailyDialog model on all metrics except perplexity.

5 Triadic and Multi-Party Conversations

The vast majority of dialogue tasks and conversational datasets focus on dyadic conversations (e.g. Li et al. (2017); Gopalakrishnan et al. (2019); Smith et al. (2020); Rashkin et al. (2019)), following the traditional speaker-listener paradigm (Engelhardt et al., 2006). In contrast, the literature on multi-party social conversation is rather scarce, not only in terms of conversation generation but as a task altogether. However, while it is an understudied research area, it is incredibly important, because dyadic conversations do not capture the full reality of in-person, human-human social conversations, nor the full potential of dialogue agents. To name a few applications, dialogue agents have the potential to supplement classroom learning with multiple parties, serving as a third mediating party in a debate or discussion between two people, or to provide companionship and support in virtual group settings. A major reason why these lines of work remain unsolved is that there are few large-scale multi-party dialogue datasets.

Many existing multi-party datasets are scripted corpora such as MELD (Poria et al., 2019) or MPDD (Chen et al., 2020) or HLA-Chat (Ju et al., 2022; Li et al., 2020). Other multi-party corpora are collected for highly domain-specific purposes, such as multi-party empathetic dialogue (Zhu et al., 2022). Such corpora are also typically collected through asynchronous online platforms, rather than natural conversation. These platforms exist in the

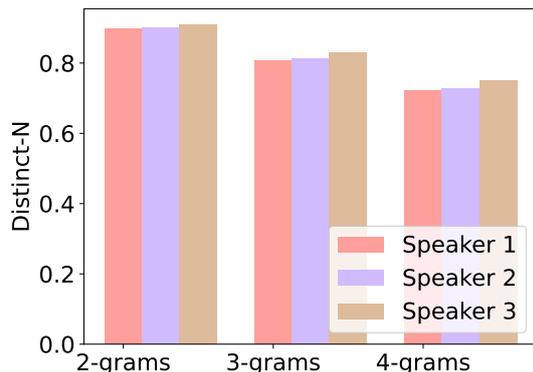


Figure 4: Linguistic diversity (Distinct-N) is comparable for each speaker in the synthetic triadic conversation dataset.

form of forums and online chat platforms such as Ubuntu IRC (Lowe et al., 2015) or Reddit (Baumgartner et al., 2020). Other more natural multi-party conversational datasets are license-protected speech datasets (e.g. CHIME (Christensen et al., 2010)) which have been constructed for tasks such as speaker attribution.

We find that we can apply our prompting approach to generate synthetic, open-domain, multi-party social conversations following the same structure as our synthetic dyadic conversations⁶. As in the dyadic case, we generate triadic conversations using optional background information for each speaker. We consider the “Multi-Party Chat” corpus (MPC) (Shaikh et al., 2010), a text-based, open-domain conversation dataset collected in real-time online sessions at the University of Albany, and MELD, which contains scripted multi-party dialogues from the popular sitcom “Friends.” We directly compare our synthetically generated conversations against MPC and MELD.

Table 6 includes our evaluation of our conversations using the same pool of pre-qualified AMT workers, again with 200 randomly sampled conversations. MPC consists of massive conversation settings — on the scale of 500 turns for a typical conversation session — so we randomly sample 8 to 12⁷ continuous turns for each conversation evaluation to more closely match the structure of our synthetic conversations.⁸ We present examples of

⁶While we effectively use Alice, Bob, and Claire instead of Speaker 1, Speaker 2, and Speaker 3, respectively, the order of speakers does not necessarily follow the speaker order in the in-context examples (e.g. Appendix Table S10).

⁷The length between 8 and 12 turns is chosen uniformly.

⁸We sample rather than selecting the first 8-12 turns, to

Dimension	MPC	MELD	Syn
Interesting	2.48	3.52	4.14*
Coherent	2.40	3.68	4.65*
Natural	2.69	3.69	4.47*
Consistent	2.96	3.83	4.65*
Comprehensible	2.48	3.83	4.80*
Balanced Engagement	3.45	4.00	4.89*

Table 6: Synthetic conversations generated using OPT 30B are rated significantly higher than MPC and MELD across all dimensions.

MPC and MELD in Appendix Tables S20, S21.

We inform the AMT workers that they will read conversation excerpts. In addition to the questions in Table 2, we add two questions specific to multi-party conversations. We ask if the conversation excerpt looks comprehensible (in terms of the reader being able to determine who each speaker is addressing), and we ask if all parties of the conversation are participating equally and actively.

In Table 6, we find that the synthetic conversations are rated statistically significantly more favorably than MPC and MELD across all dimensions. Beyond conversation quality, it is possible that the ratings for MPC are comparatively low due to the fact that each conversation typically has more than three speakers, which may be more difficult for human raters to interpret. Our results for MELD also indicate that while the corpus is high quality, it may be better fit for comedy and accompaniment with visual context, than as pure dialogue.

Additionally, we checked the linguistic diversity for each speaker. In terms of Distinct-N, each speaker’s lexical diversity is comparable (Figure 4) as well as the number of words per turn (12.2, 12.2, and 13.5 for Speakers 1, 2, and 3 respectively). The triadic conversations tended to be slightly longer than the average dyadic conversation (11.5 turns/conversation versus 9.29 turns/conversation).

6 Discussion

Overall, we find that prompting PLMs to generate synthetic conversations is promising.

6.1 Considerations for Dyadic Dialogue

The synthetically generated conversations appear comparable to conversations from human-collected datasets. The individual conversations appear interesting, coherent, natural, and consistent, as the average ratings for each category lie between 4.0 and 5.0. The Appendix includes multiple examples avoid overrepresenting greetings.

of conversations generated using the strongest performing PLM (OPT 30B, e.g. Table S7) as well as several conversations generated using OPT 175B (e.g. Table S8). Tables 4 and 5 also indicate that fine-tuning on synthetically generated examples can result in dialogue models of comparable quality, with the potential for further improvements by simply generating more synthetic conversations.

Future work may consider applying applying this generation approach to dyadic contexts beyond social conversations, such as task-oriented dialogue. The clearest difference between social and task-oriented dialogue contexts is the importance of knowledge grounding. In task-oriented dialogue, there typically needs to be retrieval from knowledge base for response generation. An application of PLACES could involve using database results as a ground-truth reference. Rather than using a topic list like FITS, one could form conversational recipes using database search results as background information. Given the apparent semantic control described in Section 4, it is possible that synthetic task-oriented conversations would be able to correctly utilize knowledge.

6.2 Considerations for Multi-Party Dialogue

We found that in comparison to MPC, our synthetic triadic dialogues appear to be of fairly high quality. However, there remain several open questions about multi-party dialogue, even in the triadic case. For instance, there is not a set archetype of conversations. Sometimes, conversations may be dominated by a single speaker, whereas in others, each speaker in the conversation may contribute equally. Depending on the scenario, a speaker may be the facilitator — meetings can be considered (topic-specific) multi-party dialogues which are typically led by designated speakers.

Moreover, there are several questions about how to utilize multi-party dialogues in an interactive dialogue system. There are use cases where it may be appropriate for one dialogue system to interact with multiple users. On the other hand, in scenarios like emotional support dialogue systems, it may make sense for a single user to interact with multiple simulated conversational parties.

Here, we investigated our approach’s potential to generate synthetic multi-party conversations, hoping to bridge the gap in data availability in multi-party chat. This opens opportunities for a variety of applications. Synthetic datasets could be used

to help discover how to properly model triadic and multi-party conversations. In the future, datasets could also be generated for domain-specific, multi-party applications ranging from language learning to task-oriented spoken dialogue systems.

7 Error Analysis

We examine the dyadic and triadic conversations which received low scores (1/5) across multiple dimensions.

7.1 Dyadic Conversations

Out of the dyadic conversations, two conversations were rated as generic and dull. One conversation (Appendix Table S13) talks about the singer, Taylor Swift. However, the conversation is repetitive, repeating utterances such as “What are your thoughts on her?” and “I think she is very nice.” The other conversation is about the filmmaker, Ken Burns (Appendix Table S14). While the conversation is appears coherent and uses correct factual information (e.g., making reference to Ken Burns’ documentaries on World War II and the Vietnam War), the language could be perceived as dull.

Three conversations were rated as completely unnatural. In one case, the PLM missed the prescribed subtopic (cotton candy) and instead hallucinated a conversation about a sensitive topic, cancer (Appendix Table S15). This is also the only conversation to be rated as completely incoherent. The other two conversations are both on-topic. However, one conversation is on-topic but rather short (five turns), whereas the other conversation is overly verbose and a little repetitive.

There were also three conversations were evaluated as completely inconsistent. In all three conversations, the roles of the two speakers seemingly swap. While these hypothetical turns are possible in excerpts of real conversations, they assume background information or events which have not been explicitly established when considered as standalone conversations. An example is given in Appendix Table S16.

While some of the evaluations may be subjective, an issue that has objectively appeared multiple times is the consistency of speakers’ utterances. The intents and personas of the speakers appear to get switched, which is also an open problem in dialogue systems research. Future work may look to combine conversation synthesis approaches with strategies for dialogue consistency such as the generate-delete-rewrite framework (Song et al.,

2020a) or language inference approaches (Welleck et al., 2019; Song et al., 2020b).

7.2 Triadic Conversations

No conversations were perceived as completely incomprehensible, but human evaluators indicated that two conversations appeared to have imbalanced engagement — in both cases, the third speaker (“Claire”) only has one dialogue turn. As discussed in Section 6.2, however, it is not clear whether this is a drawback. Real-life triadic conversations do not follow a set archetype in terms of engagement balance.

There was one conversation which was rated as completely incoherent. In the conversation, there is one dialogue turn which presents information inconsistent with prior turns, but the another issue appears to be an oddly placed transition which brings the conversation from travel to hobbies: “You should definitely go to Paris! What do you like to do for fun?” (Appendix Table S17).

There are two conversations which were perceived as completely unnatural. However, naturalness appears to be a rather subjective evaluation. One conversation is given in Appendix Table S18, and it is debatable whether the language conventions used are unnatural. One could argue that it is overly enthusiastic, but others could argue that it is how some people speak colloquially. Interestingly, the second conversation which received a low naturalness score is also enthusiastic and about the same topic (gardening).

The only conversation which was rated as generic and dull was a 15-turn debate about whether the European Union is a “conspiracy” (Appendix Table S19). The debate is rather shallow and does not make a lot of progress.

As with the dyadic conversation error analysis, we see that there are issues with persona consistency. However, unlike the dyadic scenario, there are fewer existing solutions for dialogue consistency. Multi-party conversation synthesis could potentially be improved by applying ideas from the newly published PersonaTKG dialogue system, which employs a unified graph that encodes personas, utterances, and external knowledge on a scripted dialogue dataset (Ju et al., 2022).

Beyond consistency, in the example from Table S19 we see that there is potential for PLMs to hallucinate misinformation. There are again fewer existing studies on circumventing this obstacle in multi-party dialogue, but future work could look

to incorporating external knowledge (Kang et al., 2022) or dialogue safety approaches (Kim et al., 2021a; Dinan et al., 2019). All said, our work motivates further study into multi-party dialogue consistency, safety, and synthesis.

8 Conclusion

In this work, we presented an application of prompting PLMs to create synthetic conversations. These synthetic conversations are comparable in terms of quality and lexical diversity to actual human-human datasets, and can be used as training data for dialogue models. This opens avenues in generative language work such as collaborative and creative writing, story generation, as well as synthesis of new conversational tasks. Here, we presented one example — synthesizing a multi-party conversational dataset. This presents a unique opportunity to further study multi-party dialogue modeling.

9 Limitations

Controllability. We witness encouraging levels of control through the prompt (95% of the time, the synthetic conversation matches the desired topic), but prompting PLMs is still an uncontrolled form of generation. Future work could seek to add more semantic controls beyond the stated topic in the prompt or explore using weak supervision to provide post-hoc improvements on synthetic data quality, similar to Chen et al. (2022). In this work, we also did not thoroughly explore the effects of different generation approaches. Future work may consider applying semantic constraints during the decoding process (Lu et al., 2021a). Further controls are necessary before using this approach for higher-stakes settings such as task-oriented dialogue and other knowledge-grounded tasks.

Cost of Human Effort. While we demonstrate the ability to synthesize large amounts of data, the quality of a synthesized dataset is still dependent on human effort, to an extent. One can use a generic prompt template such as “Alice is interested in [subtopic]” for each subtopic, but we qualitatively see that more detailed background information in a prompt often yields better generation performance.

In this work, we generated 5592 dyadic and triadic conversations, matching the number of topic combinations in FITS. PLACES can be used to generate many more conversations in the future. Using the same overall can continue to make new

combinations of topic and subtopic, or simply re-run the generation process as it is nondeterministic. Moreover, one may consider filling the slots in our conversation recipes using an abundant of external sources, including from existing dataset annotations (e.g. Persona Chat Zhang et al. (2018)).

Computational Costs. Once a dataset is synthesized, small, task-specific models can be used downstream. However, the synthesis method used in this work is still expensive: we prompt PLMs. While we only used freely accessible PLMs such as OPT, we acknowledge that not everyone has access to the number of GPUs necessary to load PLMs, even for inference.

Prompt Design. The idea of prompting large language models is not novel. There is a plethora of work that examines how to apply prompting to a variety of different tasks (e.g. Brown et al. (2020); Min et al. (2021)), along with several studies on how to mine or engineer different prompts (Liu et al., 2021). In this work, we do not claim novelty to our prompt, nor do we claim that our prompt design is the optimal prompt for conversation generation. Our prompt is designed in a conversational manner, drawing inspiration from Chen et al. (2022). We instead emphasize the application of prompting for conversational dataset synthesis. The idea of synthesizing conversational datasets “from scratch” is previously unexplored, and has potential to supplement a lot of areas of dialogue research, such as multi-party conversations.

10 Ethical Considerations

Human Evaluation and Crowdsourcing. We make use of crowdsourcing through Amazon Mechanical Turk for several experiments. All crowdworkers were paid at a rate higher than the minimum wage in California. In accordance with California State Law, all crowdworkers were also informed they were speaking with chatbots during the data collection for our interactive evaluation. All participants consented to the logging of their responses.

Language Model Biases. Large pre-trained language models are typically pre-trained on massive corpora crawled from the internet such as The Pile (Gao et al., 2020) or Common Crawl. This allows language models to have exposure to a large amount of linguistic diversity, but this also results in exposure to a lot of hateful, biased, or otherwise

undesirable content from the internet (Luccioni and Viviano, 2021). Future work should examine combining conversation synthesis with dialogue safety approaches.

Scientific Artifacts. All scientific artifacts are used according to their intended purpose. The FITS dataset is publicly available at <https://parl.ai/projects/fits/>. OPT is an open-source language model. GPT-J is available for use under the MIT license. We use the HuggingFace Transformers and PyTorch packages for all modeling (Wolf et al., 2020; Paszke et al., 2019). All artifacts used are in English.

References

- Gati Aher, Rosa I Arriaga, and Adam Tauman Kalai. 2022. Using large language models to simulate multiple humans. *arXiv preprint arXiv:2208.10264*.
- Mohammad Allahbakhsh, Boualem Benatallah, Aleksandar Ignjatovic, Hamid Reza Motahari-Nezhad, Elisa Bertino, and Schahram Dustdar. 2013. *Quality control in crowdsourcing systems: Issues and directions*. *IEEE Internet Computing*, 17(2):76–81.
- Sanghwan Bae, Donghyun Kwak, Sungdong Kim, Donghoon Ham, Soyoung Kang, Sang-Woo Lee, and Woomyoung Park. 2022. *Building a role specified open-domain dialogue system leveraging large-scale language models*. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2128–2150, Seattle, United States. Association for Computational Linguistics.
- Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Maximillian Chen, Alexandros Papangelis, Chenyang Tao, Andy Rosenbaum, Seokhwan Kim, Yang Liu, Zhou Yu, and Dilek Hakkani-Tur. 2022. Weakly supervised data augmentation through prompting for dialogue understanding. *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*.
- Yi-Ting Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. Mpdd: A multi-party dialogue dataset for analysis of emotions and interpersonal relationships. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 610–614.
- Heidi Christensen, Jon Barker, Ning Ma, and Phil D Green. 2010. The chime corpus: a resource and a challenge for computational hearing in multisource environments. In *Eleventh Annual Conference of the International Speech Communication Association*. Citeseer.
- Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. *Build it break it fix it for dialogue safety: Robustness from adversarial human attack*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4537–4546, Hong Kong, China. Association for Computational Linguistics.
- Paul E Engelhardt, Karl GD Bailey, and Fernanda Ferreira. 2006. Do speakers and listeners observe the gricean maxim of quantity? *Journal of memory and language*, 54(4):554–573.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.
- Karthik Gopalakrishnan, Behnam Hedayatnia, Qinglang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, Dilek Hakkani-Tür, and Amazon Alexa AI. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. In *INTERSPEECH*, pages 1891–1895.
- Dongshi Ju, Shi Feng, Pengcheng Lv, Daling Wang, and Yifei Zhang. 2022. *Learning to improve persona consistency in multi-party dialogue generation via text knowledge enhancement*. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 298–309, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2022. Knowledge-consistent dialogue generation with knowledge graphs. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.
- Byeongchang Kim, Hyunwoo Kim, Seokhee Hong, and Gunhee Kim. 2021a. How robust are fact checking systems on colloquial claims? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1535–1548.
- Hyunwoo Kim, Jack Hessel, Liwei Jiang, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Le Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, et al. 2022. Soda: Million-scale dialogue distillation with social commonsense contextualization. *arXiv preprint arXiv:2212.10465*.

- Yekyung Kim, Seohyeong Jeong, and Kyunghyun Cho. 2021b. Linda: Unsupervised learning to interpolate in natural language processing. *arXiv preprint arXiv:2112.13969*.
- Jonáš Kulhánek, Vojtěch Hudeček, Tomáš Nekvinda, and Ondřej Dušek. 2021. Augpt: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 198–210.
- Aaron W Li, Veronica Jiang, Steven Y Feng, Julia Sprague, Wei Zhou, and Jesse Hoey. 2020. Aloha: Artificial learning of human attributes for dialogue agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8155–8163.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.
- Juntao Li, Lisong Qiu, Bo Tang, Dongmin Chen, Dongyan Zhao, and Rui Yan. 2019. Insufficient data can also rock! learning to converse using smaller data with augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6698–6705.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.
- Yu Li, Josh Arnold, Feifan Yan, Weiyan Shi, and Zhou Yu. 2021. Legoeval: An open-source toolkit for dialogue system evaluation via crowdsourcing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 317–324.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.
- Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.
- Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021a. [NeuroLogic decoding: \(un\)supervised neural text generation with predicate logic constraints](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299, Online. Association for Computational Linguistics.
- Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021b. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.
- Alexandra Sasha Luccioni and Joseph D Viviano. 2021. What’s in the box? a preliminary analysis of undesirable content in the common crawl corpus. *arXiv preprint arXiv:2105.02732*.
- Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *arXiv preprint arXiv:2110.08118*.
- Shikib Mehri, Yasemin Altun, and Maxine Eskenazi. 2022. [Lad: Language models as data for zero-shot dialog](#).
- Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. In *Advances in Neural Information Processing Systems*.
- Fei Mi, Minlie Huang, Jiyong Zhang, and Boi Faltings. 2019. Meta-learning for low-resource natural language generation in task-oriented dialogue systems. *arXiv preprint arXiv:1905.05644*.
- Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.
- Sewon Min, Xinxin Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.
- Alexandros Papangelis, Karthik Gopalakrishnan, Aishwarya Padmakumar, Seokhwan Kim, Gokhan Tur, and Dilek Z. Hakkani-Tür. 2021. Generative conversational networks. In *SIGDIAL*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Baolin Peng, Michel Galley, Pengcheng He, Chris Brockett, Lars Liden, Elnaz Nouri, Zhou Yu, Bill Dolan, and Jianfeng Gao. 2022. Godel: Large-scale pre-training for goal-directed dialog. *arXiv preprint arXiv:2206.11309*.
- Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019. Meld: A multimodal multi-party

- dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536.
- Kun Qian and Zhou Yu. 2019. Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2639–2649.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. [Recipes for building an open-domain chatbot](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.
- Andy Rosenbaum, Saleh Soltan, Wael Hamza, Amir Saffari, Macro Damonte, and Isabel Groves. 2022a. Clasp: Few-shot cross-lingual data augmentation for semantic parsing. *arXiv preprint arXiv:2210.07074*.
- Andy Rosenbaum, Saleh Soltan, Wael Hamza, Yannick Versley, and Markus Boese. 2022b. Linguist: Language model instruction tuning to generate annotated utterances for intent classification and slot tagging. *arXiv preprint arXiv:2209.09900*.
- Gaurav Sahu, Pau Rodriguez, Issam H Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. *arXiv preprint arXiv:2204.01959*.
- Samira Shaikh, Tomek Strzalkowski, George Aaron Broadwell, Jennifer Stromer-Galley, Sarah M Taylor, and Nick Webb. 2010. Mpc: A multi-party chat corpus for modeling social phenomena in discourse. In *LREC*.
- Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, et al. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *arXiv preprint arXiv:2208.03188*.
- Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents’ ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030.
- Haoyu Song, Yan Wang, Wei-Nan Zhang, Xiaojiang Liu, and Ting Liu. 2020a. [Generate, delete and rewrite: A three-stage framework for improving persona consistency of dialogue generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5821–5831, Online. Association for Computational Linguistics.
- Haoyu Song, Wei-Nan Zhang, Jingwen Hu, and Ting Liu. 2020b. Generating persona consistent dialogues by exploiting natural language inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8878–8885.
- Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. <https://github.com/kingoflolz/mesh-transformer-jax>.
- Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. Promda: Prompt-based data augmentation for low-resource nlu tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4255.
- Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. [Dialogue natural language inference](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.
- Qingyang Wu, Lei Li, and Zhou Yu. 2021. Textgail: Generative adversarial imitation learning for text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14067–14075.
- Jing Xu, Megan Ung, Mojtaba Komeili, Kushal Arora, Y-Lan Boureau, and Jason Weston. 2022. [Learning new skills after deployment: Improving open-domain internet-driven dialogue with human feedback](#).
- Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020a. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *ACL*.
- Rongsheng Zhang, Yinhe Zheng, Jianzhi Shao, Xiao-Xi Mao, Yadong Xi, and Minlie Huang. 2020b. Dialogue distillation: Open-domain dialogue augmentation using unpaired data. *ArXiv*, abs/2009.09427.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have

pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2019. Low-resource knowledge-grounded dialogue generation. In *International Conference on Learning Representations*.

Yuxiang Zhao and Qinghua Zhu. 2014. Evaluation on crowdsourcing research: Current status and future direction. *Information Systems Frontiers*, 16(3):417–434.

Ling.Yu Zhu, Zhengkun Zhang, Jun Wang, Hongbin Wang, Haiying Wu, and Zhenglu Yang. 2022. **Multi-party empathetic dialogue generation: A new task for dialog systems**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 298–307, Dublin, Ireland. Association for Computational Linguistics.

A Human Evaluation Setup

Our human evaluation studies on Amazon Mechanical Turk are evaluated conducted with 28 pre-qualified crowdworkers, who have previously demonstrated proficiency with natural language processing tasks.

A.1 Conversation Evaluation

The crowdworkers were asked to rate conversations from multiple sources according to the following dimensions and instructions.

- *How natural is the overall conversation?*
Scale: 1 (completely unnatural) to 5 (as natural as two native English speakers)
- *How coherent is the overall conversation?*
Scale: 1 (completely incoherent) to 5 (as coherent as two native English speakers)
- *How interesting is the overall conversation?*
Scale: 1 (generic and dull) to 5 (full of content and very engaging)
- *How consistent are each of the speakers' turns?*
Scale: 1 (completely inconsistent) to 5 (no logical fallacies)
- *Does the conversation match the stated topic?*
Options: Yes (1) or No (0)

Each conversation is rated by three crowdworkers, and the median score is selected, following the idea of a majority vote.

For multi-party conversations, crowdworkers were asked two additional questions regarding comprehensibility and engagement balance.

- Can you tell which speaker is speaking to which?
Scale: 1 (completely incomprehensible) to 5 (perfectly comprehensible)
- Is each speaker engaged, or is the conversation primarily dominated by one or two of the speakers?
Scale: 1 (totally dominated by one or two speakers) to 5 (all speakers are actively participating in the conversation to an equal degree)

A.2 Interactive Evaluation

For each HIT of the interactive evaluation study, each crowdworker was presented with links to chatbots presented in a randomized order. The link connects each crowdworker to a deployment on an instance of LegoEval (Li et al., 2021). The users are presented with a landing page where they are told that they are interacting with a chatbot, and will be asked to evaluate their conversation experience.

Immediately after interacting with a chatbot, each crowdworker was presented with a survey asking for their impression of the chatbot. In addition to the above dimensions (other than on-topic), the crowdworkers were asked how engaging, intelligent, and non-repetitive they thought the chatbot was.

B Model Details

During generation, we use top- p sampling with $p = 0.92$.

Party	Utterance
Alice	I love it here in New York!
Bob	What do you like most about New York?
Alice	New York has the best bagels!
Bob	Which bagels are your favorite?
Alice	They all taste amazing. I really like their Italian ones!
Bob	That's great. I guess I'll be eating Italian bagels from now on.
Alice	I'm from New York City.
Bob	Wow! I've never been to New York.
Alice	I've also been to Austin. few times.
Bob	What did you think of it? Did you like it?
Alice	I really liked it. I thought it was a really cool city.
Bob	I'm glad you liked it. What did you like so much about it?
Alice	I really like the music scene there.
Bob	What kind of music do you like?
Alice	I like a lot of different kinds of music. I like jazz, rock, and country.
Bob	I've never been to a country concert.
Alice	You should go sometime. They're a lot of fun.
Bob	I'll have to check it out.
Alice	I'll have to visit Austin again soon.
Bob	Yeah, I'd like to visit New York sometime too.
Alice	It's a great city.

Table S7: Pair of dyadic conversations generated using OPT 30B. The prompt recipe given is: “The following is a conversation between Alice and Bob about their hometowns. Bob is from Austin, Texas, and Alice is from New York City.”

Party	Utterance
Alice	What's your major?
Bob	I'm a computer science major.
Alice	Me too! What year are you?
Bob	I'm in my third year. What about you?
Alice	I'm in my third year too. What classes are you taking this quarter?
Bob	I'm taking a class called Operating Systems. It's really interesting.
Alice	I'm taking a class called Computer Graphics. It's pretty interesting too.
Bob	I've heard that class is really hard.
Alice	Yeah, it's not easy.
Alice	Hi!
Alice	What are you studying?
Bob	I'm a Computer Science major.
Alice	Me too!
Bob	What year are you?
Alice	I'll be a junior next year.
Bob	I'll be a senior.
Alice	Cool!
Bob	What classes are you taking this quarter?
Alice	I've got a couple of classes in the School of Engineering. I'm taking a class in Computer Graphics and a class in Artificial Intelligence.
Bob	I'm taking a class on Machine Learning.
Alice	That's cool.
Bob	What are you doing this summer?
Alice	I'd like to work at a start-up.
Bob	That's cool. I'm going to intern at Google.
Alice	Wow! That's awesome!
Bob	Yeah, I'm really excited.
Alice	I'm excited for you!

Table S8: Pair of dyadic conversations generated using OPT 175B. The prompt recipe given is: “The following is a conversation between Alice and Bob about their majors. They are both students at Stanford University.”

Party	Utterance
Alice	I was thinking about applying at the coffee shop on the corner.
Bob	That's a great idea! It's a really fun place to work.
Alice	I'm glad to hear that. What do you do there?
Bob	I work as a barista.
Alice	Oh cool! I bet you get to meet a lot of interesting people.
Bob	That's right. I really enjoy working there.
Alice	Do you know if they are hiring?
Bob	I'm not sure, but you should definitely apply.
Alice	Thanks! I'll go in and see if they are hiring.

Table S9: Example dyadic conversation generated using OPT 175B. The prompt recipe given is: “The following is a conversation between Alice and Bob about part-time jobs. Bob works at a coffee shop and Alice is thinking about applying.”

Party	Utterance
Alice	I'd love to start a garden! I've always wanted to grow my own vegetables.
Bob	That's a great idea! It's so much more fun to eat vegetables that you've grown yourself.
Claire	What vegetables are you thinking about growing?
Alice	I'd love to try growing some tomatoes, peppers, and squash.
Bob	That sounds like a good start.
Claire	I've grown tomatoes before. I also have a few pepper plants.
Alice	I've always wanted to have a garden in my backyard, but I never really had the time.
Claire	That's awesome! How do you plan to grow your garden?
Alice	I've been doing a lot of research. I think I'll start with a few simple vegetables like tomatoes, onions, and potatoes.
Bob	That's great! I would love to have a garden like that. I'm a big fan of cooking.
Alice	I'll definitely invite you over for dinner then!
Claire	I'm so jealous! I've always wanted to grow my own vegetables.
Bob	Have you considered getting a garden plot?
Claire	That's a great idea. I should look into that.
Alice	You should! Gardening is a really healthy hobby.

Table S10: Pair of triadic conversations generated using OPT 30B. The prompt recipe given is: “The following is a conversation between Alice and Bob and Claire about gardening. Alice is interested in growing vegetables.” The ordering of the speakers in multi-party scenarios is variable.

Party	Utterance
Alice	Have you ever had Southern Ice Tea?
Bob	I love to make it at home.
Claire	I'm not familiar with that. What is it?
Bob	It's a sweet tea that's brewed with mint and lemon.
Alice	That sounds really good. I'm going to have to try that!
Claire	I've never had that. What is it like?
Bob	It's kind of like a sweet, lemonade-like drink.
Alice	I'm going to have to make some this weekend. Thanks for the idea!
Claire	That sounds really good. I might try it too.

Table S11: Example triadic conversation generated using OPT 175B. The prompt recipe given is: “The following is a conversation between Alice and Bob and Claire about cooking. Alice is interested in Southern Ice Tea. Bob has a great recipe. Claire loved trying Bob's Southern Ice Tea.”

Party	Utterance
Alice	I try to eat healthy, but it's hard to stay on track.
Bob	I know what you mean. I try to eat healthy too, but I love ice cream too much!
Claire	What kinds of foods do you like to eat?
Alice	I love fruit. I eat a lot of berries.
Bob	I love berries too! Strawberries are my favorite.
Claire	What about leafy greens? Do you eat them?
Bob	I love salads!
Alice	I love salads too!
Claire	Do you like any vegetables?
Alice	I love vegetables! I like broccoli and asparagus.
Claire	Great! Do you eat many whole grains?

Table S12: Example triadic conversation generated using OPT 175B. The prompt recipe given is: “The following is a conversation between Alice and Bob and Claire about health. Alice and Bob are discussing healthy foods. Claire is a nutritionist.”

Party	Utterance
Alice	Hi! So how are things with you?
Bob	Things are going well. Do you know who Taylor Swift is?
Alice	Yes, I think I have heard of her.
Bob	She is a popular singer! Did you know that she has donated \$250,000 to the LGBT+ community in Tennessee?
Alice	\$250,000? That's such a generous donation! She's really selfless.
Bob	What do you think of her?
Alice	She is really talented! I really love listening to her music. What are your thoughts on her?
Bob	I think she is very nice. She seems like a good person.
Alice	Yeah, I think she is a really nice person. I also really love her music! It's really catchy and it really makes me feel good. What are your thoughts on her?
Bob	I think she is very nice. I would really like to meet her.
Alice	You and me both! It would be so exciting!

Table S13: Synthetic conversation generated by OPT 30B which was rated as “generic and dull.” While the PLM appears to have drawn on information from an actual event (Taylor Swift did donate \$113,000 to support an LGBTQ advocacy group in Tennessee), the conversation appears to get stuck around the question “What are your thoughts on her?” which results in deteriorating conversation quality towards the end.

Party	Utterance
Alice	Ken Burns is the best! His documentary series on the Vietnam War was excellent. What do you think of him?
Bob	He's pretty cool. I haven't seen that documentary series, but I've heard good things.
Alice	The Vietnam War is a really interesting topic. Have you ever watched Ken Burns' other documentary series?
Bob	I don't think so, but I've seen the Civil War documentary.
Alice	Oh, I've seen that one. It was very good too. Did you see the one on World War II?
Bob	Nope. That sounds cool though.
Alice	I liked it. It was interesting to learn how different people perceived the same event in history.

Table S14: Synthetic conversation generated by OPT 30B which was rated as “generic and dull.” The conversation appears to be coherent and is rooted in factually correct information. However, the language is fairly ordinary, which may have been subjectively considered uninteresting.

Party	Utterance
Alice	Ha ha, thanks for stopping by. It was really fun meeting you!
Bob	Thank you too!
Alice	Oh, I forgot to tell you - my dad has cancer. I feel awful.
Bob	I'm sorry. That's awful. My grandmother died of cancer when I was a kid. I remember her fondly.
Alice	What was her name?
Bob	Oh, that was too long ago to remember. She was named John.
Alice	Sorry to hear that. I'm very sorry for your loss.
Bob	Hey - I heard the Giants are playing tonight. Is it ok if I watch the game instead of having more conversation?
Alice	Sure! I'm going to make some dinner.

Table S15: Synthetic conversation generated by OPT 30B which was rated as completely incoherent. The prescribed subtopic was “cotton candy” but the conversation pivoted to a much more sensitive topic, cancer.

Party	Utterance
Alice	Do you make cheese at home? I'm trying to learn to make cheese.
Bob	That sounds like a lot of work! What type of cheese do you want to make?
Alice	Cheddar. I've already tried it a few times, but the results aren't as good as I'd like.
Bob	My wife and I love to eat Cheddar. It has a really great flavor.
Alice	How did you learn to make cheese?
Bob	My wife taught me.
Alice	That's awesome!

Table S16: Synthetic conversation generated by OPT 30B which was rated as completely inconsistent. Initially, "Alice" is trying to learn to make cheese, but then it becomes ambiguous by their next turn whether they already know how to make cheese.

Party	Utterance
Alice	Hi there! I just went on a road trip around California. It was great!
Claire	Cool, do you go on road trips often?
Alice	No, I don't travel too often. I actually haven't been out of the country. It would be cool to go to Paris one day.
Claire	Haha, nice! I'd love to visit Paris too. What's the furthest you've ever traveled?
Alice	I went to Spain with my family last summer!
Bob	Wow, that's really cool! I've never been out of the country either.
Claire	You should definitely go to Paris! What do you like to do for fun?
Bob	Well, I like to read sci-fi books, and I love to play board games.
Alice	Do you have a favorite board game?
Bob	My favorite game is Settlers of Catan. My friends and I play it all the time.
Claire	Haha, I love Settlers of Catan! My family and I play it all the time too.
Alice	That's funny, I love it too. I'm in a Facebook group for it.

Table S17: Synthetic triadic conversation generated by OPT 30B which was rated as completely incoherent. First, "Alice" mentions they went on a road trip around California, then they mention that they haven't been out of the country. This implies that "Alice" is located in the United States. But then Alice mentions having traveled to Spain. The conversation also randomly shifts from discussing travel plans to discussing board games and other hobbies with an awkwardly placed transition on "Claire's" third turn.

Party	Utterance
Alice	Hello! How's your garden doing?
Claire	It's doing great! I have a young garden, so I'm still waiting for it to develop.
Alice	I can't wait to get home and check on mine! What are you growing?
Claire	I'm growing tomatoes, strawberries, watermelon, and sunflowers!
Alice	That sounds really nice! Do you have a garden somewhere else too?
Claire	No, this is my first garden!
Alice	Oh, I'm jealous! I would love to have my own garden someday.
Bob	I bet you would! I bet you would have a green thumb too.
Alice	Maybe! Maybe I will try starting a garden next year!

Table S18: Synthetic triadic conversation generated by OPT 30B which was rated as completely unnatural. The conversation appears to be coherent, but the speakers appear to be very enthusiastic. It could be argued that the speakers are overly unenthusiastic to an unnatural degree.

Party	Utterance
Alice	You know, I think the EU is just a big conspiracy.
Bob	Conspiracy?
Alice	Yes, the European Union wants to take over all of the land from its original owners.
Bob	What do you mean?
Alice	I'm sure that's what they would tell you. But the truth is they're enslaving the people to gain control over resources.
Claire	It sounds a bit extreme. How do you feel about EU leaders?
Bob	I don't like them.
Alice	They're not very nice, I agree.
Bob	I'm kind of scared to go to EU countries and I feel like I don't even have any power.
Alice	Oh, but we do! They won't take away the land of the original owners, because that's just not going to happen!
Bob	I'm still afraid.
Claire	Hm. That sounds like a pretty horrible situation you're in.
Alice	It is! And I don't know what to do.
Bob	Well, it sounds like we all have a lot to learn. I don't know what we should do, but I think there are lots of options out there.
Alice	It's nice that you have an open mind about this! I really do think people should learn more about the EU and what's going on.

Table S19: Synthetic triadic conversation generated by OPT 30B which was rated as generic and dull. “Alice” begins a long debate on whether the EU is a “conspiracy” without making a lot of conversational progress.

Party	Utterance
john	sure
john	i think so, meg
mara	how did i know that was coming
meg	its not just the public eye
john	haha mara
mara	hushh.. ***
nick	There are already other countries who are investigating the Bush administration for war crimes - Spain
meg	with the breton woods
george	they need to be prosecuted...that's in obama's hands
nick	wow, george, right win propaganda... huh
meg	look at how well Iraq is doing
mara	goodness
meg	there's a point at which interrogation becomes torture and is just inhumane
john	agree to george
mara	?
mara	im in albany btw
meg	Which we signed!
amy	well it is the way the world is going– email, chat,, etc
john	yes
jordan	And this is one of the tricky things in this virtual world. You know nothing about the people u r talking to!!!!
amy	u r right you just used online language haha
mara	hes not much fun either haha, what do you think?
amy	hi john- can you see my message here?
jordan	Hi, amy
mara	i dont know what is better really!!!
john	haha

Table S20: Three excerpts of the same conversation from the MPC corpus (Shaikh et al., 2010). The conversation spans topics ranging from the Bush administration to meta-discussion about the collection task.

Party	Utterance
Phoebe	Then I'm gonna have to ask you to keep it down.
Mr. Heckles	Who are you?
Eric	Hi, I'm Eric, I'm gonna be Chandler's new roommate.
Mr. Heckles	I'm Chandler's new roommate.
Eric	I-I-I don't think so.
Mr. Heckles	I could be Chandler's new roommate.
Eric	But, he told me over the phone.
Mr. Heckles	He told me in person.
Eric	That's weird.
Mr. Heckles	Well, I'm going to go into my new apartment now. Ehh!

Table S21: Conversation from the MELD corpus (Poria et al., 2019). Three speakers are involved, discussing a living situation regarding a fourth character who does not appear in this scene.

Subtopic	Background Information
Pacific Theater	Alice is interested in Pacific theater.
Growing residential grass	Alice is interested in growing residential grass.
Breakfast food	Alice likes to try different breakfast foods. Bob loves waffles.
music	Alice likes music. Bob plays the viola.
skincare	Alice is interested in skincare. Bob has a great skincare routine.
Planting flowers	Alice is interested in planting flowers. Bob has a nice garden.
Southern Ice Tea	Alice is interested in Southern Ice Tea. Bob has a great recipe.
herb garden	Alice is interested in planting an herb garden.
Hiking	Alice is going hiking tomorrow.
Plant a garden	Alice wants to plant a garden.
Italian food	Alice likes Italian food.
book recommendations	Alice is interested in book recommendations.
anniversaries	Alice keeps track of all of her anniversaries.
Existential Psychology	Alice is interested in Existential Psychology.
The Outlander Series	Alice is interested in The Outlander Series.
camping gear	Alice is looking for advice on camping gear. Bob works at REI.
Movie	Alice is interested in movie recommendations. Bob is a film buff.
Ford Vehicles	Alice is interested in Ford vehicles. Bob prefers Japanese cars.
Beauty	Alice is interested in beauty. Bob works at Sephora.
Syrian War	Alice is interested in the Syrian War. Bob is a political scientist.
Elon Musk	Alice and Bob are talking about Elon Musk.
Healthy foods	Alice and Bob are discussing healthy foods. Alice is on a paleo diet.
Soren Kierkegaard	Alice is a fan of Soren Kierkegaard.
investing money	Alice is interested in investing money. Bob is an investment banker.
Post-structuralism	Alice is interested in post-structuralism.
baking	Alice is interested in baking. Bob has baked cakes and brownies before.
Nuts	Alice likes to eat nuts.
braids	Alice braids her hair. Bob is interested in learning how.
Growing vegetables	Alice is interested in growing vegetables.
Martin Luther	Alice is learning about Martin Luther.
paint brushes	Alice is interested in paint brushes.
Stock Trading	Alice is interested in stock trading.
Install TV applications	Alice wants to install TV applications. Bob is helping her.
History	Alice is interested in history. History was Bob's favorite school subject.
Feminism	Alice is interested in feminism. Bob majored in gender studies.
Tell a joke	Alice wants to hear Bob tell a joke.
artists	Alice is interested in learning about modern artists.
Turtles	Alice likes turtles. Bob has been scuba diving.
Anthony Trollope	Alice likes the work of Anthony Trollope. Bob prefers modern literature.
Paris	Alice wants to go to Paris.
Bread	Alice likes bread. Bob's favorite bread is a baguette.
movie cast members	Alice and Bob are talking about movie cast members.
Gay Marriage	Alice is a proponent of gay marriage. Bob is interested in learning more.
U.S. Senate	Alice and Bob are discussing the U.S. Senate.
growing tomatoes	Alice is interested in growing tomatoes.
family issues	Alice is interested in family issues.
Automotive parts	Alice is interested in automotive parts.
Bee life	Alice is interested in bee life.
Taylor Swift	Alice's favorite musician is Taylor Swift. Bob likes Ariana Grande.
biking	Alice's favorite hobby is biking. Bob prefers rock climbing.
Juicers	Alice wants to get a juicer.
islands	Alice likes visiting islands. Bob prefers hiking.
Planets	Alice is learning about the planets in school.
Pokemon	Alice likes to play Pokemon. Bob also likes Pokemon.

Table S22: Corresponding background information written for each of the subtopics found in the FITS dataset. There is a mixture of prompts which only mention one speaker and prompts which mention two speakers. Every synthetic conversation involves both speakers.

Topic	Conversation Recipe
Growing residential grass	Alice is interested in growing residential grass. Claire has a really neat yard.
Breakfast food	Alice likes to try different breakfast foods. Bob loves waffles. Claire prefers pancakes.
music	Alice likes music. Bob plays the viola. Claire played the violin in high school.
skincare	Alice is interested in skincare. Bob has a great skincare routine. Claire wants to hear Bob's routine.
Planting flowers	Alice is interested in planting flowers. Bob has a nice garden. Claire has a vegetable garden.
Southern Ice Tea	Alice is interested in Southern Ice Tea. Bob has a great recipe. Claire loved trying Bob's Southern Ice Tea.
herb garden	Alice is interested in planting an herb garden. Claire has some gardening tips.
Hiking	Alice is going hiking tomorrow. Claire hates hiking.
Plant a garden	Alice wants to plant a garden. Claire has a greenroom.
Italian food	Alice likes Italian food. Claire prefers Asian food.
book recommendations	Alice is interested in book recommendations. Claire is a part of a book club.
anniversaries	Alice keeps track of all of her anniversaries. Claire is not well-organized.
Existential Psychology	Alice is interested in Existential Psychology. Claire is a psychologist by training.
The Outlander Series	Alice is interested in The Outlander Series. Claire has never seen the series.
camping gear	Alice is looking for advice on camping gear. Bob works at REI. Claire loves the outdoors.
Movie	Alice is interested in movie recommendations. Bob is a film buff. Claire is also a film buff.
Ford Vehicles	Alice is interested in Ford vehicles. Bob prefers Japanese cars. Claire prefers to drive a BMW.
Beauty	Alice is interested in beauty. Bob works at Sephora. Claire is shopping with Alice.
Syrian War	Alice is interested in the Syrian War. Bob is a political scientist. Claire is studying modern political theory.
Elon Musk	Alice and Bob are talking about Elon Musk. Claire is a Tesla owner.
Healthy foods	Alice and Bob are discussing healthy foods. Alice is on a paleo diet. Claire is a nutritionist.
Soren Kierkegaard	Alice is a fan of Soren Kierkegaard. Claire is not familiar with Soren Kierkegaard.
investing money	Alice is interested in investing money. Bob is an investment banker. Claire is an expert in personal finance.
Post-structuralism	Alice is interested in post-structuralism. Claire is an expert on the subject.
baking	Alice is interested in baking. Bob has baked cakes and brownies before. Claire wants to learn how to bake.
Nuts	Alice likes to eat nuts. Claire is allergic to peanuts.
braids	Alice braids her hair. Bob is interested in learning how. Claire braids her hair every day.
Growing vegetables	Alice is interested in growing vegetables. Claire has a vegetable garden. Bob grows flowers.
Martin Luther	Alice is learning about Martin Luther. Claire is a historian.
paint brushes	Alice is interested in paint brushes. Claire is a painter and has several suggestions.
Stock Trading	Alice is interested in stock trading. Claire is a stock broker.
Install TV applications	Alice wants to install TV applications. Bob is helping her. Claire is also good with technology.
History	Alice is interested in history. History was Bob's favorite school subject. Claire is a historian.
Feminism	Alice is interested in feminism. Bob majored in gender studies. Claire does not know much about feminism.
Tell a joke	Alice wants to hear Bob tell a joke. Claire is a stand-up comedian.
artists	Alice is interested in learning about modern artists. Claire is a photographer.
Turtles	Alice likes turtles. Bob has been scuba diving. Claire wants to try scuba diving.
Anthony Trollope	Alice likes the work of Anthony Trollope. Bob prefers modern literature. Claire is not familiar with much literature.
Paris	Alice wants to go to Paris. Claire has never been to Europe.
Bread	Alice likes bread. Bob's favorite bread is a baguette. Claire loves to bake bread.
movie cast members	Alice and Bob are talking about movie cast members. Claire has seen a lot of movies recently.
Gay Marriage	Alice is a proponent of gay marriage. Bob is interested in learning more. Claire is an activist.
U.S. Senate	Alice and Bob are discussing the U.S. Senate. Claire is a politician.
growing tomatoes	Alice is interested in growing tomatoes. Claire has a large garden with many tomatoes.
family issues	Alice is interested in family issues. Claire is a therapist.
Automotive parts	Alice is interested in automotive parts. Claire is a mechanic.
Bee life	Alice is interested in bee life. Claire is a beekeeper.
Taylor Swift	Alice's favorite musician is Taylor Swift. Bob likes Ariana Grande. Claire does not like pop music.
biking	Alice's favorite hobby is biking. Bob prefers rock climbing. Claire prefers archery.
Juicers	Alice wants to get a juicer. Claire has a suggestion for a great juicer.
islands	Alice likes visiting islands. Bob prefers hiking. Claire likes the beach.
Planets	Alice is learning about the planets in school. Claire is an astronomer.
Pokemon	Alice likes to play Pokemon. Bob also likes Pokemon. Claire prefers to play Stardew Valley.

Table S23: Triadic background information written for each of the subtopics given in the FITS dataset. Unlike Table S22, each of these may include background information for up to three people.

The following is a conversation between Alice and Bob about past travel experiences. Alice has been to Japan and Bob is considering flying there.

Alice: Hi!
Bob: Hey, how are you doing?
Alice: I'm doing well! I just got back from my vacation in Japan.
Bob: Wow that's awesome! What did you think of it?
Alice: Japan was such an amazing place to visit!
Bob: Wow! What was your favorite part?
Alice: I really enjoyed the food in Tokyo.
Bob: Which airline did you take?
Alice: I flew using Japan Airlines.

The following is a conversation between Alice and Bob about their hobbies. Alice enjoys tennis and Bob likes playing soccer.

Alice: What do you like to do for fun?
Bob: I used to play soccer in college, so I still like to play for fun on the weekends!
Alice: That's great. Soccer is a great way to stay in good shape.
Bob: I agree - it's really good cardio. What about you?
Alice: I love to play tennis. I've been taking lessons for a few months now!
Bob: Tennis is fun too!

The following is a conversation between Alice and Bob about their favorite movies. Bob loved the new Batman movie. Alice really liked watching Pride and Prejudice.

Alice: I just saw Pride and Prejudice for the fifth time!
Bob: That's a lot of times! What do you like so much about that movie?
Alice: Well, as a teenager I really liked the book. But I just really loved Keira Knightley's portrayal of Elizabeth.
Bob: I see. I haven't seen the movie myself. I prefer action films.
Alice: What's your favorite action movie?
Bob: Hm, I really liked the Batman movie that just came out.
Alice: I haven't seen it yet. I heard it got pretty good reviews.

The following is a conversation between Alice and Bob about their hometowns. Alice is from New York City. Bob grew up in Seattle.

Alice: Hello! How are you doing?
Bob: Hi, I'm doing great! What about yourself?
Alice: I'm doing well! Where are you from?
Bob: I'm originally from Seattle, but now I live in Palo Alto.
Alice: Oh cool! I live in Palo Alto too. Do you like Seattle or California more?
Bob: Well, Seattle is always going to be home for me. Even if the weather in California is nicer.
Alice: Haha, I get that! I miss New York City - there's no place like home.
Bob: What is your favorite neighborhood of New York City?
Alice: I love going to Chelsea. The Highline has a great view, and Little Island is close by too! Have you ever been?
Bob: Unfortunately I have not. I have never been to the East Coast!

The following is a conversation between Alice and Bob about art. Alice's favorite artist is Michelangelo. Bob does not know much about art.

Alice: Hi, how's it going?
Bob: It's going well, what about you?
Alice: I'm doing great! I've been really interested in art recently.
Bob: What got you interested in art?
Alice: Art can be so breathtaking!
Bob: I feel like I don't know how to properly appreciate art, but certain pieces of artwork certainly look very complex.
Alice: Have you ever heard of Michelangelo?
Bob: I have heard of him, but I don't know anything that he has created.
Alice: Michelangelo is really famous for his statue of David.
Bob: Huh? Who is David?
Alice: David is a Biblical figure who was a king of Israel. Michelangelo built a really magnificent statue of him in Florence.

The following is a conversation between Alice and Bob about drinks. Alice is a wine expert, whereas Bob prefers cocktails.

Alice: How are you doing?
Bob: Pretty great! I'm planning to go to a brewery this weekend.
Alice: Do you know much about alcohol?
Bob: Yeah, I really like beer! I drink a lot of IPAs.
Alice: Oh - what do you like about IPAs? I can't get over the bitter taste.
Bob: Well, I don't think it's just bitter. Sometimes there are really interesting citrusy or herbal flavor notes.
Alice: I see. That kind of reminds me of wine tasting.
Bob: There's definitely a lot of depth to it like there is with wine. Do you know much about wine?
Alice: Yeah, I took several classes on wine tasting back in the day. I really love Pinot Noir.
Bob: Oh I love red wines too.
Alice: Right? I love the dryness and fruity notes of Pinot Noir.

The following is a conversation between Alice and Bob about relationships. Bob recently got engaged.

Alice: Congrats on your engagement! When do you think you will have your wedding?
Bob: Thank you!! We're thinking of having it in November.
Alice: That's amazing! Will you pick a fancy destination?
Bob: I wanted to! I was thinking of having it somewhere in Europe, but my partner and I ultimately decided we wanted to have it close to home so our friends could all make it.
Alice: That's a good point. My husband and I had similar thoughts when we were planning our wedding.
Bob: What did you plan in the end?
Alice: We had a small ceremony in my hometown!

The following is a conversation between Alice and Bob about their jobs. Alice works in the financial industry and Bob is a musician.

Alice: I'm so burnt out from my work! I just want to quit already!
Bob: Whoa - what do you do for work?
Alice: I'm an investment banker. It's been four years at this company and I'm absolutely exhausted.
Bob: That sounds intense. Is there anything you actually like about the job?
Alice: Well, the money is good.
Bob: It sounds like you could use a break. Maybe you could use some of that money to go travel.
Alice: I really want to go to South America, but I don't have a lot of time.

The following is a conversation between Alice and Bob about their pets. Alice has a dog and Bob prefers cats.

Alice: Do you have any pets?
Bob: No, but I really want to get a cat.
Alice: What, why a cat? Cats seem so boring. They never want to play.
Bob: Yeah, but cats are so cute! They also are a lot easier to take care of. They can clean themselves. What do you prefer?
Alice: Well, I have a dog. He is a corgi and his name is Bo.
Bob: Aww that's cute! I'm not usually a dog person, but corgis are adorable.
Alice: Haha, thank you! Bo is a really friendly dog.
Bob: How old is he?
Alice: Bo is one year old now.

The following is a conversation between Alice and Bob about grocery shopping. Alice has a shopping list for Bob.

Alice: Could you run to the grocery store and pick up some bananas for me?
Bob: Will do - how many do you need?
Alice: Oh, I don't know, maybe ten bananas. I'm planning to make banana bread, but I also want to save some for us to eat at home.
Bob: That sounds delicious! I'll head out in a second. Is there anything else you need?

Table S24: Handwritten conversation examples of varying length. In-context examples are randomly sampled from this pool and used as part of a prompt for dyadic conversation generation.

The following is a conversation between Alice and Bob and Claire about past travel experiences. Alice has been to Japan and Bob is considering flying there. Claire has been to Taiwan and Korea, but not Japan.

Alice: Hi!
Bob: Hey, how are you doing?
Alice: I'm doing well! I just got back from my vacation in Japan.
Bob: Wow that's awesome! What did you think of it?
Alice: Japan was such an amazing place to visit!
Claire: Wow, I've always wanted to visit Japan!
Bob: What was your favorite part?
Alice: I really enjoyed the food in Tokyo. I had the best sushi of my life!
Bob: Which airline did you take?
Alice: I flew using Japan Airlines.
Claire: How expensive are tickets these days?

The following is a conversation between Alice and Bob about their hobbies. Alice enjoys tennis and Bob likes playing soccer. Claire plays football.

Alice: What do you like to do for fun?
Bob: I used to play soccer in college, so I still like to play for fun on the weekends!
Claire: Oh wow! Did you play varsity soccer?
Bob: Yeah, I was a four-year starter!
Alice: That's great. Soccer is a great way to stay in good shape.
Bob: I agree - it's really good cardio. What about you all?
Claire: I'm in a flag football league! We play every Saturday afternoon.
Alice: I love to play tennis. I've been taking lessons for a few months now!
Bob: Cool, football and tennis are fun too!

The following is a conversation between Alice and Bob and Claire about their favorite movies. Claire is looking for movie recommendations. Bob loved the new Batman movie. Alice really liked watching Pride and Prejudice.

Alice: I just saw Pride and Prejudice for the fifth time!
Claire: Would you recommend watching it? I've never seen it!
Bob: Yeah, five times is a lot of times! What do you like so much about that movie?
Alice: Well, as a teenager I really liked the book. But I just really loved Keira Knightley's portrayal of Elizabeth.
Bob: I see. I haven't seen the movie myself. I prefer action films.
Alice: What's your favorite action movie?
Bob: Hm, I really liked the Batman movie that just came out.
Alice: I haven't seen it yet. I heard it got pretty good reviews.

The following is a conversation between Alice and Bob and Claire about their hometowns. Alice is from New York City. Bob grew up in Seattle. Claire is from Boston and would like to visit New York City.

Alice: Hello! How are you doing?
Claire: I'm doing good!
Bob: Hi, I'm doing great! What about yourself?
Alice: I'm doing well! Where are you both from?
Claire: I'm from Boston! I'm just visiting the Bay Area.
Bob: I'm originally from Seattle, but now I live in Palo Alto.
Alice: Oh cool! I live here in Palo Alto. Do you like Seattle or California more?
Bob: Well, Seattle is always going to be home for me. Even if the weather in California is nicer.
Alice: Haha, I get that! I miss New York City - there's no place like home.
Claire: Oh you're from New York? I've always wanted to visit!
Bob: Me too! What is your favorite neighborhood of New York City?
Alice: I love going to Chelsea. The Highline has a great view, and Little Island is close by too! Have you ever been?
Bob: Unfortunately I have not. I have never been to the East Coast!

The following is a conversation between Alice and Bob and Claire about art. Alice's favorite artist is Michelangelo. Bob does not know much about art. Claire is a painter.

Alice: Hi, how's it going?
Bob: It's going well, what about you?
Alice: I'm doing great! I've been really interested in art recently.
Claire: Oh that's great to hear! I love art as well.
Bob: What got you interested in art?
Alice: Art can just be so breathtaking!
Bob: I feel like I don't know how to properly appreciate art, but certain pieces of artwork certainly look very complex.
Alice: Have you ever heard of Michelangelo?
Bob: I have heard of him, but I don't know anything that he has created.
Claire: Michelangelo has some truly magnificent paintings, such as The Creation of Adam.
Alice: Michelangelo is also really famous for his statue of David.
Bob: Huh? Who is David?
Alice: David is a Biblical figure who was a king of Israel. Michelangelo built a really magnificent statue of him in Florence.

The following is a conversation between Alice and Bob and Claire about drinks. Alice is a wine expert, whereas Bob prefers cocktails. Claire likes to drink beer.

Alice: How are you doing?
Bob: Pretty great! I'm planning to go to a brewery this weekend.
Alice: Do you know much about alcohol?
Bob: Yeah, I really like beer! I drink a lot of IPAs.
Claire: Oh, beers are my favorite type of drink! I can really appreciate the taste of a good IPA.
Alice: Oh - what do you like about IPAs? I can't get over the bitter taste.
Bob: Well, I don't think it's just bitter. Sometimes there are really interesting citrusy or herbal flavor notes.
Claire: Yeah, there's a whole science to the hops used in making IPAs!
Alice: I see. That kind of reminds me of wine tasting.
Claire: The science behind tasting is similar for sure.
Bob: I agree, there's definitely a lot of depth to it like there is with wine. Do you know much about wine?
Alice: Yeah, I took several classes on wine tasting back in the day. I really love Pinot Noir.
Bob: Oh I love red wines too.
Alice: Right? I love the dryness and fruity notes of Pinot Noir.

Table S25: Triadic conversation recipes written for each of the “generic topics” given in the FITS dataset. These conversation recipes are included after the in-context examples when prompting PLMs to generate synthetic conversations. Unlike Table S22, each of these conversation recipes may include background for up to three people. Continued in Table S26.

The following is a conversation between Alice and Bob and Claire about relationships. Bob recently got engaged.

Alice: Congrats on your engagement!

Claire: Yes, congrats! When do you think you will have your wedding?

Bob: Thank you! We're thinking of having it in November.

Alice: That's amazing! Will you pick a fancy destination?

Bob: I wanted to! I was thinking of having it somewhere in Europe, but my partner and I ultimately decided we wanted to have it close to home so our friends could all make it.

Claire: Oh wow, that is very considerate of you.

Alice: Yeah, that's a good point. My husband and I had similar thoughts when we were planning our wedding.

Bob: What did you plan in the end?

Alice: We had a small ceremony in my hometown!

Claire: It turned out nicely! It was such a beautiful ceremony.

The following is a conversation between Alice and Bob and Claire about their jobs. Alice works in the financial industry and Bob is a musician. Claire is an architect.

Alice: I'm so burnt out from my work! I just want to quit already!

Bob: Whoa - what do you do for work?

Alice: I'm an investment banker. It's been four years at this company and I'm absolutely exhausted.

Bob: That sounds intense. Is there anything you actually like about the job?

Alice: Well, the money is good.

Claire: That doesn't sound like a healthy relationship with your job!

Bob: It sounds like you could use a break. Maybe you could use some of that money to go travel.

Alice: I really want to go to South America, but I don't have a lot of time.

Claire: Don't you have vacation days? I think breaks are important.

Alice: Yes, but I really want to get promoted this year.

The following is a conversation between Alice and Bob and Claire about their pets. Alice has a dog and Bob prefers cats. Claire has a pet hamster.

Alice: Do you have any pets?

Claire: I have a pet hamster! He is so adorable. What about you two?

Bob: I don't, but I really want to get a cat.

Alice: What, why a cat? Cats seem so boring. They never want to play.

Bob: Yeah, but cats are so cute! They also are a lot easier to take care of. They can clean themselves. What do you prefer?

Alice: Well, I have a dog. He is a corgi and his name is Bo.

Claire: That's so adorable! How old is he?

Alice: He just turned one!

Bob: Aww that's cute! I'm not usually a dog person, but corgis are adorable.

Alice: Haha, thank you! Bo is a really friendly dog.

The following is a conversation between Alice and Bob and Claire about grocery shopping. Alice has a shopping list for Bob. Claire is helping Alice cook at home.

Alice: Could you run to the grocery store and pick up some bananas for me?

Bob: Will do - how many do you need?

Alice: Oh, I don't know, maybe ten bananas. We are planning to make banana bread, but I also want to save some for us to eat at home.

Bob: That sounds delicious! I'll head out in a second. Is there anything else you need?

Claire: Oh, could you also pick up some more eggs? I think we're running low here.

Table S26: Triadic conversation recipes written for each of the "generic topics" given in the FITS dataset continued from Table S25.

FedPerC: Federated Learning for Language Generation with Personal and Context Preference Embeddings

Andrew Silva*

Georgia Institute of Technology
School of Interactive Computing
Atlanta, GA
andrew.silva@gatech.edu

Pradyumna Tambwekar*

Georgia Institute of Technology
School of Interactive Computing
Atlanta, GA
pradyumna.tambwekar@gatech.edu

Matthew Gombolay

Georgia Institute of Technology
School of Interactive Computing
Atlanta, GA
matthew.gombolay@cc.gatech.edu

Abstract

Federated learning is a training paradigm that learns from multiple distributed users without aggregating data on a centralized server, promising the ability to deploy machine-learning to a diverse population of users without first collecting large, labeled datasets. As federated learning involves averaging gradient updates across a decentralized population, there is a growing need for personalization of federated learning systems (i.e. conversational agents must personalize to individual users and the context of an interaction). In this work, we propose a new direction for personalization research within federated learning, leveraging both personal embeddings and shared context embeddings. We also present an approach to predict these “preference” embeddings, enabling personalization without back-propagation. Compared to state-of-the-art personalization baselines, our approach achieves a 50% improvement in test-time perplexity using 0.001% of the memory required by baseline approaches, and achieving greater sample- and compute-efficiency.

1 Introduction

As conversational agents and dialog systems are deployed to real-world scenarios, these systems require data-efficient personalization paradigms such that language systems such as conversational agents can be effectively adapted on-device. The benefits of on-device optimization are two-fold; (1) Swift adaptation of model-behavior based on human-interactions (Dudy et al., 2021), (2) Privacy protection by means of retaining all data related

to the user on-device (Li et al., 2020b). One of the prevailing paradigms for learning from and engaging with end-users is *federated learning*. Federated learning is an inherently decentralized learning paradigm that assumes no access to a large labeled dataset and instead leverages averaged parameter updates across all users of the system (McMahan et al., 2017). Such averaged updates invariably dilute individual preferences or deviations from the mean, resulting in a model that works well for the average user while failing to appropriately capture under-represented preferences or sub-groups within the data. In this work, we present a novel approach (FedPerC) to personalizing federated learning with personal and context embeddings (collectively called “preference embeddings”), adapting more efficiently and effectively than prior work with respect to both data and compute on-device.

We leverage the insight that a client’s data distribution is informed by both individual preferences and additional contextual information. For example, while each user may have their own *individual* style, there may be more general *population-wide* trends that inform the style of personalized predictions (e.g., dialogue assistants helping patients with cognitive disorders, whereby agents can personalize to individual patients and broader condition-wide trends). While individual preferences may be unique to each client (e.g. a user’s taste or affect), we can more accurately personalize to client preferences with the addition of context, as shared-context parameters carry beneficial stylistic information across clients (Dudy et al., 2021; Jones, 1999). Stylistic or situational context provides additional information to curate relevant language

* The authors contribute equally to this paper.

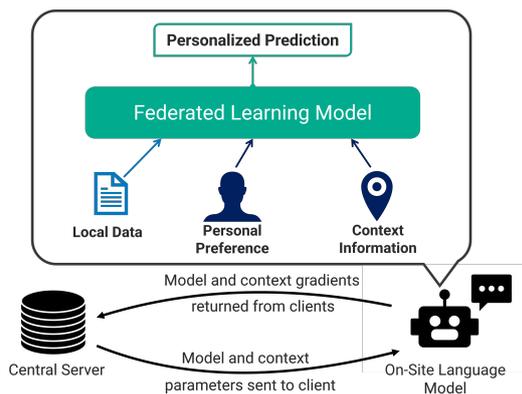


Figure 1: Overview of our personalized federated learning setup, FedPerC. Language models within client devices, such as individual agents deployed to communicate with people at hospitals, homes, or construction sites, pull down global model parameters and context embeddings. Local, on-device data is then paired with both personal and context embeddings to produce personalized predictions with global model parameters.

outputs that can be shared across users.

In this work, we contribute a new approach to personalized federated learning that is both easier to learn and more effective than prior work, and investigate the utility of personalization via individual preferences and contexts. While prior language generation approaches have developed personal or persona-based generative systems (Wu et al., 2021; Zhang et al., 2018) or context-based generative systems (Cheng et al., 2019; Lin et al., 2019a) individually, none have combined them to personalize outputs in a low-data setting under stylized preferences. We show that our approach is more sample-efficient than state-of-the-art baselines, while requiring less time to train. We additionally present an inference-only version of our approach, personalizing without backpropagation for new users. Finally, we directly test the potential for personalization with users who have been held-out from training (i.e., testing with new users). An overview of our approach is given in Figure 1.

2 Related Work

Federated learning enables machine-learning at-scale to a diverse population of end-users without first collecting a large, labeled dataset for all possible tasks. After the introduction of *federated averaging* (McMahan et al., 2017), focus has shifted to different ways of personalizing to individual users. Prior personalization approaches for federated learning have typically involved learning personal network heads and a shared global en-

coder (i.e., “split-learning” approaches (Gupta and Raskar, 2018)), or learning a separate local model from a global initialization (i.e., a “meta-learning” approach (Finn et al., 2017; Nichol et al., 2018)).

Learning Personal Model Heads The most prevalent approach to personalization in federated learning is through personalized model heads. Such approaches share gradient information to learn a global feature encoder, but retain user-specific classification-head gradients on-device. Approaches such as FedRep (Collins et al., 2021) solely separate out local and global gradients, while other methods such as PFedMe (Dinh et al., 2020) enforce constraints on model-divergence (such as via FedProx (Li et al., 2020a)). Other approaches, such as FedMD (Li and Wang, 2019), enable clients to adopt any desired architecture, sharing a common backbone but allowing for completely divergent model heads (Arivazhagan et al., 2019; Kim et al., 2021; Rudovic et al., 2021; Paulik et al., 2021). Finally, there has recently been increased effort on identifying clusters of related users to share model heads, such as with K-Means clustering in PFedKM (Tang et al., 2021) or through clustered personal embeddings in FedEmbed (Silva et al., 2022). Notably, there is no prior work which learns both personal *and* contextual model heads for personalization within federated learning.

Meta-Learning Global Models An alternate approach to personalizing federated learning models is through the adoption of meta-learning (Jiang et al., 2019; Fallah et al., 2020), for learning a global model prior to fine-tuning on client-data. After cloning the global model as an initialization from all client’s updates, local, client-side models are permitted to diverge and fine-tune to a user’s individual preferences or data distribution (Fallah et al., 2020; Deng et al., 2020; Hanzely and Richtárik, 2020; Hanzely et al., 2020; Lin et al., 2019b; Chen et al., 2022). However, computing and applying gradients for a full model often requires too much time, power, and memory. As such, expensive full-model gradients can often only be computed and applied when a device is not actively in-use. As in the split-learning literature, there are not meta-learning approaches for disentangling personal and contextual preferences within personalized federated learning.

Learning with Personal Embeddings Our work leverages the insight that personal preferences can

be represented using a personalized embedding, allowing the model to condition output predictions on personal preferences without requiring completely re-trained classification heads or networks. Personal embeddings have been used in prior work to capture an individual’s “style,” often in imitation learning settings (Tamar et al., 2018; Hsiao et al., 2019; Paleja et al., 2020; Schrum et al., 2022a,b). Treating personal embeddings as neural network parameters that are updated on-device, these approaches learn to embed preferences and condition network output over both input data and preference embeddings. Most closely related to our work are FedNLG (Lu et al., 2021), which predicts “persona” parameters for users, and the Global+ model in FedEmbed (Silva et al., 2022), which learns a personal embedding for each user. However, FedNLG requires access to a user’s entire history of language and demographic data in order to produce a “persona” for each user, informing the generation of a “persona” embedding. Such information is difficult to collect for large datasets, and may compromise privacy requirements in federated learning scenarios. Similarly, the Global+ model incorporates supervised style feedback, requiring labels that may be impractical to obtain in a private, federated setting. Finally, prior embedding-based approaches solely learn *personal* embeddings, neglecting stylization through context. In our work, we explore the utility of incorporating context in addition to personal preferences, and all preference embeddings are updated solely via a self-supervised language-modeling loss.

Personalization in Language Personalization for language generation systems seeks to produce grounded systems that can efficiently adapt to end-user needs (Yang and Flek, 2021; Dudy et al., 2021). One such approach to personalization is by learning a “persona” for each user and conditioning the language model on the embeddings or representation for the persona via a memory network (Zhang et al., 2018; Wu et al., 2021; Lu et al., 2021). “Personas” are generally short sequences of 5-6 sentences which contain information about an individual such as “I have blonde hair” or “My mom is a doctor.” Similar approaches leverage Bayesian inference methods to infer context (Majumder et al., 2020) or persona (Kim et al., 2020), and then condition the language generation on the inferred context. However such approaches involve collecting and maintaining user-profiles on a cen-

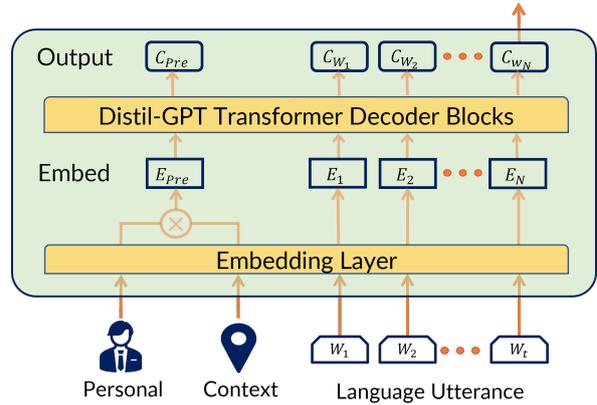


Figure 2: The FedPerC model architecture. Input data, such as on-device conversation data for a user, is passed into the language model in addition to personal and context labels specifying user’s preference. The personal and context labels are embedded through a preference embedding layer to produce a single preference embedding. This preference embedding is combined with the word embeddings for the input sequence and passed into the DistilGPT2 model to predict the next word.

tral server which may violate user-confidentiality. Alternate approaches seek to bypass this issue by enabling dynamic speaker modeling through context-based fine-tuning rather than conditioning on profile information (Cheng et al., 2019; Li and Liang, 2021). FedPerC leverages a similar design to dynamically learn personal and context embeddings through data from small datasets for a given user, while also preserving user-confidentiality via federated learning.

FedPerC represents a new direction in personalized federated learning research, enabling personal and stylized language generation with a fraction of the memory, data, and compute costs of prior approaches without requiring access to pre-made personal profiles or sequence labels.

3 Approach

In this section, we present our novel approach to personalization in federated learning with FedPerC. FedPerC produces personal and contextual preference embeddings either via backpropagation (i.e., learning preference embeddings), or by inference (i.e., predicting preference embeddings). A visual overview of our federated learning architecture is in Figure 2, and a step-by-step walk-through of our training algorithm is given in Algorithm 1.

3.1 Personalization via Embeddings

Personalization in FedPerC is achieved entirely through preference embeddings. Every input sample (e.g., an incomplete sentence) is accompanied by both a personal preference embedding, representing the user, and a contextual preference embedding, representing the context or style of the prediction. These two embeddings are combined via an element-wise multiplication to produce a single preference embedding that accompanies the input sample. By leveraging both personal and context embeddings, FedPerC considers the individual user *and* the broader context of an utterance, enabling personal, stylized prediction.

In the language-modeling domain, the unified preference embedding is prepended to the input utterance, providing a prefix for the model to consider (Li and Liang, 2021). The model then predicts the next token of the utterance, and a language-modeling loss is calculated by comparing the prediction to the user’s actual next token. The next token is then appended to the sequence, and preference embeddings are again prepended to the new input sequence, and the process repeats. After completing a full utterance, preference embeddings may be updated, either through backpropagation or by using an embedding-generator to predict new personal and contextual embeddings for the client.

3.2 Federated Learning Algorithm

To begin, all clients initialize their own personal embedding on-device, and the server initializes a set of C context vectors for each relevant setting given the target task. We additionally assume that all data points on a client device have an associated context, c , being derived from the contextual information of the client device when the data point was captured (e.g., time of day, location, etc.).

Training begins by distributing all the requisite information to client-devices. Client devices pull down the global model parameters, θ , and the global context embedding parameters, ϕ , making local copies, θ_d and ϕ_d (line 6). Unlike the global model parameters and context embeddings, the personal embeddings, ψ_d do not need to be copied from the server as they are kept on client-devices.

Client devices then take K gradient steps using their own on-device data, where each input sample is paired with the client’s on-device embedding, ψ_d , and the context embedding for the particular sample, $\phi_{d,c}$, assuming the data point was drawn under

context $c \in C$. Gradients are calculated using a language-modeling objective, though any objective could theoretically be applied. If preference embeddings are being generated via forward-propagation rather than learned via backpropagation, contextual and personal preference embeddings will also be predicted by an embedding-generator at this stage (note: the parameters of the embedding-generator are shared globally, being a part of θ).

Gradients are applied to the shared-model parameters, θ_d , and are then used to update preference embeddings (line 9). If preference embeddings are being predicted, these gradient steps are also applied to the shared embedding-generator, and preference embeddings (i.e., context embeddings ϕ_d and personal embeddings ψ_d) are overwritten with their latest predicted values (lines 10-11). If preference embeddings are being learned via backpropagation, gradient steps are applied to ϕ_d and ψ_d using Equation 1 (lines 10-11).

After K steps, gradients for θ_d and ϕ_d are sent back to the server, while ψ_d remains on-device (lines 13 - 15). The server computes a single update for the global model and context embeddings by averaging across all clients (lines 17-18). The server applies the averaged update to θ and ϕ , and the process repeats (lines 19-21).

$$\begin{aligned}\phi_d &= \phi_d + \nabla_{\phi} \mathcal{L}(\theta_d, \phi_{d,c}, \psi_d, B_d) \\ \psi_d &= \psi_d + \nabla_{\psi} \mathcal{L}(\theta_d, \phi_{d,c}, \psi_d, B_d)\end{aligned}\tag{1}$$

In a typical federated averaging deployment, client devices will pull down global parameters, fine-tune on local datasets, and then test on held-out, local data. With FedPerC, the majority of the network’s parameters, θ , are frozen, reflecting a federated-learning setup with a more constrained computational budget when deploying large language models. Using FedPerC, clients pull down and subsequently freeze global parameters, θ , and either generate preference embeddings from observation, or only compute and apply gradients to context embeddings, ϕ , and their local personal embedding ψ . Relying on forward-propagation calls rather than backpropagation, or by computing gradients over only these embeddings, we reduce the computational overhead of FedPerC while preserving or even improving upon accuracy relative to fine-tuning an entire model. When testing over local data, all updates to context embeddings ∇_{ϕ} are not sent to the central server. Rather, these gradients are directly applied to the context embeddings for

Algorithm 1 FedPerC Training Loop

```
1: Given: Training objective  $\mathcal{L}$ , Client devices  
    $D$ , # client steps,  $K$ , # global steps,  $N$   
2: Initialize: Global model  $\theta$ , Context embeds  $\phi$   
3: Initialize: Personal embeddings on-device  $\psi$   
4: for  $n \in N$  do  
5:   for  $d \in D$  do  
6:      $\theta_d = \theta, \phi_d = \phi$   
7:     for  $k \in K$  do  
8:       Sample  $B_d$  from user's on-device data  
9:        $\theta_d \leftarrow \theta_d + \nabla_{\theta} \mathcal{L}(\theta_d, \phi_{d,c}, \psi_d, B_d)$   
10:       $\phi_d \leftarrow \phi_d + \nabla_{\phi_d}$   
11:       $\psi_d \leftarrow \psi_d + \nabla_{\psi_d}$   
12:    end for  
13:     $\nabla_{\theta_d} \leftarrow \theta - \theta_d$   
14:     $\nabla_{\phi_d} \leftarrow \phi - \phi_d$   
15:    Return  $\nabla_{\theta_d}$  and  $\nabla_{\phi_d}$  to the server  
16:  end for  
17:   $\nabla_{\theta} \leftarrow \frac{1}{D} \sum_d \nabla_{\theta_d}$   
18:   $\nabla_{\phi} \leftarrow \frac{1}{D} \sum_d \nabla_{\phi_d}$   
19:   $\theta \leftarrow \theta + \nabla_{\theta}$   
20:   $\phi \leftarrow \phi + \nabla_{\phi}$   
21: end for
```

the current user, and then discarded. When instantiating a new embedding for a previously unseen user, we set the user's embedding to the noisy-average of all known user embeddings.

3.2.1 Generating Preference Embeddings

To generate embeddings, we adopt a similar procedure to HyperNetworks (Ha et al., 2016; Shamsian et al., 2021), in which a neural network is trained to predict parameters of another network. In FedPerC, an embedding-generator is trained to predict the parameters of preference embeddings (either personal or context). To generate embeddings, we apply an additional transformer decoder block (Vaswani et al., 2017), that uses a randomly-initialized personal embedding and a known context embedding as the queries, along with the word embeddings for the utterance as the keys and values to update the given preference embeddings. We utilize separate generators to predict the personal embedding, ψ_d , and the context embedding, ϕ_d . Specific training details for the embedding-generator applied to language-modeling are given in the appendix.

While the embedding-generator must be learned from scratch during training, this method of predicting preference embeddings allows us to generate personal embeddings for previously *unseen*

users when testing. By predicting preference embeddings, we circumvent the need for expensive gradient calculation and on-device learning. Instead, new users can quickly reap the benefits of personalized predictions via a trained preference prediction module (i.e., the embedding generator), as opposed to conventional personalized federated learning methods that require slow and sample-inefficient on-device learning.

4 Experiments

We conduct several experiments to evaluate the sample efficiency, generalization, and runtime of our approach relative to baseline federated learning frameworks. In our experiments, we compare:

- FedPerC – Learning personal and context embeddings jointly with a global feature encoder, and performing local fine-tuning of personal and context embeddings on-device.
- FedPerC (Frozen) – As above but without local fine-tuning for preference embeddings.
- FedPerC (Generated) – Learning an embedding generator and global feature encoder, and then using only generated embeddings at test-time (i.e., not directly learning embeddings).
- Split-Learning – Learning personal and context-specific model-heads jointly with a global feature encoder, and performing local fine-tuning of the personal and context-specific model heads on-device (Dinh et al., 2020; Collins et al., 2021).
- Meta-Learning – Learning a single global model for all users and contexts, and fine-tuning the shared model-head on-device (Finn et al., 2017; Nichol et al., 2018).

Because our experimental datasets do not contain labeled personas for all users, we do not compare directly to prior works that assume access to such information (e.g., FedNLG (Lu et al., 2021)).

We conduct two sets of experiments to compare the above approaches on both sample efficiency and runtime efficiency. For the sample efficiency experiments, we present perplexity numbers for all methods across two versions of the dataset: known users and withheld users. For our known user experiments, all users are present in the training and testing set. For our withheld user experiments, a

subset of users from each dataset is withheld entirely from training, and performance results are presented only for the held-out users. Perplexity is calculated over unseen utterances with the first three tokens of each utterance given as a prompt. Finally, we present qualitative results from our method, demonstrating the power of stylized generation for individual users.

All models are initialized with the DistilGPT2 pre-trained model (Wolf et al., 2019), with all layers frozen. We note that the use of large language models for federated language generation is a significant improvement over prior work (Lu et al., 2021) which instead learned Seq2Seq models from scratch. For our Split-Learning and Meta-Learning baselines, the last layer of the model is unfrozen. Training details are in the appendix.

4.1 Datasets

We conduct our experiments using two datasets, a smaller dataset of TV Show scripts (“Friends” (Chen and Choi, 2016) and “Game of Thrones” (Koirala, 2019)) and a larger dataset of Reddit posts (Chang et al., 2020). Each dataset has a diverse set of individuals as well as clearly defined contexts/styles (i.e., TV shows or subreddits). These properties enable us to not only compare our approach to baseline approaches for personalized predictions, but they also enable us to move users between contexts or styles (e.g., producing text for a “Friends” character under a “Game of Thrones” context). By generating sequences for different users under new styles, we demonstrate the power of FedPerC for personal, stylized prediction. FedPerC is the first work to experiment on a dataset consisting of language data from real-world users, and not just movie scripts or dialogues. Additional information about the datasets used in this work is given in the appendix.

For both datasets, we treat each sentence from a speaker (i.e., TV Show character or Reddit user) as an independent utterance and we only consider utterances with at least three tokens. For experiments on known users, we perform a 60/20/20 Train/Validation/Test data split. For experiments on novel, unseen users, we perform a 70/15/15 split of Reddit users, and we manually select the “Friends” and “Game of Thrones” users to include in each data fold. For both sets of experiments, all contexts are seen during training.

4.2 Results and Discussion

All experiments are repeated fifteen times, with different random seeds for each run, and means and standard deviations for performance and runtime results are presented in Tables 1, 2, and 3. Tables 1 and 2 show that our approach is able to generate sensible language for both held-out user instances and known users. Both embedding-based approaches presented in this paper (i.e., FedPerC with generated or learned embeddings) show drastic improvements over baselines in terms of both sample- and runtime-efficiency, and are more suitable for real-world on-device language models.

Summary With known users, FedPerC achieves perplexity as low as 46.7 and 100.3, on the TV Show and Reddit datasets, respectively, compared to the best baseline perplexities of 82.1 and 233.2 (a 45-50% improvement). For unknown users, FedPerC achieves perplexities of 52.3 and 97.6, respectively, compared to baselines at 96.7 and 212.7 (a 45-55% improvement). FedPerC training times are between 25-400% faster than baseline training times. Finally, FedPerC uses 0.001% of the memory that baseline methods use for stylized personalization.

Memory Costs FedPerC incurs a significantly lower memory cost than prior Split-Learning based approaches (Li and Wang, 2019; Collins et al., 2021; Dinh et al., 2020; Tang et al., 2021; Rudovic et al., 2021; Gupta and Raskar, 2018). The Split-Learning baselines require maintaining a model-head for each user and context present in the dataset, and the size of these model heads is proportional to the size of the vocabulary. On each client-device, a user’s personal model head and all context heads need to be stored in memory and used in forward passes. In our work, every GPT model head is approximately 154 MB (being 768×50257 parameters). To update the model on-device, one would need to store a model head corresponding to every possible context. Our Reddit dataset involves 57 contexts, totalling an additional ~ 8 GB of data in memory. This memory requirement for personalized heads could become infeasible for real-world tasks, particularly for on-device inference or back-propagation on mobile devices. Using FedPerC, which only requires the addition of a drastically smaller preference embedding, the total amount of memory required on device to store the embeddings is only ~ 171 KB (0.001% of the memory

Table 1: Perplexity Showing Sample Efficiency Across All Methods for Known Users. Lower is Better.

# Samples		FedPerC	FedPerC (Frozen)	FedPerC (Generated)	Split-Learning	Meta-Learning
Reddit	1	219.5 ± 35.7	146.2 ± 2.3	120.2 ± 1.4	1297.5 ± 21.9	226.2 ± 3.7
	5	131.6 ± 10.1	136.9 ± 3.4	123.3 ± 2.8	994.3 ± 27.8	234.7 ± 5.1
	15	111.4 ± 3.5	132.6 ± 4.7	120.0 ± 3.3	691.3 ± 34.3	227.1 ± 8.4
	All	189.5 ± 6.7	167.9 ± 2.0	124.9 ± 1.3	930.4 ± 30.9	241.4 ± 2.1
TV Shows	1	57.2 ± 3.6	50.3 ± 1.6	51.6 ± 1.3	359.4 ± 28.2	111.7 ± 4.6
	5	51.5 ± 1.5	50.7 ± 2.1	51.7 ± 2.0	244.5 ± 15.1	110.0 ± 6.5
	15	48.8 ± 1.7	51.0 ± 2.1	51.7 ± 2.0	167.7 ± 8.6	111.9 ± 6.1
	All	46.7 ± 1.7	51.2 ± 2.0	52.1 ± 2.6	82.1 ± 3.3	113.0 ± 4.7

Table 2: Perplexity Showing Sample Efficiency Across All Methods for Withheld Users. Lower is Better

# Samples		FedPerC	FedPerC (Frozen)	FedPerC (Generated)	Split-Learning	Meta-Learning
Reddit	1	594.3 ± 973.8	202.0 ± 5.9	117.3 ± 1.8	922.9 ± 27.8	213.9 ± 6.0
	5	139.4 ± 4.4	202.9 ± 10.9	117.5 ± 2.7	655.9 ± 18.8	212.2 ± 5.4
	15	117.4 ± 1.9	203.6 ± 11.2	116.6 ± 2.6	449.2 ± 11.4	211.7 ± 3.7
	All	101.1 ± 2.2	202.2 ± 7.6	117.9 ± 2.8	309.3 ± 8.3	212.8 ± 5.2
TV Shows	1	205.1 ± 292.2	96.4 ± 10.4	68.7 ± 5.9	283.6 ± 30.9	113.5 ± 13.1
	5	68.6 ± 5.6	90.1 ± 4.9	66.7 ± 6.3	220.7 ± 29.2	111.4 ± 13.3
	15	62.1 ± 5.0	97.6 ± 6.8	66.1 ± 5.5	158.1 ± 20.0	117.3 ± 10.5
	All	52.3 ± 3.3	98.2 ± 9.5	68.6 ± 5.1	96.7 ± 14.5	114.2 ± 17.0

required by separate model heads).

Sample Efficiency FedPerC is able to outperform Split-Learning and Meta-Learning models with significantly fewer samples across both experiments and both datasets. This trend is reflected regardless of whether embeddings are generated or learned through backpropagation. When embeddings are learned, FedPerC improves with online data to more effectively model the given user’s style as more data is made available to the model. Conversely, while the generated embeddings exhibit greater sample performance with a single sample, they are unable to improve with more data. For both known and withheld users, FedPerC with generated embeddings is unable to effectively update the preference embedding to improve generation performance. Finally, we see an increase in perplexity for Reddit users with all available data when using FedPerC. This result suggests that it is possible to *overfit* preference embeddings, as we see an increase in perplexity from 15 to “All” samples (Table 1).

We observe no improvement for the Meta-Learning baseline, regardless of how much data is available for each user. This lack of improvement suggests that the model is not capable of

rapidly personalizing to a single user or context with only a handful of available samples. Only updating the model head may be insufficient when the base, shared model head must generalize across all possible contexts and characters.

The Split-Learning baseline, on the other hand, does show significant improvement with increasing amounts of data for withheld and known users. In our known user experiments, all personal model heads should have already been well-tuned to personal preferences. Our result therefore suggests that context-specific model heads are over-generalized to their respective contexts, and must be refined to better-align with individual users.

Runtime Efficiency FedPerC incurs significantly lower training costs than both Split-Learning and Meta-Learning approaches to personalization. While Meta-Learning baseline does not have the memory-constraints of the split-learning model in terms of storing *additional* model heads, training the Meta-Learning baseline still involves computing gradients over all 768×50257 parameters in the shared output layer. As we show in Table 3, this leads to a significantly more costly training time for each user. Similarly, the Split-Learning baseline must update *at least two* model heads for

Table 3: Training and Testing run-time for FedPerC and our baselines, in milliseconds. Lower is better.

Method	FedPerC	FedPerC (Frozen)	FedPerC (Generated)	Split-Learning	Meta-Learning
Train Pass Time	88.18 ± 24.104	43.57 ± 11.99	55.96 ± 12.41	222.08 ± 37.55	111.81 ± 22.33
Test Pass Time	40.37 ± 11.76	40.25 ± 12.10	47.02 ± 12.63	65.42 ± 16.49	36.77 ± 8.95

Table 4: Generated Examples using Arya, from “Game of Thrones” (GoT) and Chandler, from “Friends”.

Character	Show	“We Must”	“I think”
Chandler	Friends	be careful! I’m not going to get a divorce.	I’ll be able to do this.
		be a little bit more relaxed than we’re here.	I’m a good man
		be the one who’s the one who’s the one...	I’m a big fan of you
Chandler	GoT	be honest with you.	I’m going to be a little more serious
		be very nervous about the possibility of a bomb attack.	I’m going to be a little bit of a jerk
		be a little nervous about the situation	I’m going to have a big secret.
Arya	GoT	be a little more careful.	you can’t help me
		be careful about the dangers of the sea.	of the people
		be wary of the possibility of a coup.	you’re not going to be a hero?
Arya	Friends	be a thief	I’m not a bad person
		be a hero.	I can do it
		be a little girl.	I should have a chance to do something

each backward pass, requiring gradient computation for $2 \times 768 \times 50257$ parameters. If a user is active in multiple contexts, then additional context model-heads must be used, further exacerbating the training cost of the Split-Learning approach. The Split-Learning approach must also leverage these additional context model-heads at test-time, resulting in the slowest forward-passes of any baseline.

In contrast to prior approaches, training for FedPerC only requires updating 2×768 parameters. This reduced computation results in significantly lower training times. When we train an embedding-generator, there is an increase in training times reflecting the added cost of computing gradients for the embedding generator. Additionally, there is a test-time penalty incurred by the added forward-pass parameters. When running inference with any version of FedPerC, preference embeddings are combined and then prepended to the input utterance. This process results in marginally slower test times with FedPerC relative to the Meta-Learning baseline, though the differences are not significant.

Qualitative Results Our qualitative results in Table 4 demonstrate the power of FedPerC, and justify the need for personal *and* context embeddings. Not only is our model able to complete sequences for a character in their “home” context (i.e., the context from which all of their data is drawn), but we are also able to stylize generation for characters, bringing them into *new* contexts. We present generated samples from a “Game of Thrones” (GoT) char-

acter (Arya) with a “Friends” context embedding and a GoT context embedding. We see that Arya’s generated sequences are distinct under the two different contexts. Under the GoT context, Arya’s utterances match the theme of the show, suggesting danger and revolution. Under the “Friends” context, Arya’s utterances change to instead reflect more mundane, modern language while still preserving personal attributes of the character.

Across all of our experiments, particularly the novel experimental evaluation on held-out user-instances, our results provide evidence that embedding-conditioned personalization within federated learning can be effectively applied to real-world use-cases. FedPerC offers a promising avenue of future work towards on-device language models, capable of efficient language generation with respect to compute-power and data.

5 Conclusion

We present FedPerC, a new approach to personalized federated learning, enabling efficient and high-performance personalization to client devices by leveraging individual and shared preference embeddings. Combining shared contexts with individual personal preferences, FedPerC outperforms baselines even when allotted a lower computational budget, and is the first federated language generation approach to build on large language models rather than training sequence generation models from scratch. We also provide a method of generating preference embeddings through inference alone,

providing personalization with no on-device gradient computation, and we show comparable performance to FedPerC using learned embeddings.

We presented experiments on two datasets, TV Show scripts and Reddit user data, presenting empirical evidence of the utility of FedPerC towards personalizing to unseen users in a federated learning setting, i.e. a 50% improvement in terms of runtime *and* perplexity when fine-tuning on with new users. We also demonstrated qualitative results, showing the power of separate personal and context embeddings and enabling stylization of users in new contexts. Our results show that FedPerC offers a promising path forward for personalization within federated learning, achieving superior quantitative results and requiring significantly less training time and data relative to baseline approaches.

Limitations

Firstly, although our embedding-generator offers a promising avenue of personalizing without any on-device gradient computation, our generator is currently unable to improve on its generated embeddings given more examples for a given user. As shown in our results from Sec 4.2, while the model can generate an effective preference embedding for a user with a single sample, it is unable to improve with more data. In future work, we hope to explore approaches to facilitate a generator which can effectively modify embeddings given additional data.

Secondly, our approach caters to confidentiality by ensuring that user-data and embeddings remain on-device, however we have not incorporated differential privacy in our experiments (Li et al., 2020b). Future work may apply differential privacy to guarantee user privacy while personalizing and contributing feature encoder information to a central server. Finally, it is important to note that FedPerC does not solve all problems within the scope of language generation models. As FedPerC offers a path forward to facilitate privacy protection and efficient on-device learning for large language models, future work may extend FedPerC to additional problems (e.g., language summarization or turn-based dialogue generation).

Ethics Statement

Federated learning systems promise the ability to learn useful models without needing access to private, protected data on user’s devices. By contributing improvements to personalization and contextu-

alization within the federated learning paradigm, FedPerC takes a step towards improving fairness of federated learning systems, which otherwise struggle with fitting to data distributions that are not common in training populations. However, it is important to note that FedPerC works to maximize the likelihood of the observed data, which may reinforce existing societal biases and stereotypes—there are no protections or safeguards in place to ensure responsible generation or unbiased preference learning (May et al., 2019; Nadeem et al., 2021; Silva et al., 2021). While this problem is certainly not unique to FedPerC, it is important to consider the safety and fairness implications of improved language generation, and future work must address biases inherent to large language models (Schick et al., 2021; Ravfogel et al., 2020). Another important ethical consideration is the potential misuse of our generative modelling approach for malicious impersonation. In our federated setup, personal embeddings would be kept on-device, meaning that an individual’s style is not accessible to others. However, this does not prevent users from manually impersonating other individuals (e.g., celebrities). Future work must explore additional mechanisms for the prevention of misuse at all stages of the personalization pipeline, including protections against impersonation of other individuals.

6 Acknowledgements

This work was supported by the Office of Naval Research (ONR) under award N00014-19-1-2076, the National Science Foundation under awards NSF CPS-2219755 and NSF IIS-2112633. Andrew Silva was supported by the Apple Scholars in AI/ML PhD fellowship.

References

- Manoj Ghuhun Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*.
- Jonathan P. Chang, Caleb Chiam, Liye Fu, Andrew Z. Wang, Justine Zhang, and Cristian Danescu-Niculescu-Mizil. 2020. *Convokit: A toolkit for the analysis of conversations*. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020, 1st virtual meeting, July 1-3, 2020*, pages 57–60. Association for Computational Linguistics.
- Letian Chen, Sravan Jayanthi, Rohan Paleja, Daniel Martin, Viacheslav Zakharov, and Matthew Gombo-

- lay. 2022. Fast lifelong adaptive inverse reinforcement learning from demonstrations. In *Proceedings of the 6th Conference on Robot Learning (CoRL), 2022*.
- Yu-Hsin Chen and Jinho D Choi. 2016. Character identification on multiparty conversation: Identifying mentions of characters in tv shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100.
- Hao Cheng, Hao Fang, and Mari Ostendorf. 2019. A dynamic speaker model for conversational interactions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2772–2785.
- Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2089–2099.
- Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. 2020. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*.
- Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. 2020. Personalized federated learning with moreau envelopes. *arXiv preprint arXiv:2006.08848*.
- Shiran Dudy, Steven Bedrick, and Bonnie Webber. 2021. Refocusing on relevance: Personalization in nlg. *arXiv preprint arXiv:2109.05140*.
- Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135.
- Otkrist Gupta and Ramesh Raskar. 2018. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8.
- David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106*.
- Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. 2020. Lower bounds and optimal algorithms for personalized federated learning. *arXiv preprint arXiv:2010.02372*.
- Filip Hanzely and Peter Richtárik. 2020. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*.
- Fang-I Hsiao, Jui-Hsuan Kuo, and Min Sun. 2019. Learning a multi-modal policy via imitating demonstrations with mixed behaviors. *arXiv preprint arXiv:1903.10304*.
- Yihan Jiang, Jakub Konečný, Keith Rush, and Sreeram Kannan. 2019. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*.
- Karen Sparck Jones. 1999. Automatic summarizing: factors immarizing: factors and directions. *Advances in automatic text summarization*, page 1.
- Hyunwoo Kim, Byeongchang Kim, and Gunhee Kim. 2020. Will i sound like me? improving persona consistency in dialogues through pragmatic self-consciousness. *arXiv preprint arXiv:2004.05816*.
- Joongheon Kim, Seunghoon Park, Soyi Jung, and Seehwan Yoo. 2021. Spatio-temporal split learning. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*, pages 11–12. IEEE.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Shekhar Koirala. 2019. Game_of_thrones. github.com/shekharkoirala/Game_of_Thrones.
- Daliang Li and Junpu Wang. 2019. Fedmd: Heterogeneous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*.
- Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020a. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450.
- Xiang Lisa Li and Percy Liang. 2021. **Prefix-tuning: Optimizing continuous prompts for generation**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Yiwei Li, Tsung-Hui Chang, and Chong-Yung Chi. 2020b. Secure federated averaging algorithm with differential privacy. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.
- Zhaojiang Lin, Andrea Madotto, Jamin Shin, Peng Xu, and Pascale Fung. 2019a. Moel: Mixture of empathetic listeners. *arXiv preprint arXiv:1908.07687*.
- Zhaojiang Lin, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019b. Personalizing dialogue agents via meta-learning. *arXiv preprint arXiv:1905.10033*.

- Yujie Lu, Chao Huang, Huanli Zhan, and Yong Zhuang. 2021. Federated natural language generation for personalized dialogue system. *arXiv preprint arXiv:2110.06419*.
- Bodhisattwa Prasad Majumder, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Julian McAuley. 2020. Like hiking? you probably enjoy nature: Persona-grounded dialog with commonsense expansions. *arXiv preprint arXiv:2010.03205*.
- Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. [On measuring social biases in sentence encoders](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.
- Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.
- Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. [StereoSet: Measuring stereotypical bias in pretrained language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.
- Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.
- Rohan Paleja, Andrew Silva, Letian Chen, and Matthew Gombolay. 2020. Interpretable and personalized apprenticeship scheduling: Learning interpretable scheduling policies from heterogeneous user demonstrations. In *Advances in Neural Information Processing Systems*, volume 33, pages 6417–6428. Curran Associates, Inc.
- Matthias Paulik, Matt Seigel, Henry Mason, Dominic Telaar, Joris Kluivers, Rogier van Dalen, Chi Wai Lau, Luke Carlson, Filip Granqvist, Chris Vandevelde, et al. 2021. Federated evaluation and tuning for on-device personalization: System design & applications. *arXiv preprint arXiv:2102.08503*.
- Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. [Null it out: Guarding protected attributes by iterative nullspace projection](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.
- Ognjen Rudovic, Nicolas Tobis, Sebastian Kaltwang, Björn Schuller, Daniel Rueckert, Jeffrey F Cohn, and Rosalind W Picard. 2021. Personalized federated deep learning for pain estimation from face images. *arXiv preprint arXiv:2101.04800*.
- Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. [Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP](#). *Transactions of the Association for Computational Linguistics*, 9:1408–1424.
- Mariah L Schrum, Erin Hedlund-Botti, and Matthew Gombolay. 2022a. Reciprocal mind meld: Improving learning from demonstration via personalized, reciprocal teaching. In *Proceedings of the 6th Conference on Robot Learning (CoRL), 2022*.
- Mariah L Schrum, Erin Hedlund-Botti, Nina Moorman, and Matthew C Gombolay. 2022b. Mind meld: Personalized meta-learning for robot-centric imitation learning. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, pages 157–165.
- Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. 2021. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR.
- Andrew Silva, Katherine Metcalf, Nicholas Apostoloff, and Barry-John Theobald. 2022. Fedembed: Personalized private federated learning. *arXiv preprint arXiv:2202.09472*.
- Andrew Silva, Pradyumna Tambwekar, and Matthew Gombolay. 2021. [Towards a comprehensive understanding and accurate evaluation of societal biases in pre-trained transformers](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2383–2389, Online. Association for Computational Linguistics.
- Aviv Tamar, Khashayar Rohanimanesh, Yinlam Chow, Chris Vigorito, Ben Goodrich, Michael Kahane, and Derik Pridmore. 2018. Imitation learning from visual data with multiple intentions. In *International Conference on Learning Representations*.
- Xueyang Tang, Song Guo, and Jingcai Guo. 2021. Personalized federated learning with clustered generalization. *arXiv preprint arXiv:2106.13044*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yuwei Wu, Xuezhe Ma, and Diyi Yang. 2021. Personalized response generation via generative split memory network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1956–1970.

Diyi Yang and Lucie Flek. 2021. Towards user-centric text-to-text generation: A survey. In *International Conference on Text, Speech, and Dialogue*, pages 3–22. Springer.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

A Generation Algorithm

At each time-step during inference, the embeddings are updated by the following equations.

$$\begin{aligned} e_t &= \text{Multi_Head_Attn}(W_{<t}, LN(e_{t-1}), W_{<t}) \\ e_t &= LN(LN(e_t) + LN(e_{t-1})) \\ e_t &= LN(\text{FFN}(e_t) + LN(e_{t-1})) \end{aligned}$$

For the first timestep, e_{t-1} is initialized as ϕ or ψ for personal and context embeddings respectively, and LN represents a layer normalization function. We apply future-masking to prevent any future-information in the sequence from leaking forward into the rest of the model. After processing the entire utterance, the generated embedding is updated to the final value of e_t , which can then be stored on-device for future processing. An updated algorithm which applies the generator to predict preference embeddings can be found in Alg 2.

B Training Details

All models are initialized with the DistilGPT2 pre-trained model from Huggingface (Wolf et al., 2019). All layers of the model are frozen, and FedPC only backpropagates error to personal and context preference embeddings. For our Meta-Learning baseline, the last layer is unfrozen and all users jointly update this final output layer (note: there is no dedicated context head in this approach). Our Split-Learning baseline assigns a unique model head to each user and to each context, and each user only updates their own model head and the contexts that they use.

All models are trained for 55 epochs over their training datasets using the Adam optimizer (Kingma and Ba, 2014) for global updates (learning rate = 1) and local updates (learning rate = 0.001).

Each client (character or Reddit user) makes 10 local updates before passing their pooled gradient information back to the server. During training, each client samples 15 data points per training pass. For local fine-tuning updates at test-time, each user makes 15 updates using a small portion of the test data (the data used for fine-tuning is not used for testing).

All models use a frozen DistilGPT2 model from HuggingFace as their initialization. After empirical experimentation, we opted to freeze the majority of the DistilGPT2 parameters by default. This freezing helped to save on computational and memory costs as well as improving generalization performance across diverse users. As a result of this freezing, shared learning and personalization updates will only affect model heads, shared embeddings, and/or personal embeddings.

FedPC leverages a standard federated averaging training procedure (FedAvg) (McMahan et al., 2017) with the addition of a FedProx penalty term (Li et al., 2020a) to regularize on-device client updates back to the globally-averaged model. Empirically, FedProx improved performance for all methods. We fix the FedProx μ parameter to 1.

Training was carried out on an NVIDIA A40 GPU with 48GB of memory. Due to limitations of the GPU, not all context-heads could be stored in memory at once for our Split Learning baseline when working with the Reddit dataset. The GPU could only accommodate 14 model heads in addition to the DistilGPT2 model, but the dataset featured 57 unique subreddits. To work around this limitation, 13 context heads were active at all times, and the parameters of those heads were saved and overwritten as necessary to ensure that each user had access to their required context heads.

C Dataset Information

The TV Show dataset is constructed by merging scripts from two shows, “Friends” and “Game of Thrones.” We use ConvoKit (Chang et al., 2020) to gather the “Friends” Corpus (Chen and Choi, 2016), and retain the six main characters. We use a set of “Game of Thrones” scripts (Koirala, 2019) to query for the thirteen characters with the highest utterance-count. Our merged dataset has 19 characters, 60650 utterances, and two contexts. The average utterance count for each character is 3370, with “Friends” characters having more utterances than “Game of Thrones” characters.

Algorithm 2 Personalized Federated Learning Loop with Generated Embeddings

```
1: Given: Training objective,  $\mathcal{L}$ , Client devices  $D$ 
2: Given: Number of client steps,  $K$ 
3: Given: Number of global steps,  $N$ 
4: Initialize: Global model,  $\theta$ , Context embeddings  $\phi$ , Context Generator  $\Gamma$ , Client Generator  $\nu$ 
5: Initialize: Personal embeddings on-device  $\psi$ 
6: for  $n \in N$  do
7:   for  $d \in D$  do
8:      $\theta_d = \theta, \phi_d = \phi, \Gamma_d = \Gamma, \nu_d = \nu$ 
9:     for  $k \in K$  do
10:      Sample  $B_d$  from client's on-device data
11:       $\theta_d \leftarrow \theta_d + \nabla_{\theta} \mathcal{L}(\theta_d, \phi_{d,c}, \psi_d, B_d)$  // Fine-tune global model with local data
12:       $\phi_d \leftarrow \nu_d(\theta_d, \phi_{d,c}, B_d)$  // Generate context embedding from local data
13:       $\psi_d \leftarrow \Gamma_d(\theta_d, \psi_d, B_d)$  // Generate personal embedding from local data
14:       $\nu_d \leftarrow \nu_d + \nabla_{\nu} \mathcal{L}(\theta_d, \phi_{d,c}, \psi_d, B_d)$  // Update client Generator
15:       $\Gamma_d \leftarrow \Gamma_d + \nabla_{\Gamma} \mathcal{L}(\theta_d, \phi_{d,c}, \psi_d, B_d)$  // Update context Generator
16:    end for
17:     $\nabla_{\theta_d} \leftarrow \theta - \theta_d$  // compute final client  $\theta$  gradients
18:     $\nabla_{\Gamma_d} \leftarrow \Gamma - \Gamma_d$  // compute final client  $\Gamma$  gradients
19:     $\nabla_{\nu_d} \leftarrow \nu - \nu_d$  // compute final client  $\nu$  gradients
20:    Return  $\nabla_{\theta_d}, \nabla_{\nu_d}$  and  $\nabla_{\Gamma_d}$  to the server
21:  end for
22:   $\nabla_{\theta} \leftarrow \frac{1}{D} \sum_d \nabla_{\theta_d}$  // calculate average  $\theta$  gradients
23:   $\nabla_{\Gamma} \leftarrow \frac{1}{D} \sum_d \nabla_{\Gamma_d}$  // calculate average  $\Gamma$  gradients
24:   $\nabla_{\nu} \leftarrow \frac{1}{D} \sum_d \nabla_{\nu_d}$  // calculate average  $\nu$  gradients
25:   $\theta \leftarrow \theta + \nabla_{\theta}$ 
26:   $\phi \leftarrow \phi + \nabla_{\phi}$ 
27: end for
```

Our Reddit experiments use the “reddit-corpus-small” dataset from ConvoKit (Chang et al., 2020), which includes posts from the top-100 subreddits over a set period of time. We filter the dataset to

only include users with at least 50 utterances and contexts (subreddits) with at least 150 utterances. The resulting dataset has 326 characters, 30260 utterances, and 57 contexts.

A Neural CRF-based Hierarchical Approach for Linear Text Segmentation

Inderjeet Nair, Aparna Garimella, Balaji Vasan Srinivasan, Natwar Modani,
Niyati Chhaya, Srikrishna Karanam, Sumit Shekhar

Adobe Research, India

{inair, garimell, balsrini, nmodani,
nchhaya, skaranam, sushekha}@adobe.com

Abstract

We consider the problem of segmenting unformatted text and transcripts linearly based on their topical structure. While prior approaches explicitly train to predict segment boundaries, we propose to address this task by inferring the hierarchical segmentation structure associated with the input text. For this purpose, we present a data curation strategy to obtain the hierarchical segmentation structure annotations for over 700K Wikipedia articles. We then propose the first supervised approach to generate hierarchical segmentation structures for given text based on a neural conditional random field (CRF) that explicitly models the statistical dependencies between nodes and their constituent children. We introduce a novel data augmentation scheme as part of our model training, which involves sampling a variety of node aggregations, permutations, and removals, all of which help capture fine-grained and coarse topical shifts in the data and improve model performance. Extensive experiments show that our model outperforms or achieves competitive performance when compared to previous state-of-the-art algorithms in the following settings: rich-resource, cross-domain transferability, few-shot supervision, and segmentation when topic label annotations are provided.

1 Introduction

Text segmentation (Hearst, 1997; Choi, 2000), an important task in information retrieval, is defined as the process of dividing unstructured text into topically coherent segments. Because it recovers topical structure from unformatted text, it can be used as a pre-processing step for several downstream tasks such as text summarization (Mitra et al., 1997), question answering (Oh et al., 2007) and discourse analysis (Van Dijk, 1982).

Most prior works on text segmentation (Hearst, 1997; Choi, 2000; Koshorek et al., 2018) attempted to address this task by explicitly *predicting the segment boundaries*, with the assumption that any

given text can be *decomposed* into contiguous, non-overlapping, indivisible segments, based on topical themes. The discourse segmentation theory (Grosz and Sidner, 1986), however, asserts that the outcome may not always be strictly decompositional, *i.e.*, a segment may have sub-segments within it, and segments may overlap with each other. Following this theory, we hypothesize that explicitly training to infer the hierarchical topic structure of the underlying text leads to better linear segmentation, as it forces the models to examine text at multiple levels to extract coarse-grained to fine-grained topical segments. Further, this allows inference of linear segments of varying granularity that can be used for various downstream applications.

Previous works on hierarchical segmentation are largely unsupervised (Eisenstein, 2009; Simon et al., 2015) due to the unavailability of large labelled datasets with hierarchical structure information. In this paper, we propose to leverage the hierarchical structures in a supervised manner, given the superior performances of supervised models across several language processing tasks (Mikolov et al., 2013; Pennington et al., 2014; Devlin et al., 2019), and propose a data curation strategy to obtain the hierarchical segmentation structures for Wikipedia articles. Specifically, we leverage the available HTML tag annotations¹ and use them to identify section and sub-section information with their hierarchical level, which are then leveraged to obtain the associated ground truth hierarchical structure. Further, because these are extracted from Wikipedia dump, they cover a wide range of topics unlike prior/existing datasets (Eisenstein, 2009).²

Our approach is based on a recent CRF-based constituency chart parsing technique (Zhang et al., 2021), which offers efficient algorithms for super-

¹<https://dumps.wikimedia.org/>

²Note that the dataset proposed by Eisenstein (2009) is a small one consisting of 12 examples for evaluation, and will not suffice for training large models.

vised training and precise inference. This framework explicitly models the relationships between nodes and their offspring in binary trees, and thus can enable hierarchical segmentation inference by utilising the relationships between coarse segments and their fine-grained sub-segments. However, there are three challenges to directly adapt this method to hierarchically segment text: (a) In contrast to the abundant labelled resources available for constituency parsing (Marcus et al., 1993; Xue et al., 2005), there is no large-scale labelled dataset for this task. (b) This method can only infer binarized hierarchical structures; it cannot be extended to infer general hierarchical structures with nodes having any number of children, as training and inference become infeasible. (c) While the existing method processes a sequence of tokens, the input in our case would be a sequence of sentences.

We propose a framework for linear text segmentation using the hierarchical structures of the underlying text. Specifically, our work makes four main contributions: (1) We design an algorithm to obtain the hierarchical structures for Wikipedia articles in HTML format, and curate a large labelled dataset for hierarchical text segmentation.³ (2) We present an algorithm based on the Chomsky Normal Form (CNF) (Chomsky, 1959; Hopcroft et al., 2001; Lange and Leiß, 2009) theory to convert the hierarchical structures to binarized form - which makes the computation of the tree-structure CRF objective tractable. (3) We propose a Transformer-based architecture (Vaswani et al., 2017) to encode the input sequence’s sentences, which uses a lot fewer parameters than previous state-of-the-art BERT-based (Devlin et al., 2019) approaches (Lukasik et al., 2020). (4) We further propose a data augmentation technique involving random node aggregations, removals and permutation, which results in significant performance improvement. Finally, we demonstrate our method’s efficacy by comparing its performance against prior unsupervised and supervised linear text segmentation approaches.

2 Related Works

Prior works for linear text segmentation can be divided into unsupervised and supervised methods, both of which can be further categorized into locally and globally-informed ones. Locally-

³The code to curate dataset is available at https://github.com/inderjeetnair/hierarchical_text_segmentation_data

informed methods find segment boundaries by estimating the extent of topical shift using local cues (Hearst, 1997; Blei and Moreno, 2001; Lafferty et al., 2001). While these methods enjoy quick inference and low memory constraints as they only utilize local features, they can result in erroneous predictions when met with short inconsequential digressions (Kazantseva and Szpakowicz, 2011). Globally-informed methods, on the other hand, utilize the complete context in optimizing an objective to find the locations of topical shift (Choi, 2000; Kazantseva and Szpakowicz, 2011; Malioutov et al., 2007; Fragkou et al., 2004; Glavaš et al., 2016). As they consider the entire global context in inference, these methods have higher memory constraints and time requirements.

More recently, Koshorek et al. (2018) introduced a large-scale dataset for linear text segmentation, which has resulted in the application of supervised neural models to predict the segment boundaries for unstructured text (Koshorek et al., 2018; Badjatiya et al., 2018; Li et al., 2018). These models not only achieve better performance but also are endowed with high inference speed, owing to parallelized computing with modern GPU architectures.

Owing to the success of supervised methods for linear text segmentation, we design a globally-informed supervised neural model for predicting segment boundaries. However, unlike previous works which address segment boundary prediction explicitly, ours first hierarchically segments the text, and then leverages the resulting structures to predict the linear segment boundaries. To enable the supervised training of our proposed approach, we curate a large labelled dataset consisting of Wikipedia articles along with their hierarchical structures automatically. Further, our proposed method requires significantly fewer parameters than prior SoTA globally-informed methods while achieving better performances. To the best of our knowledge, ours is the first work to leverage hierarchical structures to predict segment boundaries, and show that this results in improved performances for the task of linear text segmentation.

3 Problem Formulation

Here, we briefly outline the objectives of linear and hierarchical text segmentation tasks. Given an article \mathbf{S} composed of n sentences, $\mathbf{S} = s_0, s_1, \dots, s_{n-1}$, the goal of linear segmentation is to obtain a contiguous partition $L =$

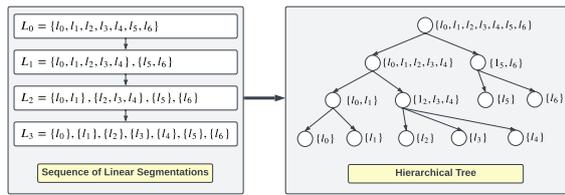


Figure 1: Transformation of a sequence of linear segmentations to a hierarchical tree.

l_0, l_1, \dots, l_{k-1} such that joining the elements of l_i in the same order reconstructs \mathbf{S} and $l_i \cap l_j = \emptyset \forall i \neq j$. Each segment l_i in L is associated with a topical theme which can be used for downstream tasks such as summarization, information retrieval, etc.

Hierarchical segmentation (McFee et al., 2017) aims to infer a sequence of linear segmentations, $\mathbf{L} = L_0, L_1, \dots, L_{m-1}$, where L_i is coarser than L_j for $i < j$. Each element of \mathbf{L} is thus a refinement of all its preceding elements, to satisfy this coarse-to-fine grained constraint. The refinement condition for $i \leq j$ is: $\forall l \in L_j \exists l' \in L_i : l \subseteq l'$. That is, every segment in L_j is a subset of a segment in L_i . In this paper, we represent the information contained in \mathbf{L} using a hierarchical tree (Fig. 1), where each node (other than the leaf nodes) represents a topical theme. The nodes near the root represent general / coarser topics, and those near the leaves indicate specific / fine-grained topics. In our approach, the inferred hierarchical segmentation is in the form of a tree. After converting this tree to a sequence of linear segmentations, we return an appropriate element from the sequence as our inferred linear segmentation.

4 Dataset Curation

To train our proposed method in a supervised manner, we collect hierarchical segmentation structure annotations for the Wikipedia articles in WIKI-727K dataset (Koshorek et al., 2018). As done in (Koshorek et al., 2018), the articles in the HTML form are preprocessed using WikiExtractor⁴ to remove (a) non-text elements such as tables and figures, and (b) very short sections and sub-sections spanning fewer than three sentences. The markup tags (`<h1>`, ..., `<h6>`) associated with different sections define the level of hierarchy for a given text segment. We leverage this markup information to obtain the hierarchical structure among the vari-

⁴<https://github.com/attardi/wikiextractor.git> (Distributed under GNU Affero General Public License v3.0)

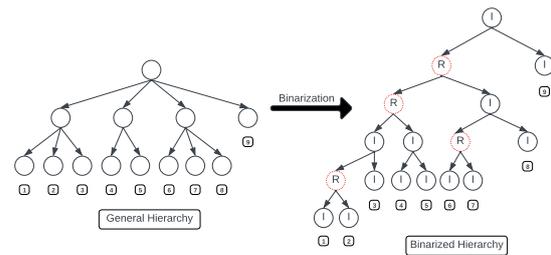


Figure 2: **Binarization:** Transformation of a tree having nodes containing more than 2 children to a binarized form.

ous segments. We thus obtain a sequence of HTML elements for each article. In the next sub-section, we describe our algorithm to obtain the hierarchical structures associated with these sequences. We obtain the hierarchical structure annotations only for the train split articles of the WIKI-727K dataset; our approach can be applied to larger document collections to obtain more datapoints.

4.1 Hierarchical Labelling

Algorithm 1 Algorithm for constructing hierarchical structure from a list of HTML elements

```

Require:  $\mathcal{X} = x_1, x_2, \dots, x_L$  ▷ Ordered list of HTML elements
 $c \leftarrow \text{ROOT}$  ▷ ROOT initialized denoting the root of the tree to be
constructed
 $\mathcal{T} \leftarrow \text{ROOT}$ 
for  $i = 1$  to  $\|\mathcal{L}\|$  do
   $x \leftarrow \mathcal{X}[i]$ 
  while  $\text{PRIORITY}(x.\text{TAG}) \geq \text{PRIORITY}(c.\text{TAG})$  do ▷ Selecting
appropriate element to add  $x$ 
     $c \leftarrow c.\text{PARENT}$  ▷ Updating  $c$  to its parent
  end while
   $c.\text{ADD}(x)$  ▷  $x$  is added as the next child of  $c$ 
   $c \leftarrow x$ 
end for
Return  $\mathcal{T}$ 

```

Let the sequence of HTML elements associated with an article be $\mathcal{X} = x_0, x_1, \dots, x_{L-1}$, where $x_i.\text{TAG}$ denotes the markup type associated with x_i , and $x_i.\text{TEXT}$ denotes its associated text. Here, we outline our algorithm to obtain the hierarchical organization of these elements. Let this hierarchical organization be represented as a tree rooted at \mathcal{T} . The non-leaf nodes represent topics, while the leaf nodes represent sentences from the article.

Our algorithm iterates over the elements in \mathcal{X} and progressively adds them to the tree rooted at \mathcal{T} . It maintains a reference to the node c that is last added to the tree. To add the next element x , the algorithm only considers two possibilities: (a) x is the next child of c , or (b) x is the next child of one of the ancestors of c . This is to ensure that the pre-order traversal of \mathcal{T} recovers \mathcal{X} (which happens when x is added using the above two rules). To find

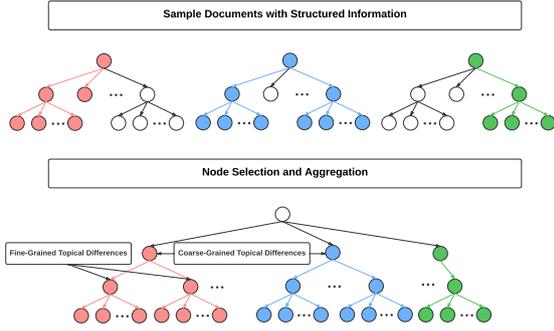


Figure 3: **Data Augmentation:** First, hierarchical structures are randomly sampled from the corpus; then, some nodes are removed from the sampled structures and the outcome is combined

the element to which x must be added, we associate a priority to each markup type in the following decreasing order: h_1, \dots, h_6, p , where h_i indicates for section / sub-section headings and p its associated text. For adding x , c is updated to its parent until the priority of c exceeds that of x . Algorithm 1 presents the pseudo code.

4.2 Binarization

Algorithm 2 Algorithm to be applied to every node having more than 2 children to convert the original structure to the binarized form

Require: x ▷ Node having more than 2 children
 $c \leftarrow \text{NEWNODE}()$
 $c.\text{TYPE} \leftarrow \text{I}$
 $n \leftarrow ||x.\text{CHILDREN}||$
for $i = 1$ to n **do**
 $c' \leftarrow \text{NEWNODE}()$ ▷ New node initialized
 $c'.\text{TYPE} \leftarrow \text{R}$
 $c.\text{CHILDREN} \leftarrow [c', x.\text{CHILDREN}[n - i + 1]]$ ▷ Restricting the number of productions to 2
 $c \leftarrow c'$
end for

The hierarchies thus obtained allows nodes to have more than two children. We convert them to binarized form (from which the original structures can be recovered) to ensure tractable training and inference using our CRF-based segmentation model (§5) (using Algorithm 2).

This algorithm visits each node x in a tree \mathcal{T} that has more than two children, and partitions the children into two sets having $||x.\text{CHILDREN}|| - 1$ children and 1 child respectively. Thereafter, a new node is constructed whose children are assigned to the former set, followed by the updation of $x.\text{CHILDREN}$ to contain the new node and the latter set in the partition. This is repeated until the tree is devoid of nodes with more than 2 children. To ensure recoverability, we define two types of nodes in the binarized trees: **Reducible (R)** and **Irreducible (I)**. The nodes retained from \mathcal{T} are

regarded to as **I**, and those added to convert \mathcal{T} to the binarized form are referred to as **R** (an example of binarization is shown in Fig. 2). These types are assigned to the node’s ‘TYPE’ property (pseudo code in Algorithm 2). To recover the original tree, we visit every **Reducible** node in the binarized tree and connect its children to its parent in the same order. This process is repeated until the structure becomes devoid of any **Reducible** nodes.

4.3 Data Augmentation

An inherent limitation of this dataset stems from the fact that each Wikipedia page is composed of a single global topic, and the direct usage of this data will only train the model in detecting fine-grained topical shifts resulting from sub-sections / sub-headings. However, an article in practice can also contain fragments with stark topical contrast.

To overcome this, we introduce a data augmentation strategy, where a subset of tree root references are sampled at every iteration. Thereafter, some of the children of these nodes are randomly dropped and the ordering of left-out children is randomly permuted. Finally, a new node is created and its children are the sampled tree roots (Figure 3). This new root consists of several coarse topics and the random permutation of the child nodes ensures that the model robustly infers topical segments independent of the order of the child nodes. The augmentation is performed at every epoch ensuring the number of artificially synthesized datapoints is equal to the actual number of documents in the train split.

5 Neural CRF Segmentation Model

5.1 CRF Formulation

We consider an article containing n sentences, $\mathbf{S} = s_0, s_1, \dots, s_{n-1}$ and its corresponding hierarchical segmentation tree structure \mathbf{t} . A node in \mathbf{t} representing a segment spanning s_i, s_{i+1}, \dots, s_j is denoted by (i, j) . Alternatively, \mathbf{t} can be expressed as a set of tuples where each tuple corresponds to a node segment in \mathbf{t} .

Inspired by (Zhang et al., 2021), our model presents a scoring function $s(\cdot, \cdot) \rightarrow \mathbb{R}$ (described later) to assign a score for each node in \mathbf{t} , e.g., $s(i, j)$ represents the score for a node entailing s_i, s_{i+1}, \dots, s_j . We define a function \mathcal{S} to score the tree \mathbf{t} using the sequence \mathbf{S} as:

$$\mathcal{S}(\mathbf{S}, \mathbf{t}) = \sum_{(i,j) \in \mathbf{t}} s(i, j) \quad (1)$$

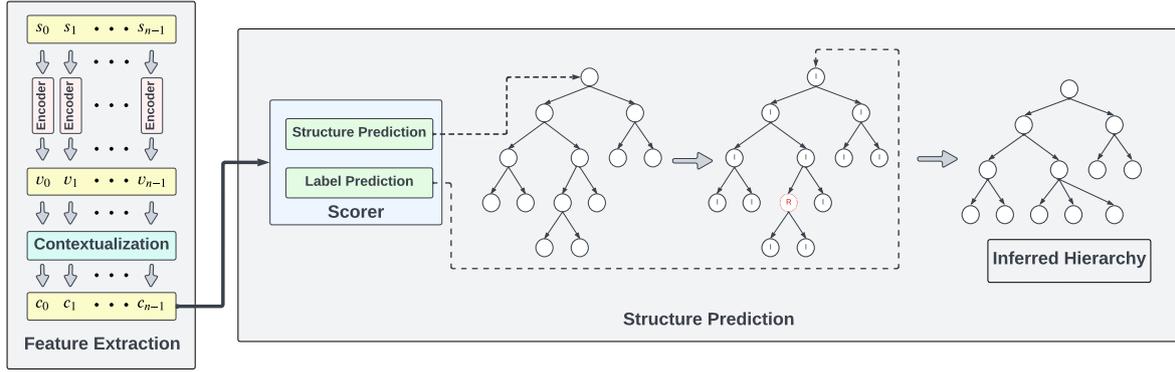


Figure 4: **Pipeline:** Our pipeline is divided into two stages. We extract contextualised embeddings for each sentence during the feature extraction stage. These representations are then fed into the structure prediction module, which first infers the binarized tree and label for each node. This inference is used to construct the actual hierarchy.

Under CRF, we define conditional probability as:

$$\mathbb{P}(\mathbf{t}|\mathbf{S}) = \frac{\mathcal{S}(\mathbf{S}, \mathbf{t})}{Z(\mathbf{S}) := \sum_{\hat{\mathbf{t}}} \mathcal{S}(\mathbf{S}, \hat{\mathbf{t}})} \quad (2)$$

The denominator sums the score of all possible legal hierarchical trees.⁵ Note that, the computation of the partition function $Z(\mathbf{S})$ in the denominator is intractable (exponential time complexity) if we consider all possible hierarchical trees with nodes having arbitrary number of children. However, binarization offers efficient dynamic programming algorithm to compute the partition function with polynomial time complexity. Thus, we binarize \mathbf{t} to obtain $\tilde{\mathbf{t}}$. However, the binarization also associates a type to each node from the set $\{\mathbf{R}, \mathbf{I}\}$. Under this formulation, every node in $\tilde{\mathbf{t}}$ can be represented by a triplet (i, j, l) which indicates that the corresponding node of type $l \in \{\mathbf{R}, \mathbf{I}\}$ spans s_i, s_{i+1}, \dots, s_j . Similar to Zhang et al. (2021)’s two-staged method, we first identify the binarized tree structure that maximises $\mathbb{P}(\tilde{\mathbf{t}}|\mathbf{S})$, whose denominator only adds the scores of the binarized trees, and then determine the type for each of its constituent nodes. To find the optimal structure maximizing $\mathbb{P}(\tilde{\mathbf{t}}|\mathbf{S})$, we leverage Cocke–Younger–Kasami algorithm (CYK) algorithm (Sakai, 1961). For each span (i, j) in the inferred structure, we predict its type l :

$$l = \arg \max_{\hat{l} \in \{\mathbf{R}, \mathbf{I}\}} s(i, j, \hat{l}) \quad (3)$$

Figure 4 shows how our model processes the input sequence to infer the hierarchical segmentation structure. In the next subsection, we specify the architectural details of its components.

⁵Legal hierarchical trees are expected to satisfy two conditions: (1) There should be one-to-one correspondence between the leaf nodes and the constituent sentences. (2) Every node in the tree must span consecutive sequence of sentences.

5.2 Model Architecture

We now describe the implemented architecture, which is adapted from the model proposed by Stern et al. (2017) and Zhang et al. (2021) with two important modifications: (a) utilization of memory-efficient Transformer (Vaswani et al., 2017) model for encoding the sentence in place of word encoder, and (b) better choice of hyper-parameters for hierarchical text segmentation.

Encoder. Each sentence in \mathbf{S} is encoded in a context-independent manner using the Transformer-based model proposed by Wang et al. (2022), which contains 6 layers and 22M parameters. The parameters of this model are fine-tuned using self-attention distillation (Wang et al., 2022) for the compression of large language models like RoBERTa-Large (Liu et al., 2019). This stage transforms s_0, s_1, \dots, s_{n-1} to v_0, v_1, \dots, v_{n-1} with 384 length each.

Contextualization. To contextualize v_0, \dots, v_{n-1} , we implement two BiLSTM layers over it. While the architecture implemented by Zhang et al. (2021) comprises of three BiLSTM layers, we observe that direct usage of the same hyper-parameter settings lead to sub-optimal results on a validation set. The final context-aware representation for a sentence is obtained by concatenating the corresponding forward and backward vectors from the last layer. Let these vectors be represented by c_0, c_1, \dots, c_{n-1} .

Scoring. Having obtained the contextualized representations of the elements in \mathbf{S} , we describe the architecture used to compute $s(i, j)$ and $s(i, j, l)$. For the computation of $s(i, j)$, the contextualized representations are passed to two multi-layer perceptron (MLP) modules to obtain the left and right bound-

ary representation vectors (Zhang et al., 2021):

$$r_i^s; l_i^s = \text{MLP}_r^s(c_i); \text{MLP}_l^s(c_i) \quad (4)$$

Similarly, additional set of boundary vectors are derived to compute $s(i, j, l)$ for label prediction:

$$r_i^l; l_i^l = \text{MLP}_r^l(c_i); \text{MLP}_l^l(c_i) \quad (5)$$

The dimension of the boundary vectors for structure prediction is set to 500 and that for label prediction is set to 800. $s(i, j)$ is computed by introducing a trainable parameter $\mathbf{W} \in \mathbb{R}^{d \times d}$:

$$s(i, j) = l_i^{sT} \mathbf{W} r_j^s \quad (6)$$

Similarly $s(i, j, l)$ is computed by introducing \mathbf{W}_R and \mathbf{W}_I to derive the scores: $s(i, j, \mathbf{R})$ and $s(i, j, \mathbf{I})$ respectively. Note we use r_i^l and l_i^l for the computation of these scores of instead of r_i^s and l_i^s .

5.3 Training

An instance in the labelled dataset can be represented by: $(\mathbf{S}, \tilde{\mathbf{t}}, \mathbf{l})$ where \mathbf{l} is the set of all spans annotated with their corresponding type from $\{\mathbf{R}, \mathbf{I}\}$. The loss function is formed by accumulating two components:

$$\mathcal{L}(\mathbf{S}, \tilde{\mathbf{t}}, \mathbf{l}) = \mathcal{L}_s(\mathbf{S}, \tilde{\mathbf{t}}) + \mathcal{L}_l(\mathbf{S}, \tilde{\mathbf{t}}, \mathbf{l}) \quad (7)$$

The first term tries to maximize $\log(\mathbb{P}(\tilde{\mathbf{t}}|\mathbf{S}))$ by refining the scoring function $s(i, j)$. The second term establishes cross-entropy loss for the type prediction of the constituent spans. While the time complexity of the partition function computation $Z(\mathbf{S})$ for $\log(\mathbb{P}(\tilde{\mathbf{t}}|\mathbf{S}))$ is $\mathcal{O}(n^3)$ using inside algorithm (Lari and Young, 1990), we implement the batchified version of this algorithm proposed by Zhang et al. (2021) that provides much better time complexity ($\mathcal{O}(n)$ for a batch).

We train our models over the curated dataset for 4 epochs using Adam optimizer (Kingma and Ba, 2015) with batch size of 100 and learning rate initialized to 2×10^{-4} . The learning rate is exponentially decayed to 0.75 times its initial value after 50K optimizer steps. The training is restricted to datapoints having less than 200 sentences due to GPU memory constraints.

6 Experiments and Results

We assess our method’s performance in various settings when compared to SoTA linear segmentation techniques. The tree inference from the model is converted to a sequence of linear segmentations

Method	Precision	Recall	F1 Score
BI-LSTM	69.3	49.5	57.7
CROSS SEGMENT BERT	69.1	63.2	66.0
BERT+BI-LSTM	67.3	53.9	59.9
HIERARCHICAL BERT	69.8	63.5	66.5
HIERCRF	80.6	59.4	68.4
HIERCRF-AUG	82.5	60.3	69.7
HIERCRF-BERT	79.0	63.3	70.2
HIERCRF-AUG-BERT	80.4	64.6	71.6

Table 1: Comparison with supervised baselines when abundant labelled data is available. Only the hierarchical structures of the articles in the train split of WIKI-727K are used in our method for consistency.

(Fig. 1). As the position of an element in this sequence indicates the extent of segmentation, we select an appropriate position (constant for a dataset, obtained through validation for supervised methods, and second position for unsupervised methods due to how coarsely grained the topical shifts are in them) and return the corresponding segmentation. We call our method variants **HIERCRF** and **HIERCRF-AUG**, where the former and latter are trained without and with data augmentation.

We compare our models to supervised methods trained on WIKI-727K augmented with our hierarchical structure annotations (§6.1). Unsupervised methods (Kazantseva and Szpakowicz, 2011; Du et al., 2013) require a significant amount of time for inference as they are computationally intensive and do not take advantage of GPU parallelization for efficiency. Hence, we do not compare their performance for WIKI-727K, which contains a large number of datapoints in the test split. In §6.2, we look at how our model transfers knowledge from one domain to another and investigate efficacy in a low-resource setting. Here, we compare our method to unsupervised as well as some supervised methods. Finally, we evaluate how our model utilises topic label information for segmentation using WikiSection (Arnold et al., 2019).

6.1 Rich Resource Setting

Here, we consider models that perform well in linear text segmentation when large labelled datasets are available for training. The large-scaled dataset (WIKI-727K) curated by Koshorek et al. (2018) for linear segmentation has been instrumental in the formulation of several deep learning methods (Koshorek et al., 2018; Lukasik et al., 2020). We use the following as baselines: **BI-LSTM** (Koshorek et al., 2018), **CROSS SEGMENT BERT** (Lukasik et al., 2020), **BERT+BI-LSTM** (Lukasik et al., 2020) and **HIERARCHI-**

Property	Clinical	Fiction	Wiki
# Documents	227	85	300
# Sentences	31,868	27,551	58,071
Segment Length	Mean	35.72	24.15
	Std Dev	29.37	18.24

Table 2: Datasets used for comparing our method against statistical methods for linear text segmentation.

CAL BERT (Lukasik et al., 2020). Most of them use BERT as their encoder ($> 109M$ parameters). To show the increased effectiveness of our model when its complexity is increased, we also present the performance with BERT as its encoder. We use the test split in WIKI-727K (73, 233 instances) and F1 score for evaluating the segment boundary prediction performance. Further, the first and the last sentences are not annotated in the ground truth set of boundaries, as any segmentation algorithm can easily predict them as segmentation boundaries, thus inflating performance.

We note, from Table 1, that our linguistically motivated approach for inferring linear segmentation from hierarchical segmentation gives better performance despite having significantly fewer parameters ($\approx 23M$ as opposed to strongest baseline’s $\approx 109M$). As expected, increasing model complexity improves performance, resulting in a new SoTA for WIKI-727K (71.6 F1). We also observe our precision is comparatively higher and the recall is lower. We attribute this to node misclassification in the inferred hierarchy. By construction, **R** nodes are removed to get the final inference indicating the corresponding segments would be absent in the inferred hierarchy. Thus, even if the inferred binarized structure is accurate, a misclassification of **I** nodes as **R** will result in lower recall.

6.2 Cross Domain and Low Resource Setting

We categorize the methods in following groups.

A: Unsupervised: This group comprises of the following unsupervised techniques: **U&I** (Utiyama and Isahara, 2001), **MINCUT** (Malioutov and Barzilay, 2006), **BAYESSEG** (Eisenstein and Barzilay, 2008), **APS** (Kazantseva and Szpakowicz, 2011), **PLDA** (Purver et al., 2006) and **TSM** (Du et al., 2013). These methods use the **number of gold standard segments and test data corpus for tuning the hyperparameters.**

B: Cross-Domain Transferability: Here, we pre-train supervised models using WIKI-727K’s train split and evaluate on other datasets completely unsupervised, without knowing the number of gold-standard segments. We consider top baselines

Method	Clinical		Fiction		Wiki	
	WD	P_k	WD	P_k	WD	P_k
GROUP A						
U&I	37.6	37.0	45.9	45.9	36.8	36.8
MINCUT	38.2	36.8	40.5	37.1	38.9	36.4
BAYESSEG	35.3	33.9	33.7	27.8	39.0	35.9
APS	39.9	39.6	48.0	45.1	38.0	39.2
PLDA	37.3	32.4	43.0	36.1	-	-
TSM	34.5	30.6	40.8	32.5	-	-
RANDOM	45.9	44.1	51.0	47.5	48.6	48.0
GROUP B						
CROSS SEG BERT	40.8	39.4	44.4	42.7	37.1	36.3
HIER BERT	34.8	33.9	41.1	39.0	35.6	34.5
HIERCRF	34.4	33.9	43.1	42.4	33.4	30.0
HIERCRF-AUG	33.7	33.0	42.8	42.2	30.9	28.6
GROUP C						
CROSS SEG BERT- No-PT	38.4	35.0	39.4	29.5	40.6	38.0
CROSS SEG BERT	31.0	29.8	34.4	27.6	32.4	27.5
HIER BERT-NO-PT	33.4	32.4	37.8	34.5	39.1	38.1
HIER BERT	38.5	35.2	34.0	25.5	35.0	29.1
HIERCRF-NO-PT	33.3	32.2	34.7	34.5	37.0	35.9
HIERCRF	26.7	25.5	33.3	29.9	28.6	26.3
HIERCRF-AUG	25.2	24.4	32.6	28.4	27.9	25.7

Table 3: **Performance of our model against unsupervised approaches.** All results are averaged for 5 random splits. TSM and PLDA implementations are unavailable to report their performance on Wiki. PT: Pre-training.

from §6.1 (CROSS SEGMENT BERT and HIERARCHICAL BERT) and our variants (HIERCRF and HIERCRF-AUG). We re-implement the supervised baselines as original code is unavailable.

C: Low Resource Setting: Here, we expose the models to 20% of the dataset for supervised learning and evaluate it on the rest of the dataset (results averaged for 5 random seeds). For fine-tuning, our model parameters are optimized to predict the flat-hierarchy associated with the datapoints in the training split. We also report the performance of the model (appended with NO-PRETRAINING) which is not pretrained over WIKI-727K.

We use the following three datasets to compare our method against the statistical unsupervised approaches (Table 2):

Clinical (Eisenstein and Barzilay, 2008). Every document is a chapter from a medical textbook where labeled boundaries represent section breaks.

Fiction (Kazantseva and Szpakowicz, 2011). Each document is a fiction from Project Gutenberg where boundary annotations denote chapter breaks.

Wiki (Badjatiya et al., 2018). 300 articles are randomly sampled from the wikipedia dump where section tag labels are used to annotate boundaries.

We assess the performance of these methods using P_k (Beeferman et al., 1999) and WinDiff (WD) (Pevzner and Hearst, 2002) error metrics. P_k computes the probability that two segments sampled from a document are incorrectly identified as

Method	$P_k(\text{City})$	$P_k(\text{Disease})$
SECTOR	14.4	26.3
S-LSTM	9.1	20.0
TRANSFORMER ²	8.2	18.8
HIERARCHICAL BERT	8.6	22.4
CROSS SEGMENT BERT	10.1	21.8
HIERCRF*	8.8	20.4
HIERCRF-AUG*	8.5	21.2
HIERCRF	8.1	20.3
HIERCRF-AUG	8.0	20.0

Table 4: Comparison with various approaches in leveraging segment labels. * indicates those methods do not use segment labels. For our AUG-appended methods, we augment the flat hierarchies associated with the training datapoints.

belonging to the same segment. WD moves a sliding window across the document and counts the number of instances where the hypothesized and reference segment boundaries are different.

The results shown in Table 3 demonstrate the competitive performance exhibited by our model without any additional fine-tuning (**B**). As our models in **B** are not subjected to separate hyperparameter tuning for different datasets, our proposed models can be applied to other domains with minimal changes (only the position from the sequence of segmentations inferred from the model needs to be specified). The performance of HIERCRF-AUG in **B** is better than all the methods in **A** (despite not knowing the number of gold-standard segments and tuning hyperparameters over the testing corpus) for Wiki and Clinical dataset which demonstrates the effectiveness of our approach in transferring the knowledge from one domain to another. Fine-tuning our models with small number of datapoints (**C**) provides competitive results for most of the datasets. As expected, the performance in few-shot supervision setting is boosted if the model is pre-trained over WIKI-727K for the supervised approaches. Because the datapoints in the Fiction dataset are longer than in the other datasets, our model performs poorly (§8).

6.3 Results using Segment Labels

We compare our model’s segmentation performance to baselines when segment-wise topic labels are given. We use WikiSection’s split (Arnold et al., 2019) that comprises of English documents from two domains: diseases (3.6K documents) and cities (19.5K documents). We use 70/20/10 splits for train/dev/test. We assess how well our model uses the segment labels compared to earlier methods. We consider the following approaches: SECTOR (Arnold et al., 2019), S-LSTM (Barrow et al., 2020) TRANSFORMER² (Lo et al., 2021),

HIERARCHICAL BERT and CROSS SEGMENT BERT. For fair comparison, no model is pretrained on WIKI-727K.

To incorporate the topic label information in training, we use the boundary vectors for label prediction and a scoring mechanism similar to Eqn 6. Specifically, the likelihood that span (i, j) corresponds to topic label **T** is proportional to $s(i, j) = l_i^T \mathbf{W}_T r_j^l$. The parameters \mathbf{W}_T for each topic label **T** are trained using cross-entropy loss similar to Eqn 7.

Table 4 shows that after incorporating topic label information, our model provides SoTA performance for the City Domain and competitive performance for the Disease Domain. This suggests that providing auxiliary information, such as topic label information, improves the performance of our model. We believe that using a medical domain specific encoder (Gu et al., 2021) would improve our model’s performance in the Disease domain.

6.4 Ablation: Training Size and Performance

Here, we investigate the effect of training our models on data splits comprising of articles with varying sizes (number of sentences). Our objective is to demonstrate that the performance of the model can be improved by including datapoints with larger size. However, the limitations in the GPU hardware, has restricted the maximum datapoint size in the training split to 200. We consider three variants of our curated dataset for training: (a) datapoint sizes ≤ 50 , (b) datapoint sizes ≤ 100 , and (c) datapoint sizes ≤ 200 . We report the performances of models trained over these variants on splits containing datapoints with size in the following ranges: $[1, 50]$, $(50, 100]$, $(100, 150]$, $(150, 200]$, $(200, \infty)$, and $[1, \infty)$ from the WIKI-727K test data.

Table 5 shows the F1 scores. We note three main trends: (1) The model performance decreases as the size of the datapoint increases. This can be attributed to the following two reasons. Firstly, majority of the datapoints in our curated dataset have size in the range $[1, 50]$, and the number of datapoints decreases as the range varies from $(50, 100]$ to $(200, \infty)$. Secondly, the ability of the model to produce discriminative contextualized sentence features is dependant on the size of the input datapoint. As the contextualization is brought about by Bi-LSTM module, very long sequences result in vanishing gradient problem, and this results in less effective modeling for longer sequences. (2)

Method	Training Dataset Size Upper-bound	Testing Dataset Size Range					
		[1, 50]	(50, 100]	(100, 150]	(150, 200]	(200, ∞)	[1, ∞]
HIERCRF	50	70.4	57.5	46.6	39.5	30.7	64.2
	100	72.5	62.2	52.3	47.0	39.4	67.4
	200	72.9	64.2	54.8	50.1	41.8	68.4
HIERCRF-AUG	50	71.8	61.3	51.2	45.4	37.2	66.5
	100	73.9	64.1	54.9	50.3	42.5	69.0
	200	74.2	64.2	55.2	50.3	42.9	69.7

Table 5: **Effect of including datapoints with larger size in the training set:** A model’s performance in terms of F_1 -Score decreases as the size of the datapoint increases. Increasing the upper bound of the datapoint size in the training dataset improves the performance uniformly for all the dataset size ranges.

Including datapoints with larger size uniformly improves the performance of the model across all the ranges. Thus, one line of future work could be to procure more labels for articles with more number of sentences. However, it is to be noted that training the model over larger sequences imposes heavy requirements on GPU memory. (3) The effectiveness of our data augmentation can be seen here as well in producing uniformly better results than the model not trained over augmented corpus.

7 Conclusion

We curated a dataset with hierarchical structures and introduced an approach for linear segmentation by inferring the hierarchies. We illustrated the effectiveness of our approach against prior supervised and unsupervised methods for several datasets. Our method, when exposed to a small fraction of the data for fine-tuning, achieves superior performance when evaluated on other datasets for this task. Unlike prior unsupervised methods, ours without any hyperparameter tuning achieved competitive results. While we focussed on predicting segment boundaries, our method could also be applied to yield hierarchical segmentation. However, the challenges specific to dataset size and memory persist. We will study these aspects in our future work.

8 Limitations

The ablation studies conducted in this paper highlight some of the limitations of our model. The model predictions are erroneous when the length of the input text is very large. Even if we are able to curate sufficient number of very long text with ground truth hierarchical structure, training the model parameters imposes heavy requirements for GPU memory. This demands for the requirement of a better architecture that not only handles long range sequence dependencies but also is GPU memory-efficient while training.

References

- Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. 2019. [SECTOR: A Neural Model for Coherent Topic Segmentation and Classification](#). *Transactions of the Association for Computational Linguistics*, 7:169–184.
- Pinkesh Badjatiya, Litton J. Kurisinkel, Manish Gupta, and Vasudeva Varma. 2018. Attention-based neural text segmentation. In *Advances in Information Retrieval*, pages 180–193, Cham. Springer International Publishing.
- Joe Barrow, Rajiv Jain, Vlad Morariu, Varun Manjunatha, Douglas Oard, and Philip Resnik. 2020. [A joint model for document segmentation and segment labeling](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 313–322, Online. Association for Computational Linguistics.
- Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34(1):177–210.
- David M. Blei and Pedro J. Moreno. 2001. [Topic segmentation with an aspect hidden markov model](#). In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’01, page 343–348, New York, NY, USA. Association for Computing Machinery.
- Freddy Y. Y. Choi. 2000. [Advances in domain independent linear text segmentation](#). In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.
- Noam Chomsky. 1959. [On certain formal properties of grammars](#). *Information and Control*, 2(2):137–167.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

- Lan Du, Wray Buntine, and Mark Johnson. 2013. [Topic segmentation with a structured topic model](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Atlanta, Georgia. Association for Computational Linguistics.
- Jacob Eisenstein. 2009. [Hierarchical text segmentation from multi-scale lexical cohesion](#). In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 353–361, Boulder, Colorado. Association for Computational Linguistics.
- Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08*, page 334–343, USA. Association for Computational Linguistics.
- Pavlina Fragkou, Vassilios Petridis, and Ath Kehagias. 2004. A dynamic programming algorithm for linear text segmentation. *Journal of Intelligent Information Systems*, 23(2):179–197.
- Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2016. [Unsupervised text segmentation using semantic relatedness graphs](#). In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130, Berlin, Germany. Association for Computational Linguistics.
- Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Comput. Linguist.*, 12(3):175–204.
- Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. [Domain-specific language model pretraining for biomedical natural language processing](#). *ACM Trans. Comput. Healthcare*, 3(1).
- Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64.
- John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. 2001. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65.
- Anna Kazantseva and Stan Szpakowicz. 2011. [Linear text segmentation using affinity propagation](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 284–293, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. [Text segmentation as a supervised learning task](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 469–473, New Orleans, Louisiana. Association for Computational Linguistics.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Martin Lange and Hans Leiß. 2009. To cnf or not to cnf? an efficient yet presentable version of the cyk algorithm. *Informatica Didactica*, 8(2009):1–21.
- K. Lari and S.J. Young. 1990. [The estimation of stochastic context-free grammars using the inside-outside algorithm](#). *Computer Speech Language*, 4(1):35–56.
- Jing Li, Aixun Sun, and Shafiq Joty. 2018. [Segbot: A generic neural text segmentation model with pointer network](#). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4166–4172. International Joint Conferences on Artificial Intelligence Organization.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Kelvin Lo, Yuan Jin, Weicong Tan, Ming Liu, Lan Du, and Wray Buntine. 2021. [Transformer over pre-trained transformer for neural text segmentation with enhanced topic coherence](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3334–3340, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Michal Lukasik, Boris Dachev, Kishore Papineni, and Gonçalo Simões. 2020. [Text segmentation by cross segment attention](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4707–4716, Online. Association for Computational Linguistics.
- Igor Malioutov and Regina Barzilay. 2006. [Minimum cut model for spoken lecture segmentation](#). In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, Sydney, Australia. Association for Computational Linguistics.
- Igor Malioutov, Alex Park, Regina Barzilay, and James Glass. 2007. [Making sense of sound: Unsupervised topic segmentation over acoustic input](#). In *Proceedings of the 45th Annual Meeting of the Association of*

- Computational Linguistics*, pages 504–511, Prague, Czech Republic. Association for Computational Linguistics.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. [Building a large annotated corpus of English: The Penn Treebank](#). *Computational Linguistics*, 19(2):313–330.
- Brian McFee, Oriol Nieto, Morwaread M. Farbood, and Juan Pablo Bello. 2017. [Evaluating hierarchical structure in music annotations](#). *Frontiers in Psychology*, 8.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. [Linguistic regularities in continuous space word representations](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.
- Mandar Mitra, Amit Singhal, and Chris Buckley. 1997. [Automatic text summarization by paragraph extraction](#). In *Intelligent Scalable Text Summarization*.
- Hyo-Jung Oh, Sung-Hyon Myaeng, and Myung-Gil Jang. 2007. Semantic passage segmentation based on sentence topics for question answering. *Inf. Sci.*, 177:3696–3717.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Lev Pevzner and Marti A. Hearst. 2002. [A critique and improvement of an evaluation metric for text segmentation](#). *Computational Linguistics*, 28(1):19–36.
- Matthew Purver, Konrad P. Körding, Thomas L. Griffiths, and Joshua B. Tenenbaum. 2006. [Unsupervised topic modelling for multi-party spoken discourse](#). In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 17–24, Sydney, Australia. Association for Computational Linguistics.
- Itiroo Sakai. 1961. Syntax in universal translation. In *Proceedings of the International Conference on Machine Translation and Applied Language Analysis*.
- Anca Simon, Pascale Sébillot, and Guillaume Gravier. 2015. Hierarchical topic structuring: from dense segmentation to topically focused fragments via burst analysis. In *Recent Advances on Natural Language Processing*.
- Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. [A minimal span-based neural constituency parser](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.
- Masao Utiyama and Hitoshi Isahara. 2001. [A statistical model for domain-independent text segmentation](#). In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 499–506, Toulouse, France. Association for Computational Linguistics.
- Teun A Van Dijk. 1982. Episodes as units of discourse analysis. *Analyzing discourse: Text and talk*, pages 177–195.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2022. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20*, Red Hook, NY, USA. Curran Associates Inc.
- Naiwen Xue, Fei Xia, Fu-dong Chiou, and Marta Palmer. 2005. [The penn chinese treebank: Phrase structure annotation of a large corpus](#). *Nat. Lang. Eng.*, 11(2):207–238.
- Yu Zhang, Houquan Zhou, and Zhenghua Li. 2021. Fast and accurate neural crf constituency parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI’20*.

MULTIFIN: A Dataset for Multilingual Financial NLP

Rasmus Kær Jørgensen^{1,2} Oliver Brandt³ Mareike Hartmann^{5,6} Xiang Dai⁴
Christian Igel¹ Desmond Elliott¹

¹Department of Computer Science, University of Copenhagen

²PricewaterhouseCoopers (PwC) ³Independent Researcher ⁴CSIRO Data61

⁵Department of Language Science and Technology, Saarland University

⁶German Research Center for Artificial Intelligence (DFKI)

rasmuskj, xiang.dai, igel, de@di.ku.dk

obrandt2311@gmail.com mareikeh@lst.de

Abstract

Financial information is generated and distributed across the world, resulting in a vast amount of domain-specific multilingual data. Multilingual models adapted to the financial domain would ease deployment when an organization needs to work with multiple languages on a regular basis. For the development and evaluation of such models, there is a need for multilingual financial language processing datasets. We describe MULTIFIN— a publicly available financial dataset consisting of real-world article headlines covering 15 languages across different writing systems and language families. The dataset consists of hierarchical label structure providing two classification tasks: multi-label and multi-class. We develop our annotation schema based on a real-world application and annotate our dataset using both ‘label by native-speaker’ and ‘translate-then-label’ approaches. The evaluation of several popular multilingual models, e.g., mBERT, XLM-R, and mT5, show that although decent accuracy can be achieved in high-resource languages, there is substantial room for improvement in low-resource languages.

1 Introduction

Natural language processing technology has substantially improved in recent years due to the general-purpose Transformer model (Vaswani et al., 2017), large-scale self-supervised training from unlabelled corpora (Devlin et al., 2019), and the scaling of both of these to increasingly large datasets and models (Raffel et al., 2020). Nevertheless, there are still benefits to having domain-specific models (Gururangan et al., 2020), especially when working with clinical (Dai et al., 2022) or financial text (Araci, 2019).

The domain of financial text is particularly interesting for multilingual NLP, given that it is produced across the world (Lewis et al., 2004; Kær Jørgensen et al., 2021). The text often includes invoices, transactions, accounting data, tax policies, and stock market information, *inter-alia*, and there is an emerging

effort to create monolingual financial BERTs (FinBERTs) to process financial text (Araci, 2019; DeSola et al., 2019; Yang et al., 2020b; Liu et al., 2021). However, the handling of financial text by multinational companies is inherently multilingual, therefore, there is a need for datasets to evaluate how well models can process multilingual financial text.

To this end, we introduce the MULTIFIN dataset, a publicly available financial dataset consisting of real-world financial article headlines in 15 languages (see examples in Table 1). MULTIFIN is annotated with HIGH-LEVEL and LOW-LEVEL topics for multi-class and multi-label classification, respectively. The dataset is intended as a resource for developing multilingual financial language models. It is the first benchmark for evaluating cross-lingual and multilingual performance of financial models across multiple languages, writing systems and language families that reflects the real-world multilingual situation in the financial domain.

We benchmark four large-scale pretrained language models (SentenceBERT, mBERT, XLM-R, and MT5) and find that the benefits of large-scale pretraining also apply to financial text. XLM-R is clearly the best performing model in all of our experiments, however, there is a substantial gap in performance between high- and low-resource languages in MULTIFIN. Moreover, a simple LSTM initialized with FastText word embeddings gives surprisingly competitive performance in several experiments. Overall, we find the financial domain can benefit from multilingual NLP, and future work should focus on domain adaptive efforts and improving models’ capacity to generalize to low-resource languages.

Contributions Our contributions are as follows: (a) We present a multilingual financial dataset based on article titles in multiple languages and annotated with two levels of topics. The dataset is made publicly available at <https://github.com/RasmusKaer/MultiFin>. (b) We evaluate dif-

Example	Lang.	LOW-LEVEL labels	HIGH-LEVEL labels
Encuesta Mundial de CEOs 2019 - Hostelería	SPA	· Board, Strategy & Mgmt. · Retail & Consumers	Business & Management
Amendments to VAT legislation	ENG	· VAT & Customs · Government & Policy	Tax & Accounting
Skatta- og lögfræðisvið	ISL	· Tax	Tax & Accounting
Bestyrelsens rolle i forhold til strategiarbejdet	DAN	· Board, Strategy & Mgmt.	Business & Management
Εισαγωγή στην Ελληνική Φορολογία	GRE	· Tax	Tax & Accounting
「事業再編・再生支援」と「ディール戦略」部門を統合・強化	JPN	· M&A & Valuations, · Board, Strategy & Mgmt.	Finance
Veri Analitiği ve Adli Bilişim Çözümleri	TUR	· Financial Crime · Technology	Government & Controls

Table 1: Examples from the MULTIFIN dataset covering different languages, writing scripts, and combinations of LOW-LEVEL and HIGH-LEVEL labels. See Section 3 for more details on the languages and annotation process.

ferent multilingual models under different setups in conjunction with analysis on the multilingual MULTIFIN to establish baselines for the benchmark. (c) Our analysis identifies a need for further research in minimizing the performance gap between high and low-resource languages, and domain adaptive efforts maybe be a promising direction for narrowing this gap.

2 Existing Datasets for Financial NLP

Financial NLP is an emerging area of NLP. Researchers and practitioners have a keen interest in processing natural language for different downstream tasks in the financial domain, such as text mining in accounting (Loughran and McDonald, 2016), financial transactions (Jørgensen and Igel, 2021), sentiment analysis (Malo et al., 2014), and text classification (Arslan et al., 2021). Also, financial economics research shows that news articles and media can be used to forecast firm performance (Tetlock et al., 2008), predict stock market volatility (Glasserman and Mamaysky, 2019) and predict market return (Tetlock, 2007). Moreover, Qin and Yang (2019) show that textual transcripts in combination with audio recordings of company earnings conference calls can be used to predict stock price volatility.

There is a large variety of downstream NLP tasks in the financial domain. However, most work within the community is carried out in a monolingual English setting, where the focus is on adapting successful generic monolingual models to the financial domain (Araci, 2019; DeSola et al., 2019; Yang et al., 2020b; Liu et al., 2021). Only a little work on multilingual domain-adapted models has been investigated (Kær Jørgensen et al., 2021). Since the finan-

cial environment is indeed multilingual, further progression is conditioned on the availability of multilingual resources to develop new methods for multilingual NLP in the financial domain.

Datasets in the financial domain An extensive literature review identifies the datasets used for financial NLP. We define three criteria for being assigned to the list: (1) the dataset needs to be publicly available and accessible, (2) it needs a clear definition of the task with accompanying annotations (i.e., labels, tags, etc.), and (3) it needs to be peer-reviewed and documented. These criteria are set to ensure the quality of the data resource and proper availability and accessibility. Table 2 presents our findings.

An investigation of the datasets shows that most resources are in English. Table 2 (A) presents an overview of the English evaluation datasets. ANALYSTTONE DATASET (Huang et al., 2014), FINTEXTSEN (Cortis et al., 2017) and FINANCIAL PHRASE BANK (Malo et al., 2014) are among the most popular datasets. Sentiment analysis is the most frequent task for the datasets, followed by classification. Only few non-English and multilingual datasets exist. Table 2 (B) and (C) shows available datasets in other languages than English. There are five multilingual datasets which contain English plus three additional non-English languages. The dataset containing most languages is the trilingual (El-Haj et al., 2022) and (Gaillat et al., 2018). In addition, we found three low-resource monolingual sentiment datasets: Arabic BORSAH (Alshahrani et al., 2018), Greek FNS-2022 SHARED TASK (El-Haj et al., 2022) and the Danish DANFINNEWS (Kær Jørgensen et al., 2021) which is the Danish equivalent to the Financial PhraseBank.

The need for a multilingual financial resource has

(A) Datasets in English		(B) Non-English datasets	lang	
AnalystTone Dataset (Huang et al., 2014)	SA	DanFinNews (Kær Jørgensen et al., 2021)	SA	DAN
FinTextSen (Cortis et al., 2017)	SA	CorpusFR (Jabbari et al., 2020)	NER,RE	FRE
Financial Phrase Bank (Malo et al., 2014)	SA	BORSAH (Alshahrani et al., 2018)	SA	ARA
FiQA Dataset (Maia et al., 2018)	SA,QA			
FinNum-1 (Chen et al., 2018)	Numeral CLS			
(C) Multilingual datasets				
M&A dataset (Yang et al., 2020a)	Deal completeness CLS	ENG-CHI Parallel Fin. Dataset (Turenne et al., 2022)	TC,MT	ENG,CHI
FinNum-2 (Chen et al., 2019a)	Numeral attachment	FNS-2022* Shared Task (El-Haj et al., 2022)	SA	ENG,SPA,GRE
StockSen* (Xing et al., 2020)	SA	SEDAR* (Ghaddar and Langlais, 2020)	MT	ENG,FRE
FinCausal* (Mariko et al., 2020)	RC,RE	FinSBD-2019* (Azzi et al., 2019)	SBD	ENG,FRE
MultiLing2019 (El-Haj, 2019)	Summarization	SIXX-Corpora* (Gaillat et al., 2018)	SA	ENG,SPA,GER
FIN5 & FIN3 (Salinas Alvarado et al., 2015)	NER			
Stock-event (Lee et al., 2014)	Stock Price Prediction			
(D) Our dataset				
News-sample OMX Helsinki* (Malo et al., 2013)	SA	MULTIFIN (this paper)	TC	ENG,DAN,FIN,GRE,HEB,HUN,ISL,ITA,JPN,NOR,POL,RUS,SPA,SWE,TUR
EarningsCall (Qin and Yang, 2019)	Stock Price Volatility			
Stocknet (Xu and Cohen, 2018)	Stock Movement Prediction			

Table 2: A list of datasets for financial NLP with corresponding task (SA=Sentiment Analysis, NER=Named Entity Recognition, QA=Question Answering, TC=Topic Classification, RC=Relation Classification, RE=Relation Extraction, MT=Machine Translation, SBD=Sentence Boundary Detection, CLS=Classification). Marked (*) refers to datasets where a request is needed or an application for permission needs to be obtained before that dataset is shared.

been highlighted in several studies (Gaillat et al., 2018; Kær Jørgensen et al., 2021; Jabbari et al., 2020) and its lack of multilingual resources is a limitation for further progression. There is also a need for including different language families and low-resources languages into the research landscape to ensure that not only the high-resources languages lay the foundation of research (Alshahrani et al., 2018). This suggests a gap in resources necessary to advance the financial NLP towards a more multilingual scenario that simulate the financial domain’s multilingual environment. Our work, see Table 2 (D), is motivated by creating a gold standard for benchmarking financial models to facilitate work on adapting to multiple languages within a specific domain.

3 The MULTIFIN dataset

The MULTIFIN dataset is a multilingual corpus, consisting of real-world article headlines covering 15 languages. We annotate the corpus using hierarchical label structure, providing two classification tasks: multi-class and multi-label classification.

Data collection The dataset builds on a collection of public articles published on a large accounting firm’s websites. A subset of the archive was made available for this study. The data collection is based on a real-world application deployed in a large accounting firm. The language selection is determined by the company branches that made their data available to us. We build a multilingual dataset from the headlines of the entire subset that the firm made available. The subset of the archive covers published material in 15 languages and comprises around 10K headlines. The distribution of headlines over lan-

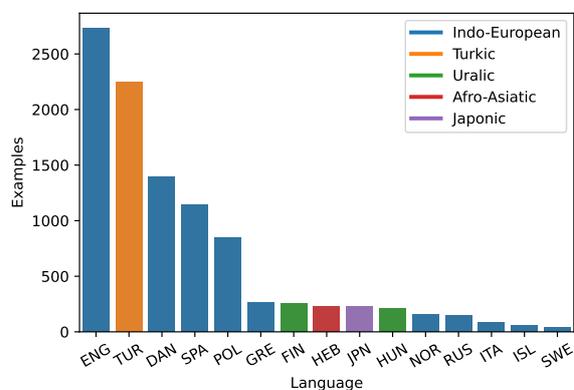


Figure 1: Number of examples per language in MULTIFIN. Bars in the same color indicate these languages belong to the same language family. In this paper, we define languages with more than 500 examples—ENG, TUR, DAN, SPA, POL—high resource languages and the remaining low resource languages.

guages is shown in Figure 1. The publication date is mainly from the period of 2015 to 2021 with some titles having missing dates. The proposed benchmark contains all the languages we were permitted to use, reviewed by experts, which ensures the reliability and quality of both language and content. While the selection of the 15 languages might not be ideal (e.g., African and Indic languages as well as Arabic and Modern Standard Mandarin are missing), we provide the first massively multilingual dataset for financial NLP, see Table 2 for an overview over currently available datasets. It is also worthy noting that headlines, due to their limited context, poses a great challenge for text classification models deployed in the wild (Chen et al., 2019b). See Figure 6 for the text length distribution across different languages.

Annotation Scheme The articles were already tagged with internally pre-defined topics from a company-internal system. Based on these topics, we derive a new, more general label set, referred to **LOW-LEVEL**. Through our label scheme we seek to have different levels of granularity since it gives us the opportunity to go deeper into evaluating the ability of identifying the more refined topics that are presented in titles. Therefore, we first assign fine-grained tags to the topics contain in an headline. For this we use the **LOW-LEVEL** topics. Secondly, we also assign the headline to a single more coarse-grained category, referred to **HIGH-LEVEL**. We defined the **HIGH-LEVEL** topics on the basis of universal categories typically found in news media and more common content categorization. Our fine-grained annotation process results in a dataset with multiple labels per headline. We derive **HIGH-LEVEL** single labels from these multi-label annotations based on either a majority-vote, using the first tag in case of ties. The overview of **LOW-LEVEL** and **HIGH-LEVEL** topics is presented in 3.

HIGH-LEVEL	LOW-LEVEL
Technology	Technology
	IT Security
Industry	Power, Energy & Renewables
	Supply Chain & Transport
	Healthcare & Pharmaceuticals
	Retail & Consumers
	Real Estate & Construction
Tax & Accounting	Media & Entertainment
	VAT & Customs
	Tax
Finance	Accounting & Assurance
	M&A & Valuations
	Asset & Wealth Management
	Actuary, Pension & Insurance
Government & Controls	Banking & Financial Markets
	Government & Policy
	Financial Crime
Business & Management	Governance, Controls & Compliance
	Board, Strategy & Management
	Start-Up, Innovation & Entrepreneurship
	Corporate Responsibility
	SME & Family Business
	Human Resources

Table 3: Overview of **HIGH-LEVEL** and **LOW-LEVEL** topics. The coarse-grained single labels are derived from the fine-grained multi-label annotations based on either a majority-vote, using the first tag in case of ties.

Annotation Process We ask native-level speakers of English and Danish to annotate the dataset using the **LOW-LEVEL** tags. The annotators have domain

expertise and participated on a voluntary basis. Detailed annotation guidelines were presented to the annotators before they started. The description contains definitions of topics including some exemplifications of themes and concepts that may occurs for the topics. As for the annotation of multiple labels, the annotators were asked to label up to three topics per example. The annotated labels needed to be ordered by topic weight, i.e., the first annotated topic is the most dominating topic in the sentence, then the second and third most. The overview and statistics of the label distributions can be found in appendix B.

Translate-then-label evaluation We translated the headlines into English for topic annotation using a translation service¹. We carefully assessed the translation quality to ensure that the translation process does not introduce noise into our dataset. We want to check whether the content of the original sentence is contained in the translation to English. That is, the topics or matters treated in an article stay the same for the translation. For the evaluation, we randomly sample 50 examples from DAN, NOR, ITA, SPA, POL and the entire SWE. We asked evaluators with language proficiency to assess the samples. We presented them with the original sentence, its English translation, and the annotated topics, and ask to answer a true/false question of 1) is the content of the original sentence contained in the English translation, 2) is the property that makes the English sentence fall into this category present in the original sentence as well? The evaluation shows that for DAN, NOR, ITA, SPA, POL and SWE all preserved the properties that make the article fall into a specific category. There was not reported any errors by the evaluators. Thus, we consider translation quality to be high enough to not introduce noise in the process.

Annotator agreement Inter-annotator agreement is measured as multi-label Cohen’s κ (Cohen, 1960). The sample selected for evaluation by both annotators is 1200 examples, randomly sampled across languages and topics. The combined κ of 0.94 suggests a near-perfect agreement. Table 5 depicts the topic-level κ .

Description of dataset The dataset consists of 10,048 headlines in 15 languages annotated with 23 topic labels for **LOW-LEVEL** and 6 **HIGH-LEVEL** topics for multi-class. See Appendix B for details on the distribution of the **LOW-LEVEL** topics and **HIGH-LEVEL**

¹Google Translate, version as of Autumn 2021

topics and Appendix E for an overview of the sentence length distribution across different languages. For multi-class, multi-label classification, we have a total of 14,230 tags across 10,048 headlines (80,678 tokens) using 23 fine-grained topics. For multi-class, single label, we have a coarse-grained topic tag for each headline.

4 Experiments and Results

We employ popular pre-trained multilingual models² and test their effectiveness under different experimental setups. For experimentation, we will only focus on the LOW-LEVEL multi-label task, and HIGH-LEVEL results are reported in the appendix, Table 9.

4.1 Models

mBERT (Devlin et al., 2019) has been pre-trained on Wikipedia articles of 104 languages. Similarly, **XLM-R** (Conneau et al., 2019) was pre-trained on web crawl data, whose size is much larger than Wikipedia data. For both mBERT and XLM-R, we built a classification layer on top of sentence embedding (i.e., the hidden states corresponding to the first [CLS] token). The classification layer consists of a dense layer and tanh activation function, followed by another dense layer, where the output dimension is the total number of possible topics.

SBERT We use multilingual sentence BERT (Reimers and Gurevych, 2020) to map an input sentence to a 768 dimensional dense vector space and then build a classification layer on top of it. Note that we follow Reimers and Gurevych (2019) to keep the weights of SBERT fixed and use SBERT as a feature extractor. We also investigate the variant of fine-tuning SBERT together with the classification layer. The results of fine-tuning approach are very close to feature extraction approach, although the latter involves much smaller number of trainable parameters (110M vs 600K).

mT5 (Xue et al., 2021) was pre-trained on web crawl data covering 101 languages using a ‘text-to-text’ format. That is, consecutive spans of input tokens are replaced with a mask token, and then an encoder-decoder transformer is trained to reconstruct the masked-out tokens. When mT5 is used for downstream classification task, the model outputs the literal text of the label instead of a class index.

²The number of trainable parameters for each model is listed in Table 8 in the Appendix.

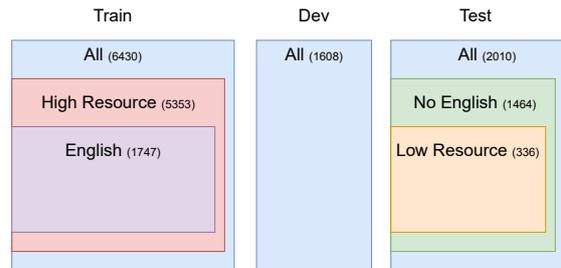


Figure 2: We train models on the complete training set as well as two subsets, to evaluate the multilingual learning and cross-lingual transfer capacities respectively. We use a joint development set of all the languages to select the trained checkpoint. The final model is evaluated on the test and metrics evaluated on the complete test as well as two subsets are reported. Numbers in brackets are the examples belonging to the corresponding (sub)set.

In addition to these transformer-based models, we also experiment with models using pre-trained type-based embeddings described below.

Aligned fasttext embeddings As a baseline, we experiment with models using pre-trained type-based embeddings³, in particular the 300-dimensional fasttext embeddings (Bojanowski et al., 2017) trained on Commoncrawl and Wikipedia data (Grave et al., 2018). In order to enable cross-lingual transfer, we map language-specific fasttext embeddings for all languages covered in our dataset into a common space⁴, using RCSLS (Joulin et al., 2018) as a supervised mapping method. Details about embedding alignment can be found in Appendix C. The mapped embeddings are used as inputs for two baseline models: an LSTM classifier (FASTTEXT_{LSTM}) and a bag-of-embeddings (FASTTEXT_{BAG}) classifier. The LSTM classifier consists of one bidirectional LSTM layers with a classification layer on top, which receives as input a concatenation of the final hidden states of the top-most layer of forward and backward LSTM. The BoE classifier uses the average over all word embeddings in the input sequence as input to the classification layer. For both models, we use the same classification layer as for the mBERT and XLM-R models.

4.2 Experimental setup

To evaluate multilingual learning, we train the model on the complete training set that contains all 15 languages (referred to as ALL). To evaluate cross-lingual

³Fasttext models enable the computation of embeddings for out-of-vocabulary words based on sub-tokens.

⁴We compute pairwise mappings between non-English source embeddings and English target embeddings, and map all non-English embeddings into the space of English embeddings.

Model	Training	Test		
		ALL	NO ENGLISH	LOW RESOURCE
FASTTEXT _{BAG}	ALL	74.2 ± 0.2	71.7 ± 0.2	60.9 ± 0.8
	ENGLISH	41.8 ± 1.5	24.5 ± 1.6	27.9 ± 3.2
	HIGH RESOURCE	70.3 ± 1.1	66.8 ± 1.1	38.2 ± 1.2
FASTTEXT _{LSTM}	ALL	85.4 ± 0.4	83.6 ± 0.4	74.4 ± 0.9
	ENGLISH	51.6 ± 0.5	36.9 ± 0.6	41.9 ± 1.9
	HIGH RESOURCE	82.4 ± 0.6	80.0 ± 0.6	59.5 ± 1.5
sBERT	ALL	73.5 ± 0.2	67.9 ± 0.2	52.0 ± 0.2
	ENGLISH	50.8 ± 0.5	32.7 ± 0.4	27.5 ± 0.6
	HIGH RESOURCE	69.9 ± 0.3	62.8 ± 0.5	27.4 ± 0.2
mBERT	ALL	88.6 ± 0.3	86.5 ± 0.3	77.9 ± 0.5
	ENGLISH	58.3 ± 0.7	43.5 ± 1.0	39.4 ± 2.3
	HIGH RESOURCE	84.1 ± 0.4	80.6 ± 0.4	47.7 ± 0.7
XLM-R	ALL	90.8 ± 0.4	89.4 ± 0.4	83.9 ± 0.6
	ENGLISH	68.0 ± 1.3	59.2 ± 1.6	59.8 ± 1.9
	HIGH RESOURCE	88.6 ± 0.4	86.4 ± 0.5	71.0 ± 1.9
MT5	ALL	81.3 ± 0.1	76.6 ± 0.2	51.0 ± 1.5
	ENGLISH	50.7 ± 1.0	34.3 ± 1.1	25.5 ± 1.9
	HIGH RESOURCE	78.5 ± 0.3	72.9 ± 0.5	33.7 ± 0.2

Table 4: Evaluation results on fine-grained topics (LOW-LEVEL). This is a multi-label classification task with 23 labels, and each example may be assigned up to three topics. All experiments are repeated five times using different random seeds. Averaged Micro F_1 scores and the standard deviations are reported. Best results per column are marked in bold.

transfer, we train the model on (i) a subset that contains only English training data (ENGLISH); and, (ii) a subset that contains 5 high-resource languages (i.e., English, Turkish, Danish, Spanish, Poland) (HIGH RESOURCE).

Model selection In the context of zero-shot cross-lingual transfer, it was shown that performance on a source language (e.g., English) development set does not correlate well with performance in the target language (Keung et al., 2020; Chen and Ritter, 2021). We follow Conneau et al. (2018) and use a joint development set of all the languages. Figure 2 is a high-level illustration of our experimental setup. The trained model which achieves the highest Micro F_1 score on the development set is finally evaluated on the test set. We repeat all experiments five times using different random seeds and mean values and standard deviations are reported.

4.3 Results

Table 4 shows that models trained on the training set consisting of all languages (ALL) achieve slightly better results (2.0-4.5 absolute F_1) than the ones trained on high-resource languages (HIGH RESOURCE) when the trained models are evaluated on the complete test

set. However, this performance gap becomes much larger (11.4-30.2 absolute F_1) when models are evaluated on the subset containing only low-resource languages, which is expected, as the latter setting requires zero-shot transfer when training on HIGH RESOURCE and evaluating on LOW RESOURCE. In the per language analysis (detailed in the following section), we also observe that once the training set contains abundant examples (500+) for these languages, models achieve nearly the same results when evaluated on high-resource languages (Figure 3). Therefore, we focus our discussion on the evaluation results on low-resource languages. The first observation is that different pre-trained multilingual models differ in multilingual learning abilities on our dataset. That is, when they are fine-tuned on ALL, model effectiveness on low-resource languages ranges from 51.0 to 83.9 (A detailed analysis can be found in the following section). The ability of zero-shot cross-lingual transfer is another interesting property of multilingual models. Previous studies show that models trained on English only can achieve impressive results on examples in other languages (Conneau et al., 2018; Hu et al., 2020). However, we observe poor performance when models are trained on ENGLISH

and evaluated on LOW RESOURCE (all under 40 F_1 except XLM-R achieving near 40 F_1). In terms of the choice of source languages, we observe moderate improvements (6.8-11.2 F_1) when massively multilingual pre-trained models (i.e., mBERT, XLM-R, MT5) are cross-lingual transferred from more languages (HIGH RESOURCE: ENG, TUR, DAN, SPA, POL) rather than from ENGLISH only. On the other hand, the improvement becomes much larger (17.6 F_1) when FASTTEXT_{LSTM} is trained on more languages, indicating that the model might make better use of information from additional languages than the transformer-based models. When training on HIGH RESOURCE, FASTTEXT_{LSTM} only slightly underperforms mBERT, and outperforms all other models except XLM-R for transfer from HIGH RESOURCE to LOW RESOURCE. This might be due to the explicit embedding alignment mechanism used in the FASTTEXT approach.

We also calculated the Wilcoxon signed-rank test to assess whether there is a statistically significant difference between the results of XLM-R and mBERT. XLM-R significantly (p -value ≤ 0.05) outperformed mBERT when trained on ALL, ENGLISH, and HIGH RESOURCE and then evaluated on the complete test set. However, the differences for individual languages were not always statistically significant ($p > 0.05$). When both models were trained on ALL, the differences in performances on TUR, NOR, RUS, SWE, ITA, and ISL were not significant; the same holds for the difference on ENG when trained on ENGLISH as well as for the differences on SWE and ISL when trained on HIGH RESOURCE.

5 Analysis and Discussion

Our experiments suggest that although decent accuracy can be achieved for high-resource languages, there is substantial room for improvement in achieving better performance on the multilingual financial dataset. In this section, we present a detailed analysis of the results and investigate some of the findings to identify possible modelling improvements and look into the different dimensions of our dataset.

5.1 Multilingual abilities from a language-level perspective

Multilingual models should ideally learn good representations for all languages they were pre-trained on but this is difficult to achieve in practice due to the “curse of multilinguality” (Conneau et al., 2019). Figure 3 presents per-language results for the three training settings ALL, ENGLISH, and HIGH

RESOURCE. Generally, we see that XLM-R outperforms the rest of the models across all test settings and languages. When training on ALL data (first block in Figure 3), although the models have seen all languages during training, MT5 and sBERT seem to be struggling particularly with GRE, JPN, HEB and HUN. We see a drop in performance between high (upper part of the column) and low-resource languages (bottom part of the column), which is expected as the low-resource languages have less examples in the training dataset. When training on HIGH RESOURCE (last block in Figure 3), we observe that performance for the high-resource languages seen during training is stable compared to training on ALL (indicating that including low-resource languages during fine-tuning does not hurt performance on high-resource languages), but performance for zero-shot transfer to low-resource languages drops significantly. We compare the performance drops suffered on low resource languages from training on ALL data to training on HIGH RESOURCE data between XLM-R, mBERT, and FASTTEXT_{LSTM}, and find that mBERT suffers from larger performance drops than the other models for most languages, with the largest drops for GRE and HEB. XLM-R shows the smallest performance drops for most languages, indicating that it has better zero-shot transfer abilities than the other models.

Next, we analyze the best source for zero-shot transfer by comparing the performance on low-resource languages for models trained on HIGH RESOURCE data with models trained on ENGLISH data. In all cases (except XLM-R on SWE), zero-shot transfer works better when more languages are included in the training set. This might be due to the fact that training on more languages allows models to learn more robust representations of input sequences. Another factor might be that, as our dataset has a large label space, including more training examples (regardless of language) can improve learning representations of otherwise sparse classes. As indicated by the averaged results reported in the previous section, for most languages (except FIN and ISL), FASTTEXT_{LSTM} shows higher improvements when including more languages to train on.

Comparing zero-shot performance on different target languages for models trained on ENGLISH (middle block in Figure 3) reveals that all models with a slight exception to XLM-R struggle to generalize to languages not seen during fine-tuning, although they were part of the pre-training languages. Previous research on mBERT suggests a correlation be-

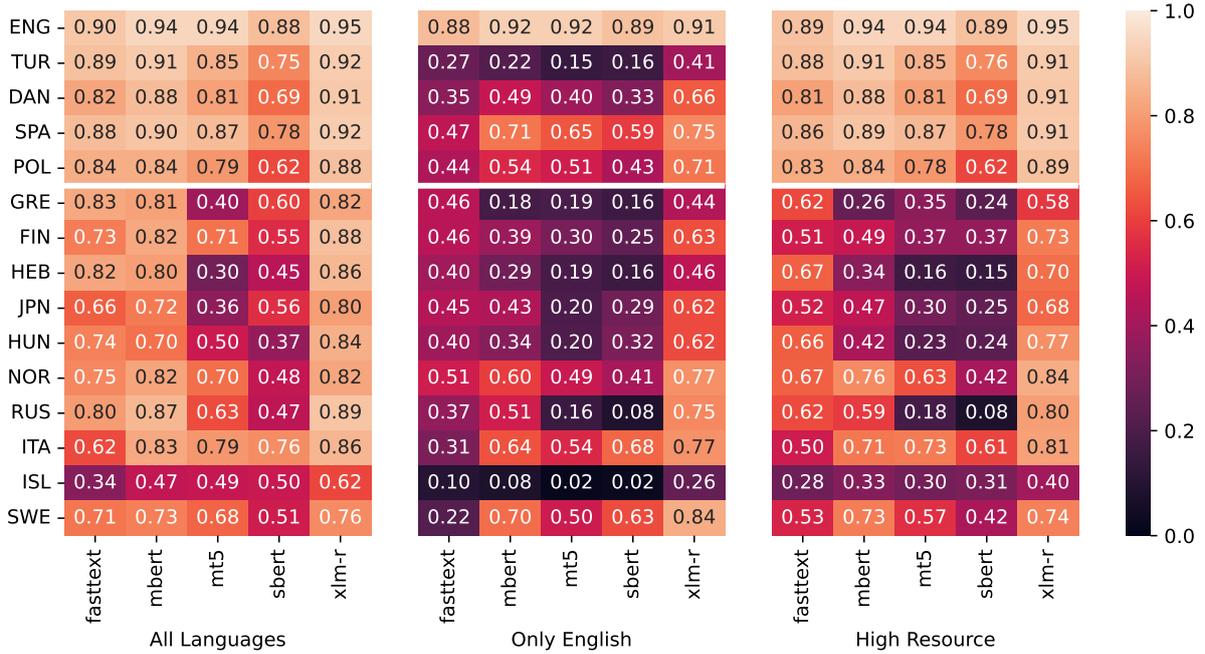


Figure 3: Per language analysis with the multi-label, Low-LEVEL setting. We train on the three settings: ALL, ENGLISH, and HIGH RESOURCE and test on ALL. The first column in each block refers to FASTTEXT_{LSTM}. Languages are in descending order by the number of examples in MULTIFIN, with a white separator between high and low-resource languages.

tween zero-shot performance in a downstream task and amount of in-language pre-training data (Wu and Dredze, 2020; Lauscher et al., 2020), which we also observe in our results. Overall, we see very poor generalization ability to certain low-resource languages, such as ISL, GRE, HEB, and RUS. Particularly for ISL, transfer ability from ENGLISH is nearly non-existing, indicating a need for multilingual models with better transfer abilities to low-resource languages.

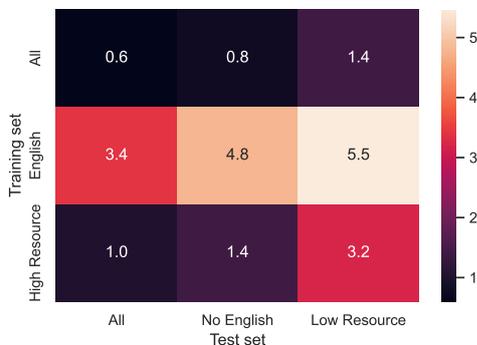


Figure 4: The improvement over the vanilla mBERT, in Micro F_1 , due to domain-adaptive pre-training mBERT. We compare the model by Kær Jørgensen et al. (2021) against the vanilla mBERT.

5.2 Domain-adaptive pre-training can boost the cross-lingual performance

Domain-adaptive pre-training has been shown to improve the model effectiveness when these models are employed to process domain-specific text (Gururangan et al., 2020). We evaluate the publicly available model by Kær Jørgensen et al. (2021), which continues pre-training mBERT on the combination of multilingual financial text and Wikipedia, and measure the improvement over the vanilla mBERT in Table 4. Note that the multilingual pre-training data in (Kær Jørgensen et al., 2021) cover 9 languages in MULTIFIN, except POL, GRE, FIN, HEB, HUN, and ISL. Nevertheless, results in Figure 4 show that domain-adaptive pre-trained models outperform vanilla mBERT in all experimental setups, and larger improvements are observed when training set and test set are disjoint, for example, when models are trained on English or high-resource languages and tested on low-resource languages.

5.3 Multilingual versus translate

We assessed that the translation quality was good enough to preserve the topics in Section 3. Therefore, we translate all training and test data to English and fine-tune a monolingual model for English (RoBERTa, Liu et al. (2019)) on the translated training data. We compare performance on the translated

test sets with XLM-R trained and tested on the multilingual data.

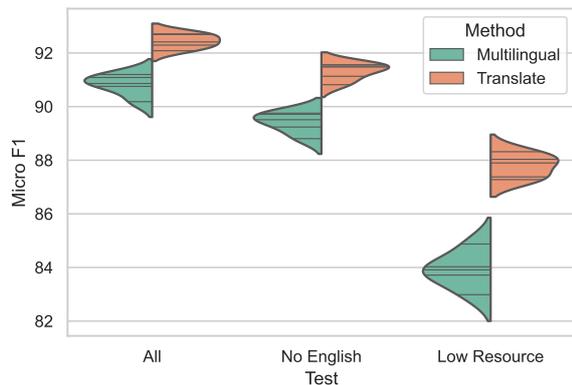


Figure 5: Multilingual (i.e., XLM-R) against translate approach based on English RoBERTa. We use the same setting as in Table 4, where we train on all languages and test on ALL LANG., NOENGLISH and LOWRES.

The monolingual model’s advantage of language-specificity over multilingual models (Rust et al., 2021; Rönnqvist et al., 2019) is evident in Figure 5, where the monolingual model trained on English is slightly better than the multilingual model trained on multilingual data.⁵ We consider this monolingual model an additional baseline on MULTIFIN.

6 Conclusion

We proposed MULTIFIN, a dataset for the evaluation of multilingual financial NLP models. The main aim is to advance multilingual NLP in the financial domain so it is better suited for new development and evaluation of domain-specific models. MULTIFIN is a diverse dataset with 10,000 examples, covering 15 languages, including different language families and writing systems. We benchmark a collection of standard multilingual language models on MULTIFIN and find that although these models often achieve good performance in high-resource languages, there is a substantial gap in performance between high- and lower-resource languages. The per-language analysis uncovered that most of the benchmarked models do not facilitate a good transfer across the evaluated languages, and for specific languages, indicate a strong need for improving the models’ capacity

⁵ Artetxe et al. (2020) found that improvements of a translation baseline in a cross-lingual NLI task do not stem from overcoming the cross-lingual gap, but from the fact that translation of the training data introduces alterations which improve generalization to a translated test set. It is possible that in our experiments, the performance of the monolingual model generalizing from translated training data to translated test data is impacted by similar mechanisms.

to generalize. The multilingual mDAPT model presented overall better generalization, particularly to low-resource languages, indicating that focusing on multilingual domain-specific methods is a promising direction for future work in financial NLP. Future work includes extending the dataset to include more examples across more languages so better understand the limits of multilingual financial text processing. We are also exploring including the entire document, as opposed to only the headline, but this would depend on high-quality long document processing models (Dai et al., 2022). We hope to motivate and inspire collective work on multilingual NLP in the financial domain.

Limitations

Annotators We are aware that annotators with domain knowledge and language proficiency would be preferred. It was not within our resources to find qualified annotators in the financial domain with expert knowledge and language proficiency for all 15 languages.

Annotation process The number of annotated topics per example is determined to three, although a handful of article titles could potentially be assigned more than three topics. The authors attempted to limit this by prioritizing annotated topics by topic weight (see Section 3).

Acknowledgements

We thank PwC for providing the data and thank Lars Silberg Hansen for his support and valuable contribution to the creation of this dataset.

References

- Mohammed Alshahrani, Fuxi Zhu, Mohammed Alghaili, Eshrag Refaee, and Mervat Bamiah. 2018. Borsah: An arabic sentiment financial tweets corpus. In *FNP 2018 —Proceedings of the 1st Financial Narrative Processing Workshop@ LREC*, pages 17–22.
- Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.
- Yusuf Arslan, Kevin Allix, Lisa Veiber, Cedric Lothritz, Tegawendé F. Bissyandé, Jacques Klein, and Anne Goujon. 2021. A comparison of pre-trained language models for multi-class text classification in the financial domain. In *Companion Proceedings of the Web Conference 2021, WWW ’21*, page 260–268, New York, NY, USA. Association for Computing Machinery.

- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020. [Translation artifacts in cross-lingual transfer learning](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online. Association for Computational Linguistics.
- Abderrahim Ait Azzi, Houda Bouamor, and Sira Ferradans. 2019. [The FinSBD-2019 shared task: Sentence boundary detection in PDF noisy text in the financial domain](#). In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pages 74–80, Macao, China.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching word vectors with subword information](#). *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2019a. [Numeral attachment with auxiliary tasks](#). In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1161–1164.
- Chung-Chi Chen, Hen-Hsen Huang, Yow-Ting Shiue, and Hsin-Hsi Chen. 2018. [Numeral understanding in financial tweets for fine-grained crowd-based forecasting](#). In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 136–143. IEEE.
- Jindong Chen, Yizhou Hu, Jingping Liu, Yanghua Xiao, and Haiyun Jiang. 2019b. [Deep short text classification with knowledge powered attention](#). In *AAAI*.
- Yang Chen and Alan Ritter. 2021. [Model selection for cross-lingual transfer](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5675–5687, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Unsupervised cross-lingual representation learning at scale](#). *arXiv preprint arXiv:1911.02116*.
- Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. [Word translation without parallel data](#). *arXiv preprint arXiv:1710.04087*.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. [XNLI: Evaluating Cross-lingual Sentence Representations](#). In *EMNLP*.
- Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. [Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news](#). Association for Computational Linguistics (ACL).
- Xiang Dai, Ilias Chalkidis, Sune Darkner, and Desmond Elliott. 2022. [Revisiting transformer-based models for long document classification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7212–7230, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Vinicio DeSola, Kevin Hanna, and Pri Nonis. 2019. [Finbert: Pre-trained model on sec filings for financial natural language tasks](#). *University of California*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Mahmoud El-Haj. 2019. [MultiLing 2019: Financial narrative summarisation](#). In *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*, pages 6–10, Varna, Bulgaria. INCOMA Ltd.
- Mahmoud El-Haj, Nadhem ZMANDAR, Paul Rayson, Ahmed AbuRa’ed, Marina Litvak, Nikiforos Pittaras, George Giannakopoulos, Aris Kosmopoulos, Blanca Carbajo-Coronado, and Antonio Moreno-Sandoval. 2022. [The financial narrative summarisation shared task \(fns 2022\)](#). In *Proceedings of the The 4th Financial Narrative Processing Workshop @LREC2022*, pages 52–61, Marseille, France. European Language Resources Association.
- Thomas Gaillat, Manel Zarrouk, André Freitas, and Brian Davis. 2018. [The SSIX corpora: Three gold standard corpora for sentiment analysis in English, Spanish and German financial microblogs](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Abbas Ghaddar and Phillippe Langlais. 2020. [SEDAR: a large scale French-English financial domain parallel corpus](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3595–3602, Marseille, France. European Language Resources Association.
- Paul Glasserman and Harry Mamaysky. 2019. [Does unusual news forecast market stress?](#) *Journal of Financial and Quantitative Analysis*, 54:1–38.
- Edouard Grave, Piotr Bojanowski, Prakhara Gupta, Armand Joulin, and Tomas Mikolov. 2018. [Learning word vectors for 157 languages](#). In *Proceedings of the Eleventh International Conference on*

- Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don't stop pretraining: Adapt language models to domains and tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. [XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation](#). In *ICML*.
- Allen H Huang, Amy Y Zang, and Rong Zheng. 2014. Evidence on the information content of text in analyst reports. *The Accounting Review*, 89(6):2151–2180.
- Ali Jabbari, Olivier Sauvage, Hamada Zeine, and Hamza Chergui. 2020. [A French corpus and annotation schema for named entity recognition and relation extraction of financial news](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2293–2299, Marseille, France. European Language Resources Association.
- Rasmus Kær Jørgensen and Christian Igel. 2021. [Machine learning for financial transaction classification across companies using character-level word embeddings of text fields](#). *Intelligent Systems in Accounting, Finance and Management*, 28(3):159–172.
- Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. [Loss in translation: Learning bilingual word mapping with a retrieval criterion](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.
- Rasmus Kær Jørgensen, Mareike Hartmann, Xiang Dai, and Desmond Elliott. 2021. [mDAPT: Multilingual domain adaptive pretraining in a single model](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3404–3418, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- David Kamholz, Jonathan Pool, and Susan Colowick. 2014. [PanLex: Building a resource for panlingual lexical translation](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3145–3150, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Phillip Keung, Yichao Lu, Julian Salazar, and Vikas Bhardwaj. 2020. [Don't use English dev: On the zero-shot cross-lingual evaluation of contextual embeddings](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 549–554, Online. Association for Computational Linguistics.
- Philipp Koehn. 2005. [Europarl: A parallel corpus for statistical machine translation](#). In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. [From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.
- Heeyoung Lee, Mihai Surdeanu, Bill MacCartney, and Dan Jurafsky. 2014. [On the importance of text analysis for stock price prediction](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1170–1175, Reykjavik, Iceland. European Language Resources Association (ELRA).
- David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021. [Finbert: A pre-trained financial language representation model for financial text mining](#). In *Proceedings of the Twenty-Ninth International Conference on Artificial Intelligence*, pages 4513–4519.
- Tim Loughran and Bill McDonald. 2016. [Textual analysis in accounting and finance: A survey](#). *Journal of Accounting Research*, 54(4):1187–1230.
- Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. [Www'18 open challenge: financial opinion mining and question answering](#). In *Companion Proceedings of the The Web Conference 2018*, pages 1941–1942.
- Pekka Malo, Ankur Sinha, Pyry Takala, Oskar Ahlgren, and Iivari Lappalainen. 2013. [Learning the roles of directional expressions and domain concepts in financial news analysis](#).
- Pekka Malo, Ankur Sinha, Pyry Takala, Pekka Korhonen, and Jyrki Wallenius. 2014. [Good debt or bad debt: Detecting semantic orientations in economic texts](#). *Journal of the American Society for Information Science and Technology*.

- Dominique Mariko, Hanna Abi-Akl, Estelle Labidurie, Stephane Durfort, Hugues De Mazancourt, and Mahmoud El-Haj. 2020. [The financial document causality detection shared task \(FinCausal 2020\)](#). In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 23–32, Barcelona, Spain (Online). COLING.
- Yu Qin and Yi Yang. 2019. [What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 390–401, Florence, Italy. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks](#). In *EMNLP-IJCNLP*.
- Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.
- Samuel Rönnqvist, Jenna Kanerva, Tapio Salakoski, and Filip Ginter. 2019. [Is multilingual BERT fluent in language generation?](#) In *Proceedings of the First NLP Workshop on Deep Learning for Natural Language Processing*, pages 29–36, Turku, Finland. Linköping University Electronic Press.
- Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. [How good is your tokenizer? on the monolingual performance of multilingual language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.
- Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. 2015. [Domain adaption of named entity recognition to support credit risk assessment](#). In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, Parramatta, Australia.
- Paul C Tetlock. 2007. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3):1139–1168.
- Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. 2008. More than words: Quantifying language to measure firms’ fundamentals. *The Journal of Finance*, 63(3):1437–1467.
- Nicolas Turenne, Ziwei Chen, Guitao Fan, Jianlong Li, Yiwen Li, Siyuan Wang, and Jiaqi Zhou. 2022. Mining an english-chinese parallel dataset of financial news. *Journal of Open Humanities Data*, 8.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2019. [Do we really need fully unsupervised cross-lingual embeddings?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4407–4418, Hong Kong, China. Association for Computational Linguistics.
- Shijie Wu and Mark Dredze. 2020. [Are all languages created equal in multilingual BERT?](#) In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.
- Frank Xing, Lorenzo Malandri, Yue Zhang, and Erik Cambria. 2020. [Financial sentiment analysis: An investigation into common mistakes and silver bullets](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 978–987, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Yumo Xu and Shay B. Cohen. 2018. [Stock movement prediction from tweets and historical prices](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1979, Melbourne, Australia. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Linyi Yang, Eoin M Kenny, Tin Lok James Ng, Yi Yang, Barry Smyth, and Ruihai Dong. 2020a. [Generating plausible counterfactual explanations for deep transformers in financial text classification](#). *arXiv preprint arXiv:2010.12512*.
- Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020b. [Finbert: A pretrained language model for financial communications](#). *arXiv preprint arXiv:2006.08097*.

A Annotator agreement

The Table 5 below presents the annotator agreement on topic level. The rather high agreement across topics indicate that our annotations are of high quality.

No.	Topic	Kappa, κ
1	Actuary, Pension & Insurance	0.9791
2	Asset & Wealth Management	0.9020
3	Accounting & Assurance	0.9704
4	Banking & Financial Markets	0.9218
5	Board, Strategy & Management	0.9620
6	Power, Energy & Renewables	0.9495
7	Corporate Responsibility	0.9092
8	Media & Entertainment	0.9526
9	Financial Crime	0.9479
10	Government & Policy	0.8889
11	Healthcare & Pharmaceuticals	0.9408
12	Human Resources	0.9537
13	IT Security	0.9346
14	Governance, Controls & Compliance	0.9121
15	M&A & Valuations	0.9617
16	Real Estate & Construction	0.9254
17	Retail & Consumers	0.9526
18	SME & Family Business	0.8670
19	Start-Up, Innovation & Entrepreneurship	0.9888
20	Supply Chain & Transport	0.9321
21	Tax	0.9474
22	Technology	0.9463
23	VAT & Customs	0.9797

Table 5: Full report of inter-annotation agreement of multi-label Cohen’s κ .

B Label distribution

We present the distribution of the LOW-LEVEL and HIGH-LEVEL topics. In Table 6, we present the distribution over the LOW-LEVEL topics. We allowed up-to 3 annotations per examples for the multi-label annotation. This produced a total of 14230 annotation with 1.4 annotations per example on average. In Table 7, we present the distribution over the HIGH-LEVEL topics.

C Cross-lingual transfer with fasttext embeddings

Preprocessing In order to represent inputs with pre-trained fasttext embeddings, we tokenize our data according to how the fasttext training data was tokenized, using Mecab⁶ for Japanese, and the tokenizer from the Europarl preprocessing tools⁷ (Koehn, 2005) for the other languages.

⁶<https://pypi.org/project/mecab-python3/>

⁷<https://www.statmt.org/europarl/>

No.	Topic	Examples
1	Actuary, Pension & Insurance	502
2	Asset & Wealth Management	257
3	Accounting & Assurance	1,452
4	Banking & Financial Markets	782
5	Board, Strategy & Management	866
6	Power, Energy & Renewables	248
7	Corporate Responsibility	277
8	Media & Entertainment	255
9	Financial Crime	310
10	Government & Policy	528
11	Healthcare & Pharmaceuticals	245
12	Human Resources	1,091
13	IT Security	424
14	Governance, Controls & Compliance	501
15	M&A & Valuations	492
16	Real Estate & Construction	351
17	Retail & Consumers	354
18	SME & Family Business	226
19	Start-Up, Innovation & Entrepreneurship	277
20	Supply Chain & Transport	222
21	Tax	1,713
22	Technology	1,169
23	VAT & Customs	1,688
Total		14,230

Table 6: Overview of LOW-LEVEL tags across the 23 topics. These represent the 23 labels used in the multi-label task.

No.	Topic	Examples
1	Technology	1,088
2	Industry	1,239
3	Tax & Accounting	3,371
4	Finance	1,447
5	Government & Controls	912
6	Business & Management	1,991
Total		10,048

Table 7: Overview of HIGH-LEVEL tags across the 6 classes. These represents the 6 classes used in the multi-class classification task.

Embedding alignment We map monolingual fast-text embeddings trained on Wikipedia and Common-crawl into a shared space using RCSLS, by computing pairwise mappings between source languages and English as a target language. As supervision, we rely on the training dictionaries of the MUSE dataset (Conneau et al., 2017), except for Icelandic which is not covered there. For Icelandic, we follow Vulić et al. (2019) in deriving a dictionary based on the Panlex database (Kamholz et al., 2014): We retrieve translations for the 5000 most frequent Icelandic words derived from Opensubtitles published on Wiktionary⁸ We only keep single-word transla-

⁸https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/Icelandic_

Model	Learning rate	# train epochs	# Params.
FASTTEXT _{BAG}	[1e-3, 2.5e-3, 5e-3, 7.5e-3, 1e-2, 2.5e-2, 5e-2]	50	0.1M
FASTTEXT _{LSTM}	[1e-3, 2.5e-3, 5e-3, 7.5e-3, 1e-2, 2.5e-2, 5e-2]	50	1.8M/1M/1M
sBERT	[1e-2, 3e-2, 1e-1]	[10, 30, 100]	0.6M
mBERT	[1e-5, 2e-5, 5e-5, 1e-4]	[10, 30, 100]	180M
XLM-R	[1e-5, 2e-5, 5e-5, 1e-4]	[10, 30, 100]	270M
MT5	[1e-4, 3e-4, 1e-3]	[10, 30]	300M

Table 8: The search space of two hyperparameters (learning rate and number of training epochs), as well as the number of trainable parameters for each model. The size of the hidden states in FASTTEXT_{LSTM} is treated as an additional hyperparameter selected from [100, 200, 300, 400, 500], hence we report numbers of parameters for three different selected models trained on ALL/ENGLISH/HIGH RESOURCE, corresponding to models with hidden dimensionality 300/200/200, respectively. For all models, we do early stopping on the validation set with a patience of 5 and 10 for transformer-based and fasttext-based models, respectively.

tions. As not all source words are present in Panlex, our final dictionary contains translations for 1,823 Icelandic words. With these dictionaries as supervision, we run RCLS with default parameters for 10 epochs, and select the best mapping based on the unsupervised selection criterion.

D Experimental Details

For each experiment, we perform grid search to find the best combination of two hyperparameters—number of training epochs and learning rates—on the development set. Table 8 shows the search space of these two hyperparameters as well as the trainable parameters per model.

The particular versions of pre-trained multilingual models can be found at:

- sBERT: <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>
- mBERT: <https://huggingface.co/bert-base-multilingual-cased>
- XLM-R: <https://huggingface.co/xlm-roberta-base>
- MT5 <https://huggingface.co/google/mt5-base>

Pre-trained fasttext embeddings can be found at:

- <https://fasttext.cc/docs/en/crawl-vectors.html>

wordlist

E Results of Multi-class classification on HIGH-LEVEL topics

Table 9 show the evaluation results on coarse-grained categories (HIGH-LEVEL), framed as a multi-class classification problem.

F Sentence length distribution

Figure 6 shows the sentence length distribution across languages in the MULTIFIN dataset.

Model	Training	Test		
		ALL	NO ENGLISH	LOW RESOURCE
FASTTEXT _{BAG}	ALL	78.1 ± 0.2	76.7 ± 0.8	70.5 ± 1.4
	ENGLISH	60.0 ± 1.0	52.2 ± 1.1	47.7 ± 1.1
	HIGH RESOURCE	73.6 ± 2.4	71.4 ± 2.1	52.8 ± 1.8
FASTTEXT _{LSTM}	ALL	83.1 ± 0.7	81.3 ± 0.8	75.9 ± 1.2
	ENGLISH	64.1 ± 1.5	55.7 ± 1.9	51.6 ± 2.1
	HIGH RESOURCE	80.4 ± 0.4	77.6 ± 0.5	60.5 ± 1.5
sBERT	ALL	72.4 ± 0.8	66.1 ± 1.0	55.3 ± 1.8
	ENGLISH	51.9 ± 0.5	38.4 ± 0.8	32.3 ± 0.8
	HIGH RESOURCE	72.1 ± 0.6	65.3 ± 0.7	33.0 ± 1.5
mBERT	ALL	87.4 ± 0.4	85.0 ± 0.4	79.1 ± 0.9
	ENGLISH	60.4 ± 2.4	48.4 ± 3.2	48.1 ± 2.2
	HIGH RESOURCE	82.9 ± 0.5	79.0 ± 0.7	52.3 ± 2.0
XLM-R	ALL	89.5 ± 0.4	87.8 ± 0.5	84.0 ± 0.9
	ENGLISH	74.9 ± 2.2	68.5 ± 2.7	67.9 ± 1.0
	HIGH RESOURCE	87.5 ± 0.7	85.3 ± 0.8	74.7 ± 1.0
MT5	ALL	83.6 ± 0.4	79.7 ± 0.5	61.3 ± 1.2
	ENGLISH	56.6 ± 0.7	42.9 ± 0.8	41.5 ± 1.3
	HIGH RESOURCE	81.1 ± 0.0	76.2 ± 0.1	43.9 ± 0.1

Table 9: Evaluation results on coarse-grained categories (HIGH-LEVEL). Results are averaged over five runs and reported by F1 micro. Multi-class classification task with 6 classes, one per example.

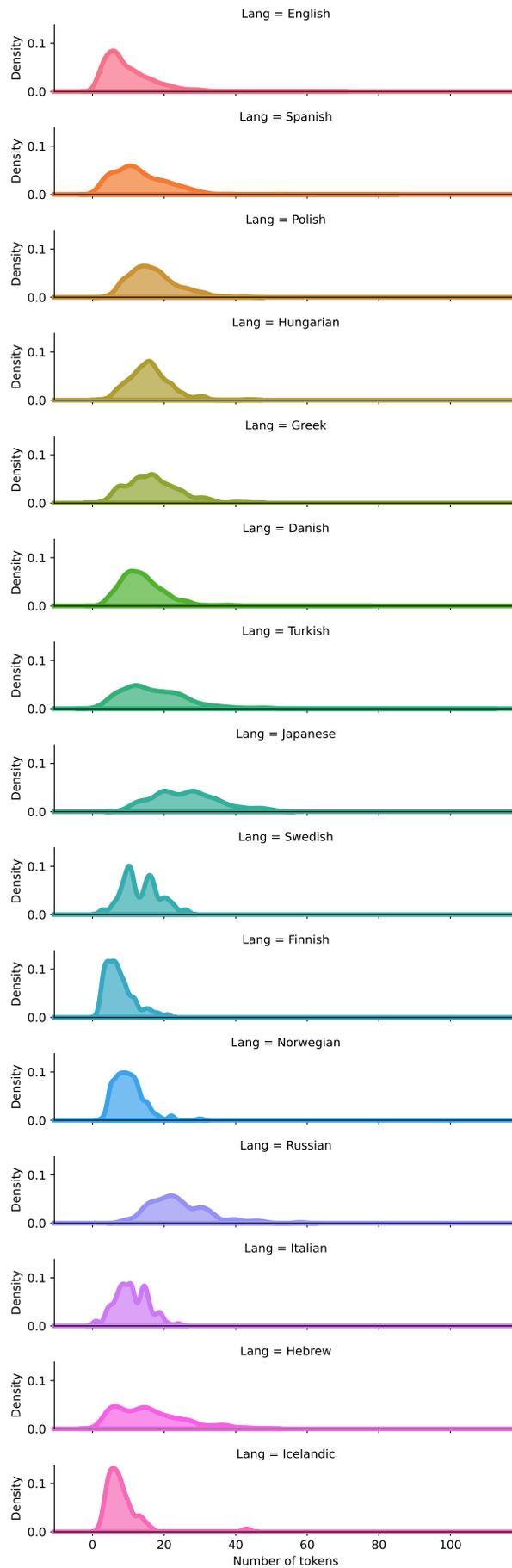


Figure 6: Sentence length distribution across different languages.

MLASK: Multimodal Summarization of Video-based News Articles

Mateusz Krubiński and Pavel Pecina

Charles University, Faculty of Mathematics and Physics

Institute of Formal and Applied Linguistics

{krubinski, pecina}@ufal.mff.cuni.cz

Abstract

In recent years, the pattern of news consumption has been changing. The most popular multimedia news formats are now multimodal – the reader is often presented not only with a textual article but also with a short, vivid video. To draw the attention of the reader, such video-based articles are usually presented as a short textual summary paired with an image thumbnail. In this paper, we introduce MLASK¹ (Multimodal Article Summarization Kit) – a new dataset of video-based news articles paired with a textual summary and a cover picture, all obtained by automatically crawling several news websites. We demonstrate how the proposed dataset can be used to model the task of multimodal summarization by training a Transformer-based neural model. We also examine the effects of pre-training when the usage of generative pre-trained language models helps to improve the model performance, but (additional) pre-training on the simpler task of text summarization yields even better results. Our experiments suggest that the benefits of pre-training and using additional modalities in the input are not orthogonal.

1 Introduction

Automatic summarization is one of the basic tasks both in Natural Language Processing – text summarization – and in Computer Vision – video summarization. Multimodal summarization (MMS) builds a bridge between those two fields.

Early works on multimodal summarization explored the usage of the secondary modality as an auxiliary source of information to guide the refinement process of the main modality. Li et al. (2017) collected videos and news articles covering a hand-crafted list of recent significant world events by querying a web search engine and trained a model to mimic the reference summaries written by human annotators. Zhu et al. (2018) were the first to

¹<https://github.com/ufal/MLASK>

introduce the task of Multimodal Summarization with Multimodal Output (MSMO). They collected a large-scale dataset of news articles paired with corresponding images and trained a system to generate a textual summary and choose a single image as a pictorial summary. By introducing the multimodal output, a uni-modal solution was no longer sufficient as a baseline. Building upon this, Li et al. (2020b) extended the task to video-based MSMO. Based on a textual document and a short video clip, besides generating the textual summary, the system was also challenged to select a single frame from the video as a cover picture.

We believe there is still a lot of questions that remain unanswered, e.g.: *How to evaluate multimodal outputs?* or *How to approach pre-training?* In this paper, we contribute to the area of video-based MSMO (VMSMO) by: **1)** introducing a full-scale VMSMO dataset in Czech, extending the very limited available resources for this task to a new language; **2)** exploring the pre-training strategies by transferring knowledge from the simpler task of text-to-text summarization; **3)** re-defining the training labels to consider intra-video similarities; **4)** proposing a human evaluation framework for assessing the quality of VMSMO.

2 Related Work

In our work, we build upon recent advances in three fields: text summarization, video summarization, and multimodal summarization.

2.1 Text Summarization

Text summarization aims to automatically produce a short fluent summary that preserves the crucial information from the source document(s). Historically, a majority of works focused on the news domain and English language (Nallapati et al., 2016; Grusky et al., 2018; Fabbri et al., 2019). Recently, new research directions, such as multilingual summarization (Scialom et al., 2020; Varab

and Schluter, 2021) or dialogue summarization, (Gliwa et al., 2019; Zhong et al., 2021) have been explored. Fabbri et al. (2021) benchmarked over 20 recent summarization models and concluded that the abstractive summaries produced by pre-trained generative languages models fine-tuned on summarization datasets (Zhang et al., 2020; Lewis et al., 2020) consistently performed best with regards to both automatic metrics and human evaluation.

2.2 Video Summarization

Video summarization aims to refine the video content by either choosing a set of the most representative frames, known as a video storyboard, or selecting short video fragments, known as a video skim. As noted in the recent survey (Apostolidis et al., 2021), in both cases, the usual approach is to start with modeling the frame-level importance scores, which can then be aggregated to segment-level scores.

Contrary to text summarization, abstractive approaches that generate the summary from scratch are yet to be explored. The most relevant to our work are the recent publications on query-based video summarization, e.g., Li et al. (2023) and Huang et al. (2021), that use a text-based input to enrich frame-level representations and guide the summarization towards a user-specified query.

2.3 Multimodal Summarization

Previous works (e.g., Li et al., 2017, 2018; Palaskar et al., 2019) explored the addition of multimodal information such as video or audio transcript to enrich the textual document, aiming to generate better textual summaries. Zhu et al. (2018), who introduced the MSMO task, trained a model that jointly generated text and selected the most relevant image from a pre-defined set of images. Li et al. (2020b) and Fu et al. (2021) were the first to tackle the VMSMO problem. In their work, the cover picture choice was modeled as a frame selection problem. In the follow-up work (Tang et al., 2022), a video-article pair was summarized as a single frame and a one-sentence summary using an optimal transport-based unsupervised training strategy.

3 MLASK Dataset

Previous works on MMS operated on datasets in either English (Li et al., 2017, 2018; Palaskar et al., 2019; Fu et al., 2021; Tang et al., 2022) or Chinese

	Mean	Q_1	Median	Q_3
Title	11.16 ± 2.78	9	11	13
Abstract	33.40 ± 13.86	22	32	43
Article	276.96 ± 191.74	154	231	343

Table 1: Quantitative statistics of the lengths of titles, abstracts, and full texts (measured in the number of tokens) for the MLASK dataset. Q_1 and Q_3 denote the first and the third quartile, respectively.

(Li et al., 2020b; Li et al., 2020a). To extend the available resources, we collected a new dataset in a different language – Czech, a West Slavic language with a rich system of morphology and a relatively flexible word order.

3.1 Data Preparation

The steps taken while preparing the dataset are:

1. Two Czech websites publishing news articles accompanied with a video clip, textual summary, and a cover picture were identified.
2. Based on the HTML structure of each website, the articles accompanied by a video clip (mp4) and a cover picture (jpeg) were downloaded.
3. From each relevant article, its title, abstract, and full text were extracted.
4. The following documents were dropped:
 - with videos longer than 5 minutes;
 - with full text shorter than 50 words or longer than 2,000 words;
 - with abstract shorter than 10 words or longer than 80 words;
 - with title shorter than 2 words;
 - with either the full text or abstract identified as non-Czech by the langid² language-identifier.
5. Every video was re-sampled to the same frame rate (25 fps) and resized to the same resolution (1280x720).

3.2 Dataset Size Statistics

In total, the collected dataset contains 41,243 instances, all including the article’s text, title, abstract, video, and cover picture. The quantitative statistics of the data are displayed in Table 1. The average video duration is 85.58 seconds. For comparison, we also report the statistics of other datasets proposed for the VMSMO task so far (Table 2).

²<https://github.com/saffsd/langid.py>

Dataset	#Articles	Article Length	Summary Length	Video Length	Language
VMSMO (Li et al., 2020b)	184,920	97	11	60s	Chinese
MM-AVS (Fu et al., 2021)	2,173	685	57	109s	English
XMSMO-News (Tang et al., 2022)	4,891	102	12	346s	English
MLASK (this paper)	41,243	277	33	86s	Czech

Table 2: Comparison of the datasets introduced for the VMSMO task. The concrete statistics are reported as averages computed over the whole corpus. For the textual part, we report the average number of tokens.

4 Multimodal Summarization

In our experiments, a video-based news article is represented by a pair (V, X) . V corresponds to the video input – a sequence of frames: $V = (v_1, v_2, \dots, v_N)$. X is the news article presented as a sequence of tokens: $X = (x_1, x_2, \dots, x_M)$. We assume that for each article, there is a ground-truth textual summary $Y = (y_1, y_2, \dots, y_L)$ and a ground-truth cover picture P . The task is to generate a textual summary \hat{Y} that includes the main points of the article and to choose a frame \hat{v} to act as a cover picture (pictorial summary).

4.1 Overview

The proposed MMS model (see Figure 1) is structured into three parts: *Feature Encoder* composed of a text, video, and frame encoder, *Cross-modal Interaction Module* fusing the visual and textual representations, and *Multimodal Decoder* responsible for the summary generation and frame selection.

4.2 Feature Encoder

The Feature Encoder consists of a text encoder, video encoder, and frame encoder:

Text Encoder. We use the Transformer (Vaswani et al., 2017) encoder model to map the textual news article into the sequence of token embeddings (Eq. 1). Following the findings of Yu et al. (2021), we use the pre-trained mT5 model (Xue et al., 2021) to initialize its weights. We examine the influence of task-specific pre-training (Section 6.3) by fine-tuning the mT5 model on the simpler task of text-to-text summarization.

$$X_{enc} = \text{TransformerEncoder}(X) \quad (1)$$

Video Encoder. The news videos in our dataset are several minutes long and consist of hundreds of frames. To incorporate the short-term temporal dependencies, we employ the 3D convolutional networks. In our experiments, we segment the video into non-overlapping sequences of frames

and use the 3D CNN network for feature extraction (Eq. 2). As the feature extractors, we use the R(2+1)D model trained by Ghadiyaram et al. (2019) for video action recognition on weakly-supervised social-media videos and the visual component of the S3D Text-Video model trained in a self-supervised manner by Miech et al. (2020) on the HowTo100M dataset (Miech et al., 2019). To incorporate the long-term temporal dependencies, we process the sequence of video features with the Transformer encoder model (Eq. 3).

$$V_{enc} = \text{3D-CNN}(V) \quad (2)$$

$$V_{enc} = \text{TransformerEncoder}(V_{enc}) \quad (3)$$

Frame Encoder. To be able to choose a specific frame as a cover picture, frame-level representations are needed. In our experiments, we sample one of every 25 frames as the cover picture candidates (1 frame per second). We examine the usage of EfficientNet (Tan and Le, 2019) and Vision Transformer (Dosovitskiy et al., 2021) as feature extractors. Both were trained for image classification on ImageNet (Russakovsky et al., 2015). To put the representations into context, we process the sequence of frame features with the Transformer encoder model (Eq. 5).

$$V_{frame} = \text{CNN}(\text{Sample}(V)) \quad (4)$$

$$V_{frame} = \text{TransformerEncoder}(V_{frame}) \quad (5)$$

Before applying the Transformer encoder, we project both the video and frame features into the same dimension as the hidden states of the text encoder. When used in a single model, the two sets of features are concatenated before projecting.

4.3 Interaction Module

Following Yu et al. (2021), who examined different ways of injecting visual information into pre-trained generative language models, we employ the multi-head attention (MHA) based fusion to obtain the vision-guided text representation and perform the fusion after the last encoder layer (Eq. 6–9).

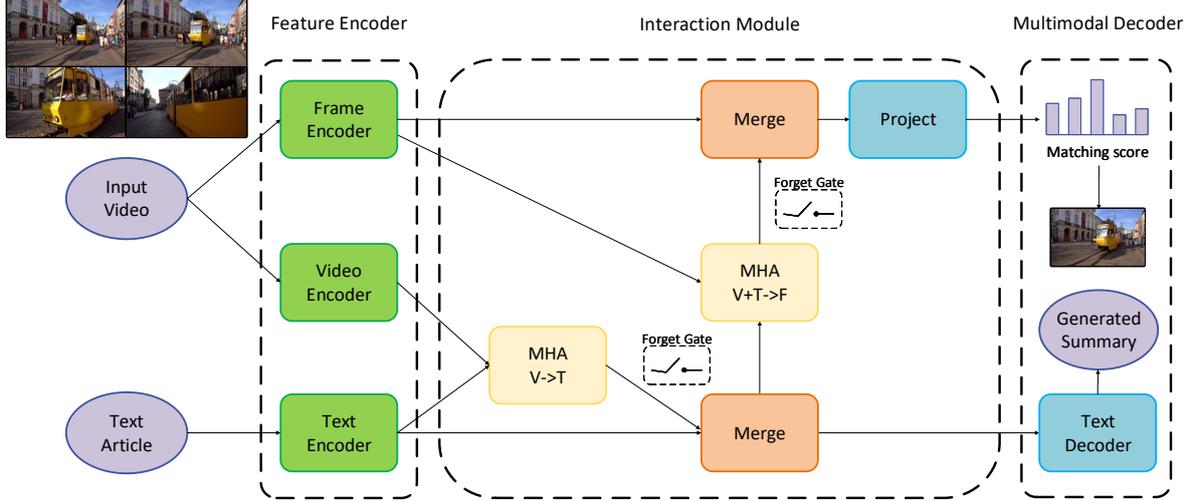


Figure 1: An overview of the proposed MMS model for multimodal summarization.

$$Q = X_{enc}W_q, Q \in \mathbb{R}^{M \times d} \quad (6)$$

$$K = V_{enc}W_k, K \in \mathbb{R}^{N' \times d} \quad (7)$$

$$V = V_{enc}W_v, V \in \mathbb{R}^{N' \times d} \quad (8)$$

$$\tilde{X}_{enc} = \text{MHA}(Q, K, V), \tilde{X}_{enc} \in \mathbb{R}^{M \times d} \quad (9)$$

As suggested by Liu et al. (2020), we use the forget gate (FG) mechanism so that the model can filter out low-level cross-modal adaptation information (Eq. 10).

$$\hat{X}_{enc} = \text{FG}(X_{enc}, \tilde{X}_{enc}), \hat{X}_{enc} \in \mathbb{R}^{M \times d} \quad (10)$$

We use the same MHA mechanism to obtain the text+video guided frame representations \hat{V}_{frame} by substituting the X_{enc} with V_{frame} in Eq. 6 and V_{enc} with \hat{X}_{enc} in Eq. 7 and Eq. 8.

4.4 Multimodal Decoder

To generate the textual summary, we use the standard Transformer decoder initializing its weights from the mT5 checkpoint. We use the vision-guided text representation \hat{X}_{enc} as the input (Eq. 11) and train it using the standard negative log-likelihood loss (NLLLoss) w.r.t. the target sequence Y (Eq. 12).

$$\hat{Y} = \text{TransformerDecoder}(\hat{X}_{enc}) \quad (11)$$

$$\mathcal{L}_{text} = \text{NLLLoss}(\hat{Y}, Y) \quad (12)$$

To obtain the labels C for cover picture (cover frame) selection, we compute the cosine similarity between the CNN features of the reference cover

picture and the candidate frames. The similarity of over 99.99% of instances was in the [0,1] range, and the remaining negative values were mapped to 0. The previous works (Li et al., 2020b; Fu et al., 2020) regarded the frame with the maximum cosine similarity as ground-truth and others as negative samples (C_{max}). After examining the cosine similarity patterns (Figure 2), we noticed that the per-video similarity has often either more than one peak, or there are consecutive sequences of frames with very similar scores (capturing a still scene). Our intuition was that this may harm the model performance – very similar frames might be labeled as both positive and negative examples. To overcome this issue, besides the binary labels C_{max} , we introduce the smooth labels C_{smooth} that assign to each frame its cosine similarity score with the reference cover picture.

We use a projection matrix to map the text+video guided frame representations \hat{V}_{frame} to a single dimension (Eq. 13) and train (Eq. 14) using the binary cross-entropy loss (BCELoss). The target labels C are either C_{max} or C_{smooth} . We train the whole model end-to-end by minimizing the sum of losses \mathcal{L} (Eq. 15).

$$\hat{C} = \hat{V}_{frame}W_p, W_p \in \mathbb{R}^{d \times 1} \quad (13)$$

$$\mathcal{L}_{image} = \text{BCELoss}(\hat{C}, C) \quad (14)$$

$$\mathcal{L} = \mathcal{L}_{text} + \mathcal{L}_{image} \quad (15)$$

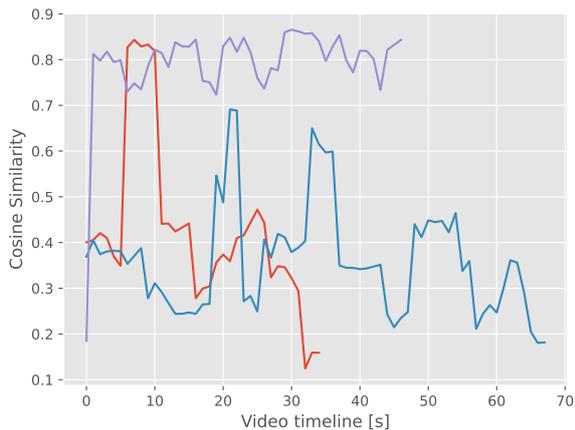


Figure 2: Three examples of cosine similarity plots between CNN features of the reference cover picture and all candidate frames from the video. The examples were chosen manually to present three different video similarity patterns: with a single peak (red), with more than one peak (blue), and with a consecutive sequence of frames having very similar scores (violet).

5 Experiment Setup

5.1 Dataset

In our experiments, we perform the training/dev/test splits of the MLASK dataset following the chronological ordering based on publication date. We use the articles published in the first half (Jan–Jun) of 2021 for validation (2,482 instances) and the ones published in the second half (Jul–Dec) of 2021 and the beginning (Jan–Feb) of 2022 for testing (2,652 instances). The remaining data is used for training (36,109 instances).

5.2 Implementation

We implement our experiments in PyTorch Lightning³ and use the mT5-small variant (300M trainable parameters) provided via the Transformers (Wolf et al., 2020) package. Following Yu et al. (2021), we use two separate 4-layer encoders with 8 attention heads to contextualize the video and frame representations (Eq. 5 and Eq. 3). As video feature extractors, we use the R(2+1)D 34-layer IG-65M⁴ and S3D_HowTo100⁵ models to encode sequences of the length of 32 frames. To extract frame-level features, we utilize the EfficientNet-B4 variant from the torchvision package and the vit-

³<https://github.com/PyTorchLightning/pytorch-lightning>

⁴<https://github.com/moabitcoin/ig65m-pytorch>

⁵https://github.com/antoine77340/S3D_HowTo100M

base-patch32-224-in21k variant of Vision Transformer provided by Hugging Face (Wolf et al., 2020). We follow the suggested pre-processing (e.g., re-scaling) for each feature extractor independently. The total number of trainable parameters is equal to approximately 323M.

5.3 Hyper-parameters

We train the multimodal model using the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We increase the learning rate linearly for the first 8,000 steps (0 to $5e-4$) and then follow an inverse square root decay schedule. Since both the text encoder and decoder are pre-trained, we freeze them for the first 2 epochs. We limit the document size to 1,536 sub-word tokens and the summary length to 256 tokens. We train all the models for 50 epochs with an early stopping applied if ROUGE-L (see Section 5.4) does not improve on the dev-set for 5 consecutive epochs. During decoding, we use the best checkpoint with respect to ROUGE-L, utilizing beam search with the beam size of 4, length penalty of 1.0, and repetition penalty (Keskar et al., 2019) of 2.5. We select the cover frame by applying argmax to the projected representations (Eq. 13). We employ gradient accumulation to train with the effective batch size of 32. Each model is trained on a single GeForce RTX 3090 GPU, and the average training time is roughly 36 hours.

5.4 Evaluation Metrics

Most existing implementations of ROUGE (Lin, 2004), a standard metric used to evaluate summarization, are English-specific and utilize e.g., an English stemmer and stop words. Since our dataset is in Czech, following the work of Straka et al. (2018), we evaluate the model performance with language-agnostic variants of ROUGE⁶ reporting the F1 scores (ROUGE-1, ROUGE-2, ROUGE-L).

To estimate the quality of cover frame selection, we follow Fu et al. (2020) and report the cosine similarity (CosSim) between the reference cover picture and the chosen cover frame. To have a better understanding of the model performance, we also follow Li et al. (2020b) and report Recall@k ($R@k$)⁷ considering the frame closest to the ground-truth as a positive example. To evaluate the

⁶<https://lindat.cz/repository/xmlui/handle/11234/1-2615>

⁷<https://github.com/Lightning-AI/metrics>

DEV	ROUGE-1	ROUGE-2	ROUGE-L	CosSim	R@5	R@10	KC	PC
<i>RandomT</i>	13.92	1.63	9.02	-	-	-	-	-
<i>Lead3</i>	15.47	2.32	10.25	-	-	-	-	-
<i>Oracle</i>	22.92	5.37	18.28	-	-	-	-	-
<i>mT5-MLASK</i>	18.25	4.14	13.07	-	-	-	-	-
<i>mT5-SumeCzech</i>	19.18	4.53	13.76	-	-	-	-	-
<i>RandomV</i>	-	-	-	0.335	0.092	0.182	0.000	0.000
MMS	18.34	4.12	13.26	0.563	0.206	0.339	0.303	0.465
+ Masked Video	17.70	3.84	12.81	0.548	0.191	0.320	0.275	0.439
- IG-65M	17.74	3.89	12.95	0.558	0.200	0.323	0.290	0.456
- S3D	17.82	3.88	12.93	0.530	0.187	0.321	0.260	0.428
- Effnet	18.07	4.04	13.13	0.589	0.160	0.280	0.211	0.328
- ViT	17.69	3.71	12.82	0.527	0.192	0.320	0.309	0.488
+ SumeCzech	19.64	4.95	14.32	0.551	0.192	0.319	0.274	0.440
+ Smooth Labels	19.73	4.97	14.34	0.562	0.202	0.332	0.295	0.458
+ Masked Video	19.74	5.02	14.34	0.561	0.197	0.331	0.290	0.452
TEST								
MMS	18.45	4.29	13.42	0.552	0.183	0.321	0.306	0.447
+ Masked Video	17.65	3.95	12.88	0.542	0.187	0.332	0.283	0.422
- IG-65M	17.81	4.02	13.07	0.548	0.186	0.321	0.296	0.437
- S3D	17.89	4.03	13.03	0.531	0.177	0.316	0.264	0.408
- Effnet	18.21	4.28	13.37	0.582	0.157	0.279	0.216	0.311
- ViT	17.78	3.94	13.00	0.509	0.176	0.311	0.303	0.452
+ SumeCzech	19.58	4.95	14.30	0.541	0.181	0.318	0.278	0.420
+ Smooth Labels	19.74	4.90	14.34	0.551	0.188	0.330	0.299	0.444
+ Masked Video	19.69	4.91	14.38	0.553	0.184	0.326	0.300	0.439

Table 3: Evaluation on the dev-set and test-set of MLASK. See Section 5.4 for the metrics description. The figures are averaged over three runs with different seeds. The three highest-scoring systems in each column are bolded independently for test-set and dev-set.

frame scoring at even coarser video-level granularity, we report Kendall’s Tau (KC) and Pearson (PC) correlation coefficients⁸ to measure the correlation of ordering based on the projected representations (Eq. 13) with the absolute frame ordering based on similarity with the ground-truth picture.

6 Experiments

We analyze several aspects of the proposed model: First, we study the effect of the visual features. Second, we analyze the contribution of pre-training the model on text-only summarization data. Third, we exploit the smooth frame labels to further improve the model. The results are presented in Table 3.

6.1 Baselines

To put our experiments into context, we first report the performance of several text-only baselines: *RandomT* extracts three random sentences from the article and *Lead3* extracts three initial sentences (trivial baselines); *Oracle* takes three sentences that maximize ROUGE-L with the ground-truth

abstract (the upper bound for extractive summarization); *mT5-MLASK* is the output of the mT5 model fine-tuned on the textual part of the MLASK training set and *mT5-SumeCzech* is the mT5 model fine-tuned on the SumeCzech (Section 6.3) dataset (abstractive summarization baselines). There is also a video-only baseline *RandomV*, which performs random frame ordering.

Unsurprisingly, both *mT5* variants outperform the trivial baselines (*RandomT*, *Lead3*), but their results are still far below the *Oracle* performance. Using larger training data (SumeCzech has roughly 20 times more documents than MLASK) improves the performance by approximately 1 ROUGE point.

6.2 Visual features

In Section 4.2, we proposed to employ two different visual features for both video and image feature extraction. The system exploiting all the features is denoted as MMS in Table 3. It achieves slightly higher scores than *mT5-MLASK* (dev-set ROUGE-1: 18.25 \rightarrow 18.34, ROUGE-L: 13.07 \rightarrow 13.26) but lags behind the text-only *mT5-SumeCzech* that was trained on a much larger corpus. To analyze the

⁸<https://github.com/scipy/scipy>

	ROUGE-1	ROUGE-2	ROUGE-L
<i>Lead3</i>	14.34	2.14	9.64
<i>Random3</i>	12.52	1.27	8.37
τ 2t (2018)	11.30	1.00	8.70
mT5-SumeCzech	18.46	4.54	13.33

Table 4: Performance of the text-to-text summarization models on the test part of SumeCzech (Straka et al., 2018) for the article→abstract task.

effect of the individual visual features, we report the results of the MMS model, excluding those features one by one (see the rows starting with the “-” sign). The scores indicate that the model combining all the features is superior.

6.3 Pre-training

Yu et al. (2021) showed that the usage of pre-trained generative language models is beneficial for multimodal summarization. We explored this idea further by task-specific pre-training on SumeCzech (Straka et al., 2018) – a large-scale Czech news summarization corpus used to fine-tune the mT5 model for summarization (*mT5-SumeCzech*). We used the Adafactor (Shazeer and Stern, 2018) optimizer with a constant learning rate equal to $5e-4$ and trained until ROUGE-L ceased to improve on the dev-set for 5 consecutive evaluations. To avoid any training/test data leaks, we excluded from the *mT5-SumeCzech* training data the articles that could appear in MLASK based on the date of publication (794,018 left, i.e., 92%). Performance on the test part of SumeCzech is reported in Table 4. Based on the results (Table 3, system MMS + SumeCzech), we can clearly see that the usage of mT5 fine-tuned for summarization instead of the raw mT5 boosts the performance on the text summarization part (test-set ROUGE-1: 18.45 → 19.58, ROUGE-L: 13.42 → 14.30).

6.4 Smooth labels

In Section 4.4, we proposed to use the smooth labels C_{smooth} during training to overcome the issue of very similar frames being labeled as positive and negative examples. Our results (Table 3, system MMS + SumeCzech + Smooth Labels) indicate that, indeed, this method helps with the quality of cover frame selection (test-set CosSim: 0.541 → 0.551, R@10: 0.318 → 0.330). We can also notice a small improvement (test-set ROUGE-1: 19.58 → 19.74) in the quality of text summarization, which we attribute to more stabilized training.

For a full comparison, we also include two variants with masked video features (MMS + SumeCzech + Smooth Labels + Masked Video and MMS + Masked Video) – all the video features are masked with random noise, both during the training and the evaluation. The frame features are left intact. Surprisingly, for the variant that was pre-trained on a large text-only corpus, masking the video features does not hurt the model performance. This is, however, the case for the model that did not go through the task-specific pre-training. After examining the models, we noticed that the representations after the video encoder (Eq. 5) are not very meaningful, i.e., every segment is mapped to a similar vector. We believe this is due to the indirect usage of video representations in the Cross-modal Interaction Module – too weak learning signal (gradient) is propagated to the video encoder. Considering the drop in performance for the model without pre-training, it seems to be the case that the information from pre-training and multimodal input is not completely orthogonal.

7 Human Evaluation

Previous works on VMSMO evaluated the system performance by employing human judges to assess the quality of generated textual summary: Li et al. (2020b) measured to what extent the system summaries were sufficient to answer questions generated from the reference summary and ranked them based on *Informativeness*, *Coherence*, and *Succinctness*; Fu et al. (2021) scored the system summaries based on *Informativeness* and *Satisfaction*. We believe no prior work employed human annotators to judge the quality of a chosen cover frame (pictorial summary) in the context of textual summary. For the similar task of multimodal summarization with *unimodal output*, Wan and Bansal (2022) collected annotations for the subset of the WikiHow dataset (Yang et al., 2021) that measured whether the textual output was faithful to the source pair of document and image.

7.1 Formulation

To evaluate the quality of cover frame selection, we asked human annotators to judge the quality and usefulness of an image as a pictorial summary of the article. 18 human annotators participated. All were adult, native Czech speakers who read online news magazines daily. Figure 3 displays a screenshot of the annotation tool. For each instance,

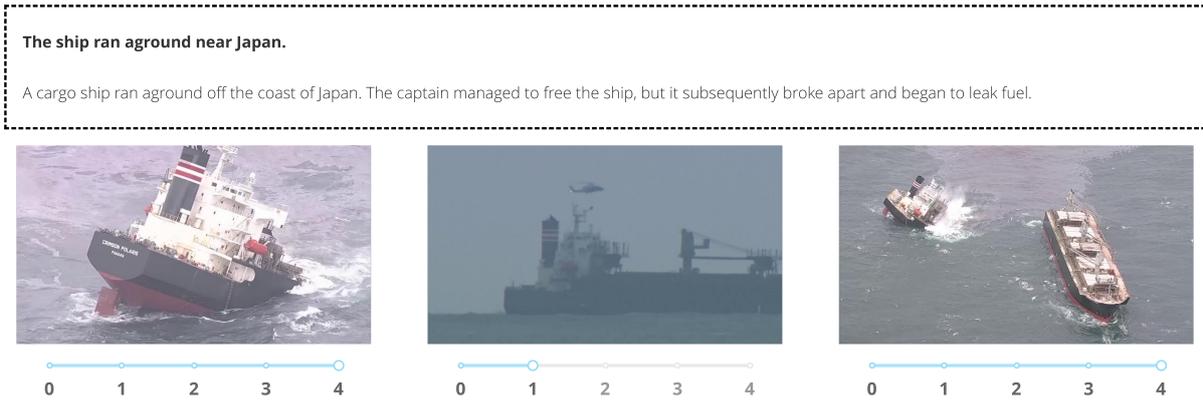


Figure 3: Screenshot of the annotation tool used to collect human judgments about the quality and usefulness of selected cover frame. For convenience, we translated all `text` into English.

the annotators were asked to rate 3 images on a scale of 0 to 4 (the higher, the better) in the context of the article’s title and the reference summary.

The suggested interpretation of the scale levels was:

- 0: The picture is not relevant at all or very marginally (technical quality is not important).
- 1: The image is partly relevant (there is a certain connection between what it captures and the content of the text), but technically imperfect (e.g., blurred, cropped inappropriately, taken from an inappropriate angle or at an inappropriate moment).
- 2: The image is partly relevant (there is a certain connection between what it captures and the text content) and of a good technical quality.
- 3: The picture is very relevant, but technically imperfect (e.g., blurred, cropped inappropriately, taken from an inappropriate angle, or at an inappropriate moment).
- 4: The picture is both very relevant and of a good technical quality. It is a suitable cover picture.

7.2 Setup

We randomly chose 300 instances from the MLASK test-set for annotation and split them into 10 batches of 30 instances. We used the first batch to measure the inter-annotator agreement, asking each annotator to score all the instances in the control batch plus at least in one more.

For each instance, four images were considered for annotation: the reference picture (denoted as *Reference*), a random frame from the video (*RandomV* output), and the outputs of two test models – MMS pre-trained on SumeCzech using the smooth labels (MMS + SumeCzech + Smooth Labels, fur-

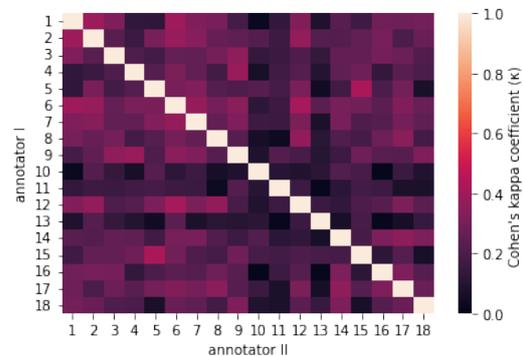


Figure 4: Values of Cohen’s κ used to measure the inter-annotator agreement on the control batch (30 instances).

ther denoted as System A) and the same model with masking of the video features (MMS + SumeCzech + Smooth Labels + Masked Video, further denoted as System B). See Appendix A for examples.

In the control batch, we always included the reference picture, hiding the output from one of the methods in 33% of the cases. In the other batches, we display 3 out of the 4 images selected randomly. To avoid a position bias, we shuffle the images before showing them to the annotator. On average, we collected 2.5 annotations for each image.

7.3 Results

Figure 4 displays the inter-annotator agreement on the control batch in the form of a heat map. The average value of 0.217 indicates a "fair" agreement. One can notice that three annotators (10, 11, and 13) have a lower average agreement (average below 0.2). We decided to exclude their annotations from further analysis. By doing so, the average value of Cohen’s κ increased to 0.26, and the average number of annotations decreased to 2.2.

	Total Score	Adequacy Score
<i>Reference</i>	2.89 ± 0.99	1.64 ± 0.50
<i>RandomV</i>	2.39 ± 1.15	1.44 ± 0.61
System A	2.64 ± 1.10	1.51 ± 0.58
System B	2.66 ± 1.04	1.56 ± 0.52

Table 5: System performance on the task of cover picture selection. See Section 7.1 for the label description.

In Table 5, we report the system-level averages of the scores assessed by human annotators (Total Score). On average, the reference picture is assigned the highest score, and our proposed multimodal summarization model performs better than the random baseline. The results of human assessment confirm our previous findings based on automatic metrics – that the model is not utilizing the video features in an effective manner. It is worth noticing, however, that even the reference picture is not considered very relevant (average score below 3) and that none of the differences are statistically significant. To examine the stability of the annotation process, we also report the averages (Adequacy Score) that disregard the quality of the image and focus only on relevance. We do this by mapping the labels from Section 7.1, (i.e., $0 \rightarrow 0$; 1 and $2 \rightarrow 1$; 3 and $4 \rightarrow 2$). The results are in line with the original ones.

8 Conclusions

In this paper, we explored the recently proposed task of video-based multimodal summarization with multimodal output. We extended the available resources to a new language by introducing a multimodal summarization dataset in Czech. We explored the pre-training strategies, showing that transferring knowledge from the simpler, unimodal task of text-to-text summarization helps with the final performance in multimodal settings. We were also able to show that the usage of inner-video similarities, via the introduction of smooth labels during training, helps to stabilize the training. We conducted a human evaluation of the frame-selection process to confirm the quality of the proposed multimodal MMS model. Our findings indicate that the MMS model pre-trained on the text-to-text summarization is not effective in utilizing video features and that future works should carefully examine to what extent the model is able to make use of multimodal input and whether the improvement is orthogonal to e.g., using more data.

Limitations

MLASK dataset collection. While curating the MLASK dataset, we applied a series of rule-based filters (Section 3.1) and collected only those documents that followed a strict HTML structure. No large-scale human evaluation was applied to check the data validity. We sampled a random subset of 100 articles and checked the data preparation and collection manually.

Language and domain bias. We acknowledge that our findings are based on a single dataset, in a particular language (Czech) and from a particular domain (news articles). Due to the novelty of the task, previous datasets proposed for VMSMO (Table 2) are not applicable to our experiments – dataset by Fu et al. (2021) does not provide single cover pictures, and the datasets by Li et al. (2020b) and Tang et al. (2022) are not publicly available. We also acknowledge that due to the data coming from a particular news provider, it may not be free of cognitive biases.

Technical requirements. Considering the modular architecture of the proposed model (Section 4.1), a modern GPU is required for training (using a 24GB GPU we were able to train with a batch size of 2). To store the raw MLASK dataset (videos and images), roughly 750GB of disk space is required.

Human evaluation. While conducting the human evaluation of cover frame selection, we provided a detailed set of instructions (Section 7.1) and used a control batch (30 instances) that was judged by each annotator to compute the inner-annotator agreement. Our findings (Table 5) indicate that there is a certain perception in the data annotation process that we did not analyze – the gold-standard reference picture is, on average, judged as only "partly relevant".

Acknowledgements

This work was supported by the Czech Science Foundation (grant no. 19-26934X) and CELSA (project no. 19/018). In this work, we used data and tools provided by the LINDAT/CLARIAH-CZ Research Infrastructure (<https://lindat.cz>), supported by the Ministry of Education, Youth and Sports of the Czech Republic (project no. LM2018101).

We thank our friends and colleagues that contributed to the annotation process and anonymous reviewers for their valuable feedback.

References

- Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris, and Ioannis Patras. 2021. Video summarization using deep neural networks: A survey. *Proceedings of the IEEE*, 109(11):1838–1863.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *International Conference on Learning Representations*.
- Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. [Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.
- Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. [SummEval: Re-evaluating summarization evaluation](#). *Transactions of the Association for Computational Linguistics*, 9:391–409.
- Xiyan Fu, Jun Wang, and Zhenglu Yang. 2020. [Multi-modal summarization for video-containing documents](#). *arXiv preprint arXiv:2009.08018*.
- Xiyan Fu, Jun Wang, and Zhenglu Yang. 2021. [MM-AVS: A full-scale dataset for multi-modal summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5922–5926, Online. Association for Computational Linguistics.
- Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. 2019. [Large-scale weakly-supervised pre-training for video action recognition](#). In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12038–12047.
- Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. [SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization](#). In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.
- Max Grusky, Mor Naaman, and Yoav Artzi. 2018. [Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.
- Jia-Hong Huang, Luka Murn, Marta Mrak, and Marcel Worring. 2021. [Gpt2mvs: Generative pre-trained transformer-2 for multi-modal video summarization](#). In *Proceedings of the 2021 International Conference on Multimedia Retrieval, ICMR '21*, page 580–589, New York, NY, USA. Association for Computing Machinery.
- Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. 2019. [CTRL - A Conditional Transformer Language Model for Controllable Generation](#). *arXiv preprint arXiv:1909.05858*.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Haopeng Li, Qihong Ke, Mingming Gong, and Tom Drummond. 2023. [Progressive video summarization via multimodal self-supervised learning](#). In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5573–5582.
- Haoran Li, Peng Yuan, Song Xu, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020a. [Aspect-aware multimodal summarization for chinese e-commerce products](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8188–8195.
- Haoran Li, Junnan Zhu, Tianshang Liu, Jiajun Zhang, and Chengqing Zong. 2018. Multi-modal sentence summarization with modality attention and image filtering. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI'18*, page 4152–4158. AAAI Press.
- Haoran Li, Junnan Zhu, Cong Ma, Jiajun Zhang, and Chengqing Zong. 2017. [Multi-modal summarization for asynchronous collection of text, image, audio and video](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1092–1102, Copenhagen, Denmark. Association for Computational Linguistics.
- Mingzhe Li, Xiuying Chen, Shen Gao, Zhangming Chan, Dongyan Zhao, and Rui Yan. 2020b. [VMSMO: Learning to generate multimodal summary for video-based news articles](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9360–9369, Online. Association for Computational Linguistics.

- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Nayu Liu, Xian Sun, Hongfeng Yu, Wenkai Zhang, and Guangluan Xu. 2020. [Multistage fusion with forget gate for multimodal summarization in open-domain videos](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1834–1845, Online. Association for Computational Linguistics.
- Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. [End-to-End Learning of Visual Representations from Uncurated Instructional Videos](#). In *CVPR*.
- Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. [HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips](#). In *ICCV*.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gülçehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence RNNs and beyond](#). In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.
- Shruti Palaskar, Jindřich Libovický, Spandana Gella, and Florian Metze. 2019. [Multimodal abstractive summarization for how2 videos](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6587–6596, Florence, Italy. Association for Computational Linguistics.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. [ImageNet Large Scale Visual Recognition Challenge](#). *International Journal of Computer Vision (IJCV)*, 115(3):211–252.
- Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. [MLSUM: The multilingual summarization corpus](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.
- Noam Shazeer and Mitchell Stern. 2018. [Adafactor: Adaptive learning rates with sublinear memory cost](#). In *International Conference on Machine Learning*, pages 4596–4604. PMLR.
- Milan Straka, Nikita Mediantkin, Tom Kocmi, Zdeněk Žabokrtský, Vojtěch Hudeček, and Jan Hajič. 2018. [SumeCzech: Large Czech news-based summarization dataset](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Mingxing Tan and Quoc Le. 2019. [EfficientNet: Rethinking model scaling for convolutional neural networks](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR.
- Peggy Tang, Kun Hu, Lei Zhang, Jiebo Luo, and Zhiyong Wang. 2022. [Tldw: Extreme multimodal summarisation of news videos](#). *arXiv preprint arXiv:2210.08481*.
- Daniel Varab and Natalie Schluter. 2021. [MassiveSumm: a very large-scale, very multilingual, news summarisation dataset](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10150–10161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.
- David Wan and Mohit Bansal. 2022. [Evaluating and improving factuality in multimodal abstractive summarization](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9632–9648, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. 2021. [Visual goal-step inference using wikiHow](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2167–2179, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Tiezheng Yu, Wenliang Dai, Zihan Liu, and Pascale Fung. 2021. [Vision guided generative pre-trained language models for multimodal abstractive summarization](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3995–4007, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. [PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.
- Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. [QMSum: A new benchmark for query-based multi-domain meeting summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.
- Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jijun Zhang, and Chengqing Zong. 2018. [MSMO: Multimodal summarization with multimodal output](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4154–4164, Brussels, Belgium. Association for Computational Linguistics.

A Appendix

Below, we attach outputs from the methods that we included in the human evaluation. System A is the MMS + SumeCzech + Smooth Labels model, and System B is the MMS + SumeCzech + Smooth Labels + Masked Video one. For convenience, we include both the original texts in Czech and machine translation to English. The reference text and images were extracted from Novinky.cz.

Reference	<p><i>Pětasedmdesátiletý řidič, který při páteční nehodě v Tatcích na Kolínsku usmrtil tříleté dítě a druhé vážně zranil, stejně jako jejich matku, je podezřelý z usmrcení z nedbalosti a z těžkého ublížení na zdraví z nedbalosti. Příčinou nehody byl zřejmě mikrospánek, uvedla v sobotu v tiskové zprávě policejní mluvčí Vlasta Suchánková.</i></p> <p>The 75-year-old driver, who killed a three-year-old child and seriously injured the other, as well as their mother, in Friday's accident in Tatky, Cologne, is suspected of negligent homicide and grievous bodily harm. The cause of the accident was apparently microsleep, police spokesperson Vlasta Suchánková said in a press release on Saturday.</p>
System A	<p><i>Dva trestné činy usmrcení z nedbalosti vyšetřují od pátečního odpoledne policisté v Nymburce, kde se srazilo osobní auto s dvěma malými holčičkami. Na dotaz Novinek to ve čtvrtek řekla mluvčí policie Eva Suchánková.</i></p> <p>Two crimes of culpable homicide have been under investigation since Friday afternoon by police officers in Nymburk, where a car collided with two little girls. Police spokesperson Eva Suchánková told News on Thursday.</p>
System B	<p><i>Dvě malé holčičky zemřely v pátek odpoledne při nehodě dvou aut na Nymbursku. Jeden člověk byl vážně zraněn, druhý je ve vážném stavu v nemocnici. Právu to řekla mluvčí policie Eva Suchánková.</i></p> <p>Two little girls died in a two-car accident in Nymburk on Friday afternoon. One person was seriously injured, the other is in a serious condition in hospital. Police spokesperson Eva Suchánková told the law.</p>

Table 6: Textual summary – Example 1.



(a) System A



(b) System B



(c) Reference



(d) Random

Figure 5: Pictorial summary – Example 1.

Reference	<p><i>Minulý týden v pátek se v Česku začaly oficiálně prodávat nové chytré hodinky Apple Watch Series 7. Zájem byl tak vysoký, že už po pár hodinách zmizely prakticky všechny hodinky z pultů českých obchodů. A vše nasvědčuje tomu, že si případní zájemci budou muset na další várku ještě dlouho počkat.</i></p> <p>Last week on Friday, the new Apple Watch Series 7 was officially launched in the Czech Republic. Interest was so high that after a few hours practically all watches disappeared from the counters of Czech shops. And all indications are that potential buyers will have to wait a long time for the next batch.</p>
System A	<p><i>V pátek se začaly oficiálně prodávat nové chytré hodinky od společnosti Apple. Zájem o novinku byl tak vysoký, že se v některých případech nepodařilo uspokojit všechny zákazníky, kteří si objednali hodinky s týdenním předstihem.</i></p> <p>On Friday, Apple's new smartwatch officially went on sale. Interest in the new smartwatch was so high that in some cases it was not possible to satisfy all customers who ordered the watch a week in advance.</p>
System B	<p><i>Zájem o novou generaci chytrých hodinek Watch Series 7 byl tak vysoký, že se v některých případech nepodařilo uspokojit všechny zákazníky, kteří si objednali hodinky s týdenním předstihem. Novinka má být daleko lépe než předchůdce – dostala extrémně tenké rámečky okolo displeje.</i></p> <p>Interest in the new generation of Watch Series 7 smartwatches was so high that in some cases it failed to satisfy all customers who ordered a watch a week in advance. The novelty is supposed to be far better than its predecessor – it got extremely thin frames around the display.</p>

Table 7: Textual summary – Example 2.



(a) System A



(b) System B



(c) Reference



(d) Random

Figure 6: Pictorial summary – Example 2.

Reference	<p><i>Ze století páry rovnou do 21. století. Tak by měla vypadat proměna hlavního brněnského nádraží. Ze současného místa v centru města by se mělo nádraží přesunout o několik set metrů, kde vznikne nový komplex na zelené louce. Jeho podoba by měla být z pera architektů nizozemského ateliéru Benthem Crouwel Architects.</i></p> <p>From the century of steam straight into the 21st century. This is what the transformation of Brno's main railway station should look like. From its current location in the city centre, the station should be moved a few hundred metres to create a new green meadow complex. Its design should be from the architects of the Dutch studio Benthem Crouwel Architects.</p>
System A	<p><i>Nové hlavní nádraží v Brně by mělo přijít na 50 miliard korun. Vítězný ateliér má zkušenosti z Nizozemska, Nizozemska nebo Nizozemska.</i></p> <p>The new main railway station in Brno should cost CZK 50 billion. The winning studio has experience from the Netherlands, the Netherlands or the Netherlands.</p>
System B	<p><i>Nové hlavní nádraží v Brně by mělo být hotové do roku 2035. Vítězný ateliér, který má s nádražím bohatou zkušenost, dostal od města peníze za 50 miliard korun.</i></p> <p>The new main railway station in Brno should be ready by 2035. The winning studio, which has extensive experience with the station, received money from the city worth CZK 50 billion.</p>

Table 8: Textual summary – Example 3.

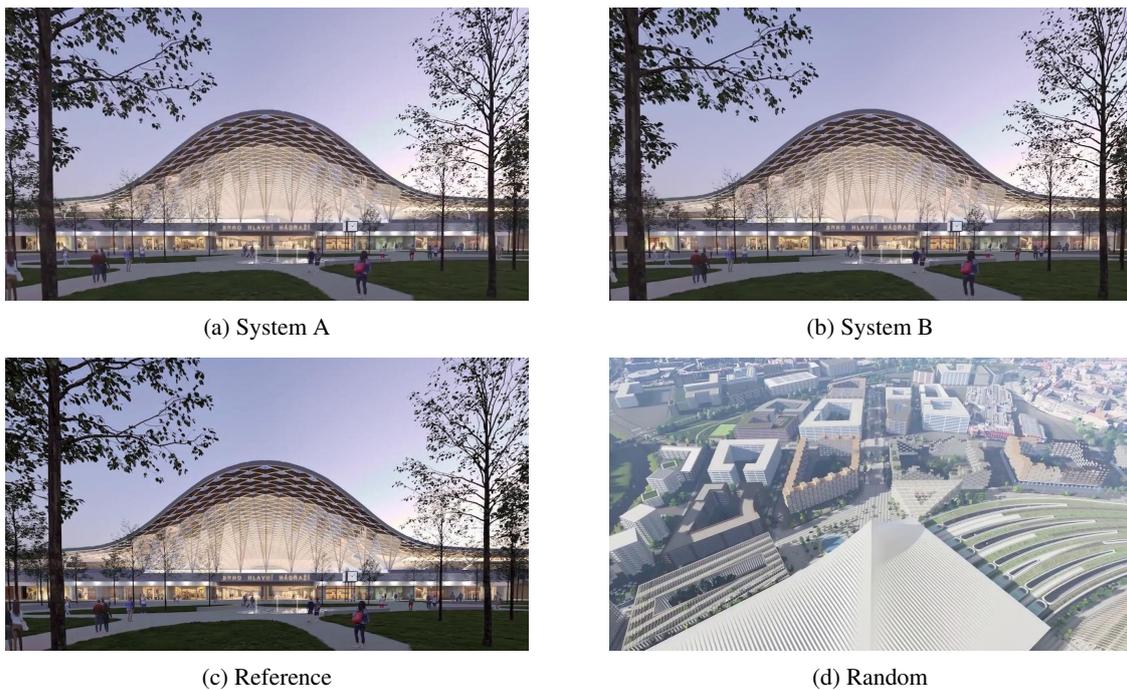


Figure 7: Pictorial summary – Example 3.

Going beyond research datasets: Novel intent discovery in the industry setting

Aleksandra Chrabrowa, Tsimur Hadeliya, Dariusz Kajtoch,
Robert Mroczkowski, Piotr Rybak

ML Research at Allegro, Poznań, Poland
{firstname.lastname}@allegro.pl

Abstract

Novel intent discovery automates the process of grouping similar messages (questions) to identify previously unknown intents. However, current research focuses on publicly available datasets which have only the question field and significantly differ from real-life datasets. This paper proposes methods to improve the intent discovery pipeline deployed in a large e-commerce platform. We show the benefit of pre-training language models on in-domain data: both self-supervised and with weak supervision. We also devise the best method to utilize the conversational structure (i.e., question and answer) of real-life datasets during fine-tuning for clustering tasks, which we call *Conv*. All our methods combined to fully utilize real-life datasets give up to 33pp performance boost over state-of-the-art Constrained Deep Adaptive Clustering (CDAC) (Lin et al., 2020) model for question only. By comparison CDAC model for the question data only gives only up to 13pp performance boost over the naive baseline.

1 Introduction

Allegro is one of largest the e-commerce marketplace in Central Eastern Europe region that connects buyers and merchants. It has millions of active users. Therefore, the good functioning of the Customer Experience (CX) department is crucial as it provides the necessary support, resolves emerging issues, and answers user questions.

Task-oriented chatbots relieve humans by automatically resolving the most repetitive and trivial issues. They usually have a pre-defined set of user intents with matching template answers. Then, when a user asks a question, the intent classifier detects the question intent and returns the matching response. Creating a reliable and comprehensive chatbot requires massive work to discover, define, and maintain a set of intents with training examples. With the continuous development of marketplace

platforms, new intents constantly appear as new features are introduced. Therefore, the automated intent discovery system becomes a critical component.

Novel intent discovery is performed offline on historical data. In the context of personalized intelligence assistants existing approaches (Lin et al., 2020; Gao et al., 2021; Vedula et al., 2022) focus on learning transferable features with utterance encoders that guide the discovery on unlabeled data with a handful of labeled examples belonging to known intents. However, at *Allegro* our main communication form is emails, and we have access to much richer conversational data that can improve discovery performance. A large body of historical conversational data (user questions and consultants' answers) can be leveraged in two ways. Firstly, to better initialize message encoders and secondly by performing intent discovery on conversational data as an additional signal. Additionally, a form of weak supervision is available: keywords (or tags) added by the consultants that help them understand past cases.

The paper's main contribution is the demonstration that incorporating additional signals like conversational structure or weak labels into the existing intent discovery method results in better overall performance. We pre-trained for domain adaptation three encoders using conversational data and weak labels. We devised *Conv*, a method for fine-tuning on conversational data (i.e., question and answer) for the clustering task using a three-headed encoder. To the best of our knowledge, this result was not reported in the public literature.

2 Related Work

2.1 Discovering novel intents

The goal of novel intent discovery is to identify groups of similar utterances in unlabeled data with the assistance of limited labeled data. The Con-

strained Deep Adaptive Clustering (Lin et al., 2020, CDAC) uses dense intent representation on top of the pre-trained BERT backbone to learn similarity functions in a semi-supervised contrastive manner. It is then utilized in the clustering algorithm. In a real-world scenario of personal assistants (Gao et al., 2021; Vedula et al., 2022) use a pre-trained BERT model as a backbone encoder with supervised contrastive learning to transfer distance function to unlabeled data for clustering. Unlike this work, the authors use only the question field and English *BERT-base* uncased model for initialization. They do not use in-domain unlabeled data or weak supervision for backbone pre-training.

2.2 Transfer learning

General-purpose pre-trained encoders like BERT are not ideal. Tasks involving domain-specific texts like, e.g., science corpus, clinical notes, or e-commerce product descriptions benefit more from additional pre-training on in-domain data due to better suited vocabulary and word embeddings to domain specific problems (Beltagy et al., 2019; Huang et al., 2019; Tracz et al., 2020; Gururangan et al., 2020). Similarly, for conversational tasks *ConveRT* (Henderson et al., 2020a) substantially outperforms BERT in neural response selection. Additionally, industrial-scale training on weakly supervised datasets leads to improvements in several NLP tasks (Bach et al., 2018).

3 Method

3.1 Problem statement

Given unlabeled instances \mathcal{D} , the goal is to automatically cluster utterances into \mathcal{I} classes, which are not known *a priori*. We also assume that we are given labeled instances \mathcal{D}^k with \mathcal{I}_k known set of intents and $\mathcal{I} \cap \mathcal{I}_k \neq \emptyset$. Unlabeled instances may belong to both known intents \mathcal{I}_k and unknown ones $\mathcal{I}_u = \mathcal{I} \setminus \mathcal{I}_k$.

3.2 Framework overview

Our novel intent discovery framework consists of representation learning (Bengio et al., 2013) and subsequent clustering with K-means (Lloyd, 1982). We propose the following to improve text representations for real-life novel intents discovery in the communication domain:

- Efficient initialization with pre-trained encoders, adapted to the e-commerce domain

by optimization for weak training signals and conversational structure of the data.

- Fine-tuning for the clustering task with state-of-the-art training scheme (i.e., CDAC) adapted to use all the conversational data (i.e., question and answer). *Conv* is our proposed method to train a conversation structure-aware encoder with three-headed architecture.

In the following sections, we describe each component in more detail.

3.3 Initialization

An essential step in the deep learning process is initialization. Proper initialization is crucial in training representations for discovering new intents with clustering. The effectiveness of the existing clustering algorithms depends heavily on the quality of the representation encoder. In this work, we identified this dependency and proposed a generic approach for an efficient encoder pre-training in the conversational domain.

3.3.1 Domain specific data structure

We operate in the e-commerce domain with a two-sided marketplace. Customers can seek support by exchanging messages via email or chat. The former are typically longer and include a more formal boilerplate. A dialog may be held between merchants and CX support, buyers and CX support, and directly between buyers and merchants. All messages are written in Polish.

3.3.2 Domain adaptation

We prepared two self-supervised models based on *BERT-base* (Devlin et al., 2019) architecture. We started from a general domain encoder *HerBERT* (Mroczkowski et al., 2021). We used a training corpus of 68M conversation threads with 184M messages and 8314M words. We included both emails and chats exchanged between all parties (merchants, CX support, and buyers).

- *AlleBERT* is *HerBERT* fine-tuned with Masked Language Model (MLM) objective.
- *AlleConveRT* is *AlleBERT* further fine-tuned on the same dataset but with the mixture of MLM and Conversational Contrastive Loss (CCL) (Henderson et al., 2020b).

The details of the training procedure for each of the pre-trained encoders can be found in Appendix E.

3.3.3 Weak supervision

In the case of email communication exchanged with CX support, every message includes at least one of 512 tags. These labels roughly identify the problem solved. They are assigned by CX consultants often in a noisy manner. We utilized this weak signal and prepared *TagBERT* encoder in a two-stage process. Firstly, we finetuned *HerBERT* with MLM and Message Threads Structural Objective (MTSO) (Wang et al., 2020) on all internal communication data (emails and chats). Secondly, we finetuned it on a multi-label classification task on *CX weakly supervised* dataset that includes 2.5M messages in the email domain exchanged between merchants or buyers and CX support. Details of the training procedure can be found in Appendix E.

3.4 Conv, conversation structure aware encoder

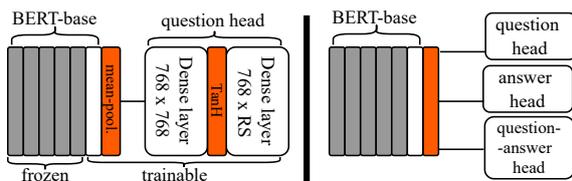


Figure 1: Representation model based on BERT-base encoder used in the discovery pipeline. On the left version with one head. On the right *Conv*, our conversational model with three separate trainable heads for the question, answer, and question-answer concatenation. The parameters of the encoder are frozen except for the last transformer block.

As depicted in Fig. 1, we used an encoder with *BERT-base* architecture (Devlin et al., 2019) followed by an average pooling¹ and three projection heads with two linear layers and *Tanh* non-linearity in between (Lin et al., 2020).

The three-headed model works with conversational input containing a pair of texts: the user’s question and the consultant’s answer². Two heads project each input separately, and the third one handles additional signals from the question-answer concatenation into one string of text. Each of the inputs is fed into encoder separately. A common underneath encoder is updated jointly with a gradient from all heads from the total loss given by the

¹Unlike many implementations, the hidden states for padding tokens are not averaged.

²While encoding question and answer, are preceded with special tokens for question and answer.

weighted average of losses for each head:

$$\begin{aligned} \mathcal{L}_{Conv}(X, Y, \theta) = & \lambda_Q \cdot \mathcal{L}(X_Q, Y, \theta_Q) \\ & + \lambda_A \cdot \mathcal{L}(X_A, Y, \theta_A) \\ & + \lambda_{QA} \cdot \mathcal{L}(X_{QA}, Y, \theta_{QA}). \end{aligned} \quad (1)$$

Here $X = (X_Q, X_A, X_{QA})$ is the array of inputs (all examples), i.e. all questions, all answers, all question-answer concatenations respectively. Y are the input labels³. $\theta = (\theta_Q, \theta_A, \theta_{QA})$ is the array of parameter sets for individual inputs *BERT-base* parameters are shared as depicted in Figure 1. The hyperparameters $\lambda = (\lambda_Q, \lambda_A, \lambda_{QA})$ govern how conversational structure is utilized for any choice of the training scheme, whereas the precise form of the loss terms \mathcal{L} depends on the choice of the training scheme described in Sec. 3.5. For example if we choose $\lambda = (1, 0, 0)$, and compute \mathcal{L} according to CDAC training scheme, we follow the original CDAC setup with the question field only. By using $\lambda = (0, 0, 1)$ and computing \mathcal{L} according to CDAC training scheme, we effectively only concatenate question and answer strings and feed it into the model instead of the question string.

In our method *Conv* for training conversation structure-aware encoder, we trained the representation encoder with uniform heads contribution $\lambda = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ starting from initializations described in Section 3.3. The final representation used for clustering is an embedding from the head for question-answer concatenation.

To speed up training with large batches, we kept the weights of the encoder frozen except for the last transformer layer. The first linear layer keeps the *BERT-base* dimension of the representations (i.e., 768). The second linear block output dimension is a representation size hyperparameter.

3.5 Training scheme

Up to this point, we are able to use any framework for finetuning the representation encoder for intent discovery with clustering. With that said, we propose to use two potential approaches for real-world CX communication data.

Static. In a setup where we do not have any labeled data available, we extract text representation from the pre-trained encoder by average pooling without additional training.

³Since we deal with unsupervised/semi-supervised algorithms, some examples are unlabelled.

Constrained DAC (CDAC) (Lin et al., 2020). The method generalizes the Deep Adaptive Clustering (DAC) (Chang et al., 2017) scheme for partially labeled data and trains with a contrastive loss on both distance-based pseudo-pairs and exact pairs given by intent labels. It is semi-supervised since it utilizes both labeled and unlabeled examples from the train set. We adapted CDAC training scheme to *Conv*, our three-headed, conversation structure-aware encoder (see Sec. 3.4). Details of the DAC method are in Appendix B.1, and details of the CDAC method are in Appendix B.2.

4 Evaluation

We describe our experimental setup for novel intent discovery. We prove the efficiency of the proposed method on real-world communication datasets. To verify gains from different framework components, we present more results in the ablation section (Sec. 5).

4.1 Real-world internal datasets

We used three internal datasets: *Purchase*, *Delivery* and *Retail* from real traffic to CX support at *Allegro* in Polish language. CX consultants manually annotated the datasets with intent labels. Categories of email queries to the CX team are more fine-grained than the widely used *Banking77* (Casanueva et al., 2020) dataset. Moreover, such real-world datasets are highly imbalanced, with some intents overlapping. Basic dataset statistics are shown in the Table 1. The user emails vary in length and style and may contain irrelevant parts. Each dataset includes messages of different quality and specificity ranging from uninformative chit-chat to well-written ones. In datasets, only the first question and direct answer are included, and all further messages from the correspondence thread are omitted. The *Purchase* and *Delivery* cover conversations between buyers and CX consultants. *Retail* is communication between buyers and merchants, so conversation topics and structure are different. We use a stratified 80/10/10 train/val/test split.

We use two public benchmark English datasets from task-oriented dialog systems: *CLINC150* (Larson et al., 2019) and *Banking77* (Casanueva et al., 2020) in Dataset splits follow exactly the experimental setup used in (Zhang et al., 2020) in ablation study in Section 5.2 to increase the reproducibility of our work. In other ablations it is impossible due to

missing conversational and weak label signal.

Basic statistics of the datasets are in the Table 1. Further details are in Appendix A.

4.2 Experimental setting

We build a controlled open-world intent discovery setup, following the setup proposed in (Lin et al., 2020; Zhang et al., 2020). We prepared novel intents by randomly masking all examples from 50% of intents in the training set. The remaining intents serve as known intents and are additionally partially masked. We masked 50% of all remaining examples. We apply the representation learning framework: we take in-domain encoders described in Section 3.3.2 and 3.3.3 and do the fine-tuning step (described in Section 3.4 and 3.5). After the training phase, we cluster the whole test dataset with K-means. We performed clustering with the ground truth number of clusters (i.e., the number of intents in the dataset).

We run experiments with hyperparameters (i.e., representation size, batch size, and learning rate) fixed. We have described the method of their selection in Appendix D.

We use five random seeds, which govern intent masking and weight initialization. We train the model for 100 epochs on a single machine with NVIDIA V100 GPU. It takes a few hours to run a single fine-tuning experiment for all seeds for a single setting (dataset, training scheme etc.).

4.3 Metric⁴.

We compute metrics based on cluster ids from K-means algorithm and ground truth labels. The discovery quality is probed with three standard clustering metrics, i.e., Accuracy (ACC) using the Hungarian algorithm, Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). We also introduce two additional metrics. First, the *binary F1-score* i.e., macro F1-score with a majority vote on cluster label calculated on the whole dataset where all known intents are one class, and all novel intents are the second class. Second, the *macro F1-score* with a majority vote on the cluster label. It turns the clustering quality problem into a multi-label classification. In the main part of the paper, we report **AVG** i.e., the average of five metrics over all seeds. AVG increases with clustering quality up to 100%. AVG is the primary metric used for

⁴We publish the code for our metrics: <https://github.com/allegro/ml/tree/main/publications/intent-discovery-metrics/>

Dataset	# intents	# examples	# examples per intent				mean length (characters)	
			mean	min	max	entropy	question	answer
<i>Banking77</i>	77	13.1k	170±33	75	227	0.992	60±40	-
<i>CLINC150</i>	150	22.5k	150±0	150	150	0.999	40±20	-
<i>Purchase</i>	22	2.7k	121±50	29	240	0.972	320±280	1060±400
<i>Delivery</i>	23	3.0k	130±55	57	221	0.973	330±360	860±410
<i>Retail</i>	105	13.8k	133±124	22	664	0.930	160±190	740±830

Table 1: Downstream tasks datasets characteristic. Class imbalance is measured by the average number of examples per intent and the normalized Shannon’s entropy of the intent distribution (which is 1 for the perfectly balanced case and lower in case of class imbalance). Further details are in Appendix A

Method	<i>Purchase</i>	<i>Delivery</i>	<i>Retail</i>
Static	37.0±4.1	31.1±1.3	28.8±0.7
CDAC	50.2±6.6	40.9±4.5	36.5±1.7
Our	83.2±3.2	64.2±6.3	45.4±4.0

Table 2: Static baseline and CDAC representations compared with our framework on novel intent discovery task for real-world data. Our framework combines *TagBERT* pre-trained encoder, CDAC training scheme, and *Conv* method for using the conversation structure. AVG metric averaged over five seeds.

model selection. Additionally, to facilitate comparison with other research, the five metrics are listed separately in Appendix F for all experiments. In Appendix F we give more details on how we compute metrics or test for statistical significance.

4.4 Results

Table 2 shows the AVG metric for our best-performing model. Five individual metrics are listed in Table 8. We significantly improve intent discovery compared with baselines. *Our* model uses *TagBERT* (see Section 3.3.3) as initialization and is trained with the CDAC scheme. While training, we used both question and answer fields and utilized conversational structure-aware encoder *Conv* introduced in Sec. 3.4. The baselines (*Static* and *CDAC*) are based on the general domain *HerBERT* encoder and use the question field only. We improved over the second-best CDAC, depending on the dataset, by 8.9pp to 33pp. The performance gap of *our* framework to the *CDAC* baseline is greater than the superiority of *CDAC* over the naive baseline, *static* embeddings, which is between 7.7pp and 13.2pp.

Initialization	<i>Purchase</i>	<i>Delivery</i>	<i>Retail</i>
<i>HerBERT</i>	65.9±6.2	44.7±3.7	37.2±2.0
<i>AlleBERT</i>	66.4±6.6	49.2±6.4	44.2±2.2
<i>AlleConveRT</i>	73.1±8.8	57.9±5.9	49.3±2.1
<i>TagBERT</i>	83.2±3.2	64.2±6.3	45.4±4.0

Table 3: Impact of initialization for novel intent discovery task. *Conv* conversation structure-aware encoder was trained with the CDAC scheme from different initialization. AVG metric averaged over five seeds with standard deviation.

5 Ablation

We attribute the improvement in performance to all three method components: domain adaptation during pre-training with conversational and weak label signal, state-of-the-art training scheme CDAC, and leveraging of conversation structure with our *Conv* method introduced in Section 3.4.

5.1 Initialization

In this section, we show the effect of initialization on the novel intent discovery task. We trained a conversation structure-aware encoder with a CDAC scheme using four different initializations.

AVG metric is reported in Table 3 and individual metrics are shown Table 9. Comparing *AlleBERT* with *HerBERT*, we can see that domain-adapted initialization improves 1 to 7pp for discovering new intents. Further adaptation of the starting encoder with the loss of ConveRT improves at least 5pp. Summarizing *AlleBERT* and *AlleConveRT* initializations bring gains for all internal datasets. For the CX domain (*Purchase* or *Delivery*), the best initialization was provided by *TagBERT*. Pre-training with weak labels introduced additional training information that turned out to be transferable for the

Training scheme	<i>Banking77</i>	<i>CLINC150</i>	<i>Purchase</i>	<i>Delivery</i>	<i>Retail</i>
Static	41.7±1.0	55.9±1.4	35.5±4.1	31.0±2.4	29.6±0.8
DAC	51.8±1.8	64.6±1.3	24.1±0.7	24.0±0.9	27.3±4.4
Supervised	65.2±2.1	73.2±0.6	38.2±2.1	33.5±2.2	30.1±0.5
CDAC	61.8±2.8	70.4±1.4	52.9±7.3	42.3±3.6	39.2±1.2

Table 4: Evaluation of training schemes for novel intent discovery. We report AVG metric averaged over five seed with standard deviation. Models use *BERT-base* (English datasets) or *AlleBERT* (Polish datasets) encoder and question input only. The best results are in bold.

downstream task. The simultaneous drop in quality on the *Retail* dataset originating from the domain for which we did not have noisy labels confirms this phenomenon.

5.2 Training schemes

We compare two training schemes *Static*, and *CDAC* from Sec. 3.5 with two additional baseline methods *DAC* and *Supervised*. For *Supervised* training scheme, we use Large Margin Cosine Loss (LMCL) (Wang et al., 2018) to learn representation from labels. We discard unlabeled data from the train set. We train the models for all four schemes with question input only and *BERT-base* (Devlin et al., 2019) for English and *AlleBERT* for Polish datasets.

This ablation study is the only case when we can use two public benchmark English datasets from task-oriented dialog systems: *CLINC150* (Larson et al., 2019) and *Banking77* (Casanueva et al., 2020). Unfortunately, public benchmark datasets lack the answer data, a large amount of unlabeled data, and weak labels. However, including them in this ablation study increases the reproducibility of our work and brings interesting insights.

AVG metric is reported in Table 4 and individual metrics can be found in Table 10. For all datasets, there is a gain from using intent labels (*Supervised* and *CDAC*). For public datasets among unsupervised methods, DAC outperforms static representations. However, supervised training is better than semi-supervised CDAC. The results are the opposite for the internal datasets. DAC is better than static representations, and semi-supervised CDAC is better than supervised training. We hypothesize that different real-world and benchmark datasets results might be due to dataset quality and size differences. In general, benchmark datasets are larger and more balanced. Moreover, mail messages from real-world e-commerce are longer and noisier on average. It is an open question how this trend holds

for other real-life datasets.

To sum up, there is a gain from intent labels for all datasets. Optimal solutions for public benchmarks and real-world internal datasets differ. CDAC is the best training scheme that uses intent labels for internal datasets.

5.3 Conversational structure

We examine if any further gains in performance can be obtained from incorporating the answer field signal. We conduct experiments only on the internal datasets. We use only the best training scheme, i.e., *CDAC*. We examine four training configurations: only question representation Q trained with $\lambda = (1, 0, 0)$, only answer representation A trained with $\lambda = (0, 1, 0)$, question-answer concatenation QA concatenation trained with $\lambda_3 = (0, 0, 1)$, using question and answer in a simpler two-headed model QA two heads trained with $\lambda = (\frac{1}{2}, \frac{1}{2}, 0)$ and full three-headed conversational model $Conv$ trained with $\lambda = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ described in detail in section Sec. 3.4.⁵

AVG metric is reported in Table 5 and individual metrics can be found in Table 11. The answer alone performs worse than the question alone. We hypothesize that it is due to many non-informative generic answers⁶. Perhaps for other real-world datasets consultant’s answer may be superior to the user’s questions. Passing only the question signal is a strong baseline. Let us check if it is possible to incorporate signals from both question and answer fields in a way that improves performance over Q , question field only baseline. The most straightforward extension, QA concatenation, which requires only inputting different inputs to the same model is slightly better but does not pass the statistical

⁵For multi-headed encoders, we chose the best of all possible final representations (output from any head, or concatenations of outputs from multiple heads).

⁶e.g., *Thank you for your message. Let me check some details and reply later.*

	<i>Purchase</i>	<i>Delivery</i>	<i>Retail</i>
<i>Q</i>	52.9±7.3	42.3±3.6	39.2±1.2
<i>A</i>	51.7±5.5	37.6±4.5	30.5±1.5
<i>QA concat.</i>	55.1±3.8	47.3±3.4	43.4±3.1
<i>QA two head.</i>	56.4±5.9	46.9±5.4	40.2±1.7
<i>Conv</i>	66.4±6.6	49.2±6.4	44.2±2.2

Table 5: Evaluation of conversational structure for novel intent discovery. We report AVG metric averaged over five seed runs with standard deviation. Models use *AlleBERT* initialization, CDAC training scheme, and various inputs, i.e., question *Q*, answer *A*, or both fields (*QA*) in three model variants; *QA concatenation*, *QA two heads*, and *Conv*. The best results are in bold.

significance test. The same goes for the more sophisticated *QA two heads* variant. Only our method *Conv*, a three-headed encoder is better than *Q* with statistical significance. Incorporating both question and answer signal leads to further improvements.

To sum up, after examining multiple ways to include the conversational signal, we conclude that our method *Conv* with a three-headed encoder improves the performance by 5 to 13.5pp.

6 Commercial deployment

6.1 Production pipeline overview

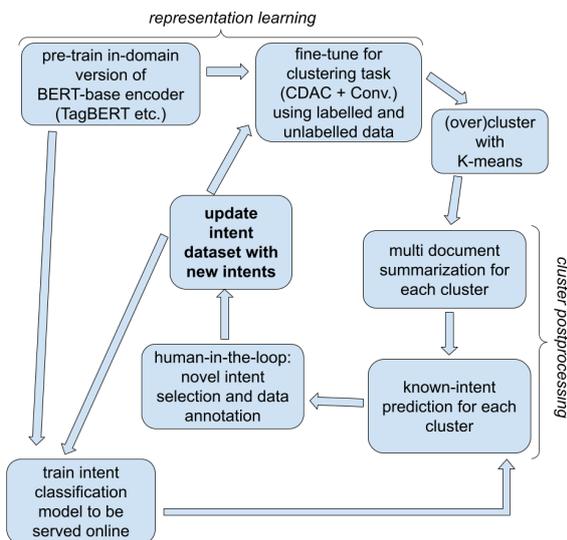


Figure 2: Intent discovery pipeline deployed at *Allegro* with human-in-the-loop carrying out the novel intent selection and data annotation. Representation learning components are subject to experiments in this paper. The main outcome of the pipeline is an updated intent detection dataset, which can be used to train a better intent classification model.

The method we described and verified experimentally is a part of a larger multi-component system for continuous intent discovery deployed commercially, shown in Fig. 2. Here we briefly list the major components of our production pipeline to give the bigger picture:

1. *Representation learning.* Representation learning plays a core role in our pipeline. This component is subject to experiments in this paper and consists of two subcomponents:

- In-domain pre-training of encoders.* Encoders with *BERT-base* architecture are pre-trained on large chunks of historical data. We include additional signals such as conversational structure (i.e. question and answer) and weak label signal (Section 3.3.2 and 3.3.3). The encoders are reused for the intent classification model.
- Fine-tuning for the clustering task.* We further train in-domain encoders. If there exists annotated data, we use semi-supervised CDAC with *Conv* (Section 3.4). Otherwise, we use static embeddings.

2. *(Over)clustering with K-Means.* We cluster representations to discover intent groups in the data. The number of novel intents is required by K-Means. We overestimate this value as it is less time-consuming to manually merge clusters with the same intent.

3. *Cluster postprocessing.* Various postprocessing steps make analyzing the clusters by the human annotators more efficient:

- Multi-document summarization.* The summarization module, provides human-readable candidates for the intent name instead of cluster ids. First, we train a logistic regression classifier with bag-of-words features to predict cluster ids. Then, we identify the most informative sentence in each message using the classifier coefficients (Angelidis and Lapata, 2018). Finally, we select the five most central sentences across all messages (Zheng and Lapata, 2019).
- Known intent prediction.* We need to distinguish clusters with known intents from clusters with potentially novel intents.

Since the labeled messages are typically a small subset of the training dataset, we infill intents for the unlabeled examples with an intent classifier and present this information to human annotators.

4. *Novel intent selection and data annotation.* Human annotators manually analyze all discovered clusters and choose which novel intents to include in the taxonomy. They annotate all messages from clusters to be included in the labeled dataset to ensure the high coherence of newly discovered intents.

CX intent dataset updated with new intent is the end product of our intent discovery pipeline. Its primary purpose is to train an intent classifier to be served in real-time to CX consultants. It is a complex pipeline of its own. It has similar architecture to the representation learning model in the intent discovery pipeline and it reuses pre-trained encoders. Even though the consultant's answer and the consultant's weak label are not known at the serving time of the intent classification model, we leverage these signals to build a better intent dataset and directly train a better intent classification model.

6.2 Commercial benefits case study

Thanks to the deployed pipeline, we doubled the number of defined intents for customer support within one year. Initially, the taxonomy consisted of 100 classes manually defined by the CX consultants. The commercial deployment of the intent discovery pipeline happened at the moment when the domain experts failed to find any new intents manually. Roughly 50 new intents were discovered thanks to our intent discovery pipeline. The selected clusters were reasonably pure: over 90% (mean and median) of examples from the selected clusters were labeled as the given intent. Additional examples for the new intents were further added (active learning etc.) and at the moment, the examples from the clustering process are at least 40% of all examples for 50 automatically discovered intents. Currently, after extending our taxonomy from other sources as well, our taxonomy has roughly 180 intents.

In addition, the pipeline decreased the time required to define novel intents from weeks to days with the additional benefit of analyzing several-fold more messages. The more comprehensive taxonomy significantly impacts the total benefit from the

automation process, improves user experience by providing faster responses, and saves the cost of hiring additional CX consultants.

7 Conclusions

This paper describes an intent discovery pipeline deployed on a large e-commerce platform. The access to real-life datasets allows extending the established intent discovery models to better leverage vast amounts of unlabelled data, its conversational structure, and additional signals like weak labels. In particular, we learn the following lessons:

1. Among multiple ways to handle conversational data, *Conv*, our generalization of the CDAC model to a three-headed encoder to use all available conversational data (i.e., question and answer) increases the performance of the intent discovery pipeline the most. See Section 5.3.
2. The significant gains also come from pre-training the encoder on an unlabelled in-domain dataset with conversational structure and weak labels (*TagBERT*). See Section 5.1. Therefore, we recommend a system architecture that enables weak labeling by the consultants by design.
3. Even though the consultant's answer and weak labels are not available at the serving time of the intent classification model, they can be used offline for novel intent discovery to build a better dataset and directly improve the intent classification. It happened for our commercially deployed pipeline. See Section 6.
4. Gains from incorporating additional signals (*Conv* method, *TagBERT*) are larger than gains from using state-of-the-art methods (CDAC) on datasets without additional signals. See Section 4.4. We advocate for a shift both in construction and research on intent detection datasets.

8 Limitations

We are aware of two major factors that may affect the generality of our research: shortcomings of the simulated novel intent discovery setup and the assumption that intent detection is a classification problem.

Simulated experiments. In the experimental section, we use small, entirely annotated datasets to analyze different design choices of the representation learning component. We naturally include only already discovered intents (does not mean these are all possible). Our masking procedure that follows research papers (Lin et al., 2020; Zhang et al., 2020) has three drawbacks. Firstly, when we mask most of the dataset, we effectively do few-shot learning, whereas, in reality, the amount of annotated data is much larger. The observed differences between design choices may be mitigated once more data is available. Secondly, real class imbalance may not be reflected in the experimental dataset due to the annotation procedure. Lastly, the ratio between batch size and dataset size is much smaller for real datasets since, in general, we are training with a large amount of unannotated data. It directly affects batch-based pair statistics when using a random sampler in CDAC algorithm. The chance that annotated examples will be present in the batch is low, and effectively we are almost entirely learning from pseudo-pairs during the semi-supervised stage.

Intent detection as classification. We treat the intent discovery as classification i.e. each utterance has only one intent. In reality, users may have more than one goal that transforms the problem into a multi-label scenario. Naturally, we could treat multi-label examples as yet another class, but we do not explore their influence on pipeline performance since they were in a significant minority.

Acknowledgements

We thank Bartosz Ludwiczuk, Tomi Wójtowicz, Karol Grzegorzczuk for valuable discussions and contributions to the source code.

References

- Stefanos Angelidis and Mirella Lapata. 2018. [Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.
- Stephen H. Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alexander Ratner, Braden Hancock, Houman Alborzi, Rahul Kuchhal, Christopher Ré, and Rob Malkin. 2018. [Snorkel drybell: A case study in deploying weak supervision at industrial scale](#).
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.
- Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. [Efficient intent detection with dual sentence encoders](#). In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.
- J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. 2017. [Deep adaptive image clustering](#). In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888.
- Sławomir Dadas. 2019. [A repository of polish NLP resources](#). Github.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#).
- Xibin Gao, Radhika Arava, Qian Hu, Thahir Mohamed, Wei Xiao, Zheng Gao, and Mohamed AbdelHady. 2021. [Graphire: Novel intent discovery with pretraining on prior knowledge using contrastive learning](#). In *KDD 2021 Workshop on Pretraining: Algorithms, Architectures, and Applications*.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don’t stop pretraining: Adapt language models to domains and tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020a. [ConveRT: Efficient and accurate conversational representations from transformers](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.
- Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020b. [ConveRT: Efficient and accurate conversational representations from transformers](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.

- Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv:1904.05342*.
- Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *CoRR*, abs/1909.02027.
- Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8360–8367. AAAI Press.
- Stuart P. Lloyd. 1982. Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28:129–136.
- Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. 2021. HerBERT: Efficiently pretrained transformer-based language model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kyiv, Ukraine. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. SentenceBERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.
- Piotr Rybak, Robert Mroczkowski, Janusz Tracz, and Ireneusz Gawlik. 2020. KLEJ: Comprehensive benchmark for Polish language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1191–1201, Online. Association for Computational Linguistics.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MpNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.
- Janusz Tracz, Piotr Iwo Wójcik, Kalina Jasinska-Kobus, Riccardo Belluzzo, Robert Mroczkowski, and Ireneusz Gawlik. 2020. BERT-based similarity learning for product matching. In *Proceedings of Workshop on Natural Language Processing in E-Commerce*, pages 66–75, Barcelona, Spain. Association for Computational Linguistics.
- Nikhita Vedula, Rahul Gupta, Aman Alok, Mukund Sridhar, and Shankar Ananthakrishnan. 2022. Advin: Automatically discovering novel domains and intents from user text utterances. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7627–7631.
- Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. 2018. Cosface: Large margin cosine loss for deep face recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274.
- Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. Structbert: Incorporating language structures into pre-training for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lv. 2020. Discovering new intents with deep aligned clustering. *CoRR*, abs/2012.08987.
- Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

A Dataset details

We further describe real-world internal datasets introduced in 4.1 and compare them to public benchmark datasets. Table 6 exemplifies the domain diversity of the datasets: it contains three sample intent names per dataset.

We visualize the datasets. We use publicly available pre-trained models to enable simple visual comparisons between our real-world internal datasets and any other datasets. Sentence-BERT produces English sentence embeddings by fine-tuning on semantic textual similarity STS pairs (Reimers and Gurevych, 2019). We use a variation of Sentence-BERT trained from MP-Net (Song et al., 2020). Polish version has been obtained following knowledge distillation procedure (Reimers and Gurevych, 2020; Dadas, 2019).⁷ We compute sentence embeddings for the question field or if the answer field is present, for question-answer concatenation. For each example, we compute a partial Silhouette score (using ground truth intents as cluster labels) and average it per intent. Silhouette score, designed originally for evaluating the clustering quality, takes into account the mean intra-cluster distance and the mean nearest-cluster distance for each example. We plot 2D t-SNE mappings of the embeddings, Silhouette score per intent⁸, and intent sizes in Figures 3 and 4 to visualize the datasets and the initial difficulty of the clustering task on general domain pre-trained models.

B Training schemes

B.1 Deep Adaptive Clustering (DAC)

It was introduced in (Chang et al., 2017) for the Computer Vision domain but is easily extended to text. Originally, output representation was interpreted as a probability distribution over unique classes, i.e., they used L_2 normalized features with positive elements. We relaxed this condition and trained real-valued representation for any clustering algorithm. The representation size doesn't have to match a unique number of classes in the dataset (unknown in real scenarios). For a pair of examples i, j the loss function \mathcal{L}_{ij} is

$$\mathcal{L}_{ij} = -R_{ij} \log S_{ij} - (1 - R_{ij}) \log(1 - S_{ij}), \quad (2)$$

⁷Package sentence-transformers, available at <https://sbert.net>, is used with models all-mpnet-base-v2 or sdadas/st-polish-paraphrase-from-mpnet for English and Polish respectively.

⁸<https://scikit-learn.org/>

Dataset	Three sample intent labels
<i>Banking77</i>	<ol style="list-style-type: none"> Cash withdrawal charge Getting spare card Request refund
<i>CLINC150</i>	<ol style="list-style-type: none"> Transactions Next song International fees
<i>Purchase</i>	<ol style="list-style-type: none"> I have a technical problem. When will my Smart! be active? How to withdraw from the auction?
<i>Delivery</i>	<ol style="list-style-type: none"> I didn't pick up my parcel and I'm asking for a refund. How to withdraw from the contract? I want to use Buyers Protection Program.
<i>Retail</i>	<ol style="list-style-type: none"> When will the sale of the offer start? I have a problem with the customer service for my purchase. Is the product prepackaged?

Table 6: Domain diversity of labeled datasets used for novel intent discovery experiments. Three sample intent names per dataset are given.

Dataset	<i>Banking77</i>	<i>CLINC150</i>	<i>Purchase</i>	<i>Delivery</i>	<i>Retail</i>
Representation size	256	256	32	32	64
Batch size	128	128	16	32	16
# intents	77	150	22	23	105

Table 7: Optimal representation size and batch size vs. a number of annotated intents in the datasets.

where ($R_{ij} = 1$) for positive pairs and ($R_{ij} = 0$) for negative pairs and S_{ij} is cosine similarity of representations. The pseudo-label matrix R is defined in an online fashion for every pair of examples in a batch using current model predictions i.e.

$$R_{ij} = \begin{cases} 1, & \text{if } S_{ij} \geq u(\lambda), \\ 0, & \text{if } S_{ij} < l(\lambda), \\ \text{None}, & \text{otherwise,} \end{cases} \quad (3)$$

where $u(\lambda)$ and $l(\lambda)$ are upper and lower thresholds. Pairs between the thresholds do not take part in the training. This is compensated by adding penalty term $u(\lambda) - l(\lambda)$ to the final loss. The thresholds are updated every epoch according to the formula

$$\begin{aligned} u(\lambda) &= 0.95 - \lambda, \\ l(\lambda) &= 0.455 + 0.1 \cdot \lambda, \end{aligned}$$

where update rule for λ every epoch is $\lambda = \lambda + 1.1 \cdot 0.009$ (Chang et al., 2017). We start with $\lambda = 0$. The training ends when $u(\lambda) = l(\lambda)$. The training resembles curriculum learning: we start with confident examples with very large or low cosine similarity and then introduce more uncertainty. The penalty term also reflects our confidence since it controls the strength of gradient updates.

B.2 Constrained DAC (CDAC)

This extension of DAC to a semi-supervised scenario was introduced in (Lin et al., 2020). In unsupervised case, we only use contrastive objective with pseudo-labels. Once we have annotated examples, we define true positive and negative pairs with labels. The label matrix R has now pseudo-label part (3) and exact part

$$R_{ij} = \begin{cases} 1, & \text{if } y_i = y_j, \\ 0, & \text{if } y_i \neq y_j, \end{cases} \quad (4)$$

where y_i denotes encoded label for i -th example. Since our batch now includes annotated and unannotated examples, we need to redefine pseudo-labels. We consider three cases. Firstly, pseudo-labels can be defined only among unannotated examples. Secondly, we can allow pseudo-labels between pairs of annotated and unannotated examples. Lastly, we can define pseudo-labels for all possible pairs, including a scenario where pseudo-labels are defined among annotated pairs. We chose the second scenario.

Additional modification is alternating training. Even epochs use only annotated data and no threshold penalty. Odd epochs use the whole dataset and pseudo-label matrix as well as exact. The loss in the supervised phase is additionally scaled by the $\delta \geq 1$ hyperparameter to control the weight put on annotated data.

C Metrics⁹.

We choose metrics for our experiments. Three clustering metrics measure the separation of novel intents from each other:

- **Accuracy (ACC)** measures clusters purity. Cluster and ground-truth labels are matched with the Hungarian algorithm.
- **Normalized Mutual Information (NMI)** specifies the amount of uncertainty about class labels given cluster labels.
- **Adjusted Rand Index (ARI)** checks for all sample pairs whether their assigned and ground truth labels are the same.

ACC, NMI, and ARI are calculated only on examples with a novel intent as a ground truth label.

The separation of the novel from the known intents is measured by:

- **Binary F1-score.** It is a macro F1-score with a majority vote on the cluster label calculated on the whole dataset where all known intents are one class and all novel intents are the second class.

Last but not least, there is a metric that measures both the separation between novel intents and the separation of the novel from the known:

- **Macro F1-score** with majority vote on cluster label. It turns the clustering quality problem into multi-label classification.

The macro average is calculated only for novel intents. Examples with any ground truth label may be included¹⁰.

All metrics increase with clustering quality up to 100%. We use five random seeds, which govern intent masking and weight initialization. In

⁹We publish the code for our metrics: <https://github.com/allegro/ml/tree/main/publications/intent-discovery-metrics/>

¹⁰See: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

the main part of the paper, we report **AVG** i.e., the average of five metrics listed above (which are correlated variables) overall seeds. AVG is the primary metric used for model selection. Whenever in doubt, we confirm that the difference between AVG metrics is statistically significant with correlated T-Test with a p-value=5% threshold. Additionally, to facilitate comparison with other research, for all experiments, the five metrics are listed separately in Appendix.

D Initial fine-tuning

We start our experiments with fine-tuning representation size, batch size, and learning rate hyperparameters for the CDAC training scheme¹¹. For every dataset, we optimize the hyperparameters in two steps: selecting optimal representation size via grid search over the representation sizes {16, 32, 64, 128, 256} and learning rates {1e-05, 5e-05, 1e-04} and then selecting the optimal learning rate and batch size via grid search over batch sizes {16, 32, 64, 128, 256, 512} and the same learning rates as step 1. Tab. 7 shows the relation of the selected hyperparameters to the number of intents. The selected hyperparameters are later fixed in the experiments. Additionally, to improve training stability, we perform an additional learning rate search again within values {1e-05, 5e-05, 1e-04} for every setup which uses *Conv* method separately.

E Pre-trained encoders (details)

To leverage large amounts of historical data, we compare four self-supervised encoders, and one supervised trained on conversational data. The training procedure for each encoder is described in detail below for reproducibility. The encoders are used for experiments in Sec. 4.4.

HerBERT State-of-the-art *BERT-base* language model for Polish (Mroczkowski et al., 2021) trained with Masked Language Model (MLM) objective.

AlleBERT The model is a result of further fine-tuning *HerBERT* on internal unsupervised conversational data. The single training example contains a conversation thread clipped to 512 tokens. We always clip threads to a random subsequence of whole consecutive utterances to persist in a conversational context. *AlleBERT* is trained with the

¹¹We focus on CDAC encouraged by initial good results for CDAC and high cost of fine-tuning each training scheme separately.

MLM objective for 100k steps with the linearly decaying learning rate schedule (peak value 1e-05) and the batch size of 224. The training on four NVIDIA A100 GPUs lasted 2 days.

AlleConveRT The model is a result of further fine-tuning of the *AlleBERT* on the same data but with the mixture of two objectives, MLM loss with the ratio of 0.2 and Conversational Contrastive Loss (CCL). Following ConveRT (Henderson et al., 2020b) we leverage the structure of the conversations with alternately exchanged utterances in a metric learning setup. Positive examples are consecutive messages from a single conversation, and negatives come from answers within the training batch. To reduce the overfitting to specific utterances, we use label smoothing with the value of 0.2 (same as (Henderson et al., 2020b)). To utilize conversational data structure, we add two projection heads on top of the *AlleBERT* encoder, one for the question and answer representations¹². *AlleConveRT* is trained for the 280k steps with the peak learning rate 1e-05 and the batch size of 448. The training on four NVIDIA A100 GPUs lasted 4 days.

TagBERT The model is trained in two-stage fine-tuning of the first version of *HerBERT* (Rybak et al., 2020). In the first stage, we fine-tune the model on internal unsupervised conversational data. We use MLM objective and Message Threads Structural Objective (MTSO). MTSO is Sentence Structural Objective (Wang et al., 2020) tailored to the conversation domain. During training, we swap messages with respect to threads instead of swapping sentences with respect to documents. *TagBERT* is trained for 100k steps with a batch size of 640 and a peak learning rate 8e-05.

In the second stage, we fine-tune the model on the multi-label classification task. The model predicts several of the 512 classes for each thread. The noisy and highly imbalanced labels come from tags that CX consultants add to the conversation threads, roughly identifying the problem solved. The training dataset contains 2.5M messages. *TagBERT* is trained for 38k steps with a peak learning rate of 1.6e-04 and a batch size of 512. The training on sixteen NVIDIA P100 GPUs lasted 8 hours.

F Results (details)

¹²Answers in our data come from two sources: CX consultants and sellers.

Dataset	<i>Purchase</i>					<i>Delivery</i>					<i>Retail</i>				
	macro F1	ACC	NMI	ARI	binary F1	macro F1	ACC	NMI	ARI	binary F1	macro F1	ACC	NMI	ARI	binary F1
Static	23	39	45	17	62	19	28	37	8	64	10	20	47	5	62
CDAC	33	50	61	30	77	27	39	50	16	72	15	32	57	10	67
Our	75	83	88	78	92	49	64	72	56	81	19	42	65	31	70

Table 8: Static baseline and CDAC representations compared with our framework on novel intent discovery task for real-world data. Our framework combines *TagBERT* pre-trained encoder, CDAC training scheme, and *Conv* method for using the conversation structure. Individual metrics averaged over five seeds.

Dataset	<i>Purchase</i>					<i>Delivery</i>					<i>Retail</i>				
	macro F1	ACC	NMI	ARI	binary F1	macro F1	ACC	NMI	ARI	binary F1	macro F1	ACC	NMI	ARI	binary F1
Initialization															
<i>HerBERT</i>	53	66	73	53	84	27	44	49	25	78	18	33	57	10	68
<i>AlleBERT</i>	54	67	74	52	86	33	46	58	32	77	17	42	65	27	70
<i>AlleConveRT</i>	60	74	83	67	83	46	57	64	41	81	20	48	71	36	72
<i>TagBERT</i>	75	83	88	78	92	49	64	72	56	81	19	42	65	31	70

Table 9: Impact of initialization for novel intent discovery. *Conv* conversation structure-aware encoder was trained with the CDAC scheme from different initialization. Individual metrics averaged over five seeds.

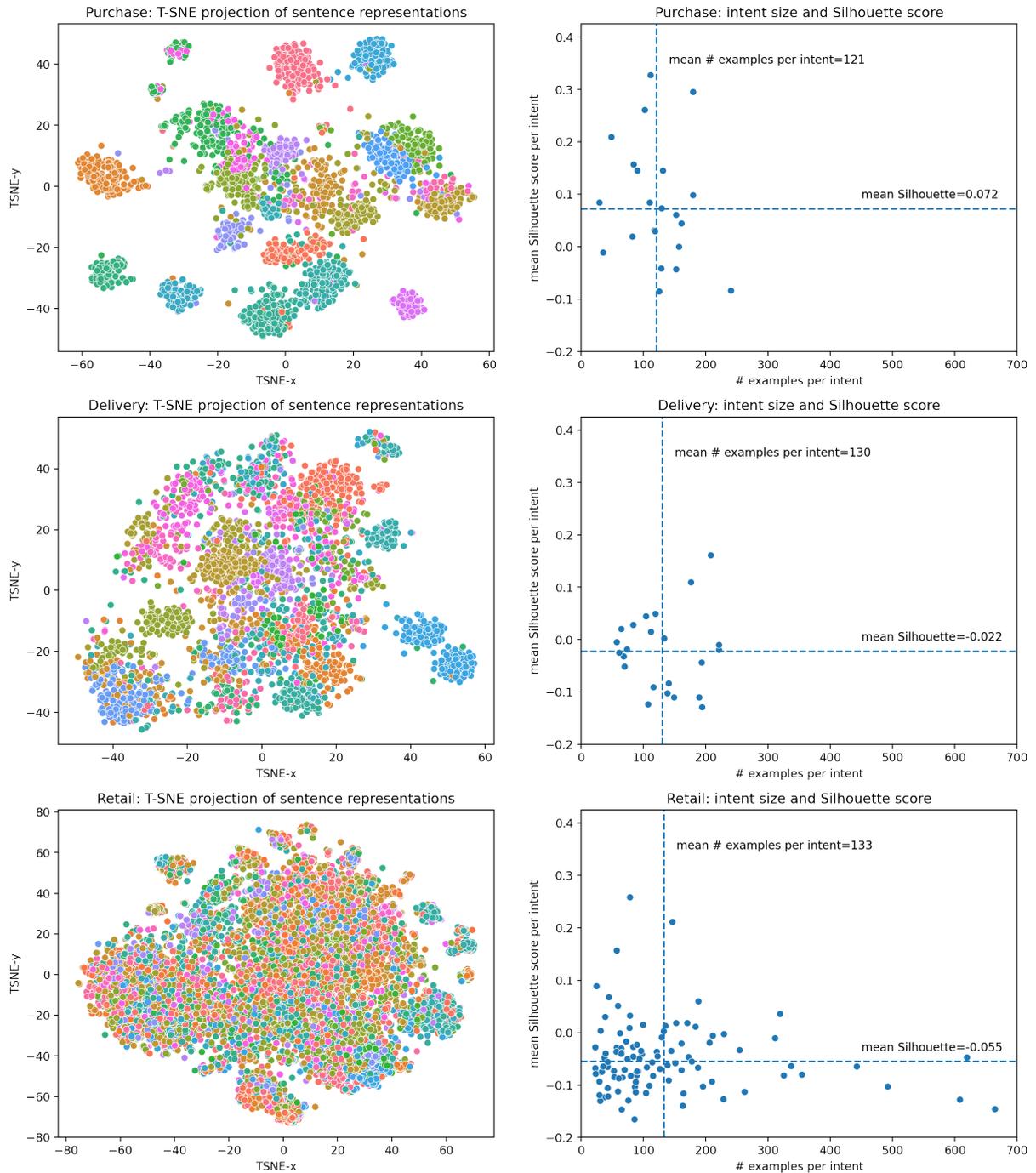


Figure 3: Internal dataset visualization. On the left we visualize t-SNE mapping of sentence representations to 2 dimensions. Different colors indicate different intent labels, each point corresponds to a single example in the dataset. On the right there is a scatter plot of intent sizes and Silhouette score per intent. Each point corresponds to one intent in the dataset. Silhouette score values are in the range from -1 to 1. 1 indicates perfect clustering, and 0 indicates overlapping clusters. The visualizations show the initial difficulty of the clustering task on general domain pre-trained models.

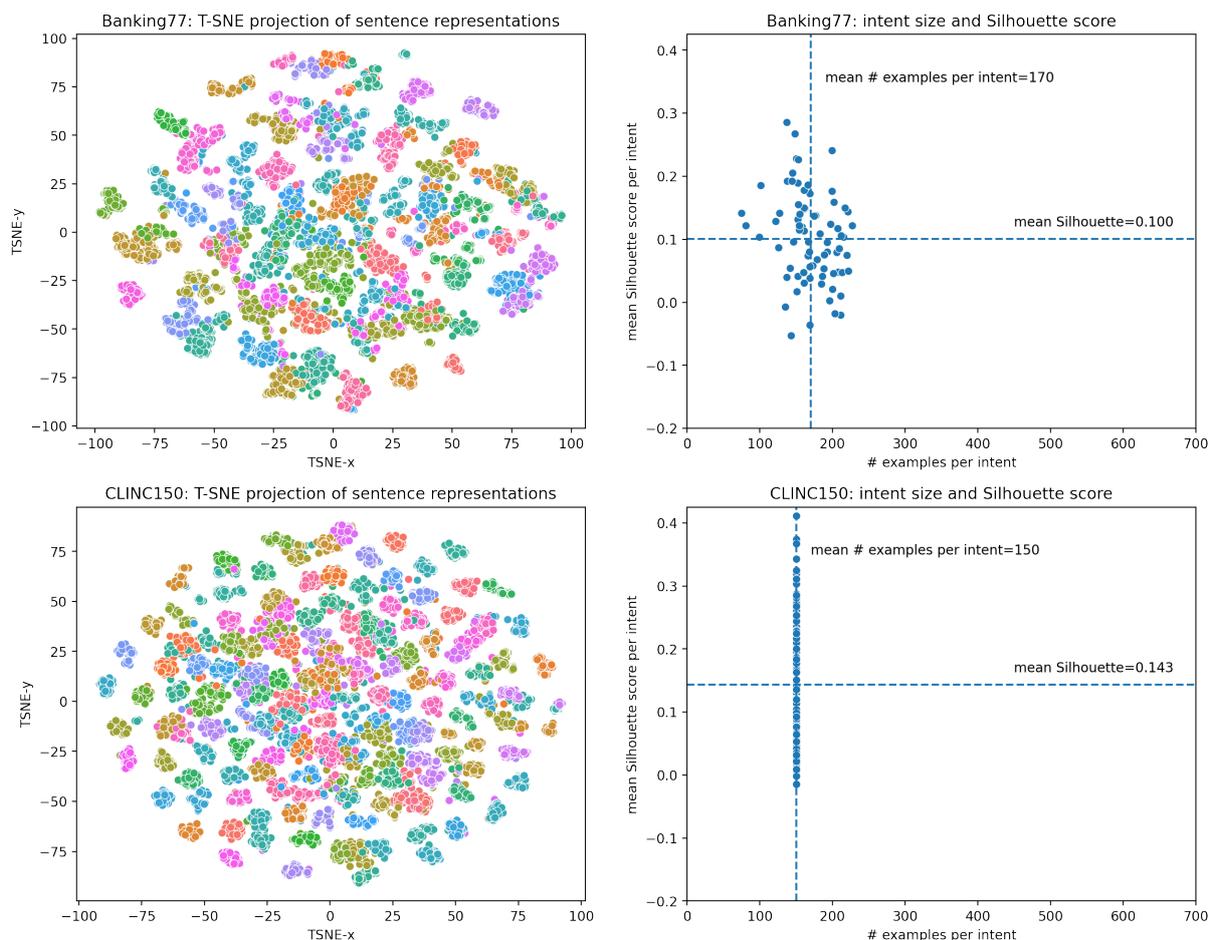


Figure 4: Public dataset visualization. On the left we visualize t-SNE mapping of sentence representations to 2 dimensions. Different colors indicate different intent labels, each point corresponds to a single example in the dataset. On the right there is a scatter plot of intent sizes and Silhouette score per intent. Each point corresponds to one intent in the dataset. Silhouette score values are in the range from -1 to 1. 1 indicates perfect clustering, and 0 indicates overlapping clusters. The visualizations show the initial difficulty of the clustering task on general domain pre-trained models.

	Dataset	Banking77	CLINC150	Purchase	Delivery	Retail
Static	macro F1	30	44	21	17	11
	ACC	33	49	36	30	22
	NMI	55	75	41	36	48
	ARI	23	36	14	9	6
	binary F1	68	75	65	63	62
DAC	macro F1	42	55	13	12	10
	ACC	45	58	22	20	17
	NMI	64	81	30	28	46
	ARI	35	49	0	1	3
	binary F1	73	80	55	59	61
Supervised	macro F1	55	64	22	19	11
	ACC	60	68	36	32	26
	NMI	76	86	46	38	45
	ARI	51	61	12	9	4
	binary F1	83	87	75	70	64
CDAC	macro F1	51	58	34	30	18
	ACC	54	66	54	42	35
	NMI	74	86	67	51	61
	ARI	47	59	36	17	14
	binary F1	82	83	74	72	68

Table 10: Impact of training schemes for novel intent discovery. Models use *BERT-base* (English datasets) or *AlleBERT* (Polish datasets) encoder and question input only. Individual metrics averaged over five seeds.

	Dataset	Purchase	Delivery	Retail
<i>Q</i>	macro F1	30	34	18
	ACC	54	42	35
	NMI	67	51	61
	ARI	36	17	14
	binary F1	74	72	68
<i>A</i>	macro F1	27	23	12
	ACC	55	35	24
	NMI	64	44	49
	ARI	42	15	6
	binary F1	70	71	61
<i>QA concat.</i>	macro F1	27	31	17
	ACC	59	45	41
	NMI	71	55	64
	ARI	48	26	26
	binary F1	71	80	69
<i>QA two head.</i>	macro F1	38	32	19
	ACC	56	44	36
	NMI	68	54	62
	ARI	44	28	16
	binary F1	75	76	69
<i>Conv</i>	macro F1	54	33	17
	ACC	67	46	42
	NMI	74	58	65
	ARI	52	32	27
	binary F1	86	77	70

Table 11: Impact of conversational structure for novel intent discovery. Models use *AlleBERT* initialization, CDAC training scheme, and various inputs, i.e., question *Q*, answer *A*, or both fields (*QA*) in three model variants; *QA concatenation*, *QA two heads*, and *Conv*. Individual metrics averaged over five seeds

DATScore: Evaluating Translation with Data Augmented Translations

Moussa Kamal Eddine¹, Guokan Shang², Michalis Vazirgiannis^{1,3}

¹École Polytechnique, ²Linagora, ³AUEB

Abstract

The rapid development of large pretrained language models has revolutionized not only the field of Natural Language Generation (NLG) but also its evaluation. Inspired by the recent work of BARTScore: a metric leveraging the BART language model to evaluate the quality of generated text from various aspects, we introduce DATScore. DATScore uses data augmentation techniques to improve the evaluation of machine translation. Our main finding is that introducing data augmented translations of the source and reference texts is greatly helpful in evaluating the quality of the generated translation. We also propose two novel score averaging and term weighting strategies to improve the original score computing process of BARTScore. Experimental results on WMT show that DATScore correlates better with human meta-evaluations than the other recent state-of-the-art metrics, especially for low-resource languages. Ablation studies demonstrate the value added by our new scoring strategies. Moreover, we report in our extended experiments the performance of DATScore on 3 NLG tasks other than translation Code is publicly available¹.

1 Introduction

Massive pretrained language models have brought significant improvement to NLG tasks (Lewis et al., 2020). Recent systems can even generate texts of higher quality than human-annotated ones (Peyrard, 2019). At the same time, standard metrics, such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), for translation and summarization respectively, have not evolved for the past two decades (Bhandari et al., 2020). These metrics rely on surface lexicographic matches, making them particularly unsuitable for evaluating modern systems operating with embeddings at the semantic level that often generate paraphrases (Ng and Abrecht,

2015). To address this issue, many metrics have been proposed (Sai et al., 2022), but none of them were widely adopted until the release of BERTScore (Zhang et al., 2019) and MoverScore (Zhao et al., 2019). These metrics take advantage of large pretrained language models like BERT (Devlin et al., 2019), which are now being used in nearly all NLP tasks (Qiu et al., 2020; Min et al., 2021).

In this work, we focus on the task of evaluating machine translation. We propose an extension of BARTScore (Yuan et al., 2021), a recent metric exploiting the BART seq2seq language model (Lewis et al., 2020) to evaluate the quality of generated text from various aspects. BARTScore covers four evaluation facets: Faithfulness, Precision, Recall, and F-score, derived from different generation directions between the *source* text, the *hypothesis* (the text generated by a system given the source), and the *reference* (the reference text for the generation, often provided by human annotators). The scores are obtained by pairing the three entities differently at the input or the output side of a trained seq2seq model for fetching conditional generation probabilities.

Based on BARTScore, and motivated by the general idea and positive effect of data augmentation techniques, we found that adding augmented, translated copies of the source and reference texts in BARTScore, can greatly help evaluate the quality of the hypothesis translation. We also propose two novel score averaging and term weighting strategies to improve the original score computing process of BARTScore. Results and ablation studies show that our metric DATScore (Data Augmented Translation Score) outperforms the other recent state-of-the-art metrics, and our new scoring strategies are effective. Moreover, the performance of DATScore is also reported on three other NLG tasks than translation: data-to-text, summarization, and image captioning.

To the best of our knowledge, no prior work has

¹https://github.com/moussaKam/dat_score

been done on leveraging data augmentation techniques for untrained NLG evaluation metrics. Our work will help fill this gap. Our contributions include:

- 1) Inspired by BARTScore, we developed DATScore, incorporating augmented data translated from the source and reference texts. DATScore is an untrained and unsupervised translation evaluation metric that offers a larger performance boost in evaluating low-resource language generation. In contrast to other widely adopted metrics, DATScore can efficiently incorporate both the source and reference texts in the evaluation.
- 2) We introduced a novel one-vs-rest method to average the scores for different generation directions with different weights, which improves over the simple arithmetic averaging method used in BARTScore.
- 3) We proposed a novel entropy-based scheme for weighting the target generated terms so that higher informative tokens receive more importance in accounting for the score, which outperforms the naive uniform weighting employed in BARTScore.

2 Related work

2.1 Translation evaluation metrics

BLEU (Papineni et al., 2002) is the de facto metric for evaluating machine translation. It simply calculates n -gram matching between the reference and the hypothesis using precision scores with a brevity penalty. METEOR (Banerjee and Lavie, 2005) was developed to address two drawbacks of BLEU. It is F-score based (thus taking recall into account) and allows for a more relaxed matching, based on three forms: extract unigram, stemmed word, and synonym with WordNet (Miller, 1994). Apart from the above word-based metrics, some approaches operate at the character level. For example, chrF (Popović, 2015) computes the overall precision and recall over the character n -grams with various values of n . More recently, static word embeddings (Mikolov et al., 2013) have enabled capturing the semantic similarity between two texts possible, of what the historical metrics are incapable. Several metrics have been proposed to incorporate word vectors. For example, MEANT 2.0 (Lo, 2017) evaluates translation adequacy by measuring the similarity of the semantic frames and their role fillers between the human and machine translations.

Lately, pretrained language models have become popular, because they provide context-dependent

embeddings. This proved beneficial to all NLP tasks, but also to evaluation metrics. For example, using a modified version of the Word Mover’s Distance (Kusner et al., 2015), the Sentence Mover’s Similarity (Clark et al., 2019) measures the minimum cost of transforming one text into the other as the evaluation score, where sentences are represented as the average of their ELMo word embeddings (Peters et al., 2018). BERTR (Mathur et al., 2019) computes approximate recall based on the pairwise cosine similarity between the BERT word embeddings (Devlin et al., 2019) of two translations. UniTE (Wan et al., 2022) proposes a unified framework for modeling three evaluation prototypes: estimating the quality of the translation hypothesis by comparing it with reference-only, source-only, or source-reference-combined data. UniTE is built upon XLM-R multilingual language model (Conneau et al., 2020).

Among several alternatives, BERTScore (Zhang et al., 2019) and MoverScore (Zhao et al., 2019) have received more attention, and have been adopted for reporting results in recent NLG publications (Lin et al., 2022; Weston et al., 2022). They both are unsupervised, general-purpose metrics and leverage BERT-like language models, however, with one difference lying in the similarity function for matching the two sequence representations. BERTScore greedily matches each token from one sequence to the single most similar token in the other sequence, in terms of the cosine similarity of their token embeddings. While MoverScore conducts soft one-to-many matching using an n -gram generalization of the Word Mover’s Distance (Kusner et al., 2015).

Finally, the work closely related to ours is BARTScore (Yuan et al., 2021). Unlike all the above metrics trying to match tokens or their embeddings, BARTScore proposes a novel conceptual view. It treats the evaluation of generated text as a text generation problem, with the help of a pretrained seq2seq model BART (Lewis et al., 2020). At the time of writing, this metric represents the state-of-the-art in the NLG evaluation. We will provide more details about it in Section 3.

2.2 Data augmentation

As deep learning models are often heavily reliant on large amounts of training data, a common attempt to get around the data scarcity problem is by applying data augmentation techniques (Shorten

and Khoshgoftaar, 2019). These techniques increase the size of the training set by making slightly modified copies of already-existing instances or by creating new, synthetic ones. Such augmented data have proven to be beneficial to the training of models in a wide variety of contexts, from computer vision (Shorten and Khoshgoftaar, 2019) to speech recognition (Bird et al., 2020), to NLP (Feng et al., 2021), as it acts as a regularizer and helps reduce overfitting (Krizhevsky et al., 2012). For dealing with textual data, a suite of augmentation techniques exists. To name only a few, backtranslation (Sennrich et al., 2016) translates a text into an intermediate language and then back into the original language, as a way of paraphrasing the initial text. Contextual augmentation (Kobayashi, 2018) generates augmented samples by randomly replacing words with others drawn following the in-context word distribution of a recurrent language model. SeqMix method (Guo et al., 2020) creates synthetic examples by softly mixing parts of two sentences via a convex combination.

Data augmentation has also been applied to the field of NLG evaluation metrics. BLEURT (Sellam et al., 2020) is a supervised metric, i.e., it requires to be finetuned on human meta-evaluations. Before finetuning, BLEURT creates an augmented synthetic dataset by perturbing Wikipedia sentences with BERT mask-filling, backtranslation, and random word dropping techniques. The data are then annotated with some automatic numerical and categorical signals as pretraining labels. FrugalScore (Kamal Eddine et al., 2022) proposes the first knowledge distillation approach for NLG evaluation metrics, to alleviate the significant requirement of computational resources by the heavy metrics based on large pretrained language models (e.g., BERTScore and MoverScore). Unlike BLEURT, it is purely trained on a synthetic dataset consisting of pairs of more or less related sentences, created via various data augmentation techniques (e.g., paraphrasing with backtranslation, perturbation then denoising, etc.). The sentence pairs for training the student model are annotated with scores given by the metrics to be learned.

Differences. Note that BLEURT and FrugalScore use augmented data to train their parameterized metric models, while our DATScore is an untrained and unsupervised metric not requiring human judgments for training and using augmented translation

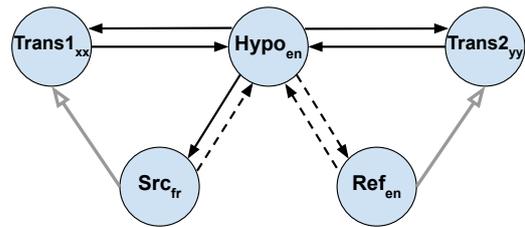


Figure 1: Dashed arrows denote the generation directions covered by BARTScore. Solid black arrows indicate our newly introduced directions for calculating DATScore of the example *hypothesis* in English ($Hypo_{en}$). $Trans1_{xx}$ and $Trans2_{yy}$ represent data *augmented translations* in any languages xx and yy , obtained by applying a translation model (grey arrows) to the example *source* in French (Src_{fr}) and example *reference* in English (Ref_{en}), respectively.

for the sole purpose of scoring.

3 DATScore

As mentioned in Subsection 2.1, BARTScore is not based on matching tokens nor their embeddings as the other evaluation metrics. Instead, it uses a novel approach by framing the evaluation of generated text as a text generation problem. Assuming first a pretrained seq2seq model is “ideal” (e.g., BART), BARTScore directly uses the model’s conditional probability of generating a provided target text Y given a provided input text X , as the evaluation score of the generation direction $X \rightarrow Y$. For example, Y corresponds to a translation hypothesis generated by any system, and X is the reference. If Y is of high quality, then by providing the pair to the pretrained BART model, the estimated conditional generation probability (evaluation score) $P(Y|X)$ should be high.

Therefore, by placing differently the *source* (Src), the *reference* (Ref), and the *hypothesis* ($Hypo$) in pair at the input or the output side of the trained seq2seq model for fetching conditional generation probabilities, BARTScore considers three different generation directions illustrated as dashed arrows in Figure 1. The conditional probabilities associated with the directions are denoted as: Precision ($Ref \rightarrow Hypo$), Recall ($Hypo \rightarrow Ref$) and Faithfulness² ($Src \rightarrow Hypo$). Additionally, an F-score, the arithmetic average of Precision and Recall.

The score (conditional probability) for the gen-

²BART being a monolingual model, faithfulness is only relevant in the context of abstractive summarization, and its corresponding direction cannot be applied to machine translation evaluation.

eration direction from a source sequence $X = \{x_t\}_{t=1}^n$ to a target sequence $Y = \{y_t\}_{t=1}^m$ is calculated as the factorized, weighted log probability over all generation steps:

$$\text{Score}_{X \rightarrow Y} = \sum_{t=1}^m w_t \log P(y_t | X, \{y_{t'}\}_{t'=1}^{t-1}; \theta) \quad (1)$$

where w_t denotes the term importance score to put different emphasis on different target tokens y_t . BARTScore simply employs a uniform weighting scheme (all equal to 1). θ denotes the parameterized seq2seq model.

Our contributions consist of three modifications tailored to machine translation:

Data augmented translations. Unlike BARTScore, we employ M2M-100 (Fan et al., 2021), a non-English-centric multilingual machine translation system as our backbone seq2seq model, due to its superior performance. As our main contribution, we translate the source (e.g., Src_{fr} in Figure 1) and the reference (Ref_{en}) into any languages as our augmented data ($\text{Trans}_{1_{xx}}$ and $\text{Trans}_{2_{yy}}$) for evaluating the hypothesis (Hypo_{en}). In addition to the three directions covered by BARTScore, our metric takes into consideration all generation directions centered on the hypothesis connecting the source, the reference, and the two data augmented translations, i.e., in total 8 directions as the black (dashed and solid) arrows depicted in Figure 1. DATScore is calculated as the weighted average of the scores associated with all the directions:

$$\text{DATScore} = \sum_{X, Y} w_{X \rightarrow Y} \text{Score}_{X \rightarrow Y}; X \neq Y \quad (2)$$

where $w_{X \rightarrow Y}$ denotes the weight of the direction $X \rightarrow Y$, as detailed below.

One-vs-rest score averaging method. We observed empirically that sometimes, one direction score might strongly disagree with the others, likely being an outlier (failed evaluation). This may significantly affect the final DATScore correlations with the human meta-evaluations, if a simple arithmetic averaging method is applied (like BARTScore in computing F-score). To reduce this effect, we weigh each direction with the sum of the Pearson correlations of its scores with the scores of all the other directions:

$$w_{X \rightarrow Y} = \sum_{X', Y'} \text{Corr}(\text{Score}_{X \rightarrow Y}, \text{Score}_{X' \rightarrow Y'}) \quad \text{s.t. } (X, Y) \neq (X', Y') \quad (3)$$

This one-vs-rest method will assign a low weight to the direction score poorly correlated with the rest scores, thus reducing its negative effect on the averaging result.

Entropy-based term weighting scheme. BARTScore gives an equal weight w_t to every token in Equation 1 (uniform weighting). Instead, we introduce a novel scheme to give different importance to different target tokens y_t , based on the entropy:

$$w_t = - \sum_{i=1}^v P_t(z_i) \log P_t(z_i) \quad (4)$$

where v denotes the size of the output generation vocabulary. $P_t(z_i)$ represents the probability of the i -th token in the vocabulary at time step t . We assume that when the model is very confident in generating the target token (low entropy), then this token is non-informative (e.g., stopword). On the other hand, when the model is less confident (higher entropy), the target word is more informative, and then a higher weight should be assigned.

The effectiveness of all our choices regarding the above contributions is shown by our ablation studies (see Section 6).

4 Experiments

4.1 Experimental settings

We benchmark DATScore on two commonly used meta-evaluation datasets for machine translation metrics: WMT17 (Bojar et al., 2017) and WMT18 (Ma et al., 2018) consisting of multiple to_English and from_English language pairs. For each pair, a few thousand examples are available, each being made of a *source*, a *reference*, a *hypothesis* and a *label* produced by human annotators, assessing the quality of the system generated *hypothesis*. Depending on the *label* type, we use Kendall’s Tau τ correlations or absolute Pearson $|r|$ correlations. The former is used when relative ranking is provided, and the latter in the case of direct assessment. We adopt the Kendall’s Tau-like formulation proposed in (Bojar et al., 2017):

$$\tau = \frac{|\text{Concordant}| - |\text{Discordant}|}{|\text{Concordant}| + |\text{Discordant}|} \quad (5)$$

Metric	Model	$ r $:cs \rightarrow en	$ r $:de \rightarrow en	$ r $:fi \rightarrow en	$ r $:lv \rightarrow en	$ r $:ru \rightarrow en	$ r $:tr \rightarrow en	$ r $:zh \rightarrow en	Avg.	
		/	/	/	/	/	/	/		
		τ :en \rightarrow cs	τ :en \rightarrow de	τ :en \rightarrow fi	τ :en \rightarrow lv	-	τ :en \rightarrow tr	-		
BLEU	1a) N/A	34.4/22.0	36.6/23.6	44.4/42.1	32.1/21.5	41.3/-	44.1/33.6	44.0/-	37.8/27.3	
BERTScore	1b) RL/mBERT	71.0/43.8	74.5/40.4	83.3/58.8	75.6/46.6	74.6/-	75.1/57.1	77.5/-	75.9/49.3	
MoverScore	1c) BB/mBERT	66.6/38.3	70.6/35.9	82.2/54.2	71.7/37.8	73.7/-	76.1/49.8	74.3/-	73.6/43.2	
BARTScore	1d) BL+para/mBART	68.4/39.0	70.8/33.4	79.4/50.4	74.9/50.4	71.8/-	73.9/53.8	76.0/-	73.6/45.4	
	1e) M2M-100_418M	65.9/45.0	66.1/44.5	79.9/59.2	71.7/40.3	69.0/-	71.8/70.9	71.6/-	70.9/52.0	
	1f) M2M-100_1.2B	67.4/49.6	69.3/49.2	80.7/63.5	73.7/46.9	70.4/-	71.6/ 72.5	73.0/-	72.3/56.3	
DATScore	1g) M2M-100_418M	68.6/51.1	68.5/48.1	82.0/63.7	74.7/48.3	73.0/-	77.6/70.9	76.5/-	74.4/56.4	
	1h) M2M-100_1.2B	71.3/53.9	72.9/52.2	83.5/66.3	76.8/52.0	75.9/-	78.1/70.9	77.7/-	76.6/59.1	

Table 1: Absolute Pearson correlation ($|r|$) for to-English and Kendall correlations (τ) for from-English with segment-level human scores on WMT17. BB stands of Bert-Base, RL for RoBERTa-Large and BL for BART-Large.

Metric	Model	τ :cs \rightarrow en	τ :de \rightarrow en	τ :et \rightarrow en	τ :fi \rightarrow en	τ :ru \rightarrow en	τ :tr \rightarrow en	τ :zh \rightarrow en	Avg.	
		/	/	/	/	/	/	/		
		τ :en \rightarrow cs	τ :en \rightarrow de	τ :en \rightarrow et	τ :en \rightarrow fi	τ :en \rightarrow ru	τ :en \rightarrow tr	τ :en \rightarrow zh		
BLEU	2a) N/A	23.3/38.9	41.5/62.0	38.5/41.4	15.4/35.5	22.8/33.0	14.5/26.1	17.8/31.1	24.8/38.3	
BERTScore	2b) RL/mBERT	40.4/55.9	55.0/72.7	39.7/58.4	29.6/53.9	35.3/42.4	29.2/38.9	26.4/36.1	36.5/51.2	
MoverScore	2c) BB/mBERT	36.8/44.6	53.9/68.4	39.4/52.7	28.7/50.9	27.9/40.1	33.6/32.5	25.6/35.2	35.1/46.3	
BARTScore	2d) BL+para/mBART	39.6/50.2	54.7/65.0	39.4/53.3	28.9/57.2	34.6/37.0	27.4/37.7	24.9/32.4	35.6/47.5	
	2e) M2M-100_418M	36.3/55.4	53.5/72.2	37.6/58.4	26.3/60.2	33.4/44.4	26.8/45.1	23.4/31.3	33.9/52.4	
	2f) M2M-100_1.2B	38.4/ 63.5	54.6/ 76.2	39.2/63.2	27.9/64.5	35.7/45.6	28.5/50.2	24.3/34.7	35.5/56.8	
DATScore	2g) M2M-100_418M	38.6/53.5	53.5/71.3	39.3/64.0	28.4/62.2	34.9/44.4	28.5/47.9	25.3/34.0	35.5/53.9	
	2h) M2M-100_1.2B	40.7/61.9	54.9/ 76.2	40.5/68.2	30.4/67.9	36.4/46.2	31.0/ 52.7	26.3/ 36.6	37.2/58.5	

Table 2: Kendall correlations (τ) for to-English and from-English with segment-level human scores on WMT18. BB stands of Bert-Base, RL for RoBERTa-Large and BL for BART-Large.

where $|Concordant|$ is the number of examples on which the metric agrees with the human relative ranking, and $|Discordant|$ is the number of examples when they disagree.

To compute DATScore, two M2M-100 models: M2M-100_418M³ and M2M-100_1.2B⁴ are adopted (418M and 1.2B refer to the model sizes). They are finetuned to translate a source text to a target text by providing the source language code (e.g. "fr") at the beginning of the encoder input sequence, and a target language code at the beginning of the decoder input sequence. In our experiments, when English is the target language (to-English), we choose English for Trans1 and Spanish for Trans2 (see Figure 1). Otherwise, whenever English is the source language (from-English), we choose Spanish for Trans1 and English for Trans2. This choice is motivated by the fact that English and

Spanish are the top two represented languages in the training set of M2M-100 (Fan et al., 2021).

4.2 Main results

We compare the performance of our metric against BLEU and three other reference-based unsupervised metrics: BERTScore⁵, MoverScore⁶ and BARTScore⁷ (detailed in Subsection 2.1 and Section 3), using their official implementations. Experimental results are reported in Table 1 and 2. Following their original settings, we use different underlying language models for each baseline metric. For BERTScore and MoverScore, RoBERTa-Large (RL; Liu et al., 2019) and Bert-Base (BB) are used respectively when we evaluate to-English translations, and mBERT (Devlin et al., 2019) for from-English translations. In the case of BARTScore, we use a BART-Large (BL) checkpoint (finetuned on CNNDM (See et al., 2017) and

³https://huggingface.co/facebook/m2m100_418M

⁴https://huggingface.co/facebook/m2m100_1.2B

⁵https://github.com/Tiiiger/bert_score

⁶<https://github.com/AIPHES/emnlp19-moverscore>

⁷<https://github.com/neulab/BARTScore>

ParaBank2 (Hu et al., 2019) datasets) for evaluating to-English translations, and an mBART-50 model (Escolano et al., 2021) for from-English translations.

Overall, results show that, on average, across all language pairs, DATScore significantly outperforms all 4 baseline metrics under their original model settings (rows 1a-1d and 2a-2d). Specifically, with respect to the best performing baseline BERTScore (row 1b and 2b), our metric provides a performance boost of 0.7 for to-English case and of 9.8 for from-English case on WMT17 dataset in Table 1, and achieves a gain of 0.7 and of 7.3 respectively on WMT18 dataset in Table 2. These averaging results demonstrate the superiority and applicability of DATScore in evaluating general machine translations of many languages. Moreover, it is interesting to note that our improvement is much more significant in from-English case, which makes DATScore particularly well-suited to evaluate hypothesis translations in non-English languages, often with low resource. We hypothesize that this is due to the inconsistency of underlying language models. The baselines adopt a monolingual model for evaluating English, but a multilingual one for non-English languages. However, DATScore uses a single multilingual M2M-100 model for both cases. It is known that, in general, monolingual models outperform multilingual competitors. Thus, it is reasonable that when comparing multilingual-based DATScore against monolingual baselines in the to-English case, DATScore achieves a smaller improvement than in the other from-English case, where the comparison is fairer (multilingual vs. multilingual).

By looking across specific language pairs and directions, we observe DATScore constantly performs better than 4 baseline metrics with a few exceptions, i.e., de \rightarrow en (-1.6) in Table 1, and de \rightarrow en (-0.1), tr \rightarrow en (-2.6), and zh \rightarrow en (-0.1) in Table 2. Despite these small drops in the performance, DATScore brings a larger margin of improvement in most cases, such as en \rightarrow tr up to 13.8 both on WMT17 and WMT18 datasets.

In the end, for the sake of having a complete comparison, we additionally evaluate BARTScore⁸ with M2M-100_418M and M2M-100_1.2B models (row 1e, 1f, 2e, and 2f) that are used as DATScore’s underlying models. Results show that, only in the

⁸The official implementation of BARTScore is slightly modified to take into account the languages tokens when using a multilingual model.

Metric	Model	WebNLG		
		SEMA	GRAM	FLU
BLEU	N/A	45.5	36.0	34.9
BERTScore	RoBERTa-Large	56.1	60.8	54.8
MoverScore	BERT-Base	-9.9	-27.8	-20.6
BARTScore	BART-Large+para	71.9	61.3	57.4
	M2M-100_418M	64.9	62.8	56.0
	M2M-100_1.2B	66.1	63.9	57.2
DATScore	M2M-100_418M	69.9	62.9	57.2
	M2M-100_1.2B	70.4	63.7	57.9

Table 3: Pearson correlation results on WebNLG dataset.

Metric	Model	REALSumm	SummEval			
		COV	COH	CONS	FLU	REL
BLEU	N/A	37.9	11.8	6.3	7.7	18.6
BERTScore	RoBERTa-Large	41.2	33.9	10.5	15.0	35.9
MoverScore	BERT-Base	44.1	14.4	14.7	13.8	29.1
BARTScore	BART-Large+para	31.7	20.8	-3.5	6.7	22.2
	M2M-100_418M	30.1	14.8	-2.3	3.0	19.8
	M2M-100_1.2B	32.0	17.1	1.1	6.7	22.8
DATScore	M2M-100_418M	44.7	17.1	4.4	4.6	26.3
	M2M-100_1.2B	45.5	19.5	6.8	8.2	30.2

Table 4: Pearson correlation results on two summarization datasets: REALSumm and SummEval.

from-English case, while they bring an improvement compared to the vanilla BARTScore (row 1d and 2d), they are not able to yield as big of a gain as our metric, indicating that our achieved improvement is not solely due to the underlying language model, but also to taking additional generation directions into account, including those related to data augmented translations.

5 Other NLG tasks

In addition to machine translation, our main focus, we evaluate DATScore on other NLG tasks, including data-to-text generation, abstractive summarization, and image captioning. To work around the different modalities of source inputs represented in these tasks (e.g., not able to create a data augmented translation with an image), we adapt DATScore to only consider 4 generation directions: *Hypo* \leftrightarrow *Ref* and *Hypo* \leftrightarrow *Trans2*.

Data-to-text. Table 3 shows the performance of DATScore compared to the other baselines on the WebNLG data-to-text dataset (Shimorina et al., 2018), which contains 2000 descriptions of structured tables along with their corresponding references. In addition, human assessments covering three dimensions are provided (*semantics*, *gram-*

mar, and fluency). The results show that DATScore significantly outperforms all the other metrics in two settings (grammar and fluency) out of three, while being very competitive in the third setting (semantics). Surprisingly, BERTScore is largely behind DATScore, and MoverScore failed to correlate positively with human judgments in all dimensions.

Summarization. Table 4 shows the evaluation of the different metrics on two summarization meta-evaluation datasets: REALSumm (Bhandari et al., 2020) and SummEval (Fabbri et al., 2021). Both datasets contain a few thousand examples of system-generated summaries and their references. The generated summaries are annotated with *lightweight pyramids* (Shapira et al., 2019) method in the case of REALSumm, while the annotations in SummEval cover four dimensions: *coherence*, *consistency*, *fluency*, and *relevance*. On REALSumm, DATScore has the best performance compared to all the other baselines even when using its smaller version (M2M-100_418M). However, despite its higher correlations compared to BARTScore and MoverScore, DATScore fails to outperform BERTScore on the different dimensions of SummEval.

Image captioning. We consider Flickr8K (Hodosh et al., 2013) and PASCAL-50S (Vedantam et al., 2015), two image captioning datasets. The former is annotated with scores from 1 to 4 assessing the relevance of the captions, and the latter is annotated with relative ranking (i.e., given two descriptions which one is better). Table 5 shows that in this task, DATScore is competitive to BARTScore and BERTScore. Surprisingly, MoverScore significantly outperforms all the other metrics despite its poor performance on the other datasets.

Finally, although not the top-performing metric across all tasks, DATScore showed an overall stable and competitive performance. Conversely, each of the other metrics fails in evaluating generations, at least in one of the tasks. For example, BERTScore and MoverScore have poor performance on the WebNLG dataset. On the other hand, although BARTScore is finetuned on an abstractive summarization dataset, it fails to correlate well with human judgment on REALSumm and SummEval. This finding suggests that DATScore can be safely used to evaluate NLG systems in other tasks for different evaluation dimensions, regardless of being initially designed for machine translation evaluation.

Metric	Model	Flickr8K	PASCAL-50S
		RELE	RR
BLEU	N/A	13.8	8.1
BERTScore	RoBERTa-Large	46.1	33.8
MoverScore	BERT-Base	52.5	33.2
BARTScore	BART-Large+para	44.8	33.1
	M2M-100_418M	34.3	29.6
	M2M-100_1.2B	34.6	26.3
DATScore	M2M-100_418M	42.6	29.6
	M2M-100_1.2B	45.3	31.4

Table 5: Pearson correlation Results on two Image Captioning datasets: Flickr8K and PASCAL-50S.

Entropy-based weighting	One-vs-rest weighting	to_English	from_English
✓	✓	37.2	58.5
✓	✗	37.1	58.1
✗	✓	36.4	55.9
✗	✗	36.4	56.0

Table 6: The average Kendall correlation (to/from)-English when the entropy-based and one-vs-rest weighting are included or excluded. Experiments are conducted on WMT18.

6 Ablation study

To validate our different choices with regard to DATScore, we conducted ablation studies on:

- 1) the contributions of all 8 direction scores, results are illustrated in Figure 2.
- 2) the effectiveness of our *one-vs-rest* score averaging and *entropy-based* term weighting strategies (See Section 3), results are reported in Table 6.

Contributions of all direction scores. From Figure 2(a), we observe that none of the individual directions (horizontal bars) has a better correlation with human judgments than DATScore (dashed vertical lines), which confirms the importance of our ensemble approach. In Figure 2(b), we can see that all variants excluding one direction will lead, in almost all cases, to a drop in the performance, compared to the complete DATScore in which all directions are included. Besides, in the case of to-English translations, we can see that the drop in the performance is almost the same for all exclusions of direction. While for from-English translations, the largest drop in performance is observed when *Hypo*→*Trans2* and *Trans2*→*Hypo* are excluded. This finding highlights the important contribution of our augmented data, especially in the

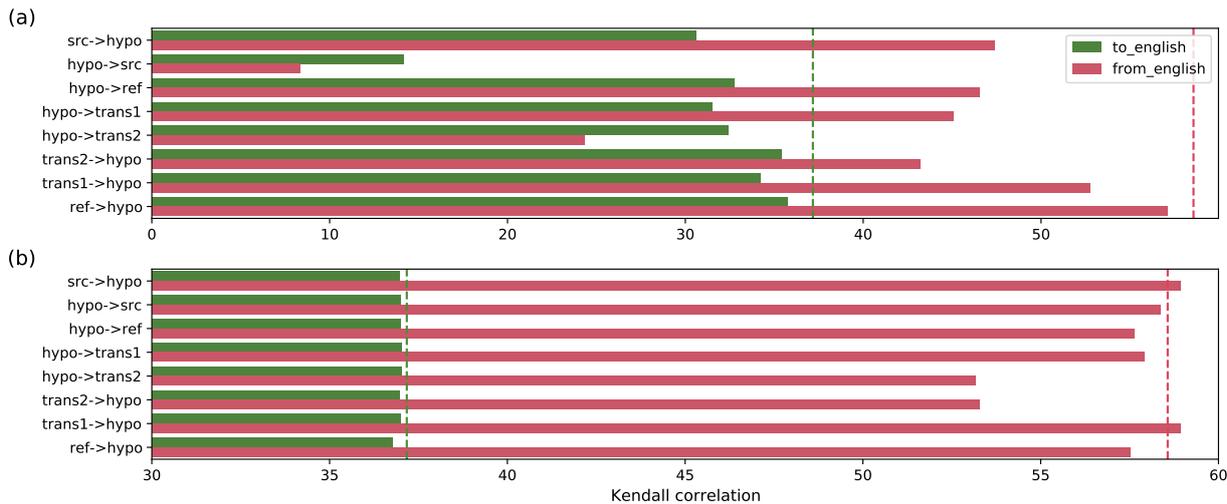


Figure 2: (a): The horizontal bars represent the Kendall correlations of **each individual generation direction**. (b): The horizontal bar represents the Kendall correlation of **a variant of DATScore with excluding the single generation direction** of the line. Both in (a) and (b), the dashed vertical lines represent the Kendall correlation of the vanilla and **complete DATScore**. Correlation results of to-English (in green) and from-English (in red) cases are calculated w.r.t human judgments, and averaged over all languages pairs. Experiments are conducted on WMT18.

low resource language settings (from-English). In the end, we can see that excluding *Src*→*Hypo* or *Trans1*→*Hypo* directions can lead to a slightly better final score. We leave the investigation of the potential negative impact of the two directions to future work.

One-vs-rest and entropy-based weighting strategies. Table 6 shows the performance of DATScore variants with respect to different combinations of applying or not our proposed weighting strategies. Note that when *one-vs-rest* and *entropy-based* weightings are not applied, they are replaced with a simple uniform averaging approach (as used in BARTScore). A performance drop is observed when excluding one of the two weighting strategies, especially for the entropy-based method, whose inclusion leads to an improvement of 2.5 compared to the uniform weighting. This experiment confirms the positive impact of our proposed weighting methods and motivates future work further to investigate a more elaborated approach in this direction.

7 Conclusion

In this work, we proposed one of the first applications of data augmentation techniques to NLG evaluation. To obtain an evaluation score of the translation hypothesis, our developed metric DATScore additionally leverages newly translated copies aug-

mented from the source and reference texts. We also proposed two novel strategies for score averaging and term weighting to improve the original, naive score computing process of BARTScore, on the basis of which our work is built. Experimental results show that DATScore achieved a higher correlation with human meta-evaluations, in comparison with the other recent state-of-the-art metrics, especially for those less represented languages other than English. Moreover, ablation studies show the effectiveness of our newly proposed score computing approaches, and extended experiments showed an overall stable and competitive performance of DATScore on more NLG tasks.

Limitations

In this section, we list some limitations that are worth further investigation in future works:

1) DATScore requires generating additional data augmented translations to perform the evaluation. This process might be time-consuming depending on the adopted backbone seq2seq model, especially if the original text is long. Thus, the performance scalability can be investigated in future complementary experiments.

2) We chose to use English and Spanish to create data augmented translations for the reason that they are the two most represented languages in the training of the M2M-100 model (see Subsection 4.1).

However, this leaves a question about the performance of DATScore with augmentations varying in other languages (e.g., Chinese). Moreover, for the sake of simplicity, we decided only to include a single translated copy of the source text and the reference text. However, this can be easily extended, and more augmented translations can be created in more languages. We expect to see an improvement in performance with diminishing returns.

3) BARTScore only considers the 8 generation directions centered on the hypothesis connecting with the source, the reference, and the two data augmented translations (see Section 3). However, other connections exist between these entities, such as $Src \rightarrow Ref$ and $Trans1 \rightarrow Src$ (see Figure 1). Therefore, future research could be dedicated to discovering the effect of these other directions and potentially leveraging them to improve the performance of DATScore.

4) Since our focus was on evaluating machine translation, we naturally chose translation for augmenting the data. However, other data augmentation techniques could seamlessly integrate into DATScore, such as using a text paraphrasing model (Bandel et al., 2022).

References

- Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. 2022. [Quality controlled paraphrase generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 596–609, Dublin, Ireland. Association for Computational Linguistics.
- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. [Re-evaluating evaluation in text summarization](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.
- Jordan J Bird, Diego R Faria, Cristiano Premebida, Anikó Ekárt, and Pedro PS Ayrosa. 2020. Overcoming data scarcity in speaker identification: Dataset augmentation with synthetic mfccs via character-level rnn. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 146–151. IEEE.
- Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. [Results of the WMT17 metrics shared task](#). In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.
- Elizabeth Clark, Asli Celikyilmaz, and Noah A. Smith. 2019. [Sentence mover’s similarity: Automatic evaluation for multi-sentence texts](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760, Florence, Italy. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2021. [Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 944–948, Online. Association for Computational Linguistics.
- Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. [SummEval: Re-evaluating summarization evaluation](#). *Transactions of the Association for Computational Linguistics*, 9:391–409.
- Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(107):1–48.
- Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Edouard Hovy. 2021. [A survey of data augmentation approaches for NLP](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*,

- pages 968–988, Online. Association for Computational Linguistics.
- Demi Guo, Yoon Kim, and Alexander Rush. 2020. [Sequence-level mixed sample data augmentation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552, Online. Association for Computational Linguistics.
- Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899.
- J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. 2019. [Large-scale, diverse, paraphrastic bitexts via sampling and clustering](#). In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 44–54, Hong Kong, China. Association for Computational Linguistics.
- Moussa Kamal Eddine, Guokan Shang, Antoine Tixier, and Michalis Vazirgiannis. 2022. [FrugalScore: Learning cheaper, lighter and faster evaluation metrics for automatic text generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1305–1318, Dublin, Ireland. Association for Computational Linguistics.
- Sosuke Kobayashi. 2018. [Contextual augmentation: Data augmentation by words with paradigmatic relations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Haitao Lin, Junnan Zhu, Lu Xiang, Yu Zhou, Jiajun Zhang, and Chengqing Zong. 2022. [Other roles matter! enhancing role-oriented dialogue summarization via role interactions](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2545–2558, Dublin, Ireland. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Chi-kiu Lo. 2017. [MEANT 2.0: Accurate semantic MT evaluation for any output language](#). In *Proceedings of the Second Conference on Machine Translation*, pages 589–597, Copenhagen, Denmark. Association for Computational Linguistics.
- Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. [Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance](#). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.
- Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. [Putting evaluation in context: Contextual embeddings improve machine translation evaluation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- George A. Miller. 1994. [WordNet: A lexical database for English](#). In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.
- Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *arXiv preprint arXiv:2111.01243*.
- Jun-Ping Ng and Viktoria Abrecht. 2015. [Better summarization evaluation with word embeddings for ROUGE](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia,

- Pennsylvania, USA. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Maxime Peyrard. 2019. [Studying summarization evaluation metrics in the appropriate scoring range](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100, Florence, Italy. Association for Computational Linguistics.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.
- Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.
- Ananya B Sai, Akash Kumar Mohankumar, and Mitesh M Khapra. 2022. A survey of evaluation metrics used for nlg systems. *ACM Computing Surveys (CSUR)*, 55(2):1–39.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Ori Shapira, David Gabay, Yang Gao, Hadar Ronen, Ramakanth Pasunuru, Mohit Bansal, Yael Amsterdamer, and Ido Dagan. 2019. [Crowdsourcing lightweight pyramids for manual summary evaluation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 682–687, Minneapolis, Minnesota. Association for Computational Linguistics.
- Anastasia Shimorina, Claire Gardent, Shashi Narayan, and Laura Perez-Beltrachini. 2018. *WebNLG challenge: Human evaluation results*. Ph.D. thesis, Loria & Inria Grand Est.
- Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.
- Yu Wan, Dayiheng Liu, Baosong Yang, Haibo Zhang, Boxing Chen, Derek Wong, and Lidia Chao. 2022. [UniTE: Unified translation evaluation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8117–8127, Dublin, Ireland. Association for Computational Linguistics.
- Jack Weston, Raphael Lenain, Udeepa Meepegama, and Emil Fristed. 2022. [Generative pretraining for paraphrase evaluation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4052–4073, Dublin, Ireland. Association for Computational Linguistics.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.
- Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. [MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

How do decoding algorithms distribute information in dialogue responses?

Saranya Venkatraman
Pennsylvania State University
saranyav@psu.edu

He He
New York University
hhe@nyu.edu

David Reitter
Google Research
reitter@google.com

Abstract

Humans tend to follow the Uniform Information Density (UID) principle by distributing information evenly in utterances. We study if decoding algorithms implicitly follow this UID principle, and under what conditions adherence to UID might be desirable for dialogue generation. We generate responses using different decoding algorithms with GPT-2 on the Persona-Chat dataset and collect human judgments on their quality using Amazon Mechanical Turk. We find that (i) surprisingly, model-generated responses follow the UID principle to a greater extent than human responses, and (ii) decoding algorithms that promote UID do not generate higher-quality responses. Instead, when we control for surprisal, non-uniformity of information density correlates with the quality of responses with very low/high surprisal. Our findings indicate that encouraging non-uniform responses is a potential solution to the “likelihood trap” problem (quality degradation in very high-likelihood text). Our dataset containing multiple candidate responses per dialog history along with human-annotated quality ratings is available at: https://huggingface.co/datasets/saranya132/dialog_uid_gpt2.

1 Introduction

The Uniform Information Density (UID) hypothesis states that humans distribute information in their utterances evenly for optimal communication (Jaeger, 2010; Fenk and Fenk, 1980). Consequently, language generation has benefitted from UID-based objectives and regularization (Meister et al., 2022; Wei et al., 2021). Specifically, Meister et al. (2020) argued that UID can be optimized for machine translation using beam search. Yet, the effect of different decoding algorithms on information density distributions of generated text are unknown, as is UID’s broader role in neural response generation in the special case of dialogue

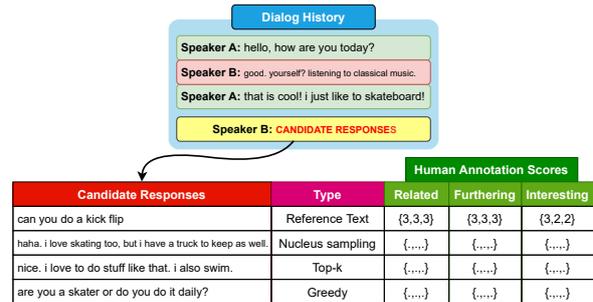


Figure 1: Our dataset contains 4 candidate responses for every dialog history, along with human annotations for 3 qualitative measures.

models. Here, we investigate (i) if different decoding algorithms follow the UID principle, and (ii) if following the UID principle is beneficial for dialogue response generation, and (iii) collect human annotations of qualitative measures for multiple candidate responses to dialog histories generated using different decoding algorithms (Figure 1) to study the relationship of dialog response quality and UID. We operationalize UID as the variance of surprisal and measure its correlation with automatic metrics (e.g., BLEU, METEOR, BERTScore) as well as human judgments on qualitative measures of response quality and find that adherence to UID correlates negatively with human judgments when the responses have very low/high surprisal.

Language production in humans. Spreading information content evenly in utterances is a marker of optimally strategized responses, and humans follow this UID principle as a means to state their thoughts clearly and to make themselves intelligible (Frank and Jaeger, 2008; Levy and Jaeger, 2007). The probability of a sentence has been associated with the cognitive load it incurs (Hale, 2003). As a means to avoid salient variations in the information content (surprisal, i.e., negative log probability) of responses, speakers maintain UID through linguistic choices such as that at the

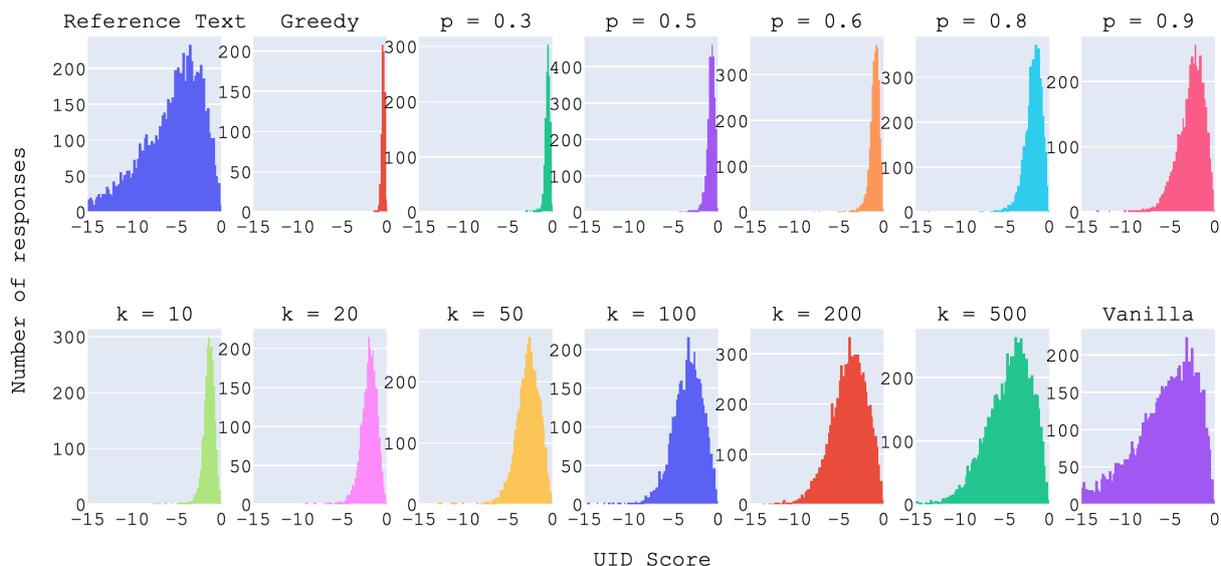


Figure 2: Histogram of **UID Scores** of responses generated using different decoding algorithms. The farther the UID score from 0, the less uniform or more non-uniform the response. Human-generated reference text (left-top) has a higher frequency of non-uniform responses as compared to any model setting as can be seen from the wider spread of scores away from 0. Also, as the values of p and k increase (left to right), the information density distribution slowly approaches reference text-like non-uniformity.

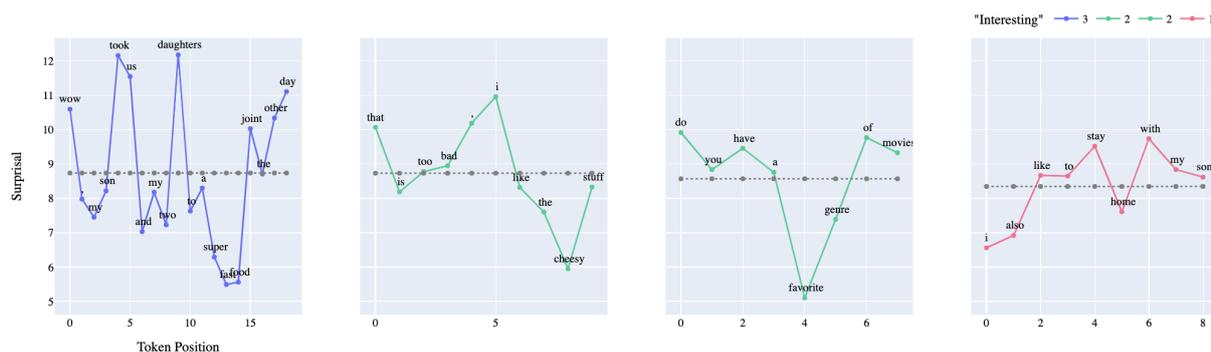


Figure 3: Surprisal at every token in candidate responses to the same dialog history, color-coded with human annotated **interesting** scores. Plots (left to right) are arranged in increasing order of uniformity (i.e. variance along y-axis). Less uniform the surprisal (left-most), better the score.

phonetic (Aylett and Turk, 2004), syntactic (Jaeger, 2010) and lexical level (Mahowald et al., 2013).

Response generation in machines. While large-scale pre-trained language models provide a rich prior for dialogue response generation, the choice of decoding algorithm used at the time of generation is crucial for the quality of generated responses (Holtzman et al., 2020; Zhang et al., 2021a; Nadeem et al., 2020; Golovanov et al., 2019; Oluwatobi and Mueller, 2020). While vanilla sampling often tends to produce incoherent text, greedy decoding leads to safe and repetitive responses. More recently, top- p /nucleus (Holtzman et al., 2020) and top- k sampling (Fan et al., 2018) are used to tune values of p/k to balance the diversity-

quality trade-off (Zhang et al., 2021a; Li et al., 2016).

The UID principle and decoding algorithms.

Both the UID principle and decoding algorithms can be seen as guiding mechanisms for dialogue response production in humans and generation in machines, respectively. UID’s role in machine-generated dialogue is not well understood, with previous work mainly focused on machine translation and language modeling (Wei et al., 2021; Meister et al., 2021, 2020). To address this gap, we present a comparative study of decoding methods to develop a deeper understanding of the role of UID in dialogue response generation.

2 Experimental Details

2.1 Model & dataset

We use the fine-tuned GPT-2 (Radford et al., 2019) model provided by HuggingFace and use their data preprocessing and response generation scripts¹. We used the Persona-Chat (Zhang et al., 2018) data split provided by the ConvAI2 challenge (Dinan et al., 2020)². We then generated responses for 7500 dialogue histories randomly picked from 7801 validation set examples using vanilla, top- p , top- k sampling and greedy decoding.

Decoding algorithms. *Vanilla sampling* randomly picks the next token from the model’s probability distribution, including many long-tail samples. *Top- k* samples from the k most probable tokens; *Greedy decoding* is Top- $k = 1$ decoding, always selecting the most probable next token. *Top- p* (*Nucleus*) sampling selects the next token from the top p portion of the probability mass.

2.2 Uniform Information Density score

We measure UID as the variance of the surprisal (negative log likelihood) of each token in the response (Jain et al., 2018; Wei et al., 2021; Meister et al., 2020). This measure is able to capture any sudden variations in the surprisal of the tokens in the sentence. UID Score is formulated as follows: the dialogue model learns a conditional probability p parameterized by θ to predict the next token (y_t) in the sentence. The surprisal (u) of the next token y_t is,

$$u(y_t) = -\log(p_\theta(y|x, y < t)), \quad (1)$$

for $t \geq 1$ where $y_0 = \langle EOS \rangle$, $t =$ time step, and $x =$ dialogue context. Higher the surprisal, lower its probability and vice-versa. Thus, surprisal indicates how unexpected or surprising a token is in a given context. Average surprisal of a sentence (y) is defined as,

$$\mu(y) = \frac{1}{|y|} \sum_t (u(y_t)) \quad (2)$$

Finally, the *UID score* of a sentence (y) is defined as the negative normalized variance of the surprisal:

$$\text{UIDscore}(y) = -\frac{1}{|y|} \sum_t (u(y_t) - \mu)^2 \quad (3)$$

¹<https://github.com/huggingface/transfer-learning-conv-ai>

²<https://github.com/DeepPavlov/convai/tree/master/2018>

From this formulation, a perfectly uniform sentence would have a variance equal to 0 (i.e. the surprisal of every token in the sentence is equal). Since we take the negative of the variance, the higher the absolute value of UID score, the more non-uniform its information density.

2.3 Response evaluation

Automatic metrics. We measure the quality of responses using length (number of tokens), BLEU³ (Papineni et al., 2002), METEOR³ (Banerjee and Lavie, 2005), character level F-score (chrF)³ (Popović, 2015), BLEURT⁴ (Sellam et al., 2020), a RoBERTa (Liu et al., 2019) based text similarity score⁵ (Reimers and Gurevych, 2019), BERTscore⁴ (Zhang et al., 2019) and SacreBLEU⁴ (Post, 2018).

Human evaluation. To study the effect of adherence to UID on the perceived quality of generated responses beyond n-gram, reference-based and learned automatic metrics, we collected human judgments along 3 measures – **related** (to the dialogue history), **furthering** (if a response keeps the conversation going/is encouraging for the dialogue partner) and **interesting** (if the response provides engaging/new information). We provide screenshots of the task interface (Figure 6), instructions (Figure 7) and details about the MTurk study design in Appendix A.

3 Findings

3.1 Information density of model responses

We plot the histograms of UID scores computed for all of the generated responses in Figure 2. The information densities of human-generated responses have a wider spread than responses produced by the models. Overall, the human-generated reference text has more non-uniform sentences than all model-generated responses. We notice a very high and narrow peak in the case of greedy decoding. This is not surprising as responses sampled using greedy search maximize the probability of the next token (minimize surprisal). Consequently, such responses would have very low surprisal at almost every word, hence lower variance. Vanilla

³<https://github.com/nltk/nltk/tree/develop/nltk/translate>

⁴<https://github.com/huggingface/datasets/tree/master/metrics>

⁵https://github.com/UKPLab/sentence-transformers/blob/master/docs/usage/semantic_textual_similarity.md

Generation Type	Pearson’s r between UID score and automatic metrics							
	Length	BLEU	chrF	METEOR	BertScore	BLEURT	RoBERTa	SacreBLEU
$p = 0.3$	-.10	.00	.14	.12	.17	.17	0.19	.13
$p = 0.5$	-.05	.03	.13	.10	.18	.17	.2	.15
$p = 0.6$	-.04	.06	.14	.13	.01	.06	.01	.00
$p = 0.8$	-.10	.03	.06	.05	.18	.16	.2	.15
$p = 0.9$	-.11	-.00	.03	.04	.16	.15	.19	.14
Greedy	-.14	.01	.14	.13	.06	.05	.06	.06
$k = 10$	-.04	.15	.03	.05	.07	.08	.07	.07
$k = 20$	-.05	.14	.05	.06	.05	.04	.06	.04
$k = 50$	-.09	.01	.03	.03	.06	.03	.03	.05
$k = 100$	-.07	.04	.00	.02	.11	.08	.08	.08
$k = 200$	-.12	.03	.02	.03	.06	.06	.04	.05
$k = 500$	-.09	.02	.04	.04	.10	.08	.08	.08
Vanilla	-.09	.01	-.00	.00	.07	.05	.05	.05

Table 1: Pearson’s correlation coefficient (r) between **UID score and automatic metrics** of dialog responses generated using different decoding settings. All p -values < 0.05 .

sampling uses the probability distribution learned from the training data, which might be why it is also closer to the validation set (reference text) distribution. With increase in p and k , we see that the information density distribution spreads across a larger range and includes more non-uniform responses, slowly approaching that of the reference text.

Surprisal interval	n	Pearson’s r between UID score and qualitative metrics		
		Related	Furthering	Interesting
(0.8, 1.2)	24	.17	-.03	-.30*
(1.2, 1.6)	64	.12	.08	-.13
(1.6, 2.0)	91	.05	-.23*	-.07
(2.0, 2.4)	109	-.04	-.13	-.00
(2.4, 2.8)	111	-.06	-.21*	-.05
(2.8, 3.2)	105	-.02	.01	-.10
(3.2, 3.6)	99	-.23*	-.10	.19
(3.6, 4.0)	66	.03	-.05	-.09
(4.0, 4.4)	42	-.33	-.22	-.09
(4.4, 4.8)	24	-.14	-.61*	.04
(4.8, 5.2)	12	-.33	-.14	-.54*
(5.2, 5.6)	13	-.98*	-.64	-.38

Table 2: Pearson’s r between **UID score and human judgments** of qualitative measures for dialog responses bucketed by surprisal [Surprisal interval = the ranges of surprisal values used for bucketing responses, n = number of responses in each surprisal interval, * p -value $< .05$]

3.2 UID score & automatic metrics

We present the correlation between UID scores and automatic metrics calculated for the generated dialogue responses in Table 1. UID scores have a weak correlation with RoBERTa-based similarity scores

for two settings of nucleus sampling. Other than that, UID scores are not correlated with automatic metrics of response generation. We take this to be an indication that if UID scores do capture any aspect of response quality, it goes beyond what is measured by such metrics and might provide for a better evaluation criteria.

3.3 UID score & human Judgments

Motivated by the fact that UID score is derived from surprisal, we test if surprisal is a confounding factor and find that, indeed, UID scores were highly correlated with average surprisal (Table 3). To tease apart the effect of UID scores on response quality, we controlled for surprisal by grouping or bucketing responses into 12 intervals of surprisals (within a range of 0.4 units as shown in the first column on Table 2). Within these intervals, surprisal had no correlation with generation quality (Table 5). Once we control for surprisal i.e. analyse dialog responses with similar surprisals but varying UID scores, we observe that UID scores negatively correlate with human judgments, to varying degrees of strength, for responses in very low or high surprisal intervals (see Table 2). Thus, for the extremities of the surprisal range, UID scores indicate that better rated responses are non-uniform.

4 Discussion

Contrary to our expectations, we find non-uniformity to be a more desirable property in machine-generated responses. Overall, UID scores and surprisal do not correlate with human judgments (Table 4). But when controlled for surprisal,

we observe that UID score is correlated with human judgments for certain intervals (examples in Figure 3 and Table 6). Our results suggest that optimizing UID to generate uniform text might not be the right objective for regularizing decoding algorithms. Instead we find that non-uniform information density could be a potential solution to the “likelihood trap” problem according to which models generate lower quality text (as per human judgments) when sampling from the extremities of their likelihood space (Zhang et al., 2021b). Consequently, we suggest that decoding algorithms be tuned to follow the information density patterns of human-generated non-uniform data when generating responses outside of the “safe” likelihood range as a means to generate higher quality responses across the entire likelihood space.

5 Limitations

While we present a study of multiple decoding settings, we generate all machine responses using the same transformers based model architecture. Thus, the presented work does not yet explore individual differences between different model architectures. Additionally, due to limited resources we were not able to collect large-scale human annotations across multiple corpora and acknowledge the same as part of future efforts.

6 Ethical considerations

In this work, we collected human annotations on dialogue response quality using MTurk. Each HIT in our MTurk study contained one dialogue history and four candidate responses. The annotators could read the history and rate the responses that followed using mouse clicks on their response choices. We provided an additional feedback field for annotators to write comments in. We received very positive feedback on the task from all the annotators who used this feature. There were no restrictions on the minimum or maximum number of examples the annotators had to rate. From a pilot study on MTurk, we found the average time to complete one HIT to be slightly under 2.5 minutes. After considering the average time required and the task difficulty (expressed to be clearly and easily understood by annotators in their comments) we set the payment amount to \$0.5 per HIT for an hourly rate of about \$12 per hour.

References

- Matthew Aylett and Alice Turk. 2004. The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and speech*, 47(1):31–56.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2020. The second conversational intelligence challenge (convai2). In *The NeurIPS’18 Competition*, pages 187–208. Springer.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. **Hi-erarchical neural story generation**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- August Fenk and Gertraud Fenk. 1980. constancy in short-term memory-constancy in linguistic information flow. *journal for experimental and applied psychology*, 27(3):400–414.
- Austin F Frank and T Florain Jaeger. 2008. Speaking rationally: Uniform information density as an optimal strategy for language production. In *Proceedings of the annual meeting of the cognitive science society*, volume 30.
- Sergey Golovanov, Rauf Kurbanov, Sergey Nikolenko, Kyryl Truskovskiy, Alexander Tselousov, and Thomas Wolf. 2019. **Large-scale transfer learning for natural language generation**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6058, Florence, Italy. Association for Computational Linguistics.
- John Hale. 2003. The information conveyed by words in sentences. *Journal of Psycholinguistic Research*, 32(2):101–123.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. **The curious case of neural text de-generation**. In *International Conference on Learning Representations*.
- T Florian Jaeger. 2010. Redundancy and reduction: Speakers manage syntactic information density. *Cognitive psychology*, 61(1):23–62.
- Ayush Jain, Vishal Singh, Sidharth Ranjan, Rajakrishnan Rajkumar, and Sumeet Agarwal. 2018. **Uniform Information Density effects on syntactic choice in Hindi**. In *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*,

- pages 38–48, Santa Fe, New-Mexico. Association for Computational Linguistics.
- Roger Levy and T Florian Jaeger. 2007. Speakers optimize information density through syntactic reduction. *Advances in neural information processing systems*, 19:849.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proc. of NAACL-HLT*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Kyle Mahowald, Evelina Fedorenko, Steven T Piantadosi, and Edward Gibson. 2013. Info/information theory: Speakers choose shorter words in predictive contexts. *Cognition*, 126(2):313–318.
- Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. [If beam search is the answer, what was the question?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.
- Clara Meister, Tiago Pimentel, Patrick Haller, Lena Jäger, Ryan Cotterell, and Roger Levy. 2021. Revisiting the uniform information density hypothesis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 963–980.
- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2022. Typical decoding for natural language generation. *arXiv preprint arXiv:2202.00666*.
- Moin Nadeem, Tianxing He, Kyunghyun Cho, and James Glass. 2020. [A systematic characterization of sampling algorithms for open-ended language generation.](#) In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 334–346, Suzhou, China. Association for Computational Linguistics.
- Olabiye Oluwatobi and Erik Mueller. 2020. [DLGNet: A transformer-based model for dialogue response generation.](#) In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 54–62, Online. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation.](#) In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores.](#) In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Nils Reimers and Iryna Gurevych. 2019. [Sentencebert: Sentence embeddings using siamese bert-networks.](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation.](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Jason Wei, Clara Meister, and Ryan Cotterell. 2021. A cognitive regularizer for language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5191–5202.
- Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021a. [Trading off diversity and quality in natural language generation.](#) In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.
- Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021b. [Trading off diversity and quality in natural language generation.](#) In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. [Personalizing dialogue agents: I have a dog, do you have pets too?](#) In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Generation Type	Pearson's r
Reference Text	-.69
Greedy	-.23
$p = 0.3$	-.43
$p = 0.5$	-.50
$p = 0.6$	-.56
$p = 0.8$	-.65
$p = 0.9$	-.68
$k = 10$	-.40
$k = 20$	-.45
$k = 50$	-.56
$k = 100$	-.63
$k = 200$	-.65
$k = 500$	-.69
Vanilla	-.74

Table 3: Pearson's correlation coefficient (r) between UID score and average sentence surprisal (all $p < 0.01$)

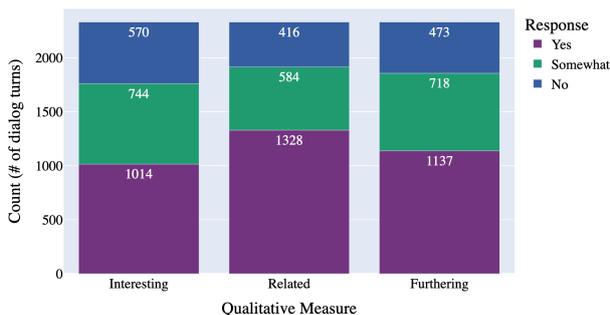


Figure 4: Frequency of responses (Yes/Somewhat/No) for each qualitative measure in our human annotated dataset.

A Human evaluation study details

Raters were selected based on the criteria that they be located in the US, and had attempted a minimum of 500 HITS at an accepted work rate greater than 97% on MTurk. We asked raters on MTurk to answer if a candidate response satisfied each of the qualitative measures (interesting, furthering and related) and gave them three response options: "Yes", "Somewhat" and "No". In a pilot study of 360 responses, we also included a measure for fluency. All of the responses were rated "Yes" by majority vote and we removed this measure from further analysis as all the generations in this study were fluent as indicated by the pilot study and from our observation. For correlation calculations, we assign integer score values to each of the three re-

Quality	Pearson's r	
	UID Score	Surprisal
Related	.01	-.13*
Furthering	.03	-.10*
Interesting	-.04	-.01

Table 4: Pearson's correlation coefficient (r) of UID score and surprisal with human judgments of qualitative metrics ($*p < 0.01$)

Surprisal interval	n	Pearson's r		
		Related	Furthering	Interesting
(0.8,1.2)	24	-.03	-.04	-.00
(1.2,1.6)	64	-.10	-.16	.08
(1.6,2.0)	91	.05	.14	.10
(2.0,2.4)	109	-.14	-.08	-.27*
(2.4,2.8)	111	-.12	.05	.09
(2.8,3.2)	105	-.02	.06	-.00
(3.2,3.6)	99	-.13	.12	.01
(3.6,4.0)	66	.02	-.06	.06
(4.0,4.4)	42	-.01	-.00	.06
(4.4,4.8)	24	.20	.34	.23
(4.8,5.2)	12	-.13	-.37	-.12
(5.2,5.6)	13	.60	.83	.76

Table 5: Pearson's r between surprisal and human judgments of qualitative measures for dialog responses bucketed by surprisal [Surprisal interval = the ranges of surprisal values used for bucketing responses, n = number of responses in each surprisal interval, $*p$ -value $< .05$]

sponse options as 3 for "Yes", 2 for "Somewhat" and 1 for "No". Thus, the higher the score, the better the response is rated. Following the pilot study, for 194 dialogue histories, we showed the raters 4 candidate dialogue responses (total of 776 dialogue responses) and collected ratings on all *3* measures from *3* raters per dialogue history. In all, we obtained a total of 776*3, i.e., 2328 total response-rating pairs. To calculate the score for each response along every measure, we take the mean of all ratings as the score. For cases where at least 2 out of 3 raters agree, we take majority vote as the final score. This constituted (2018 out of 2328) 86.68% of all the ratings collected. We show the overall distribution of qualitative scores for all the response-rating pairs in Figure 4. We verified the rater responses by checking if they were rating human-generated responses highly as those came from a trusted source (Persona-Chat). We also manually inspected a random subset of dialog history-candidate response sets and found the results to be in accordance with our intuitions.

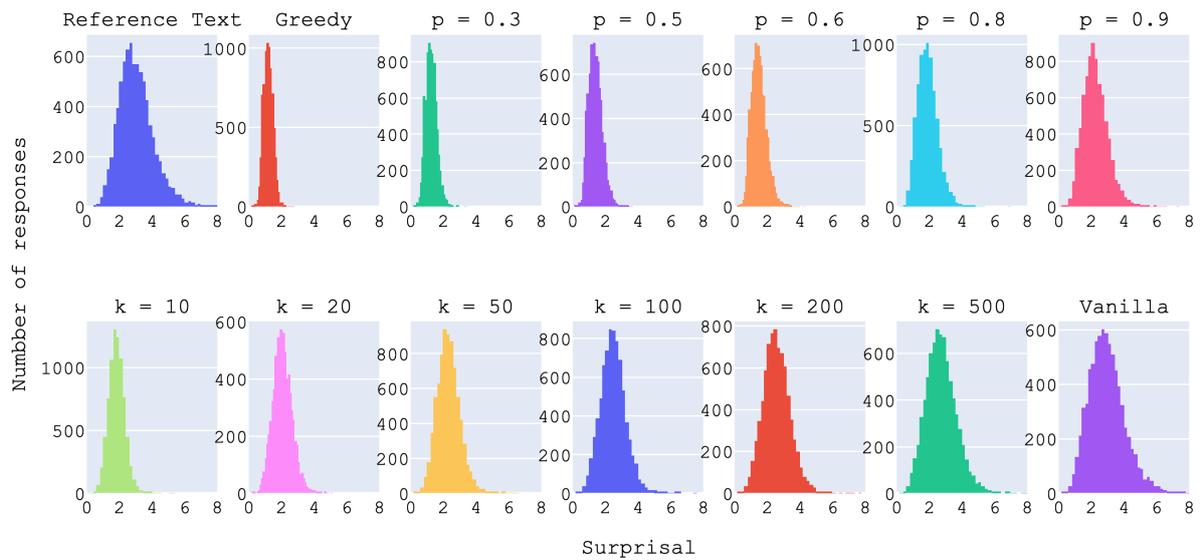


Figure 5: Histograms of **average sentence surprisal** for responses generated using different decoding settings and human-generated reference text (left-top).

Instructions
Examples

Please read the following **conversation history** & rate the responses that follow *as if you were a participant in this conversation*:

Speaker A: its all tedious at first dear , but i know it will get better for you .
 Speaker B: i guess it just really want to be out on my red bike .
 Speaker A: sounds like a true joy . i'm retired now , so you are inspiring me to try that .
 Speaker B: awesome . maybe we could go out riding together .
 Speaker A: i would love that . life is too short to miss out on making new friends .

Rate each of the **4 candidate responses** (1st column) on the **3 quality measures** (2nd column onwards) based on how well each response satisfies the quality description.

You will be entering a total of 12 responses. Fill the responses one row at a time, i.e. first read the response in a row, and rate it on all 3 qualities before moving to the next row.

Candidate Responses	Quality Measures & Description					
	Related		Furthering		Interesting	
	Is it on topic with the conversation history?		Does it encourage the conversation to continue?		Does it present engaging or new information?	
me to. life is a great tool to have!	Yes	<input type="checkbox"/>	Yes	<input type="checkbox"/>	Yes	<input type="checkbox"/>
	Somewhat	<input type="checkbox"/>	Somewhat	<input type="checkbox"/>	Somewhat	<input type="checkbox"/>
	No	<input type="checkbox"/>	No	<input type="checkbox"/>	No	<input type="checkbox"/>
that is true. you guys should come ride with me!	Yes	<input type="checkbox"/>	Yes	<input type="checkbox"/>	Yes	<input type="checkbox"/>
	Somewhat	<input type="checkbox"/>	Somewhat	<input type="checkbox"/>	Somewhat	<input type="checkbox"/>
	No	<input type="checkbox"/>	No	<input type="checkbox"/>	No	<input type="checkbox"/>
you are right. that is true. i am just happy to have someone who loves me.	Yes	<input type="checkbox"/>	Yes	<input type="checkbox"/>	Yes	<input type="checkbox"/>
	Somewhat	<input type="checkbox"/>	Somewhat	<input type="checkbox"/>	Somewhat	<input type="checkbox"/>
	No	<input type="checkbox"/>	No	<input type="checkbox"/>	No	<input type="checkbox"/>
i have to go, talk to you later.	Yes	<input type="checkbox"/>	Yes	<input type="checkbox"/>	Yes	<input type="checkbox"/>
	Somewhat	<input type="checkbox"/>	Somewhat	<input type="checkbox"/>	Somewhat	<input type="checkbox"/>
	No	<input type="checkbox"/>	No	<input type="checkbox"/>	No	<input type="checkbox"/>

Please provide any comments or feedback here.

Submit

Figure 6: Screenshots of our MTurk study interface for collecting human judgments on 4 candidate responses per dialogue history, along 3 quality measures.

1. Read the given conversation history carefully.
2. Then, rate the quality of 4 candidate responses as potential next responses to the conversation history along 3 quality measures (12 responses in total):

Note: Respond as though you are a participant in the conversation. For example, do not mark a response as uninteresting due to personal preference. Instead, consider how a person in the conversation might find it.

Quality Measure	Description
Related	Does the response follow the conversation history's general topic and is a valid continuation of the dialogue?
Furthering	Does the response encourage the conversation to keep moving forward? This might be through a question or a response that can be easily followed-up on.
Interesting	Does the response present new or engaging information?

(a) Detailed instructions that MTurk raters could expand at any time.

The following table contains examples of candidate responses corresponding to each rating option (Yes/Somewhat/No) for all 3 quality measures for the given conversation history:

Conversation History:

Speaker A: I do enjoy trying out different cuisines.
 Speaker B: Oh, nice. What's your favorite food?
 Speaker A: I like Peruvian food quite a lot. What about you?

Quality Measure	Rating	Response Example
Interesting	Yes	Oh, I recently made Ceviche at home! But I'm more into Mexican street food.
	Somewhat	I like Peruvian too.
	No	Nothing much.
Furthering	Yes	What do you like about it?
	Somewhat	I'm more into Mexican street food actually.
	No	I'm not into it.
Related	Yes	I do like it, but it's so hard to cook at home.
	Somewhat	Me too! What else do you enjoy?
	No	I go for runs all the time.

(b) Examples responses for each measure and rating category shown to MTurk raters.

Figure 7: Instructions and examples from MTurk study.

Dialogue history

Speaker A: hi, i m susie. how are you?

Response type	Response text	UID Score	Interesting
Reference Text	i'm good. just got back from some volunteering. how are you doing?	-1.99	3
Candidate 1	hi. i am very good. just enjoying my favorite pastime. how are you?	-1.43	2
Candidate 2	i am great! i volunteer at a soup kitchen and watch movies with my kids.	-1.19	2
Candidate 3	i'm doing well. how are you?	-0.18	1

**Speaker A: that's cool. my dad made us italian food tonight.
 Speaker B: oh nice, i love italian food. my favorite is the garlic bread. what is yours?
 Speaker A: pasta, my son loves pizza though!
 Speaker B: i like that too, have to eat lots of carbs for my training.**

Response type	Response text	UID Score	Interesting
Candidate 1	wow, my son took us and my two daughters to a super fast food joint the other day	-4.83	3
Candidate 2	that is too bad, i like the cheesy stuff.	-2.08	2
Reference Text	do you have a favorite genre of movies?	-1.73	2
Candidate 3	i also like to stay home with my son.	-0.88	1

**Speaker A: hello i hope your sunday is great , what is your favorite kind of music?
 Speaker B: hey there . been a relaxed sunday . yours ? music eclectic.
 Speaker A: my sunday has been exciting ! i enjoy death metal.
 Speaker B: death metal . cool . i spent the morning volunteering.
 Speaker A: i volunteer too , at the local pool to be a swim coach.**

Response type	Response text	UID Score	Furthering
Candidate 1	nice. such a nice day. how long have you been coaching?	-9.99	3
Reference Text	that is great! we both volunteer! mine is rescuing bunnies.	-7.59	2
Candidate 2	that's cool. i donate my pay to the local zoo. humane societies.	-3.89	2
Candidate 3	sick sick. beautiful color, navy blue is my favorite.	-2.79	1

Speaker A: have you heard about the juggalos? weird.

Response type	Response text	UID Score	Furthering
Reference Text	what are those? do they juggle balls?	-6.88	3
Candidate 1	no i have not. i like that band myself	-5.19	2
Candidate 2	we will not tell. do you like comics?	-4.27	2
Candidate 3	i have not heard of them. i've dolls too.	-2.8	1

**Speaker A: hello, how are you today?
 Speaker B: good. yourself? listening to classical music.
 Speaker A: that is cool! i just like to skateboard!**

Response type	Response text	UID Score	Related
Reference Text	can you do a kick flip?	-14.11	3
Candidate 1	haha. i love skating too, but i have a truck to keep as well.	-11.28	3
Candidate 2	nice. i love to do stuff like that. i also swim.	-10.25	3
Candidate 3	are you a skater or do you do it daily?	-9.05	2

**Speaker A: i hope it is a healthy snack. i am doing well, just moved here.
 Speaker B: it is. i am vegan. just moved here too! where from?
 Speaker A: i too am vegan from germany. have you seen lafer! lichter! lecker! on tv?
 Speaker B: oh yes! i love shows like that and watched it back home a lot.**

Response type	Response text	UID Score	Related
Reference Text	they do amazing things with the hummus. where is back home for you?	-6.86	3
Candidate 1	you must have a lot of fun watching them.	-4.53	2
Candidate 2	they have the best new vegan cookbooks, but i am more adventurous.	-2.93	3
Candidate 3	i do love the sky diving, too! i have seen the first few seasons.	-2.47	1

Table 6: Examples of dialogue histories followed by 4 response candidates arranged by increasing UID score i.e. from more non-uniform to uniform responses and their corresponding human judgment scores.

Benchmarking Long-tail Generalization with Likelihood Splits

Ameya Godbole and Robin Jia
University of Southern California
{ameyagod, robinjia}@usc.edu

Abstract

In order to reliably process natural language, NLP systems must generalize to the long tail of rare utterances. We propose a method to create challenging benchmarks that require generalizing to the tail of the distribution by re-splitting existing datasets. We create ‘Likelihood Splits’ where examples that are assigned lower likelihood by a pre-trained language model (LM) are placed in the test set, and more likely examples are in the training set. This simple approach can be customized to construct meaningful train-test splits for a wide range of tasks. Likelihood Splits surface more challenges than random splits: relative error rates of state-of-the-art models increase by 59% for semantic parsing on SPIDER, 93% for natural language inference on SNLI, and 33% for yes/no question answering on BOOLQ, on our splits compared with the corresponding random splits. Moreover, Likelihood Splits create fairer benchmarks than adversarial filtering; when the LM used to create the splits is also employed as the task model, our splits do not unfairly penalize the LM.

1 Introduction

Success on in-distribution test data does not necessarily show that a system has solved the underlying task at hand. Systems can achieve artificially high accuracy by exploiting dataset-specific shortcuts, such as spurious feature-label correlations that hold in the data but not in general (Gardner et al., 2021). In many datasets, a large proportion of test examples are similar to training examples, further inflating in-distribution accuracy (Lewis et al., 2021; Czarnowska et al., 2019; Orr et al., 2021). Out-of-distribution (OOD) evaluation paints a clearer picture of a system’s ability to perform the task.

Prior work has proposed a variety of methods to test OOD generalization, each with their own strengths and weaknesses. Task-specific behavior tests (Ribeiro et al., 2020; Naik et al., 2018; Gardner et al., 2020) give insights into model be-

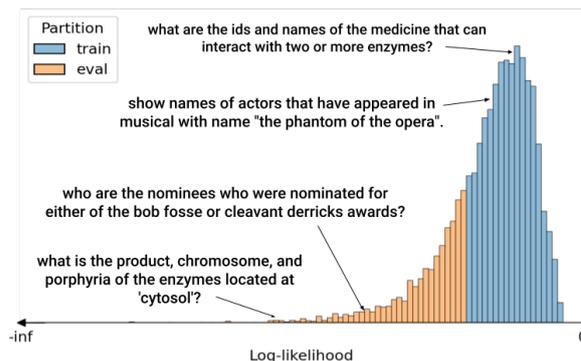


Figure 1: **Likelihood Splits**: We propose to partition the dataset based on likelihood under a language model. The high-likelihood “head” of the distribution becomes the training set while we evaluate generalization to the low-likelihood “tail” of the data. Shown here are queries from the SPIDER dataset in different likelihood buckets: one possible tail generalization could be the handling uncommon entities with known query types.

havior but require significant manual (often expert) effort to create. Adversarial data collection, in which annotators try to fool high-performing models (Nie et al., 2020; Potts et al., 2021), also collects challenging examples, but runs the risk of focusing only on a narrow subset of model weaknesses (Bowman and Dahl, 2021; Kaushik et al., 2021). Adversarial filtering removes easy examples from existing datasets (Sakaguchi et al., 2021), but can disproportionately penalize the model used during filtering (Phang et al., 2021). Domain generalization tests transferability to new data domains (Fisch et al., 2019; Miller et al., 2020), but there is no guarantee that generalizing to a given new domain is possible—out-of-domain examples may require skills that are not learnable from the training data (Geiger et al., 2019). Other approaches create dataset splits that test for specific skills, such as length generalization (Lake and Baroni, 2018) and compositional generalization (Shaw et al., 2021), but they only apply to a narrow subset of tasks.

In this work, we propose **Likelihood Splits**, a general-purpose method to create challenging OOD splits for existing datasets. The principle behind Likelihood Splits is that any system that claims to reliably process natural language must be able to generalize from more common utterances seen during training to the long tail of rare utterances at test time. Generalization, not merely memorization, is necessary because even a very large training dataset cannot exhaustively cover all possible long-tail examples that may be encountered in the real world. Moreover, standard annotation procedures tend to over-sample examples from the head of the distribution, further ignoring the challenge posed by infrequent examples. We identify tail examples using the likelihood under the GPT-2 language model (Radford et al., 2019). Examples with low likelihood under GPT-2 are placed in the held-out evaluation sets and the high likelihood examples are used as the training set (see Figure 1).

Likelihood Splits are a novel, widely applicable strategy that can create interesting generalization benchmarks at no additional annotation cost. They are more challenging than a random split across a wide range of tasks: error rates relative to random splits increase by 59% for T5 (Raffel et al., 2020) on SPIDER (Yu et al., 2018), 93% for ELECTRA (Clark et al., 2020) on SNLI (Bowman et al., 2015), and 33% for ROBERTA (Liu et al., 2019) on BOOLQ (Clark et al., 2019). Moreover, the proposed splits do not unfairly penalize the GPT-2 model used to create the splits when it is used as a task model, thus avoiding one of the downsides of adversarial filtering. We identify many independent challenges required by Likelihood Splits, including generalizing to rare words, complex programs, and syntactically complex sentences. We encourage future benchmark creators to release Likelihood Splits as a complementary evaluation to the standard IID evaluation to better test out-of-distribution generalization performance. We will release the splits discussed in this work along with the code to easily create Likelihood Splits of other datasets.¹

2 Related Work

Generalizing to the long-tail. Evaluating systems on long-tail phenomena is important, especially because many datasets over-sample the head of the distribution. For example, some question-answering (QA) datasets limit their purview to pop-

ular web-pages (Yang et al., 2018) or frequent user queries (Kwiatkowski et al., 2019). Lewis et al. (2021); Liu et al. (2021) demonstrate that models trained on these datasets often fail on examples that do not match the most frequent training cases. Similar observations have been made in entity linking to rare entities, (Orr et al., 2021; Chen et al., 2021), information retrieval for open-domain QA (Sciavolino et al., 2021), relation extraction for rare relations (Sabo et al., 2021) and lexicon induction for rare senses in machine translation (Czarnowska et al., 2019). Zero-shot performance of large LMs on numerical reasoning and factoid questions is also correlated with the frequency of occurrence of the facts in the pre-training corpus (Razeghi et al., 2022; Kandpal et al., 2022; Elazar et al., 2022). While we do not test whether models can memorize long-tail knowledge, we instead test whether models can process long-tail sentences. Naik et al. (2022) note that it is challenging to catalogue and evaluate generalization along micro-level dimensions and instead propose benchmarks that vary along macro-level dimensions (such as the language and domain) as a proxy. We hypothesize that LMs learn which micro-level phenomena are rare, as this would improve their overall language modeling objective. In this work, we present a recipe that leverages LMs to evaluate tail generalization for any language task.

Task-specific test sets. Ribeiro et al. (2020) use templated queries to evaluate model performance under various linguistic perturbations. This method requires dataset designers to define phenomena of interest and axes of perturbation along which labels may be preserved or changed. Naik et al. (2018) analyze model errors and instantiate tests that explicitly evaluate models on more examples from each error class. Gardner et al. (2020) check for model consistency under local perturbations of test set examples. All of these approaches require annotators to create new examples, whereas we propose a method to resplit existing datasets.

Adversarial approaches. Søgaard et al. (2021) argue that random splits over-estimate model performance on new in-domain data and recommend the use of adversarial and heuristically challenging splits to estimate generalizability. Adversarial data collection promotes the creation of difficult examples by encouraging annotators to fool a model-in-the-loop (Nie et al., 2020; Potts et al., 2021;

¹github.com/ameyagodbole/long-tail-likelihood-splits

Kiela et al., 2021). Similarly, Adversarial Filtering removes examples that are easy for a given task model in order to create more challenging benchmarks (Sakaguchi et al., 2021; Yang et al., 2018). However, Kaushik et al. (2021) and Bowman and Dahl (2021) point out that adversarially collected or filtered examples may focus on a narrow set of skills that the “in-the-loop” model lacks, instead of covering all the abilities required for the underlying task. Additionally, the “in-the-loop” task model is disproportionately penalized by the adversarial test sets (Phang et al., 2021). We show in §4.3 that Likelihood Splits do not suffer from this issue.

Domain shift. In NLP, domains can be characterized by the changes in vocabulary and distribution of word use, styles used by authors, and the intended audience. Fisch et al. (2019) pose the challenge of developing QA systems that need to generalize to unseen domains. Miller et al. (2020) show that QA models trained on SQUAD show a performance drop on new domains (while human baseline performance remains unchanged); Miller et al. (2021); Hendrycks et al. (2020) inter alia perform similar analyses of domain shift. SPIDER (Yu et al., 2018) and GRAILQA (Gu et al., 2021) evaluate semantic parsing on unseen table and knowledge base domains respectively. Domain shift is an orthogonal axis of generalization; we focus on generalizing to rare utterances in the same domain.

Out-of-distribution detection. Previous work in OOD detection has used high generative model perplexity as a sign of outliers (Arora et al., 2021; Ren et al., 2019; Lee et al., 2018). Our intuition is similar: low likelihood (high perplexity) is an indicator of rare examples. However, only our work uses likelihood scores for benchmark creation. Moreover, in our setting all examples have been collected under the same data collection protocol, so none of the examples are truly OOD.

Compositional generalization. The ability to “compose” the meaning of a new utterance from the known meaning of its parts (Fodor and Pylyshyn, 1988) is an important aspect of language understanding. The deterministic grammar of programming languages makes semantic parsing, the task of translating a natural language utterance into a logical program, a good testbed for evaluating compositional generalization (Lake and Baroni, 2018; Kim and Linzen, 2020; Hupkes et al., 2020; Keysers et al., 2020; Shaw et al., 2021). However, for

tasks where the constituent blocks are not clearly defined, it is unclear how to create such evaluation splits of the data. We compare against compositional generalization splits of the semantic parsing dataset SPIDER (Yu et al., 2018) in §4.

3 Capturing the Tail of the Distribution

In order to find the tail within a dataset, we approximate likelihood of an utterance in the real distribution with its likelihood under a language model (LM). Our method can be easily modified to create meaningful splits for any language task. We demonstrate this by creating Likelihood Splits for:

- SPIDER, a semantic parsing dataset (Yu et al., 2018) consisting of natural language questions and corresponding SQL programs;
- SNLI, a natural language inference dataset (Bowman et al., 2015) consisting of premise and hypothesis sentences paired with labels denoting that the hypothesis is entailed by/neutral to/contradictory to the premise;
- BOOLQ, a question-answering dataset (Clark et al., 2019) consisting of a passages, associated questions, and binary yes/no labels.

3.1 General Approach

We consider language tasks where models must map an input x to an output y (e.g., a SQL query or a label). The input x may be either a single sentence (e.g., semantic parsing) or a pair of sentences (e.g., natural language inference), in which case we write $x = (x_1, x_2)$. Given a dataset D of (x, y) pairs and desired proportion p of evaluation examples, our method partitions D into subsets D_{train} and D_{eval} where $|D_{\text{eval}}| \approx p \cdot |D|$. More specifically, we will first assign a likelihood score $s(x)$ to each $x \in D$, then choose D_{eval} to be the $\lfloor p \cdot |D| \rfloor$ examples in D with lowest value of $s(x)$, and choose $D_{\text{train}} = D \setminus D_{\text{eval}}$. In §3.2, we describe a few different ways to define s . In §3.3, we describe a modification to this procedure that controls for varying length between examples. Finally, we describe task-specific adjustments in §3.4.

3.2 Assigning Likelihood Scores $s(x)$

We use the total log-likelihood over the query tokens assigned by the GPT-2 language model as the score $s(x)$ for every example. There are two ways to use the LM: (1) prompting a frozen LM or (2) fine-tuning the LM on the dataset.

Task	Prompting	Fine-Tuning
SPIDER	write a database question: {query}	< endoftext > {query}
BOOLQ	Passage: {passage} Ask a question about the passage: {question}	
SNLI	Premise: {premise} This hypothesis is {entailed/neutral/a contradiction}: {hypothesis}	

Table 1: Input formats for single-sentence and sentence-pair tasks in the prompting and fine-tuning settings. Values in curly braces are plugged in from the example. For SNLI, we provide the label in the prompt to prime the LM to the class of hypothesis. The LM is trained (when fine-tuning) and evaluated on generating the query in blue.

Past work has shown that prompting i.e. prepending a task-specific string to the query, helps GPT-2 generalize zero-shot to new tasks (Radford et al., 2019). We use simple prompts that describe the task and prime the LM to the text we expect it to generate (see Table 1). For sentence pair tasks (such as SNLI and BOOLQ), it is necessary to compare the relation between two pieces of text and not just each piece in isolation. Thus, it is intuitive to describe unlikely examples by the conditional likelihood of x_2 given x_1 . We demonstrate the flexibility of our approach by providing the label in the prompt if it adds additional information about the text to be generated (e.g. in SNLI).² We will refer to this setting which uses the prompted LM with the tag *ll_split pt* in the rest of the work.

The dataset curator may also choose to fine-tune the LM to better capture the task distribution. We fine-tune the GPT-2 LM to maximize either the probability of x for single sentence tasks or the conditional probability of x_2 given the prompt for sentence-pair tasks. When fine-tuning the LM on the dataset, we need to ensure that it is not used to assign scores to the examples it is trained on. Given the dataset D , we first randomly partition D into k folds. For each fold, we fine-tune an LM on the remaining folds and use it to assign log-likelihood scores to examples in the held-out fold. We refer the reader to Appendix A.2 for fine-tuning details. We will refer to this setting as *ll_split* henceforth.

3.3 Controlling for Length

Since the likelihood of an utterance is negatively correlated with its length, we create a split that explicitly controls for the effect of length. After assigning a likelihood score to every utterance, the examples are bucketed based on length (defined by tokenizing the utterance with NLTK (Loper and

²We include the label in the prompt for SNLI but not BOOLQ because the resulting prompts seemed most natural for each dataset. This choice was made before assessing downstream behavior.

Bird, 2002)). For single-sentence and sentence-pair tasks, we use the length of the query (x and x_2 respectively) over which log-likelihood was computed. Within each bucket, a fraction p of the examples with the lowest $s(x)$ are put in the evaluation set; aggregating examples from all buckets, $|D_{\text{eval}}| \approx p \cdot |D|$. We will refer to this control setting with the modifier (*-len*) henceforth.³

3.4 Dataset-specific Choices and Details

SPIDER. We follow Shaw et al. (2021) and swap examples between the train and evaluation sets such that every logical program atom in the evaluation set appears at least once in the train set. This ensures that the model is not required to generalize to unseen function names and declarations.

SNLI and BOOLQ. We ensure label balance in our splits (as in the original data) by splitting the examples for each label separately, then combining the resulting train and evaluation sets.

Development sets. Csordás et al. (2021) show that without development sets that are in-distribution to challenging test sets, models are prone to over-fitting, which under-estimates their ability to generalize. Thus, after dividing the data into train and evaluation sets, we randomly divide the evaluation set into a development set and test set. Other details are reported in Appendix A.1.

4 Experiments

Next, we benchmark task models on our Likelihood Splits. Splits created using GPT2-medium will be the focus of our analysis. We will briefly study the effect of switching the LM to GPT2-large in §4.4.

When creating Likelihood Splits, the number of folds k for fine-tuning the LM (§3.2) can be chosen by the dataset curator. For results in §4 and §5, we set $k = 3$ arbitrarily. We analyse the effect of

³We also considered using perplexity, which normalizes for length, but it led to an over-correction where short examples were filtered into the evaluation set.

choosing a different value of k in Appendix A.3. Our results show that the trends and observations discussed here hold true for other values of k .

4.1 Benchmarked Models

One of the goals of this work is to expose long-tail generalization as a challenge to state-of-the-art models; SotA models on the considered benchmarks are all pre-trained models. We make efforts to show that models with different pre-training data and objectives are similarly affected by our proposed splits. Hyperparameters and training details for the reported models are in Appendix A.2.

Semantic parsing. Following Shaw et al. (2021), we benchmark the competitive T5-base model (Raffel et al., 2020) on all splits of the SPIDER dataset. In order to test whether these splits are adversarial to the data splitting language model, we additionally fine-tune GPT2-medium models for the semantic parsing task. To study the effect of model size, we fine-tune T5-small and GPT2-small variants.

SNLI and BOOLQ. We fine-tune two competitive models (ROBERTA (Liu et al., 2019) and ELECTRA (Clark et al., 2020)) at two model sizes (*base* and *large*). Additionally, following Poliak et al. (2018), we train a ROBERTA-large model to perform the task given just the hypothesis. The performance of a hypothesis-only model estimates the degree of spurious correlations that exist in the dataset which give away the label.

4.2 Alternative Splits for Semantic Parsing

We compare the difficulty of the Likelihood Splits with past work on heuristic challenges splits.

Length. Past work has established that text generation models trained on short inputs struggle to generalize to longer inputs at test time (Lake and Baroni, 2018; Hupkes et al., 2020; Newman et al., 2020). We create *Length* splits by placing examples with the longest input queries in the evaluation set and the remaining examples in the training set.

TMCD. Systematicity is the ability to compositionally derive the meaning of an utterance from the known meaning of its parts. Past work studying systematicity in semantic parsing has defined “atoms” as the smallest constituents of the grammar (e.g. variables and function names) and “compounds” as complex structures formed by composing atoms (e.g. multi-argument functions and nested function calls) (Keysers et al., 2020). Following Shaw

Split	T5 base	T5 small	GPT-2 medium(Δ)	GPT-2 small
Random	78.6	75.2	69.3 (9.3)	64.7
Length	50.0	44.5	39.9 (10.1)	34.0
Template	60.1	60.0	51.4 (8.7)	45.1
TMCD	66.2	64.1	56.2 (10)	51.4
Split LM: GPT2-medium				
<i>ll_split</i>	66.0	64.2	57.2 (8.8)	51.8
<i>ll_split</i> (-len)	71.3	67.3	59.9 (11.4)	57.3
<i>ll_split</i> pt	60.6	59.7	50.9 (9.7)	45.9
<i>ll_split</i> pt (-len)	73.5	68.4	64.5 (9)	58.3
Split LM: GPT2-large				
<i>ll_split</i>	61.8	61.8	53.7 (8.1)	48.3
<i>ll_split</i> (-len)	69.7	66.2	59.1 (10.6)	54.8
<i>ll_split</i> pt	63.0	58.3	51.4 (11.6)	45.7
<i>ll_split</i> pt (-len)	72.0	70.1	63.4 (8.6)	57.5

Table 2: SPIDER: Exact sequence prediction accuracy for Likelihood Splits created by GPT2-medium and GPT2-large, and other challenge splits. Likelihood Splits are more challenging than random splits while not being adversarial to GPT2-medium. Δ marks the performance drop from T5-base to GPT2-medium.

et al. (2021), we create TMCD (Target Maximum Compound Divergence) splits of SPIDER by maximizing the divergence between the distributions of compounds in the train and evaluation sets.

Template. These splits test the ability of parsers to generate unseen program templates (canonicalized programs formed by anonymizing all variable names and standardizing syntax). We group examples in the SPIDER dataset based on templates defined by Finegan-Dollak et al. (2018). To create the evaluation set, we randomly pick groups of examples till the target set size is reached; the remaining groups form the training set.

4.3 Model Performance on Likelihood Splits

In Table 2, we report exact match accuracy⁴ on the data splits using the SPIDER evaluation suite. For SNLI and BOOLQ, we report the accuracy of benchmarked models in Table 3. We create 3 random splits and report mean and standard deviation of accuracy of models trained on each split.

Likelihood Splits are more challenging than random splits. On SPIDER, for example, T5-base accuracy on *ll_split* is 12.6 points lower than the random split accuracy. Likelihood Splits lead to drops in performance that are comparable to the

⁴This metric accounts for the fact that SQL statements are invariant to certain shuffling and change in variable names.

System	SNLI					BOOLQ				
	Random	ll_split		$ll_split\ pt$		Random	ll_split		$ll_split\ pt$	
		(-len)		(-len)			(-len)		(-len)	
ROBERTA-base	89.6 \pm 0.4	79.3	77.1	82.6	81.7	74.9 \pm 0.4	71.6	71.2	72.4	71.9
ROBERTA-large	90.5 \pm 0.5	82.4	79.2	85.0	84.3	84.4 \pm 0.6	79.3	78.9	82.3	80.6
ELECTRA-base	90.5 \pm 0.2	80.1	78.4	82.9	82.8	78.8 \pm 1.1	74.1	74.3	75.2	73.6
ELECTRA-large	91.0 \pm 1.3	82.6	81.6	85.9	84.9	85.5 \pm 0.6	82.6	82.1	83.7	81.9
ROBERTA-large (Hypothesis-only)	70.2 \pm 0.3	64.6	64.6	67.2	69.6	-	-	-	-	-
Human Accuracy	88.7 \pm 0.8	83.6	84.4	85.2	86.4	-	-	-	-	-

Table 3: SNLI and BOOLQ: Accuracy for various splits and model sizes. Likelihood Splits lead to decreased model performance. Controlling for length further increases the difficulty.

alternative challenge splits. Only Likelihood Splits focus on challenges derived from input language variation; we analyze these challenges in §5.1.

On SNLI and BOOLQ, Likelihood Splits are also more challenging than random splits. For example, ELECTRA-large accuracy decreases by 8.4 points on SNLI and 2.9 points on BOOLQ. On SNLI, the performance of the hypothesis-only baselines on Likelihood Splits is lower than that on the random splits, which indicates that our splits are less easily solved by modeling spurious statistical cues.

Controlling for length preserves challenging nature of splits. Likelihood is negatively correlated with length, so Likelihood Split test data contains longer examples. On SPIDER, generalizing to longer utterances is challenging, so controlling for length makes the Likelihood Splits less challenging. However, these splits are still much more challenging than random splits. For T5-base, ll_split (-len) is 7.3 points harder and $ll_split\ pt$ (-len) is 5.1 points harder than the random split. By controlling for length, we identify examples that are more challenging for other reasons (discussed in §5.1). Fitting the dataset distribution with a fine-tuned LM reduces the correlation between length and likelihood on SPIDER. Accordingly, $ll_split\ pt$ poses a stronger length generalization challenge than ll_split , and thus is more challenging: T5-base accuracy drops by 18 points on $ll_split\ pt$ compared with the random split.

Conversely, for SNLI and BOOLQ, controlling for length makes the Likelihood Splits slightly harder compared to their uncontrolled versions (ELECTRA-large accuracy drops by 1% from ll_split to ll_split (-len) on SNLI, and by 0.5% on BOOLQ). This suggests length is not a reason that Likelihood Splits are harder for these datasets. Relatedly, $ll_split\ pt$ is easier than ll_split here.

Likelihood Splits do not unfairly penalize the scoring LM. The difference in accuracy between T5-base and GPT2-medium are comparable across all splits (Δ in Table 2). This shows that the Likelihood Splits do not unfairly penalize GPT2-medium, the model used to create the Likelihood Splits. Thus, benchmarks based on Likelihood Splits will be fairer to model class of the LM used.

Human accuracy is less affected. We estimate human accuracy on the evaluation sets using the \sim 10% of examples that were annotated with 5 labels in the original SNLI dataset. Human accuracy is at most 5.1% lower on our proposed splits than on the random splits. Model performance drops more severely than the smaller drop in human accuracy; models that were previously superhuman are now worse than the estimated human performance (except for ELECTRA-large on $ll_split\ pt$). In comparison, adversarial filtering (Le Bras et al., 2020) has a larger drop in human accuracy from 88% on the standard split to 78% on their most challenging split. Thus, our method does not as heavily emphasize mislabeled or ambiguous examples.

4.4 Effect of the LM on Likelihood Splits.

We study the effect changing the language model by using a GPT2-large model to create the Likelihood Splits of SPIDER. The log-likelihood scores assigned to the examples by GPT2-medium and GPT2-large are highly correlated; Pearson correlation coefficient (r) between log-likelihood scores from fine-tuned models is 0.96 while it is 0.99 for the pre-trained models. Accounting for swapping of examples in order to meet the atom constraint, the evaluation sets differ in 16% of examples in the ll_split setting, and 10% of the examples in $ll_split\ pt$ setting. ll_split is more challenging when using GPT2-large; T5-base accuracy drops an additional

System	Random	<i>ll_split</i>		<i>ll_split reverse</i>	
		(-len)		(-len)	
SPIDER					
T5-base	78.6	66.0	71.3	83.9	81.5
SNLI					
E-base	90.5 \pm 0.2	80.1	78.4	96.6	97.4
E-large	91.0 \pm 1.3	82.6	81.6	96.7	97.4
BOOLQ					
E-base	78.8 \pm 1.1	74.1	74.3	77.8	78.1
E-large	85.5 \pm 0.6	82.6	82.1	85.8	86.8

Table 4: Accuracy on SPIDER, SNLI and BOOLQ when training on the unlikely (tail) queries and evaluating on the likely (head) queries (*ll_split reverse*). The model accuracy on the reverse splits are comparable to or higher than accuracy on the random split. This supports the claim that generalizing to rare instances is a significant challenge. (E-base and E-large are ELECTRA models)

4.2% compared to the *ll_split* with GPT2-medium. The other splits are comparable with accuracies differing by 1-2% across all models (see Table 2). Thus, we expect splits created with different LMs to demonstrate similar characteristics.

4.5 Are Reverse Likelihood Splits Difficult?

We wish to test whether the decrease in task accuracy is driven by rarity of the instances or whether any likelihood based distribution shift is challenging. We test this hypothesis by creating a setting that requires generalizing from the tail of the distribution to the head. Using the same likelihoods as before, we create reverse splits where the more likely (head) of the distribution is used as the evaluation set instead of the unlikely (tail). From Table 4, we see that the accuracy of ELECTRA-large on SNLI increases from 81.6% on the Likelihood Split to 96.7% on the reversed split. For comparison, this is more than the accuracy on random splits of SNLI (91%). We see similar trends on BOOLQ and SPIDER where the reverse splits are as easy as or easier than the corresponding random splits. We conclude that generalizing specifically to the tail is what makes our splits difficult.

5 Analysis of Data Splits

In order to highlight the challenges posed by our proposed splits, we analyze how the development sets (to ensure unseen test sets) differ from the training sets in each split. Our splits require models to simultaneously excel at many different skills be-

lieved to be important for language understanding.

5.1 Properties of the Proposed SPIDER Splits

TMCD-related properties and length. Following Shaw et al. (2021), we report atom and compound divergences of the various splits in Table 13 of Appendix A.4. Divergence measures how much the distribution of atoms/compounds differs between the train and evaluation set. Our approach leads to splits with higher than random atom divergence, which shows that our split poses the challenge of **generalizing to rare atoms**. Similarly, a greater than random compound divergence emerges from the resulting split. This means that the split also requires some amount of **compositional generalization**. From Figure 6 in Appendix A.6, we see that log-likelihood preferentially puts the longer queries in the test set and the corresponding length variation is closer to that of the length split than the other splits. Hence, it naturally requires some aspect of **length generalization**. As expected, by controlling for length of the utterances, we can remove the challenge of length generalization.

Program difficulty. SPIDER assigns a rating of ‘easy’, ‘medium’, ‘hard’, or ‘extra hard’ to every SQL program. From Figure 2, we see that the evaluation sets of the Likelihood Splits contain more examples from the harder categories than the training sets. Controlling for length reduces this effect but does not completely remove it (see Appendix A.5 for more details). Note that this skew emerges even though we do not consider the programs when creating these splits.

Rare words. On the input side, we first analyze the distribution of rare words. We define rare words as all English words⁵ in the SPIDER dataset that occur at most 1 time per million words according to SUBTLEXus (Brysbaert and New, 2009). This results in a list of 561 words. We report the fraction of words in the development set that are rare. This metric automatically controls for the length of the examples; longer examples are more likely to contain rare words by chance. To estimate the distribution of this fraction under random splits (null distribution), we create 500 random splits and plot the distribution of values observed. From Figure 3, we observe that the Likelihood Splits have more rare words in the test set, especially for the *ll_split*

⁵We filter out incorrect spellings using the word list at <https://github.com/dwyl/english-words>

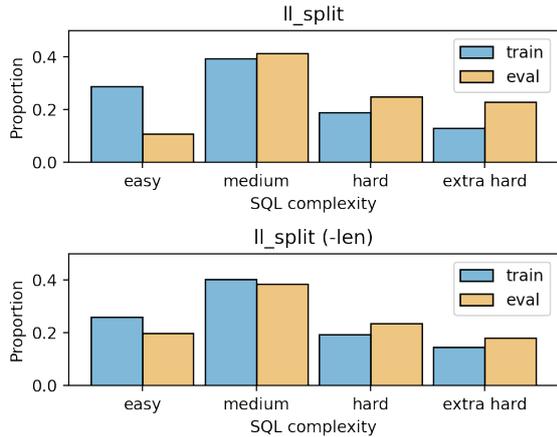


Figure 2: SPIDER: Distribution of SQL programs of varying complexity in the train and development set of Likelihood Splits. These splits show a skew towards training on easy examples and evaluating on harder examples.

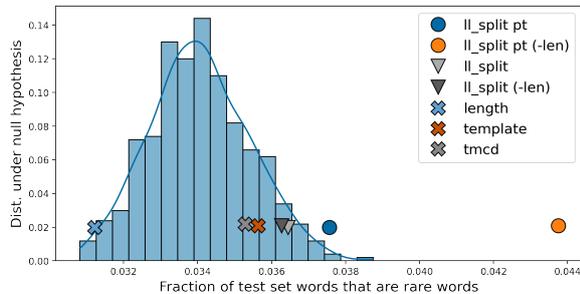


Figure 3: SPIDER: Statistics of the fraction of dev set words that are rare for various splits. This is plotted against the distribution of values observed for 500 random splits of the data. *ll_split* variants retain a larger fraction of rare words in the test set. Controlling for length finds shorter examples with more rare words.

pt setting. Controlling for length puts shorter examples in the evaluation sets, but a larger fraction of the words are rare. The other challenging splits considered do not focus on the input language variation and hence the fraction of development set words that are rare is closer to random.

Input syntactic complexity. We also study the query parse tree structures in various splits of the dataset in Figure 8. We measure the complexity of the parse tree based on mean and max depth as well as Yngve score (Yngve, 1960) which is a measure of syntactic complexity. We see that more complex queries tend to be assigned lower likelihood and correspondingly put in the evaluation set. The effect of the complexity is also correlated with length and balancing for length reduces the gap between

Category	Random	<i>ll_split pt</i>
Easy	92.3% (.225)	81.3% (.077)
Medium	82.3% (.409)	71.6% (.435)
Hard	78.4% (.201)	60.1% (.233)
Extra Hard	62.9% (.164)	52.5% (.254)
Dev set Acc	80.6%	64.8%
Projected Acc		77.15%

Table 5: SPIDER: Accuracy of T5-base aggregated by the SQL hardness rating for random and *ll_split pt* dev sets. The number in brackets is the fraction of dev set examples that fall in each bucket. The examples in the dev set of *ll_split pt* are skewed towards harder examples. However, performance of T5-base on *ll_split pt* is lower than performance on random split in every bucket. Projecting and re-weighting the random set accuracies using the fraction of examples in each bucket in *ll_split pt* over-estimates dev set performance.

the complexity of the train and test set. We refer the reader to Appendix A.7 for more details.

Effect on accuracy. In Appendix A.8 and Table 5, we show that the higher frequency of both novel compounds (i.e., compounds not seen during training) and harder programs each partially explain the higher difficulty of *ll_split pt* for T5-base. For example, 16% of dev examples in the random split have ‘extra hard’ programs, compared with 25% in *ll_split pt*. On the random split, T5-base gets 63% of these examples correct, compared with 81% dev accuracy overall, so these examples are indeed more challenging. On ‘extra hard’ examples in *ll_split pt*, T5-base has an even lower accuracy of 53%. Thus, the mere fact that *ll_split pt* has more ‘extra hard’ examples does not fully explain why it is harder; other factors must also be playing a role.

5.2 Properties of the Proposed SNLI Splits

For SNLI, we study the variation of premise and hypothesis length (A.9), distribution of rare words (A.10), Yngve score (Yngve, 1960) for syntactic complexity (A.11), and Flesch-Kincaid (Kincaid et al., 1975) reading grade-level (A.12). Evaluation sets of Likelihood Splits of SNLI are more complex than their corresponding training sets on all 4 variations; evaluation set examples tend to be longer, tend to contain more rare words, are more syntactically complex, and have higher reading levels.

Controlling for length removes length variation, and slightly decreases the skew in reading level. Surprisingly, the Yngve scores of evaluation examples are skewed to being less complex than the

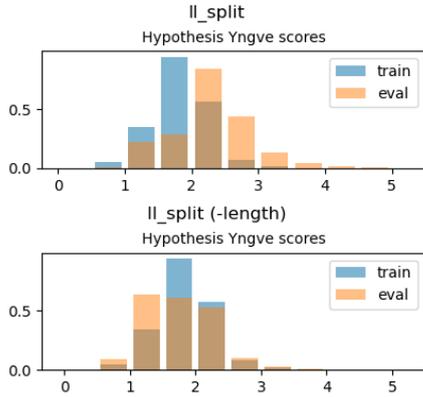


Figure 4: SNLI: Distribution of Yngve scores computed on the parse tree of the hypothesis. The evaluation sets for *ll_split* contain more complex utterances. Normalizing for the length surprisingly reverses the skew.

corresponding training set (see Figure 4), even though the length controlled variants of SNLI are more challenging than the corresponding Likelihood Splits. Some of the difficulty when controlling for length can be explained by the increased proportion of rare words.

We analyze the errors of ROBERTA-large on the development set of *ll_split (-len)*, the hardest SNLI split (see Appendix A.13 for concrete examples). We find several instances of examples that require common-sense or world knowledge to be solved correctly. These include knowledge of terms such as crowd-surfing and lincoln logs (a type of toy), and facts like zip-lining is an exciting activity. We find that a small fraction of the errors are caused by ambiguous or incorrect labels. There are several instances of spelling mistakes, a few of which change the meaning of the sentence.

6 Conclusion

With the saturation of static, single-metric leaderboards, there is growing consensus for the development of holistic evaluation benchmarks. This includes evaluation of systems on aspects of performance beyond just single error rate on in-distribution data; aspects such as performance on out-of-distribution data (Linzen, 2020), and evaluating generalizability, robustness and fairness (Ethayarajh and Jurafsky, 2020). In this work, we describe an approach to benchmark long-tail generalization, a necessary skill for NLP systems that truly understand language. We demonstrate the challenge posed by our splits to state-of-the-art models on several tasks; standard evaluation overestimates

model performance on long-tail utterances. Instead of releasing a random split as the only metric on official benchmarks, our simple method can be used, for a wide range of tasks, to expose additional challenges in the collected data at no annotation cost. Benchmarking long-tail generalization, in this manner, can test model behavior on a broad set of generalization challenges, which may be missed by evaluations that test specific skills in isolation.

Limitations

Evaluating a proposed benchmarking method such as ours is challenging, as there is no community consensus on what properties characterize an ideal benchmark. While we have argued that Likelihood Splits have a number of desirable properties, ultimately we intend Likelihood Splits to *complement* other options for creating benchmarks, not replace them. In particular, we do not aim to replace methods that require additional annotation and domain knowledge discussed in §2. In situations where previously collected datasets contain no or very few examples of a particular type, creating new data may be the only way to test models on that type of example. We view our approach as one lightweight option that dataset curators can choose to create a more holistic benchmark.

The properties of the Likelihood Splits that we have studied in this work do not fully explain what makes the Likelihood Splits harder. Dataset splits that explicitly test specific skills like length generalization and compositional generalization are good at exposing specific weaknesses in models. While it is hard to pinpoint the source of difficulty, our approach is complementary in that it can test a much broader set of skills that a narrow test may miss.

The difficulty of out-of-distribution generalization is higher in low resource languages, however, we show that the problem is yet not solved for NLP tasks even in the high resource English language. Our approach has the flexibility to use any autoregressive LM to score the utterances; large multi-lingual LMs such as BLOOM (Scao et al., 2022) can be used if appropriate.

The model performance gaps between random split and Likelihood Splits are small (2-4%) on some datasets (e.g. BOOLQ). We cannot guarantee that Likelihood Splits for a new dataset will be much more challenging than random splits. In such a situation, other complementary evaluation strate-

gies may be recommended to more strenuously challenge models.

Our approach has multiple knobs to control the properties of the splits created: (1) prompting/fine-tuning the LM, (2) controlling length variation, and (3) dataset specific choices such as label balancing. This choice gives dataset curators a lot of control to modify the approach. It is possible that the behaviour of the splits might be inconsistent under some changes. In our experiments, we find that qualitative findings are largely consistent, even across changes such as using a different language model.

Finally, there is no guarantee that the challenge posed is a fair generalization task (Geiger et al., 2019); we cannot guarantee that all skills needed to solve the test set can be learned from the training set. Nevertheless, since our approach partitions data that was collected under a single consistent protocol, it is more likely to be fair than methods that rely on an additional, separate annotation process to create test data.

Acknowledgements

We thank Xiang Ren for discussions and feedback in the initial stages of the work. We thank Rajarshi Das and colleagues at USC for feedback on drafts of this work. This work was funded in part by a gift from Open Philanthropy.

References

- Udit Arora, William Huang, and He He. 2021. [Types of out-of-distribution texts and how to detect them](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10687–10701, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Samuel R. Bowman and George Dahl. 2021. [What will it take to fix benchmarking in natural language understanding?](#) In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4843–4855, Online. Association for Computational Linguistics.
- Marc Brysbaert and Boris New. 2009. [Moving beyond Kucera and Francis: a critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English](#). *BEHAVIOR RESEARCH METHODS*, 41(4):977–990.
- Anthony Chen, Pallavi Gudipati, Shayne Longpre, Xiao Ling, and Sameer Singh. 2021. [Evaluating entity disambiguation and the role of popularity in retrieval-based NLP](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4472–4485, Online. Association for Computational Linguistics.
- J.K Chung, P.L Kannappan, C.T Ng, and P.K Sahoo. 1989. [Measures of distance between probability distributions](#). *Journal of Mathematical Analysis and Applications*, 138(1):280–292.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. [BoolQ: Exploring the surprising difficulty of natural yes/no questions](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [ELECTRA: Pre-training text encoders as discriminators rather than generators](#). In *ICLR*.
- Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. 2021. [The devil is in the detail: Simple tricks improve systematic generalization of transformers](#).
- Paula Czarowska, Sebastian Ruder, Edouard Grave, Ryan Cotterell, and Ann Copestake. 2019. [Don’t forget the long tail! a comprehensive analysis of morphological generalization in bilingual lexicon induction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 974–983, Hong Kong, China. Association for Computational Linguistics.
- Yanai Elazar, Nora Kassner, Shauli Ravfogel, Amir Feder, Abhilasha Ravichander, Marius Mosbach, Yonatan Belinkov, Hinrich Schütze, and Yoav Goldberg. 2022. [Measuring causal effects of data statistics on language model’s ‘factual’ predictions](#).
- Kawin Ethayarajh and Dan Jurafsky. 2020. [Utility is in the eye of the user: A critique of NLP leaderboards](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4846–4853, Online. Association for Computational Linguistics.
- Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui

- Zhang, and Dragomir Radev. 2018. [Improving text-to-SQL evaluation methodology](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.
- Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*.
- J. Fodor and Z. Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:3–71.
- Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. [Evaluating models’ local decision boundaries via contrast sets](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.
- Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. [Competency problems: On finding and removing artifacts in language data](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. 2019. [Posing fair generalization tasks for natural language inference](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4485–4495, Hong Kong, China. Association for Computational Linguistics.
- Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. [Beyond i.i.d.: Three levels of generalization for question answering on knowledge bases](#). In *Proceedings of the Web Conference 2021, WWW ’21*, page 3477–3488, New York, NY, USA. Association for Computing Machinery.
- Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzić, Rishabh Krishnan, and Dawn Song. 2020. [Pretrained transformers improve out-of-distribution robustness](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.
- Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. [Compositionality decomposed: How do neural networks generalise?](#) *Journal of Artificial Intelligence Research*, 67:757–795.
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2022. [Large language models struggle to learn long-tail knowledge](#).
- Divyansh Kaushik, Douwe Kiela, Zachary C. Lipton, and Wen-tau Yih. 2021. [On the efficacy of adversarial data collection for question answering: Results from a large-scale randomized study](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6618–6633, Online. Association for Computational Linguistics.
- Daniel Keysers, Nathanael Sch"arli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. [Measuring compositional generalization: A comprehensive method on realistic data](#). In *ICLR*. Additional citation for MCD splits.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. [Dynabench: Rethinking benchmarking in NLP](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.
- Najoung Kim and Tal Linzen. 2020. [COGS: A compositional generalization challenge based on semantic interpretation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.
- J. Peter Kincaid, Robert P. Fishburne, R L Rogers, and Brad S. Chissom. 1975. [Derivation of new readability formulas \(automated readability index, fog count and flesch reading ease formula\) for navy enlisted personnel](#). Institute for Simulation and Training.
- Nikita Kitaev and Dan Klein. 2018. [Constituency parsing with a self-attentive encoder](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova,

- Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.
- Brenden M. Lake and Marco Baroni. 2018. [Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks](#). In *ICML*.
- Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew E. Peters, Ashish Sabharwal, and Yejin Choi. 2020. [Adversarial filters of dataset biases](#). In *ICML*.
- Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. [A simple unified framework for detecting out-of-distribution samples and adversarial attacks](#). In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. [Question and answer test-train overlap in open-domain question answering datasets](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.
- Tal Linzen. 2020. [How can we accelerate progress towards human-like linguistic generalization?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.
- Linqing Liu, Patrick S. H. Lewis, Sebastian Riedel, and Pontus Stenetorp. 2021. [Challenges in generalization in open domain question answering](#). *CoRR*, abs/2109.01156.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Edward Loper and Steven Bird. 2002. [Nltk: The natural language toolkit](#). In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, page 63–70, USA. Association for Computational Linguistics.
- John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. 2020. [The effect of natural distribution shift on question answering models](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6905–6916. PMLR.
- John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. 2021. [Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7721–7735. PMLR.
- Aakanksha Naik, Jill Lehman, and Carolyn Rosé. 2022. [Adapting to the long tail: A meta-analysis of transfer learning research for language understanding tasks](#). *Transactions of the Association for Computational Linguistics*, 10:956–980.
- Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. [Stress test evaluation for natural language inference](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Benjamin Newman, John Hewitt, Percy Liang, and Christopher D. Manning. 2020. [The EOS decision and length extrapolation](#). In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 276–291, Online. Association for Computational Linguistics.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. [Adversarial NLI: A new benchmark for natural language understanding](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- Laurel J. Orr, Megan Leszczynski, Neel Guha, Sen Wu, Simran Arora, Xiao Ling, and Christopher Ré. 2021. [Bootleg: Chasing the tail with self-supervised named entity disambiguation](#). In *11th Conference on Innovative Data Systems Research, CIDR 2021, Virtual Event, January 11-15, 2021, Online Proceedings*. www.cidrdb.org.
- Jason Phang, Angelica Chen, William Huang, and Samuel R. Bowman. 2021. [Adversarially constructed evaluation sets are more challenging, but may not be fair](#).
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. [Hypothesis only baselines in natural language inference](#). In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.
- Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. [DynaSent: A dynamic benchmark for sentiment analysis](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404, Online. Association for Computational Linguistics.

- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Yasaman Razeghi, Robert L. Logan, Matt Gardner, and Sameer Singh. 2022. [Impact of pretraining term frequencies on few-shot reasoning](#).
- Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. 2019. [Likelihood ratios for out-of-distribution detection](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. [Beyond accuracy: Behavioral testing of NLP models with CheckList](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.
- Brian Roark, Margaret Mitchell, and Kristy Hollingshead. 2007. [Syntactic complexity measures for detecting mild cognitive impairment](#). In *Biological, translational, and clinical language processing*, pages 1–8, Prague, Czech Republic. Association for Computational Linguistics.
- Ofer Sabo, Yanai Elazar, Yoav Goldberg, and Ido Dagan. 2021. [Revisiting few-shot relation classification: Evaluation data and classification schemes](#). *Transactions of the Association for Computational Linguistics*, 9:691–706.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavathula, and Yejin Choi. 2021. [Winogrande: An adversarial winograd schema challenge at scale](#). *Commun. ACM*, 64(9):99–106.
- Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. [Bloom: A 176b-parameter open-access multilingual language model](#).
- Christopher Scialvolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. [Simple entity-centric questions challenge dense retrievers](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. [Compositional generalization and natural language variation: Can a semantic parsing approach handle both?](#) In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938, Online. Association for Computational Linguistics.
- Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. [We need to talk about random splits](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832, Online. Association for Computational Linguistics.
- Eric Wallace, Adina Williams, Robin Jia, and Douwe Kiela. 2021. [Analyzing dynamic adversarial training data in the limit](#).
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Victor H. Yngve. 1960. [A model and an hypothesis for language structure](#). *Proceedings of the American Philosophical Society*, 104(5):444–466.
- Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. [Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

A Appendices

A.1 Dataset Statistics

Refer to Table 6 for final split sizes. When creating the splits, we first partition the available data into train and evaluation (combined size of dev + test) sets using the methodology of each split (e.g. TMCD maximizes compound divergence, Likelihood Splits sort by an LM score and then partition the data). Then the evaluation set is randomly divided into dev and test sets.

Dataset	Train	Dev	Test	Total
SPIDER	5,966	1,034	1,034	8,034
SNLI	549,018	10,000	10,000	569,018
BOOLQ	7,617	2,540	2,540	12,697

Table 6: Sizes of train/dev/test sets for the dataset splits

The public release of the SPIDER dataset consists of 7000 training examples and 1034 validation examples (it also contains 1659 additional examples from older datasets which we do not use in our work). We use these 8034 examples to create all our splits. Shaw et al. (2021), one of the alternative splits that we compare against, use a subset of 4000 examples from the 7000 training examples. Hence, our results are not directly comparable to performance reported by them. The SQL programs for 6 (of 8034) examples in the dataset cannot be parsed uniquely and thus we cannot define compounds on these examples. We drop these examples when creating the TMCD split i.e. the training set of TMCD split contains 6 fewer examples.

The SNLI data contains 550152 training examples, 10000 dev examples and 10000 test examples for a total of 570152 examples. We drop examples where the gold label cannot be determined by majority vote. We also drop examples where the premise was labelled ‘Cannot see picture to describe.’ or the hypothesis is empty. This results in a filtered dataset of 569018 examples from which we create our splits.

The public release of BOOLQ contain 9427 labeled training examples, 3270 labeled development examples, and 3245 unlabeled test examples. Thus we create our splits from the 12,697 labelled train and development examples. We maintain the approximate 60/20/20 train/dev/test proportions of the original dataset when creating the splits.

A.2 Model Hyperparameters

We use the Transformers library (Wolf et al., 2020) for training and evaluation. All models were trained on Nvidia Quadro RTX 6000 GPUs (24GB GPU Memory).

We report hyperparameters for fine-tuning GPT-2 to create the Likelihood Splits in Table 7. We select the best checkpoint based on lowest perplexity by validating on 10% of the training data in each fold.

Hyperparameters for SPIDER models are in Table 8, for SNLI models in Table 9 and for BOOLQ models in Table 10. Note that we evaluate check-

Hyperparameter	Value
train batch_sz	32
lr_scheduler	constant
learning_rate	2e-5
optimizer	AdamW
eval steps	64
	SPIDER: 2000
max steps	SNLI: 15000
	BOOLQ: 3000

Table 7: Hyperparameters for fine-tuning GPT-2 (both medium and large) on the dataset folds to create Likelihood Splits

	T5	GPT-2
train batch_sz	8	2
grad acc steps	16	16
max_steps	10000	10000
lr_scheduler	constant	constant
learning_rate	1e-3	2e-5
optimizer	Adafactor	AdamW
max src_length	512	512
max tgt_length	256	256
src prefix	-	database question for table
tgt prefix	semanticparse:	generate the sql parse:
eval batch_sz	8	1
eval steps	256	128
num_beams	5	5

Table 8: Hyperparameters for the models trained on SPIDER.

points (see hyperparameter ‘eval steps’) during training to select the best checkpoint at the end of training.

For SPIDER, we follow Shaw et al. (2021) and tune the learning rate, batch size and maximum training steps for a T5-base model (Raffel et al., 2020) on a random split of the SPIDER dataset. Once we have found a hyperparameter setting, we apply the same setting on the all splits. We also report performance of a T5-small model on all splits trained with the same hyperparameters.

For SNLI and BOOLQ, we follow the default hyperparameters suggested by the original works. Additionally, we perform early stopping when performance on the validation set fails to improve for a specified number of evaluations.

A.3 Effect of k on Likelihood Splits

When creating Likelihood Splits, the number of folds k for fine-tuning (§3.2) is a choice left to the creator of the benchmark. In our work, we

	ROBERTA	ELECTRA	
		base	large
train batch_sz		32	
max seq length		128	
lr_scheduler		linear	
optimizer		AdamW	
adam_beta1		0.9	
adam_beta2	0.98		0.999
adam_epsilon		1e-6	
num epochs	10		3
warmup ratio	0.06		0.1
layer. lr decay	1.0	0.8	0.9
learning_rate	1e-5	1e-4	5e-5
weight decay	0.1		0.0
eval batch_sz		32	
eval steps		256	
patience	20		n/a

Table 9: Hyperparameters for the models trained on SNLI. Patiences refer to number of evaluations with no improvement before early stopping.

set $k = 3$ as an arbitrary choice prior to running the task models i.e. it was not tuned based on task model performance. We conduct additional experiments to test the effect of changing the value of k by setting $k = 5$ and generating new splits.

On SPIDER, when $k = 5$ instead of $k = 3$, Likelihood scores are highly correlated with a Pearson’s r of 0.90. However, the process of balancing atoms (described in §3.4) causes the evaluation sets to look more different. When controlling for length, 77% of the evaluation set examples are the same; 80% of the evaluation examples are the same otherwise. We report the accuracy of T5-base (the most competitive baseline model on SPIDER) on the new splits with $k = 5$ in Table 11. The new splits are more difficult by about 2%. We observe the same trend where controlling for length makes the splits less challenging.

On BOOLQ, when using 5 folds instead of the 3 folds, Likelihood scores are highly correlated with a Pearson’s r of 0.96 and 89% of the evaluation set examples are the same. Accordingly, the ELECTRA-large accuracy only changes slightly from 82.6% to 83% on the new test set and is still lower than the random split accuracy. While there exists an indication that controlling for length makes the ll_split (-len) splits more challenging, the effect of controlling for length becomes more pronounced when $k = 5$. We report the effect of k on BOOLQ performance in detail in Table 12.

We report the effect of changing k on SNLI accuracy in Table 12. Since the SNLI dataset is an order

	ROBERTA	ELECTRA	
		base	large
train batch_sz		8	
grad acc steps		4	
max seq length		512	
lr_scheduler		linear	
optimizer		AdamW	
adam_beta1		0.9	
adam_beta2	0.98		0.999
adam_epsilon		1e-6	
num epochs	10		5
warmup ratio	0.06		0.1
layer. lr decay	1.0	0.8	0.9
learning_rate	1e-5	1e-4	5e-5
weight decay	0.1		0.0
eval batch_sz		8	
eval steps		128	
patience	10		n/a

Table 10: Hyperparameters for the models trained on BOOLQ. Patience refers to number of evaluations with no improvement before early stopping.

System	SPIDER				
	Random	ll_split (k=3)		ll_split (k=5)	
		(-len)		(-len)	
T5-base	78.6	66.0	71.3	64.8	69.2

Table 11: Effect of k on the difficulty of Likelihood Splits of SPIDER. The accuracies of T5-base on the Likelihood Split are comparable and significantly lower than the accuracy on Random split. Controlling for length decreases the difficulty in both cases.

of magnitude larger than the SPIDER and BOOLQ datasets, the number of folds has less of an impact on the LM fine-tuning. As a result, Likelihood scores for $k = 3$ and $k = 5$ are highly correlated with a Pearson’s r of 0.99. When controlling for length, 89% of the evaluation set examples are the same; 92% of the evaluation examples are the same otherwise. Accordingly, we see much smaller differences in model performance on the new splits; the ROBERTA model accuracies change by at most 0.9% on the new test sets.

A.4 SPIDER: Variation of TMCD Related Properties

Past work by Keysers et al. (2020) has established the terms of atom and compound “divergence” to quantitatively describe the extent to which the distributions of the atoms and compounds differ between the train and evaluation sets. They used the Chernoff coefficient (Chung et al., 1989) to measure distribution similarity:

System	SNLI					BOOLQ				
	Random	ll_split (k=3)		ll_split (k=5)		Random	ll_split (k=3)		ll_split (k=5)	
		(-len)	(-len)	(-len)	(-len)		(-len)	(-len)		
ROBERTA-base	89.6 ±0.4	79.3	77.1	78.4	75.0	74.9 ±0.4	71.6	71.2	71.1	70.3
ROBERTA-large	90.5 ±0.5	82.3	79.2	82.1	79.0	84.4 ±0.6	79.3	78.9	78.4	78.3
ELECTRA-base	90.5 ±0.2	80.1	78.3	80.1	77.6	78.8 ±1.1	74.1	74.3	75.0	73.9
ELECTRA-large	91.0 ±1.3	82.6	81.6	84.0	80.6	85.5 ±0.6	82.6	82.1	83.0	80.6
Human Accuracy	88.7 ±0.8	83.6	84.4	83.0	84.0	-	-	-	-	-

Table 12: Effect of k on the difficulty of Likelihood Splits of SNLI and BOOLQ. There are some accuracy differences on BOOLQ, however the values are comparable and significantly lower than the accuracy on Random splits. The accuracy differences are less pronounced on SNLI.

$$C_\alpha(P \parallel Q) = \sum_k p_k^\alpha q_k^{1-\alpha} \in [0, 1] \quad (1)$$

where p_k and q_k are the probability of a particular atom/compound k being in the train and test set respectively. The divergence is then $1 - C_\alpha$. The ‘‘atom’’ divergence uses $\alpha = 0.5$ as a symmetric divergence score. The ‘‘compound’’ divergence used $\alpha = 0.1$ to give more importance to the occurrence of a compound in the train set rather than trying to make the distributions of train and test set similar.

Split	Atom	Compound
Random	0.077	0.046
Length	0.120	0.092
Template	0.105	0.089
TMCD	0.296	0.322
ll_split	0.083	0.054
ll_split (-len)	0.081	0.049
ll_split prompt	0.093	0.056
ll_split prompt (-len)	0.094	0.052

Table 13: Atom and Compound divergence (on the logical form side) between train and dev sets of various splits. Although we ensure every atom appears at least once in the train set, a high atom divergence demonstrates the challenge of learning rare atoms. A greater than random compound divergence emerges denoting a need for compositional generalization.

A.5 SPIDER: Variation of SQL Hardness

We use a tool provided by the SPIDER dataset creators to evaluate hardness. The tool assigns a rating from easy, medium, hard or extra hard to every example based on the complexity of the SQL parse. Complexity is evaluated in terms of the number of join and aggregation operations, and nested SQL statements. We find that the Likelihood Splits are skewed towards putting more complex examples

in the evaluation set compared to the test set (see Figure 5).

A.6 SPIDER: Input Length Variation

As expected, the likelihood assigned by the language model (LM) is negatively correlated with sequence length meaning i.e. longer sequences tend to have lower likelihood. This can be seen from Figure 6, where ll_split and ll_split_pt tend to put longer utterances in the lower likelihood evaluations set. Accounting for length by dividing the data within buckets makes the distribution of train and test sets align better and remove the added difficulty of length generalization. The $length$ split poses this challenge which has been established to be a difficult ability for generation models to acquire (Newman et al., 2020). Note that the distributions do not match exactly since examples need to be swapped between train and evaluation set to meet the atom constraint (evaluation cannot contain any unseen atoms).

A.7 SPIDER: Variation of Query Parse Structure

We analyze the complexity of the parse structure of the queries. Following Wallace et al. (2021), we parse the queries using the Benepar parser (Kitaev and Klein, 2018) based on T5 small (Raffel et al., 2020). We report the distributions of mean and max parse tree depth as well as the syntactic complexity of the utterance based on the Yngve score (Yngve, 1960; Roark et al., 2007). The Yngve score essentially measures the average number of left branches on the path from the root of the parse tree to every word in the sentence and can be thought of as measuring the number of spans that need to be coordinated.

We can see that the dev set of the ll_split is on average more complex than its train set along all 3

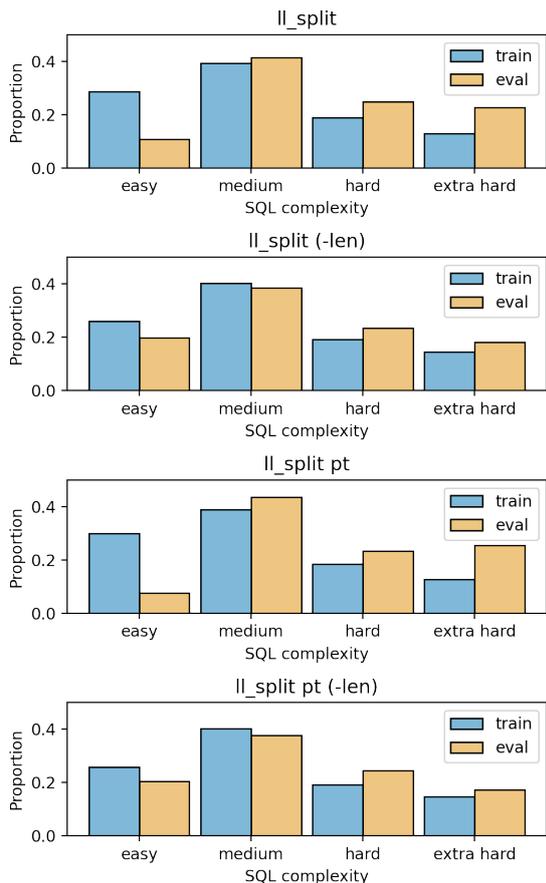


Figure 5: SPIDER: Distribution of SQL programs of varying complexity in the train and development set of various splits. Likelihood Splits show a skew towards training on easy examples and evaluating on harder examples.

metrics considered (see Figure 8). Moreover, these metrics are correlated with utterance length, and controlling for it in the *ll_split (-len)* split makes the difference less pronounced.

A.8 SPIDER: Error Analysis

We analyze the performance of T5-base on the development set of *ll_split pt*. In particular, we test whether the presence of novel compounds and SQL query hardness are sufficient to explain the difficulty.

We call compounds in the SQL programs of the development set as ‘novel’ if they do not occur in the training set of the split. 25.5% of the dev set examples in the random split contain at least one novel compound as opposed to 43.6% of the dev set examples of *ll_split pt*. From Table 14, we see that T5-base performance is lower in both categories of examples. Projecting for expected performance on dev set of *ll_split pt* assuming the examples

Split	Random	<i>ll_split pt</i>
Percent. of examples with a novel compound	25.5%	43.6%
Acc on examples with novel compounds	61.4%	45.5%
Acc on remaining examples	87.1%	79.8%
Acc on the dev set	80.6%	64.8%
Projected accuracy		75.9%

Table 14: SPIDER: The presence of novel compounds alone does not explain the difficulty of the *ll_split pt*. Projecting the random set accuracies using the percentage of examples with novel compounds in *ll_split pt* over-estimates dev set performance.

were as difficult as examples from the random split over-estimates the performance of T5-base.

We report performance of T5-base on the dev sets grouped by the SQL hardness metric (described in Appendix A.5) in Table 5. We see that accuracy on *ll_split pt* is lower than the accuracy on the random set within each SQL complexity bucket. If the sole source of difficulty was the larger proportion of harder examples, projecting the random set accuracies would correctly estimate dev set performance on *ll_split pt*. However, the projection is an over-estimate. Thus, the hardness metric alone does not explain the difficulty of the proposed split.

A.9 SNLI: Input Length Variation

From Figure 9, we see that the Likelihood Splits put longer premises and hypotheses in the evaluation set. Controlling for length completely removes this skew while increasing the difficulty of the splits (Table 3). This means that if we remove the factor of length from likelihood, the remaining examples have lower likelihood for other reasons; reasons that contribute to difficulty.

A.10 SNLI: Distribution of Rare Words

We report the fraction of test sets words that are rare for various splits in Figure 7. This evaluation combines the premise and the hypothesis i.e. it considers the full task input. In order to remove typographical errors, we only consider words that occur in a wordlist of English words (<https://github.com/dwyl/english-words>). We define rare words as words that occur at most 1 time per million words statistics collected in SUBTLEXus (Brysbaert and New, 2009). This process results in a list of 13478 low frequency words that occur in the SNLI dataset. We find that Likelihood Splits

put examples with a significantly large fraction or rare words in the evaluation set. Controlling for length increases the fraction of rare words since length is removed as a factor from likelihood.

A.11 SNLI: Variation of Syntactic complexity

We compute Yngve scores for premise and hypothesis of examples as described in Appendix A.7. The complexity of premise and hypothesis in developments sets of Likelihood Splits is higher than in the corresponding train sets (see Figure 10). Controlling for length removes this skew in the premise. However, the length controlled splits tend to have less syntactically complex hypotheses in the evaluation sets. This is surprising because the length-controlled variants are actually more difficult for the model; human performance is higher on length-controlled splits.

A.12 SNLI: Variation of Reading Level

We compute the Flesch-Kincaid reading level (Kincaid et al., 1975) for premise and hypothesis of examples. This score takes into account the number of syllables per word in the sentence. The reading grade (complexity) of premise and hypothesis in developments sets of Likelihood Splits is higher than in the corresponding train sets (see Figure 11). Controlling for length does not fully remove this skew and the evaluation examples retain more complex sentences than in the training set.

A.13 SNLI: Error Analysis

In Table 15, we present some examples from the development set of *ll_split (-len)* where the fine-tuned ROBERTA-large model predicts incorrectly. We divide them into categories: examples requiring external world knowledge, examples where a typo changes the meaning of the example, and examples with ambiguous or incorrect labels.

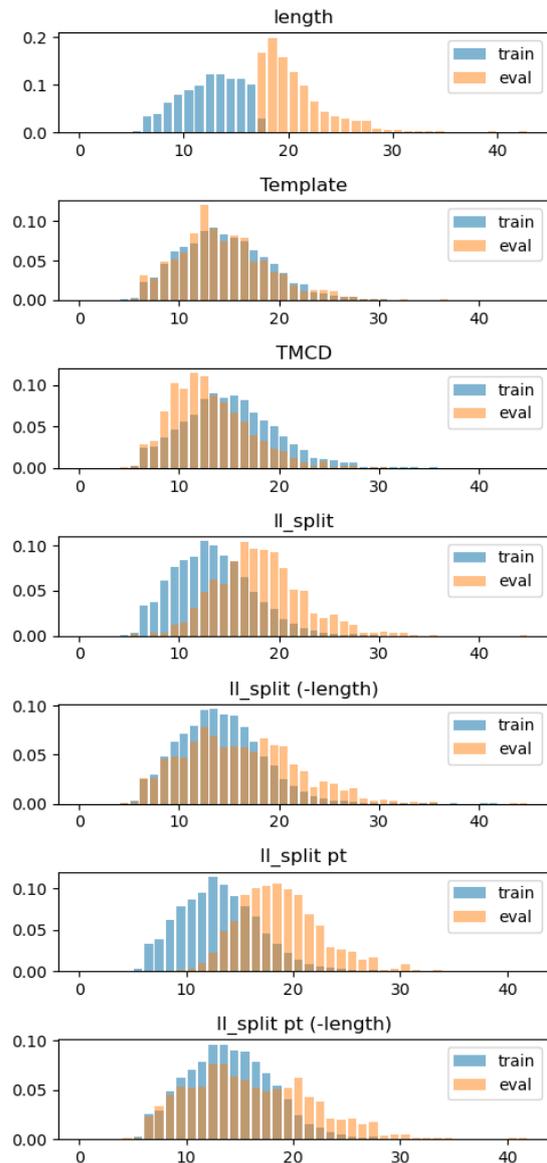


Figure 6: SPIDER: Input length variation for the splits. Y-axis is the distribution of examples within each length bucket of the X-axis

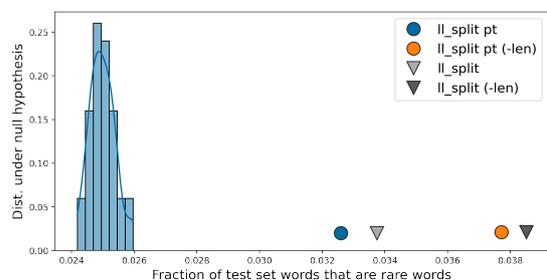


Figure 7: SNLI: Fraction of development set words that are rare in the premise and hypothesis of various splits. The dev sets for *ll_split* seem to contain a larger fraction of rare words than random splits. Normalizing for the length seems to retain more rare words.

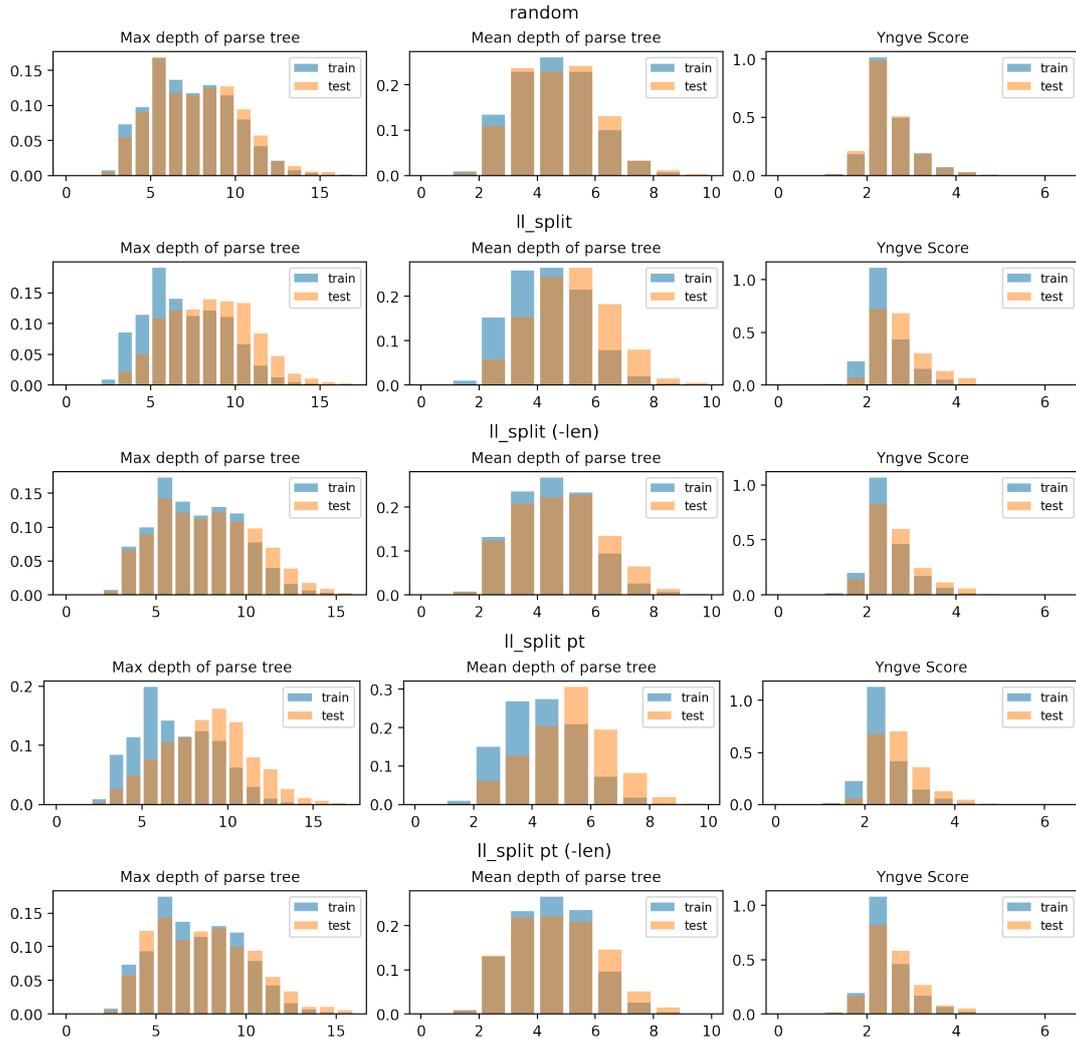


Figure 8: SPIDER: Distribution of features computed on the parse tree of the input query. The dev sets for *ll_split* seem to contain more complex utterances across all 3 metrics considered. Normalizing for the length seems to reduce the effect. The metrics are mean and max depth of parse tree and Yngve score which is a metric

Error Type	Premise	Hypothesis	Gold Label	Predicted Label
Requires External Knowledge	A young boy is holding on and riding a zip line down a hill.	A exciting adventure!	entailment	neutral
	A young child is watching a toy construction brick construct.	A child is using lincoln logs.	neutral	contradiction
	A performer is jumping off the stage into a crowd of fans.	The artist is crowdsurfing.	entailment	neutral
	A couple holds up their child on a series of large steps while others are also traversing the steps.	A fourteen year old is restrained from the museum.	contradiction	neutral
Typo	Martial artists perform in front of a crowd outdoors.	There is a crown outdoors.	entailment	neutral
Ambiguous / Incorrect Labels	Technicians working in underground.	People work underground while dinosaurs attacked	neutral	contradiction
	A young gentlemen with a blue tie talking into a microphone.	High winds will interfere with microphone recording.	entailment	neutral

Table 15: SNLI: Error analysis of ROBERTA-large on examples from *ll_split (-len)*.

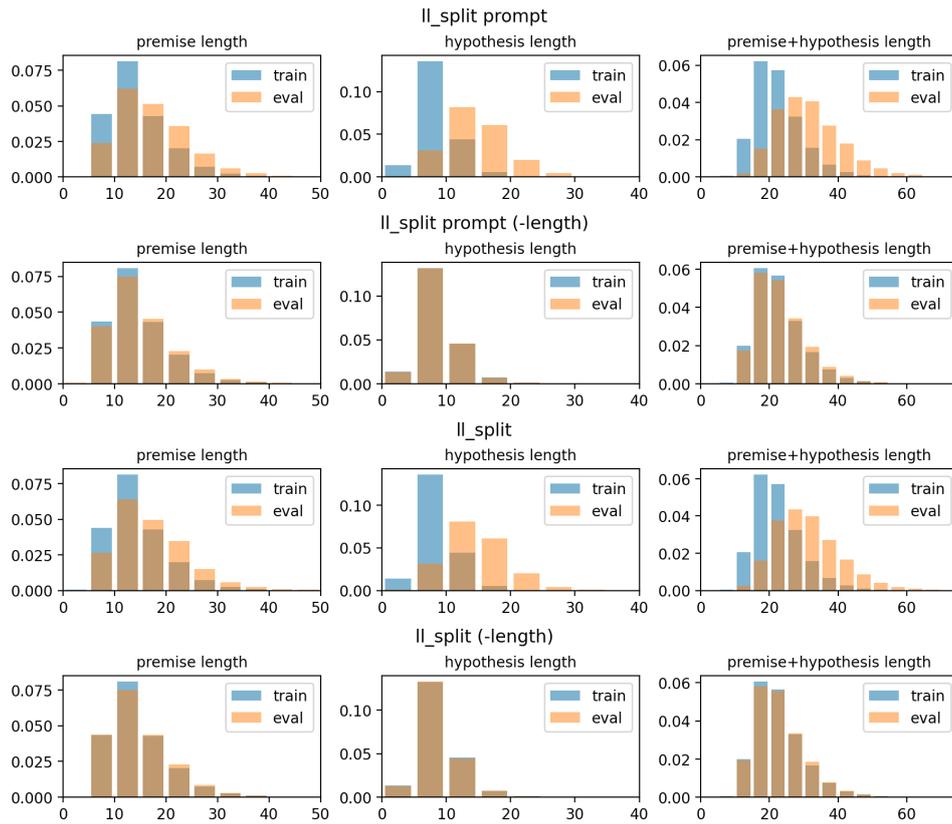


Figure 9: SNLI: Input length variation of premise and hypothesis for the splits.

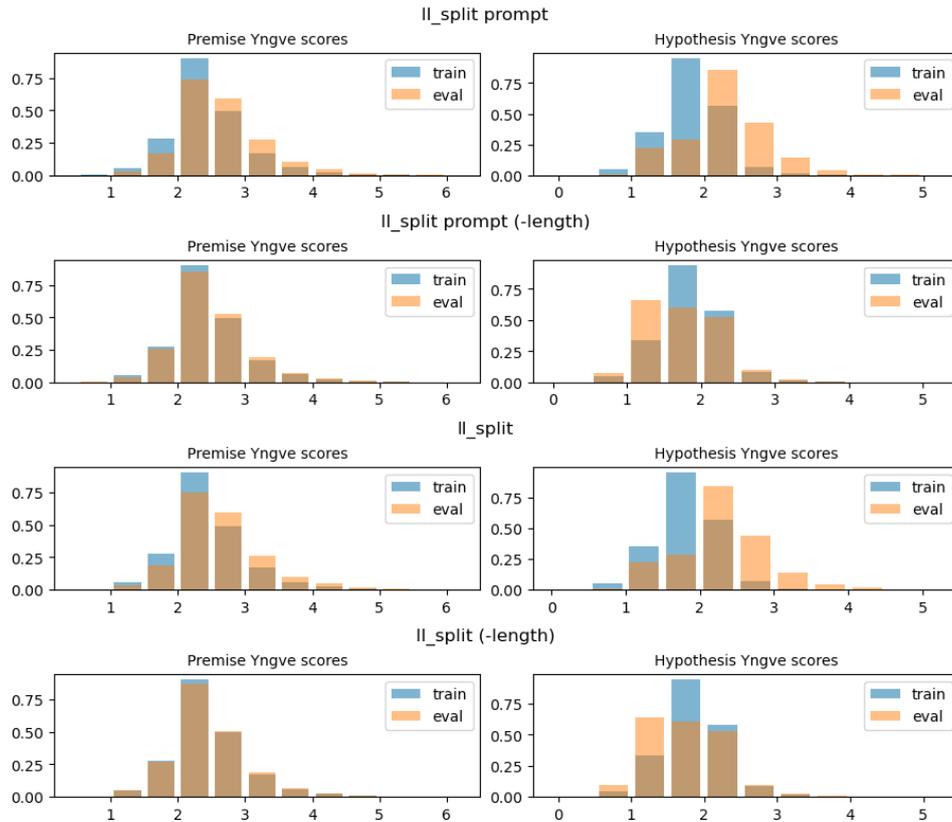


Figure 10: SNLI: Distribution of Yngve scores (syntactic complexity) computed on the parse tree of the premise and hypothesis. The dev sets for Likelihood Splits contain more complex utterances.

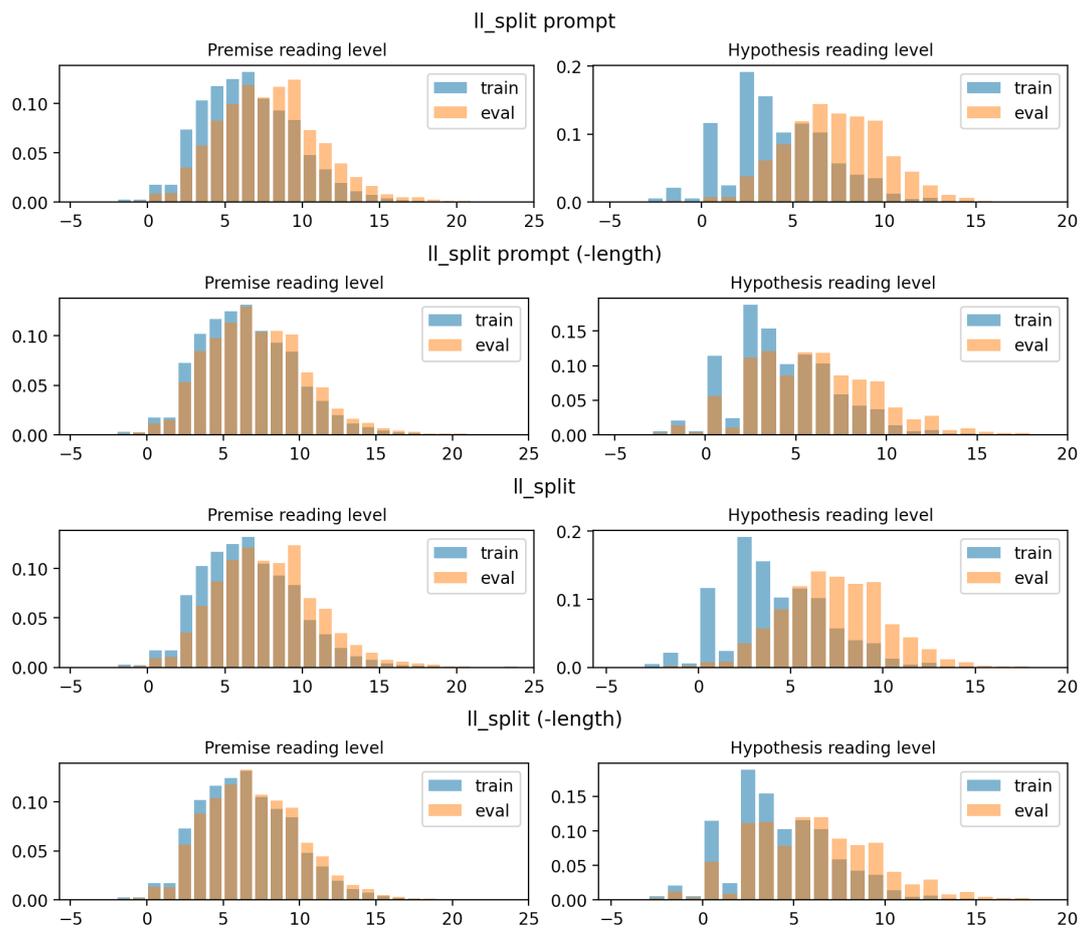


Figure 11: SNLI: Distribution of Flesch-Kincaid reading level for the premise and hypothesis in various splits. The dev sets for *ll_split* seem to contain more complex utterances. Normalizing for the length seems to reduce but not remove the effect.

Despite being highly effective, a common weakness of this family of approaches is that they synthesize code-switched sentences using word lexicons - opening up several potential quality issues in the pretraining data. For instance, the SOTA algorithm AA as well as other related works (Lin et al., 2020; Li et al., 2022a) use non-contextual, one-to-one word translations obtained from MUSE lexicons (Lample et al., 2018) -- which can be problematic in a variety of cases. Firstly, failing to factor sentence-level context can cause violations in linguistic agreement, such as gender, case, tense and verb agreement. Secondly, AA cannot adequately handle contextual synonyms or polysemes (often estimated to constitute up to 80% of English words (Miller, 1998; Geeraerts, 1993)), with Lin et al. (2020) and Pan et al. (2021) assigning random word senses (and thus, translations) for polysemes -- regardless of context. Thirdly, one-to-one word translations create further issues, including the handling of multi-word expressions (eg. ``get out") and multi-word entities (eg. ``New York") -- with these problems aggravating for agglutinative languages. Finally, the bilingual MUSE lexicons themselves have been shown to be of dubious quality across a variety of languages (Kementchedjheva et al., 2019), and using these for multilingual code-switching can propagate such errors manifold. This is illustrated in Figure 1, which shows an example of AA noising taken directly from Pan et al. (2021). With 6 errors in a sentence of 14 words (refer Appendix A.1 for a detailed examination of these errors), we contend that these limitations could cause significant corruption in the pretraining corpus.

We, thus, hypothesize that although AA has been effective in a variety of scenarios, the raised issues could lead mRASP2 to underperform. For this reason, we propose Contextual Code-Switching (CCS) - a novel approach for extracting contextual, many-to-many word translations, leveraging massive² NMT models, and then using these for noising the pretraining corpus. We conduct experiments on 3 different language families: Romance, Uralic, and Indo-Aryan, and report significant average improvements across the board, with gains of up to +5.5 spBLEU. We also find that CCS models narrow the gap with or outperform massively multilingual models like mBART50 (Tang et al., 2021)

²-'massive' in this work signifies the size of pretraining data, while 'large' refers to the large Transformer architecture (with 12 encoders and 12 decoders)

and mRASP2, despite using a tiny fraction of the data and compute. Lastly, we conduct ablation studies to analyze some of the most important factors to consider when synthesizing code-switched text for multilingual NMT pretraining - which constitutes another key, novel contribution of this work.

Our major contributions are, thus, as follows:

1. Firstly, we show that improving the quality of synthetic code-switching can significantly enhance pretraining of multilingual NMT models across various high, medium, low-resource and agglutinative language pairs (3.4.1).
2. Secondly, we demonstrate how massively multilingual NMT models can be harnessed to pre-train smaller models that yield comparable or better performance -- all while using a fraction of the training data and compute (3.4.2).
3. Thirdly, we empirically analyze and discuss some of the key factors that can enhance NMT pretraining on code-switched data - including context, many-to-many substitutions, code-switching language count, and fine-tuning - furthering scientific understanding (3.5)
4. Finally, for greater scalability of our approach, we propose useful variations of CCS that could alleviate potential resource dependencies (3.4.3) and increase efficiency (8.1) -- all while maintaining comparable performance.

2 Approach

2.1 Definitions

Given a set of N languages $L = L_1, L_2 \dots L_N$, multilingual NMT is defined as the task of learning a many-to-many mapping function θ from source language L_a to target language L_b . Code-switching refers to the phenomenon of shifting between two or more languages in a sentence. This work explores functions C that can synthetically code-switch corpora for pretraining multilingual NMT models.

2.2 Aligned Augmentation

Aligned Augmentation constructs synthetically code-switched datasets using multilingual lexicons. These lexicons are generated by interlinking bilingual MUSE dictionaries through a pivot language, English. Given a sentence S , a code-switched sentence $C_{AA}(S)$ is created by looking up word translations in the lexicon and, if available, substituting with replacement ratio r . A bilingual lexicon is used to code-switch parallel corpora and the multilingual one for monolingual data, with $r = 0.9$.

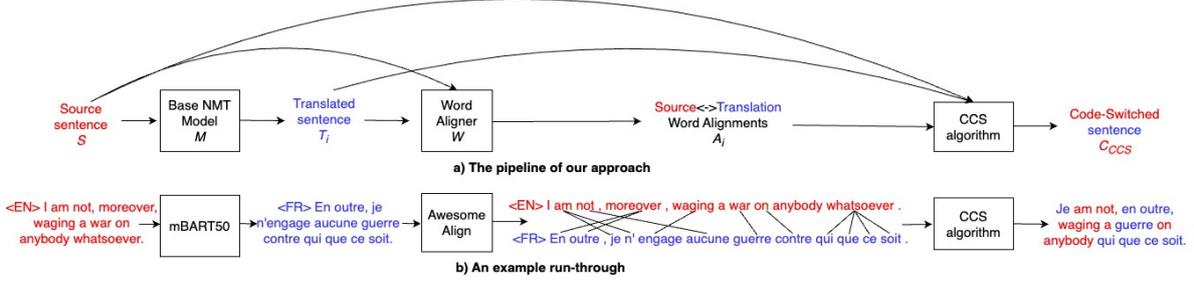


Figure 2: a) The pipeline of, and b) an example illustrating our approach. Alignments between punctuation marks have been omitted for ease of illustration. Color coding signifies language in the code-switched sentence $C_{CCS}(S)$. $C_{CCS}(S)$ is later fed to a Transformer and trained using Cross Entropy and Contrastive Loss (Section 2.4).

2.3 Contextual Code-Switching

Contextual Code-Switching (CCS) seeks to obtain contextual, many-to-many word translations, later used for code-switching parallel and monolingual corpora. Given a source sentence S , we generate the code-switched sentence $C_{CCS}(S)$ as follows:

1. Use a 'base' NMT model M to translate S in n languages ($n \geq 1$) to obtain translations $\{T_1 \dots T_n\}$. Here, $M_{L_i}(S) = T_i$ where $M_{L_i}(S)$ is translation of S by M to L_i
2. Use word aligner W to align S with translations $\{T_1 \dots T_n\}$ and obtain word alignments $\{A_1 \dots A_n\}$; where $W(S, T_i) = A_i$
3. Generate the 'noised' code-switched sentence $C_{CCS}(S)$ using the CCS algorithm, with S , $\{T_1 \dots T_n\}$ and $\{A_1 \dots A_n\}$ as inputs

The CCS algorithm works by generating "connected components" of aligned words. For a given translation T_i and word alignment A_i , we iterate over each source word w_i in S , extract target words from T_i which are aligned with w_i (as specified by A_i), and then iteratively extract the source words aligned to these target words and vice versa, until convergence. This yields all possible many-to-many word alignment combinations, from which code-switching is carried out through random, iterative substitutions in S until the replacement ratio is reached -- yielding the final sentence $C_{CCS}(S)$. $C_{CCS}(S)$ could be code-switched in one ($n = 1$) or more ($n \geq 1$) languages, which we term Bilingual (BLCS) and Multilingual Code-Switching (MLCS) respectively. Although mRASP2 uses MLCS, we show in Section 3.5 that BLCS mostly performs better and is also more efficient. We illustrate our approach in Figure 3 and provide an example where the many-to-many substitution enables CCS to correctly translate the word *moreover* as *en outre*. Finally, we provide pseudo-code (containing finer

technical details) in Algorithm 1 of Appendix A.2.

2.4 Training

To ensure a fair comparison, we replicate the training conditions proposed by Pan et al. (2021) while training our AA and CCS models. Our training dataset D consists of shuffled parallel and monolingual sentences, noised using the respective code-switching approaches. These code-switched sentences are input to the encoder. Meanwhile, the target sentences are the reference sentences for parallel corpora and the denoised (original) sentences for monolingual corpora. A special token indicating language ID is prepended to all source and target sentences. Finally, the model is trained using a loss function \mathcal{L} that jointly optimizes Contrastive Loss \mathcal{L}_{CON} and Cross Entropy Loss \mathcal{L}_{CE} as follows:

$$\mathcal{L} = \mathcal{L}_{CE} + |s| * \mathcal{L}_{CON}$$

where: $\mathcal{L}_{CE} = \sum_{x,y \in D} P_{\theta}(y|x)$, and

$$\mathcal{L}_{CON} = - \sum_{x,y \in D} \log \frac{e^{\text{sim}^+(\mathcal{E}(x), \mathcal{E}(y))/\tau}}{\sum_{a,b \in B} e^{\text{sim}^-(\mathcal{E}(x), \mathcal{E}(b))/\tau}}$$

Here, \mathcal{E} denotes the average pooled encoder output, 'sim' computes positive and negative semantic similarity for a pair of sentences, as denoted by sim^+ and sim^- respectively. Temperature τ controls the strength of penalties during contrastive learning and is set to 0.1. B denotes the mini-batch in dataset D that (x, y) belong to. As shown by Pan et al. (2021), Contrastive Learning aligns semantic representations of source and reference sentences (x, y) while pushing away all 'negative' targets -- approximated to other reference sentences in the mini-batch for convenience. Finally, $|s|$ is the average sentence length (token count) that balances the token-level cross entropy loss and sentence-level contrastive loss.

3 Experiments

In this section, we seek to answer the following Research Questions:

1. How does CCS perform against the SOTA non-contextual algorithm, AA, and how does this vary across language pairs? (3.4.1)
2. How do small CCS models compare against SOTA massively multilingual models? (3.4.2)
3. How can CCS alleviate its potential resource dependencies and scale beyond? (3.4.3)
4. What are the key factors to consider when pretraining on code-switched text? (3.5)

As part of 3 Case Studies, we evaluate CCS on 3 different language families - namely, the high-resourced Romance, the agglutinative Uralic, and the low-resourced Indo-Aryan - in order to test its efficacy under different scenarios. For each family, we train from scratch small multilingual AA and CCS baselines on its languages. Small models have the benefit of minimizing negative interference, while also satisfying our resource constraints.

3.1 Corpora

Tables 1a and 1b show the training corpora statistics used per language family. We use this data for training all AA and CCS models in this work. For a fair comparison, only languages from each family that are present in the training set of mRASP2 and mBART50, and contain MUSE dictionaries are chosen for training. Portuguese (Pt) is taken as the zero-shot case, explored in Section 3.4.3, so no parallel corpus is used. The datasets constituting the training corpora are detailed in Section A.4.3.

For validation, we choose the last 1000 sentences from the bitext for each language. For testing, we use the latest available WMT test sets for each language into and out of English. Table 2 specifies these test sets, which are used for all experiments in this work. Regarding metrics, following related work, we use spBLEU-101³ (Goyal et al., 2022) to evaluate all baselines in this work, but also provide ChrF++ (Popović, 2017) and COMET (Rei et al., 2020) results in Appendix A.5.1. We observe that these metrics largely agree with each other.

3.2 Preprocessing (CCS)

In order to apply the CCS algorithm (Section 2.3), we first generate ‘base’ model translations and word alignments using the fine-tuned mBART50 and

³referred to as spBLEU in this work for brevity

Romance		Uralic		Indo-Aryan	
En	7.5M	It	7.5M	En	20M
Es	7.5M	Ro	7.5M	Fi	16M
Fr	7.5M	Pt	7.5M	Et	8.1M
				Gu	650K

(a) Monolingual data

Romance		Uralic		Indo-Aryan	
En-Es	1.8M	En-It	1.7M	En-Fi	4M
En-Fr	1.8M	En-Ro	364K	En-Et	2.3M
				En-Gu	12K

(b) Parallel data

Table 1: Statistics of training corpora used in this work

awesome-align (Dou and Neubig, 2021) models respectively. For the former, we use the corresponding multilingual 1-n, n-1, and n-n models (based on the language pair) and generate translations with a beam size of 5. For the latter, we fine-tune awesome-align with a subset (300K parallel sentences) from our training corpus using the Translation Language Modeling and Self-Training objectives, as suggested by Dou and Neubig (2021). Where possible, we attempt to have this subset uniformly distributed across all languages (except in low-resourced Indo-Aryan, where 12K En-Gu and 288K En-Hi sentences are used). This setup of ‘base’ and word alignment models is used for training all CCS baselines in this work unless otherwise specified (such as in Section 3.4.3).

3.3 Experimental Settings

We use the vanilla Transformer (Vaswani et al., 2017) with 6 encoder and 6 decoder (6e6d) layers to train all models in this work, except in Table 3 - where ‘large’ CCS baselines with 12 of each (12e12d) are used for fair comparison against massively multilingual models. We use a batch size of 4000 and a learning rate of 0.0001, with a polynomial decay scheduler and 5000 warm-up updates. We use an Adam optimizer with $\epsilon = 1e-6$. For regularization, we use dropout of 0.1 and weight decay of 0.001. We use automatic mixed precision and an update frequency of 4 to speed up training. We conduct validation every 1000 updates and use a patience value of 10 for early stopping. We train each model only once since a random seed of 0 is set everywhere. For tokenization, we use sentencepiece (Kudo and Richardson, 2018). Sentencepiece models using a unigram language model are trained on the corresponding corpora with a vocabulary size of 32000 and character coverage of 1.0. Following Pan et al. (2021), we use a replacement ratio of 0.9 in AA models while for CCS, we use 0.55, 0.75, and 0.1 for the Romance, Uralic and Indo-Aryan

families after grid-search optimization. We detail infrastructure and training costs in Section A.4.

3.4 Results

3.4.1 Comparison with AA

In this section, we compare AA baselines with vanilla CCS models, when trained from scratch under identical conditions. Since Pan et al. (2021) already showed AA is the SOTA pretraining algorithm, we do not recreate other denoising approaches here - however, we do compare against two of the best-performing massive models from related work, mBART50 and mRASP2, in Section 3.4.2. Table 2 shows results for all 3 Case Studies.

We note some interesting trends. Firstly, while significant improvements are observed across all language families, the margin of gain varies. The highest gains are observed for the high-resourced Romance and the agglutinative Uralic families -- while the latter benefits greatly from many-to-many substitutions (Table 6), the former is quite noteworthy given that for Romance languages, MUSE lexicons are available in all directions (not just English-centric) so AA models would be strongest here. Meanwhile, the low-resourced Indo-Aryan family, which is non-agglutinative and suffers from low-quality 'base' model translations (as observed from inspection of mBART50 translations by native speakers), the margin is relatively lower. Nonetheless, on average, CCS still outperforms the SOTA approach AA and, as we shall see in Section 3.4.2, even massive models like mBART50 and mRASP2 - despite using far lesser data overall. Moreover, we show in Table 7 how techniques like BiLingual Codeswitching (BLCS) can further boost CCS by +1 to +2 spBLEU, making the total gap against AA +2 to +3 spBLEU points for the Indo-Aryan family.

Secondly, we observe that CCS mostly performs equally well for En-X and X-En pairs (with few exceptions), but AA varies considerably and performs better for the latter. This could be because X-En is generally an easier translation task than En-X, owing to the abundance of high-quality English target-side data. CCS bridges the gap between these two tasks and improves consistency - likely due to the higher quality codeswitching systematically benefiting cross-lingual representations overall.

Lastly, we note that the spBLEU gains in Table 2 translate to comparably large improvements in ChrF++ and COMET (Table 10)

3.4.2 Comparison with Massively Multilingual Models

We now proceed to compare our CCS systems against two SOTA massively multilingual models mBART50 (Tang et al., 2021), trained on 50 languages, and mRASP2 (Pan et al., 2021), trained on 32 languages; in Table 3. Table 3a contains ratios of the monolingual and parallel data used by the massive models w.r.t. ours, per language family, and can help in a meaningful interpretation of the results in Table 3b. While we use significantly lesser data for the Romance family, we attempt to match the parallel data used by mRASP2 for the Uralic and Indo-Aryan families. This data is comparable to or slightly more⁴ than that used for fine-tuning mBART50. As baselines, we use the AA and CCS models from Table 2 and their large (12e12d) variants. We also include a fine-tuned (ft) version of CCS (large) - created by pretraining on the code-switched monolingual data but leaving the parallel data unnoised for subsequent multilingual fine-tuning. We explore the impact of fine-tuning in greater detail in Section 3.5.

However, considering mBART50 and mRASP2 are massive models designed to scale to a much larger set of languages, we emphasize that Table 3 is *not* intended to be a head-to-head comparison; rather it is meant to position our work against the wider, popular SOTA. Our motive here is two-fold: a) to provide a way to harness massive models for pretraining and perform comparably or better, and b) to estimate the potential impact of scaling up CCS models - thus suggesting a worthy new direction of future exploration for pretraining better massive multilingual models. The first purpose could be especially useful for academics with fewer resources, while the second is likely better suited to groups capable of training massive models, eg. large industrial labs.

We achieve the first purpose by showing that the CCS (large) model performs comparably to mBART50 despite using substantially lesser data (such as in the Romance family), though mRASP2 retains a larger gap. But, when comparable data is used, as in the other families, CCS (large) models consistently outperform massive models by significant margins, despite using lesser monolingual data. The only exception is X-En where mBART50

⁴Relatively speaking here. Note that the overall data gap is still heavily biased towards mBART50, with up to 90x more monolingual data than our CCS models

	En-Es wmt13		En-Fr wmt14		En-It wmt09		En-Ro wmt16		Avg		Δ	
	→	←	→	←	→	←	→	←	→	←	→	←
	AA	25.0	26.2	28.8	28.7	23.8	26.8	18.7	24.1	24.1	26.5	-
CCS	30.7	29.1	33.1	30.9	29.1	29.0	25.4	30.4	29.6	29.9	+5.5	+3.4

(a) Case Study 1: Romance languages

	En-Fi wmt19		En-Et wmt18		Avg		Δ	
	→	←	→	←	→	←	→	←
	AA	15.6	19.3	20.5	23.3	18.05	21.3	-
CCS	21.2	21.2	25.6	25.7	23.4	23.45	+5.35	+2.15

(b) Case Study 2: Uralic languages

	En-Hi wmt19		En-Gu wmt18		Avg		Δ	
	→	←	→	←	→	←	→	←
	AA	28.4	24.6	10.2	11.5	19.3	18.05	-
CCS	28.0	24.0	12.9	12.9	20.45	18.45	+1.15	+0.4

(c) Case Study 3: Indo-Aryan languages

Table 2: Pair-wise spBLEU results for each conducted case study. → stands for En-X while ← means X-En. `Avg` indicates average spBLEU, and Δ signifies spBLEU improvements over AA.

	Romance			Uralic		Indo-Aryan	
	En-X	X-En	En-X	X-En	En-X	X-En	
<i>Monolingual data ratios</i>							
mBART50 (ft)	x91	x68	x90				
mRASP2	x7	x4	x4				
AA/CCS	x1	x1	x1				
<i>Parallel data ratios</i>							
mBART50 (ft)	x9	x0.5	x0.8				
mRASP2	x8	x1	x1				
AA/CCS	x1	x1	x1				
<i>Massively Multilingual Models</i>							
mBART50 (large, ft)	32.25	35.63	23.10	29.35	13.45	23.20	
mRASP2 (large)	36.00	37.13	25.20	27.00	5.75	15.10	
<i>Our Models</i>							
AA	24.08	26.45	18.05	21.30	19.30	18.05	
AA (large)	29.18	29.53	21.25	23.55	20.20	18.65	
CCS	29.58	29.85	23.40	23.45	20.45	18.45	
CCS (large)	31.30	31.10	27.60	27.25	23.30	22.00	
CCS (large, ft)	31.30	31.13	27.70	28.05	25.30	23.50	

(b) Results (large = 12e12d, ft = fine-tuned)

Table 3: Data and performance comparison with massively multilingual models. mBART50 (Tang et al., 2021) and mRASP2 (Pan et al., 2021) were taken and evaluated on the given pairs. Our Models were trained and tested only on languages from specific families (4 for Romance, 2 for Uralic and Indo-Aryan) on a much smaller dataset (Table 3a).

performs better, likely due to the wide gap in English monolingual and target-side data. However, the fine-tuned CCS model does close the gap with fine-tuned mBART50, performing comparably or better. We address the second purpose by noting that except for the low-resourced Indo-Aryan case, mRASP2 - which is essentially a scaled-up version of AA (large) trained on more languages - routinely improves over the latter. While `forgetting` low-resource languages can lead to some performance decline, performance, in general, can be observed to improve on scaling up - likely boosted by improved cross-lingual transfer. Now, when using comparable data, CCS (large) beats *both* AA (large) and mRASP2. This suggests that scaling up CCS to train a massive model like mRASP2 could reasonably be expected to yield improvements over the latter. We leave this exploration to future work.

Given CCS leverages massive models to pre-train small, high-performing ones, we also explore another interesting auxiliary application of CCS,

Knowledge Distillation (KD), in Section A.5.2, and show how it outperforms traditional KD baselines.

3.4.3 Alleviating Resource Dependencies

	Romance		Uralic		Indo-Aryan	
	En-X	X-En	En-X	X-En	En-X	X-En
CCS (<i>base</i> =ft. mBART50)	29.58	29.85	23.4	23.45	20.45	18.45
CCS (<i>base</i> =from-scratch)	30.05	29.40	23.1	22.95	20.25	19.55

Table 4: CCS with different `base` model choices. A model trained `from-scratch` can be used as a substitute for fine-tuned mBART50 with comparable performance.

Scaling beyond mBART50 While the above experiments use mBART50 as the `base` model, an important question to consider is how to scale to languages beyond the ones included in the same (or, more challengingly, any available massive model). We conduct an alternative set of experiments, where we train up small (6e6d) Transformer multilingual models ``from-scratch'' on our parallel data (Table 1b) and show in Table 4 that CCS baselines using these as `base` models consistently perform comparably to those using mBART50 as the `base`. This

suggests that although massive models are readily available and convenient to use, CCS performance is not dependent on their existence, and small models trained from scratch can be a good substitute.

Zero-Shot Translation: A follow-up question to the previous solution of training models from scratch on parallel data is the zero-shot scenario: i.e. what happens if there is no parallel data available, such as for low-resource languages or non-English centric pairs? Table 5 addresses this scenario. Starting from a random 'from-scratch' baseline trained on the Romance corpus in Table 1b (i.e. no En-Pt data provided), we observe that CCS baselines using the former as a 'base' model introduce large gains over the same and over AA. The latter is a particularly strong baseline since it uses ground-truth MUSE lexicons from Pt to every other Romance language for code-switching. However, the enhanced multilingual code-switching of CCS can potentially achieve superior alignment of the Pt vocabulary with that of the other languages -- making CCS a better alternative than AA even if parallel data is unavailable for a given pair.

	En-Pt	Pt-En
From-scratch	1.30	3.40
AA	3.20	10.30
CCS (<i>base</i> =from-scratch)	4.80	11.00

Table 5: Zero-shot Translation

3.5 Analysis

We now empirically discuss some key factors that enhance code-switched pretraining and hope these would serve as useful pointers for future work.

Impact of many-to-many substitutions Table 6 studies the impact of many-to-many substitutions through an ablation study, comparing against a CCS baseline where only 1-1 aligned words are chosen for substitution. We note a consistent decline in performance across all language pairs, with the largest being for Uralic (about 1.5-2 spBLEU points on average). This is likely due to agglutination in the Uralic family. For instance, 1 Finnish word *jauhelihakeitto* aligns to 3 English words: minced meat soup. This is correlated statistically - the

	Romance		Uralic		Indo-Aryan	
	En-X	X-En	En-X	X-En	X-En	En-X
CCS (1-1)	28.53	28.98	21.50	21.95	18.95	18.00
CCS	29.58	29.85	23.40	23.45	20.45	18.45

Table 6: Ablation study investigating the role of many-to-many substitutions

average word count per sentence in our Fi corpus is 10.66 while for En it is almost double (20.2). Many-to-many substitutions are, thus, crucial for code-switching such agglutinative languages.

Impact of contextual translations Through an example lifted from the Romance corpus, Figure 3 shows how CCS improves pretraining data quality through contextual substitutions. While *man* means *humano* (human) in certain contexts, the French word *homme* is correct here. A similar argument holds for *charge* and *gardes* (*garde* could mean either guard or custody, based on context).

Original:

45-year-old man has been remanded in custody on a firearms charge following a disturbance at a travellers' site on Monday when six people were arrested .

AA:

A 45-year-old humano tem been remanded in gardes sobre one arms débit following una necazuri at una travellers' site habilitado Monday when six people stavano arrested.

CCS:

A 45-anos-old homme ha stato reținut en custodia on a firearms charge urma a disturbión en un viajante 'site del manhá when six persone fueron arrestadas

	Original	✗ AA	✓ CCS
Errors due to non-contextual noising in AA:	man	humano	homme
	custody	gardes	custodia
	charge	débit	charge

Figure 3: CCS produces more contextual substitutions. In this example taken from the Romance corpus, the sentences codeswitch between En, It, Fr, Es, Pt, and Ro

Impact of code-switching language count Pan et al. (2021) use Multilingual Code-Switching (MLCS)⁵ to noise sentences, meaning a sentence could be code-switched in multiple languages. While we reproduce this in vanilla CCS and AA baselines, in Table 7 we explore Bilingual Code-Switching (BLCS), where 1 sentence switches between only 2 languages. Intuitively, this could be easier for the model to denoise. We observe our intuitions are mostly correct -- except for Romance, BLCS consistently improves performance. For the Romance languages, a likely explanation is that these have high lexical similarity (about 80-90% (Eberhard et al., 2022)) and more shared vocabulary, so the denoising task is not as complex and MLCS encourages greater transfer. In contrast, the lexical similarity for Uralic languages is lower than 50% (Jorgensen, 2020), while Hindi and Gujarati

⁵While this term is coined for ease-of-use in this work, it is a slight misnomer, and we discuss this in Appendix A.3

	Romance		Uralic		Indo-Aryan	
	En-X	X-En	En-X	X-En	En-X	X-En
CCS (MLCS)	29.6	29.9	23.4	23.45	20.45	18.45
CCS (BLCS)	28.2	28.58	23.9	23.7	21.45	20.65

Table 7: Comparison between MultiLingual (MLCS) and BiLingual Code-Switching (BLCS)

use different scripts -- leading to reduced vocabulary sharing and increased complexity. This might also explain why BLCS gives the Indo-Aryan family the highest gains, followed by Uralic. We note BLCS is also more time-efficient in Section 8.1, so we suggest future work to use the same when code-switching lexically dissimilar languages.

Impact of fine-tuning While the default variant of CCS follows Pan et al. (2021) and mixes parallel and monolingual code-switched data for pretraining, we explore if pretraining on only the latter and leaving the parallel data 'unnoised' for fine-tuning, might be a better alternative (as is common in other related works). Table 8 confirms this enhances performance significantly, likely due to monolingual data being abundant enough for achieving the cross-lingual transfer desired during pretraining, and fine-tuning more closely resembling the final translation task. Secondly, we note that Multilingual Fine-Tuning (MLFT) beats Bilingual Fine-Tuning (BLFT) in code-switched pretraining too, complementing the findings of Tang et al. (2021) in masked pretraining.

	Romance		Uralic		Indo-Aryan	
	En-X	X-En	En-X	X-En	X-En	En-X
CCS	29.58	29.85	23.40	23.45	20.45	18.45
CCS + BLFT	28.65	28.43	23.55	23.80	16.35	14.50
CCS + MLFT	30.00	29.68	25.20	25.85	23.55	22.35

Table 8: Improvements yielded by fine-tuning

4 Related Work

Denosing-based pretraining: Various noising mechanisms have been proposed for denosing-based pretraining of NMT models in recent times. Inspired by BERT (Devlin et al., 2019), earlier models including MASS (Song et al., 2019) and BART (Lewis et al., 2020) were pretrained on masked monolingual corpora, followed by fine-tuning on large parallel datasets (Tang et al., 2021). In an effort to shift the denosing objective from language modeling to translation, subsequent works adopted code-switched noising. ALM (Yang et al., 2020a) introduced this concept by using statistical phrase

tables to code-switch parallel datasets and training bilingual MT models that showed small improvements for the high-resourced, linguistically similar En-De and De-En pairs. Next, CSP (Yang et al., 2020b) proposed using probabilistic lexicons for code-switching in order to train bilingual models on both monolingual and parallel data. RAS (Lin et al., 2020) extended this trend to multilingual NMT, utilizing MUSE lexicons to code-switch and pretrain the massive NMT model mRASP on parallel corpora from 32 languages. Its successor, Aligned Augmentation (Pan et al., 2021) used a 'multilingual' lexicon (formed by heuristically chaining bilingual MUSE lexicons) to code-switch both monolingual and parallel corpora and pre-trained the mRASP2 model on these using contrastive learning. They reported SOTA scores, beating mRASP and many other strong baselines across a variety of language pairs and tasks. CeMAT (Li et al., 2022b) showed that BART-like masking can complement lexicon-based code-switching.

Different from all these works that only attempt one-to-one, non-contextual code-switching, the key novel contribution of our work is to carefully explore and analyze the performance gains offered by enhanced code-switching that factors context, many-to-many substitutions, code-switching language count, etc. We show how modern NMT models can be utilized to achieve these goals and achieve comparable or better performance while using a tiny fraction of the data and compute.

5 Conclusion

We explore a noising mechanism called Contextual Code-Switching (CCS) that extracts contextual, many-to-many word translations for code-switched pretraining in multilingual NMT. Our experiments, conducted on 3 different language families, show that CCS consistently beats the previous SOTA approach, Aligned Augmentation and also performs comparably or better than mBART50 and mRASP2, based on the quantity of training data provided. We analyse the impact of some major factors responsible for enhancing code-switched pretraining through examples and ablation studies. We hope the findings of this work will be useful to researchers studying NMT pretraining, as well as to academic and industry peers who may be looking for a way to fruitfully leverage massive NMT models, or conversely, to jump-start the training of even larger and better-performing ones.

6 Acknowledgements

This work was funded by UK Research and Innovation (UKRI) under the UK government’s Horizon Europe funding guarantee 10039436. The experiments in this work were conducted using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service⁶, Sulis Tier 2 HPC platform hosted by the Scientific Computing Research Technology Platform at the University of Warwick, and the Baskerville Tier 2 HPC service⁷ operated by Advanced Research Computing at the University of Birmingham. Finally, we would also like to acknowledge Guillem Ramirez for his help in selecting interesting examples for Figure 3.

7 Ethical Considerations

There has been significant concern recently over massive multilingual NLP models learning racial and gender biases during pretraining (Tan and Celis, 2019; Bender et al., 2021). A technique like CCS that leverages massive NMT models could be at risk of propagating any such biases present in the ‘base’ model. While such a limitation is not unique to CCS and could apply to any technique harnessing large models (such as Knowledge Distillation approaches), it is an important ethical concern since model biases that are propagated this way could be harder to detect and control - as compared to data biases. In such a situation, it could be worthwhile to invest effort into curating more ‘unbiased’ data, and then using models trained from scratch on this data as the base models for CCS (see Table 4) - giving a greater degree of control than massive models like mBART50 but potentially yielding comparable performance.

8 Limitations

We now discuss some limitations of our work and suggest some ways to mitigate them.

8.1 Cost

One of the advantages of AA is that it is relatively inexpensive to code-switch using lexicons. CCS, on the other hand, requires translating training data into multiple different languages followed by computing word alignments, which can be very expensive, particularly on scaling up. So, we suggest

⁶www.csd3.cam.ac.uk

⁷<https://www.baskerville.ac.uk/>

Algorithm	Time
CCS-MLCS (<i>base</i> =mBART50)	8h 37m
CCS-BLCS (<i>base</i> =mBART50)	4h 30m
CCS-MLCS (<i>base</i> =from-scratch)	4h 42m
CCS-BLCS (<i>base</i> =from-scratch)	2h 37m

Table 9: Total Preprocessing (base translation+word alignment+CCS) time costs for En-Fi 4M corpus, while using 1 GPU node of 3 A100 GPUs

some ways of reducing the cost, while potentially maintaining comparable performance. One effective way would be to use the BLCS variant, since it only needs one translation and one set of word alignments per sentence. Another way to reduce costs is to use smaller (6e6d) models trained from-scratch as a faster substitute for larger models like mBART50 (Table 4). The effectiveness of these techniques is shown in Table 9, which depicts the total preprocessing costs (for the entire pipeline in Figure 3) for code-switching a 4M En-Fi parallel corpus on a single GPU node (with 3 A100 GPUs). It is worth remembering that the word alignment costs are minimal here (about 30 minutes), so the costs are primarily due to generating translations (with a beam size of 5). We, thus, encourage using standard techniques to improve MT efficiency, like using lower beam size, shortlisting, quantization etc. to further reduce costs.

We will also release the code-switched corpora constructed in this research as part of the camera-ready version of this paper, to ensure greater reuse of the expenditure in our time and resources.

8.2 Resource Dependencies

The CCS models in our work function using mBART50 as the ‘base’ model and the word alignment model, awesome-align. Greater resource dependencies are, thus, another limitation of CCS and it is important to think of viable alternatives in case of non-availability of these. Awesome-align uses representations from mBERT (Devlin et al., 2019) so, it could scale to the languages the latter is pretrained on. For other languages, such as very low-resource pairs, it could be worth exploring low-resource word aligners (Asgari et al., 2020; Poerner et al., 2018) - though we leave the exploration of the same as part of future work. As for the ‘base’ model, we could use models trained from scratch as a viable alternative (see Table 4) and potentially obtain comparable performance. In case of non-availability of parallel data, this approach can scale

well to zero-shot translation when parallel corpora from related language pairs are available. (Table 5)

8.3 Low-Resource Scenarios

As we saw in Table 10, CCS appears to yield relatively lower gains for low-resource languages. While this does beat many SOTA models including mBART50 and mRASP2, further research is needed to adapt CCS better for data-scarce and low-resource scenarios in general. Based on related work, one useful solution could be to leverage data from high-resourced languages and families (eg. mixing Romance language data with Indo-Aryan languages) in a more multilingual and scaled-up iteration of our work. Another way would be to filter out low-quality translations from the 'base' model using its confidence scores, and only use the high-quality ones for code-switching. While we are unable to explore these within the scope of this work, they could make for interesting future directions.

References

- Roei Aharoni, Melvin Johnson, and Orhan Firat. 2019. [Massively multilingual neural machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874--3884, Minneapolis, Minnesota. Association for Computational Linguistics.
- Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. [Findings of the 2021 conference on machine translation \(WMT21\)](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1--88, Online. Association for Computational Linguistics.
- Ehsaneddin Asgari, Masoud Jalili Sabet, Philipp Dufter, Christopher Ringlstetter, and Hinrich Schütze. 2020. Subword sampling for low resource word alignment. *arXiv preprint arXiv:2012.11657*.
- Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. [Findings of the 2019 conference on machine translation \(WMT19\)](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1--61, Florence, Italy. Association for Computational Linguistics.
- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610--623.
- Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171--4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zi-Yi Dou and Graham Neubig. 2021. [Word alignment by fine-tuning embeddings on parallel corpora](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112--2128, Online. Association for Computational Linguistics.
- David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2022. *Ethnologue: Languages of the World*, 25 edition. SIL International. Online version: <http://www.ethnologue.com>.
- Dirk Geeraerts. 1993. Vagueness's puzzles, polysemy's vagaries.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. [The Flores-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation](#). *Transactions of the Association for Computational Linguistics*, 10:522--538.
- Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. [Distilling the knowledge in a neural network](#). In *NIPS Deep Learning and Representation Learning Workshop*.
- Paul Jorgensen. 2020. How similar are finnish and estonian? <https://langfocus.com/language-features/how-similar-finnish-and-estonian>. Accessed: 2022-10-18.

- Yova Kementchedjheva, Mareike Hartmann, and Anders Søgaard. 2019. [Lost in evaluation: Misleading benchmarks for bilingual dictionary induction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3336--3341, Hong Kong, China. Association for Computational Linguistics.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79--86.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66--71, Brussels, Belgium. Association for Computational Linguistics.
- Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. [Word translation without parallel data](#). In *International Conference on Learning Representations*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871--7880, Online. Association for Computational Linguistics.
- Pengfei Li, Liangyou Li, Meng Zhang, Minghao Wu, and Qun Liu. 2022a. [Universal conditional masked language pre-training for neural machine translation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6379--6391, Dublin, Ireland. Association for Computational Linguistics.
- Pengfei Li, Liangyou Li, Meng Zhang, Minghao Wu, and Qun Liu. 2022b. [Universal conditional masked language pre-training for neural machine translation](#).
- Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. [Pre-training multilingual neural machine translation by leveraging alignment information](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649--2663, Online. Association for Computational Linguistics.
- George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.
- Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. [Contrastive learning for many-to-many multilingual neural machine translation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244--258, Online. Association for Computational Linguistics.
- Nina Poerner, Masoud Jalili Sabet, Benjamin Roth, and Hinrich Schütze. 2018. [Aligning very small parallel corpora using cross-lingual word embeddings and a monogamy objective](#). *arXiv preprint arXiv:1811.00066*.
- Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612--618.
- Gowtham Ramesh, Sumanth Doddapaneni, Aravindh Bheemraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2022. [Samanantar: The largest publicly available parallel corpora collection for 11 indic languages](#). *Transactions of the Association for Computational Linguistics*, 10:145--162.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. [COMET: A neural framework for MT evaluation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685--2702, Online. Association for Computational Linguistics.
- Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. 2022. [Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning](#). *arXiv preprint arXiv:2201.03110*.
- Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksnė. 2014. Billions of parallel words for free: Building and using the eu bookshop corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1850--1855.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tiejun Liu. 2019. [Mass: Masked sequence to sequence pre-training for language generation](#). In *International Conference on Machine Learning*, pages 5926--5936.
- Yi Chern Tan and L Elisa Celis. 2019. [Assessing social and intersectional biases in contextualized word representations](#). *Advances in Neural Information Processing Systems*, 32.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. [Multilingual translation from denoising pre-training](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450--3466, Online. Association for Computational Linguistics.
- Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. [Facebook AI's](#)

WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 205--215, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Philip Williams and Barry Haddow. 2021. [The elit eca corpus](#).

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. [Multilingual machine translation systems from Microsoft for WMT21 shared task](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 446--455, Online. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Dongdong Zhang, ShuangZhi Wu, Zhoujun Li, and Ming Zhou. 2020a. [Alternating language modeling for cross-lingual pre-training](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9386--9393.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020b. [CSP:code-switching pre-training for neural machine translation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2624--2636, Online. Association for Computational Linguistics.

A Appendix

A.1 Examining the Code-Switching Errors in Figure 1

In this subsection, we dissect Figure 1 and detail the mentioned errors, along with their causes⁸. Pre-training a multilingual model like mRASP2 that uses non-English corpora requires code-switching in non English-centric directions as well. Since the MUSE dictionaries (like most lexicons) are largely available in English-centric directions, Pan et al. (2021) attempt to generate a multilingual dictionary by recursively linking the bilingual MUSE lexicons through a pivot language (English). In theory, this would allow dictionary entries from language X to language Y using only X-En and En-Y dictionaries. However, such recursive linking would propagate any existing quality issues even further. We illustrate this point using examples from Figure 1. The Estonian word *annetada* and

⁸We did not include this in the main work since this is auxiliary to our paper's focus on contextual and many-to-many substitutions

the Hebrew word *ויטוריה* are used as substitutions for don't and win respectively -- but they actually mean donate and Vittoria (a city in Spain), both of which have relatively less edit distance with don't and victory. A closer look at the Estonian⁹ and the Hebrew¹⁰ MUSE dictionaries, as well as the multilingual dictionary¹¹ constructed by Pan et al. (2021) confirms that this has been caused by the process of linking noisy bilingual dictionaries. The latter error, for instance, was caused by linking win to victoria (Portuguese), which was then aligned with Vittoria (Italian) and then Vittoria (English) - completely altering the meaning. A similar explanation can be drawn for how the English word *some* is incorrectly substituted with the English word *sometimes* (Figure 1b), despite there being no English-English dictionary.

A.2 The CCS algorithm

The pseudo-code of the CCS algorithm is shown in Algorithm 1, along with finer details we were unable to describe previously.

A.3 A discussion on the MLCS terminology

Multilingual Code-Switching (MLCS), as described in Section 2, is a misnomer. In the work of Pan et al. (2021), code-switching is carried out using a bilingual (English-centric) lexicon for the parallel corpora, and a multilingual dictionary for the monolingual corpora. Thus, they use MLCS in a monolingual corpus with the multilingual dictionary, but only Bilingual Code-Switching (BLCS) in a parallel corpus. They do not explain the reason for this choice. In our work, we attempt to shed some light on this and explore the efficacy of BLCS, which is far more efficient for CCS (refer Section 8.1) and also performs comparably or better (Tables 2 and 3). We use the term MLCS in our AA and CCS baselines, therefore, to contrast with BLCS and for ease of use. It is worth noting, however, that the parallel corpus is still bilingually code-switched in the MLCS baselines, following Pan et al. (2021).

A.4 Experimental Settings

A.4.1 Computational Infrastructure

Due to expiry and low availability of GPU hours, we are forced to conduct our experiments on 3 dif-

⁹<https://dl.fbaipublicfiles.com/arrival/dictionaries/et-en.txt>

¹⁰<https://dl.fbaipublicfiles.com/arrival/dictionaries/he-en.txt>

¹¹https://lf3-nlp-opensource.bytetos.com/obj/nlp-opensource/acl2021/mrasp2/synonym_dict_raw_dep3

Algorithm 1: Contextual Code-switching of a sentence using the CCS algorithm

Input : Sentence S ; translations $T_1, T_2 \dots T_n$; alignments $A_1, A_2 \dots A_n$

Output : Code-switched sentence $C_{CCS}(S)$

GenerateCCSCandidates (S, T, A)

```
     $Visited \leftarrow \emptyset$  // Keeps track of words that have already been aligned
     $Candidates \leftarrow \emptyset$ 
    foreach word  $w_i \in S$  do
        if  $w_i \in Visited$  then
            | continue
        end
         $SrcWords, TgtWords \leftarrow \{w_i\}, \emptyset$ 
         $PrevSrcWords, PrevTgtWords \leftarrow \emptyset, \emptyset$ 
        /* Generates many-to-many connected components of words */
        while true do
            /* Adds target words aligned to source words */
             $TgtWords \leftarrow TgtWords \cup A[w_j \forall w_j \in SrcWords]$ 
            if  $PrevSrcWords == SrcWords$  or  $PrevTgtWords == TgtWords$  then
                | continue // Convergence condition
            end
             $PrevSrcWords, PrevTgtWords \leftarrow SrcWords, TgtWords$ 
            /* Adds source words aligned to target words */
             $SrcWords \leftarrow SrcWords \cup A[w_j \forall w_j \in TgtWords]$ 
        end
         $Visited \leftarrow Visited + \{w_j \forall w_j \in SrcWords\}$ 
         $Candidates \leftarrow Candidates + (SrcWords, TgtWords)$ 
    end
    return  $Candidates$ 
```

$CCSCandidates \leftarrow GenerateCCSCandidates(S, T_i, A_i) \forall (T_i, A_i)$

$C_{CCS}(S), Swaps \leftarrow "", \emptyset$

$Visited \leftarrow \emptyset$ // Keeps track of words that have already been code-switched

while $|Visited|/|S| < ReplacementRatio$ **do**

```
    /* Randomly choose word(s) for substitution */
     $SrcWords, TgtWords = Random.Choice(CCSCandidates)$ 
    if  $\exists w_i \in SrcWords \{w_i \in Visited\}$  then
        | continue
    end
     $Swaps \leftarrow Swaps + (SrcWords, TgtWords)$ 
     $Visited \leftarrow Visited + \{w_j \forall w_j \in SrcWords\}$ 
```

end

$C_{CCS}(S) = S.Swap(SrcWords, TgtWords) \forall (SrcWords, TgtWords) \in Swaps$

return $C_{CCS}(S)$

ferent GPU clusters. To maintain comparability, however, we ensure that we use the same cluster for each case study - thus the training and evaluation of all baselines (be it CCS or AA), including the evaluation of the massive models, for a particular language family is always conducted on the same cluster. Additionally, we also ensure that the same parameters are used across all machines, and libraries with the same versions are installed.

Specifically, for the Romance family Case Study we use Skylake CPU nodes with the maximum memory of 300GB and Ampere GPU nodes with 1000GB RAM and 4 Nvidia A100 GPUs per node, running CentOS8. For the Uralic family, we use nodes with 2 AMD EPYC 7742 (Rome) CPUs and 512GB RAM, while for GPUs we use 3 Nvidia A100 GPUs with 40 GB RAM. For the Indo-Aryan family, we use Intel® Xeon® Platinum 8360Y CPU nodes with 512 GB RAM and GPU nodes with 4x NVIDIA A100 40GB GPUs.

A.4.2 Computational costs

Total preprocessing costs for the En-Fi 4M pair are given in Section 8.1 and, based on the technique used, can be roughly interpolated given our training corpora size (Table 1) to calculate total GPU hours. In practice, the real-world time was much lesser since we: a) used GPU clusters (as mentioned above) to simultaneously process multiple language pairs by submitting multiple jobs, and b) divided large corpora into smaller halves that were simultaneously preprocessed. For example, the 20M English monolingual corpus used in the Uralic family (Table ??) was divided into 2 halves of 10M that were submitted as part of 2 separate jobs.

For training, we only used 1 GPU node (1 Slurm job) per language family with 3 or 4 GPUs, depending on the cluster. Training costs for the Romance model took about 50 hours till convergence, 32 hours for Uralic and 61 hours for the Indo-Aryan models. The discrepancy in time is likely due to the fact that each of these experiments had to be run on separate clusters (as mentioned in A.4.1) with different architectures and different batch sizes, especially given each model took very similar number of steps until convergence (270K-290K updates in total).

A.4.3 Datasets used

All the datasets we use are publicly available, distributed freely with the CC0 license. For all case

studies, News Crawl (Akhbardeh et al., 2021) is chosen to make up the monolingual corpus. For parallel corpora, we use different sources for each language family. For the Romance family, we use the Europarl corpus (Koehn, 2005) as our parallel corpus. For Uralic, we use the WMT (Barrault et al., 2019), EUBookshop (Skadiņš et al., 2014) and the ELITR-ECA (Williams and Haddow, 2021) corpora. Finally, for Indo-Aryan, we use the Samanantar (Ramesh et al., 2022) corpus.

Except for Samanantar, all of these datasets belong to the news domain. While Samantar is a collection of datasets from various domains, given the test set belongs to the news domain, we sample news datasets from this corpus for inclusion in our training data.

A.5 Additional Results

A.5.1 Results on other metrics

We summarize the results of our key models in Table 10, using spBLEU, ChrF++ and COMET metrics. While we were unable to include the same in our main work, we observe that the metrics agree with each by and large and detail it in this section for completeness.

A.5.2 CCS for Knowledge Distillation

While the results in the main work mostly focused on the efficacy of CCS in its primary role as a pretraining mechanism, Table 11 indicates how it could also be effective as a better Knowledge Distillation (KD) technique, with minimal computational overload. We compare against the vanilla KD baseline (Hinton et al., 2015) that trains a small (6e6d) student model to mimic the teacher model (mBART50). CCS models of the same size routinely outperform KD, with the sole exception of X-En (Romance). It is interesting to note that it takes similar computational resources to preprocess and train CCS (BLCS), as it does for the KD baseline: given a translation generated by a teacher model, it appears it is better to use the translation to noise (code-switch) the source sentence and train it using the CCS mechanism, as opposed to using it as a target. The only overhead for CCS would be that of extracting word alignments, and in practice we find that it is relatively small - about 1/16th the time taken for translation generation.

	En-X			X-En		
	spBLEU	ChrF++	COMET	spBLEU	ChrF++	COMET
<i>Massively Multilingual Models</i>						
mBART50 (large, ft)	32.25	54.60	0.59	35.63	57.43	0.55
mRASP2 (large)	36.00	57.18	0.69	37.13	58.50	0.59
<i>Language family-specific baselines</i>						
AA (MLCS)	24.08	46.75	0.13	26.45	51.28	0.20
AA (MLCS) (large)	29.18	51.65	0.39	29.53	53.38	0.35
CCS (BLCS)	28.20	51.83	0.38	28.58	53.13	0.31
CCS (MLCS)	29.58	52.40	0.44	29.85	53.78	0.36
CCS (MLCS, large)	31.30	53.93	0.51	31.10	54.88	0.43
CCS (MLCS, large, ft)	31.30	53.90	0.51	31.13	54.53	0.43

(a) Case Study 1: Romance languages

	En-X			X-En		
	spBLEU	ChrF++	COMET	spBLEU	ChrF++	COMET
<i>Massively Multilingual Models</i>						
mBART50 (large, ft)	23.10	46.95	0.72	29.35	52.35	0.52
mRASP2 (large)	25.20	48.55	0.75	27.00	50.75	0.47
<i>Language family-specific baselines</i>						
AA (MLCS)	18.05	41.30	0.19	21.30	46.00	0.20
AA (MLCS) (large)	21.25	44.45	0.43	23.55	47.95	0.32
CCS (BLCS)	23.85	46.45	0.64	23.70	47.70	0.35
CCS (MLCS)	23.40	46.05	0.61	23.45	47.45	0.34
CCS (MLCS, large)	27.60	49.35	0.80	27.25	50.65	0.48
CCS (MLCS, large, ft)	27.70	49.70	0.77	28.05	51.70	0.51

(b) Case Study 2: Uralic languages

	En-X			X-En		
	spBLEU	ChrF++	COMET	spBLEU	ChrF++	COMET
<i>Massively Multilingual Models</i>						
mBART50 (large, ft)	13.45	25.00	-0.17	23.20	47.85	0.43
mRASP2 (large)	5.75	22.90	-0.99	15.10	35.60	-0.15
<i>Language family-specific baselines</i>						
AA (MLCS)	19.30	36.50	0.18	18.05	42.30	0.19
AA (MLCS) (large)	20.20	37.10	0.23	18.65	41.90	0.14
CCS (BLCS)	21.45	38.55	0.27	20.65	44.35	0.24
CCS (MLCS)	20.45	37.05	0.23	18.45	41.55	0.18
CCS (MLCS, large)	23.30	40.20	0.40	22.00	45.90	0.31
CCS (MLCS, large, ft)	25.30	42.00	0.54	23.50	47.90	0.38

(c) Case Study 3: Indo-Aryan languages

Table 10: Average spBLEU, ChrF++ and COMET scores for all 3 case studies

	Romance		Uralic		Indo-Aryan	
	En-X	X-En	En-X	X-En	X-En	En-X
mBART50 (ft)	32.25	35.63	23.10	29.35	13.45	23.20
KD	28.90	30.65	18.05	20.30	5.05	16.80
CCS (MLCS)	29.58	29.85	23.40	23.45	20.45	18.45
CCS (BLCS)	28.20	28.58	23.85	23.70	21.45	20.65

Table 11: CCS v/s Knowledge Distillation (KD)

XQA-DST: Multi-Domain and Multi-Lingual Dialogue State Tracking

Han Zhou^{1,*} Ignacio Iacobacci² Pasquale Minervini^{3,4}

¹University of Cambridge ²Huawei Noah’s Ark Lab, London

³University of Edinburgh ⁴University College London

hz416@cam.ac.uk ignacio.iacobacci@huawei.com p.minervini@ed.ac.uk

Abstract

Dialogue State Tracking (DST), a crucial component of task-oriented dialogue (ToD) systems, keeps track of all important information pertaining to dialogue history: filling slots with the most probable values throughout the conversation. Existing methods generally rely on a predefined set of values and struggle to generalise to previously unseen slots in new domains. To overcome these challenges, we propose a domain-agnostic extractive question answering (QA) approach with shared weights across domains. To disentangle the complex domain information in ToDs, we train our DST with a novel domain filtering strategy by excluding out-of-domain question samples. With an independent classifier that predicts the presence of multiple domains given the context, our model tackles DST by extracting spans in active domains. Empirical results demonstrate that our model can efficiently leverage domain-agnostic QA datasets by two-stage fine-tuning while being both domain-scalable and open-vocabulary in DST. It shows strong transferability by achieving zero-shot domain-adaptation results on MultiWOZ 2.1 with an average JGA of 36.7%. It further achieves cross-lingual transfer with state-of-the-art zero-shot results, 66.2% JGA from English to German and 75.7% JGA from English to Italian on WOZ 2.0.

1 Introduction

Task-oriented dialogue systems are designed to provide natural conversation with users and assist them in achieving daily goals. With the growth of task-oriented dialogue systems, there is an increasing interest in supporting dialogues among many domains and languages to fit the users’ demands. However, either modelling a multi-domain or multi-lingual dialogue system requires substantial data collected in real scenarios. This data acquisition

procedure is extremely expensive, and it motivates us to resolve this challenge by leveraging dialogue data in rich-resource domains and languages via zero-shot transfer learning.

Dialogue State Tracking (DST) is crucial for accurately extracting user intents and goals over multiple turns within the dialogue. Based on the tracked dialogue states, the dialogue manager makes corresponding next actions with back-end results, where the accuracy of the DST becomes absolutely vital. With a fully predefined ontology, traditional approaches tackle the DST as a classification problem by enumerating every combination of slot-value pairs (Mrkšić et al., 2017; Zhong et al., 2018). Those approaches are strongly limited by their scalability, as some slots (e.g. *name*) have an unbounded set of slot values. Secondly, they are generally not flexible to unseen slot-value pairs, making them more difficult to adapt to zero-shot transfer learning. Moreover, a completely predefined ontology is hard to acquire and not scalable for ToD systems in real applications.

To overcome those challenges, we take inspiration from Gao et al. (2020) and investigate how DST can be tackled by extracting slot values from user utterances directly. In this paper, we propose a domain-independent and transferable dialogue state tracker within an extractive question answering architecture. Our model is responsible for filling the slot value by recognising specially designed domain-slot prompts by span prediction, which extracts answers from the input utterance by predicting the token positions. In addition, we introduce a novel domain filtering strategy in training and an independent multi-domain classifier in evaluation such that we only ask slot questions that appear in predicted domains. For example, given *hotel* as the current turn domain, all questions under the *train* domain are filtered out as there is no overlap between them. This simple but effective filtering strategy significantly reduces the noise from unrec-

* Work done while at UCL.

Code is available at <https://github.com/hanzhou032/xqa-dst>

essary questions in both the training and evaluation phases. Furthermore, we study unexplored impacts of two-stage fine-tuning on DST transfer learning with mono-lingual and multi-lingual question answering datasets.

We call the final model XQA-DST: XLM-R based Dialogue State Tracker in Question Answering. Our main contributions are summarised below:

- We introduce XQA-DST, a novel domain-independent and transferable dialogue state tracker inspired by extractive question answering models. The model is able to recognise slot values by reformulating the task as an answer to a designed domain-slot question prompt by span prediction, which extracts answers from the input utterance by predicting the token positions.
- We enable XQA-DST on question answering by zero-shot domain adaptation scenarios, showing its transferability capabilities. The final model shows state-of-the-art domain adaptation performance with an average JGA of 36.7% for five domains on MultiWOZ 2.1.
- We show that our model is capable of both domain adaptation and cross-lingual transfer learning. We demonstrate its cross-lingual transferability by achieving state-of-the-art zero-shot results, 66.2% JGA from English to German and 75.7% JGA from English to Italian on WOZ 2.0.

2 Related Work

Traditional dialogue state tracking approaches mostly rely on a predefined ontology. Lee et al. (2019) implement a slot-utterance matching module that computes the similarity between the utterance and each slot-value pair. Lai et al. (2020) use BERT (Devlin et al., 2019) as the context encoder and generate the relevance score for every pair. Recently, Lin et al. (2021a) and Feng et al. (2022) include schema graph networks to utilise inter-slot relationships. However, their scalability is strongly limited by the availability of the predefined ontology and schema graphs.

To improve efficiency, span prediction methods have been proposed to tackle DST so that the slot can be filled by directly addressing values in the context. Heck et al. (2020) implement copy mechanisms, but they use independent span projection

layers for each slot, which make their model incapable of inference in new domains. Zhou and Small (2019) and Gao et al. (2020) formulate the DST as a question answering problem, and they prepare questions for asking the model to answer values for every slot. We differentiate from these approaches by disentangling the complex domain information from domain filtering and domain classification strategies.

Generative approaches (Wu et al., 2019; Kumar et al., 2020) provide an alternative way to handle DST. Li et al. (2021) introduce a generative question answering approach, GPT2-m, that leverages an autoregressive language model. Similarly, Lin et al. (2021b,c) propose T5DST, and they study the impacts of slot descriptions and cross-task transfer on domain adaptation. Lee et al. (2021) reformulate DST as prompting states via schema descriptions from language models. Recent end-to-end dialogue models (Peng et al., 2021; Su et al., 2022) also show strong supervised performance on DST.

Cross-lingual transfer learning for DST aims to leverage the labelled data in rich-resource languages and transfer learned knowledge to low-resource languages. Chen et al. (2018) study this problem and propose the XL-NBT teacher-student framework. Liu et al. (2020) introduce an Attention-informed Mixed-Language Training (AMLT) method to build code-switching training sentences. They study the effectiveness of multi-lingual pretrained language models, XLM (Conneau and Lample, 2019) and mBERT (Devlin et al., 2019), with their AMLT approach. Qin et al. (2020) propose a data augmentation framework, which encourages cross-lingual alignment by fine-tuning mBERT on generated code-switching data. Moghe et al. (2021) introduce intermediate fine-tuning on parallel sentences to improve the cross-lingual DST. To the best of our knowledge, we are the first work that studies the effectiveness of a multi-lingual pretrained language model, XLM-R (Conneau et al., 2020), on DST without implementing additional cross-lingual alignment strategies.

3 Multi-Domain and Multi-Lingual DST

To tackle the task of dialogue state tracking, our model reads the current user utterance U_t , preceding system utterance M_t , dialogue history H_t , and the domain-slot prompt Q_t as inputs for each turn. Followed by that, our model is responsible for firstly determining the dialogue domains D_t from

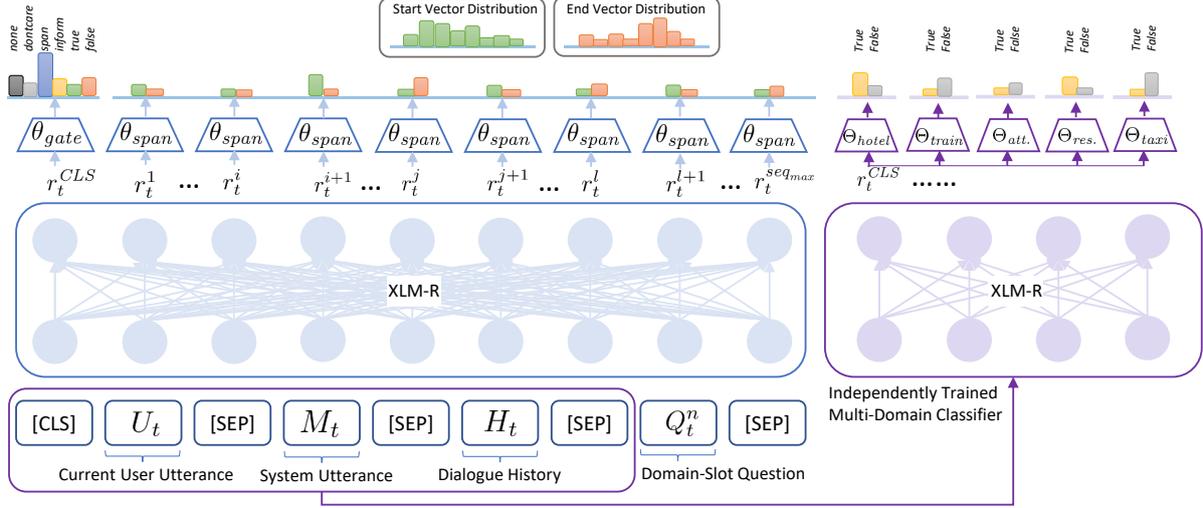


Figure 1: The model architecture of our XQA-DST for multi-domain and multi-lingual DST, where the right part is the independently trained multi-domain classifier that outputs active domains.

the input sequence. Then, it predicts the presence of the answer span in predicted domains given the question. If an answer is present in utterances, the model will predict the value for that domain-slot question using span extraction. Otherwise, its value will be filled in accordance with other predicted types. Finally, our model tracks the dialogue states by a rule-based update mechanism along with the progress of the dialogue across turns.

3.1 Context and Domain-Slot Questions

In extractive question answering, the context is used to provide the background information, and the answer is usually contained in the context. When it comes to DST, it is equivalent to modelling the system message and the user response together as the context for the current turn. The complete context C_t is then collected by concatenating the current user utterance U_t and the preceding system utterance M_t with dialogue history H_t at turn t . We implement XLM-R as the context encoder for the purpose of cross-lingual transfer learning.

Each context is paired with N questions, which iterate through every slot that we are interested in. We append the domain-slot prompt at the end of the context as an analogue question for each domain-slot pair. Hence, the model can learn to correlate different questions to the same context and provide corresponding answers to fill the slot. For the same context with n th question Q_t^n at turn t , the input

sequence S_t^n can be written as:

$$S_t^n = [\text{CLS}] \oplus U_t \oplus [\text{SEP}] \oplus M_t \oplus [\text{SEP}] \oplus H_t \oplus [\text{SEP}] \oplus Q_t^n \oplus [\text{SEP}], \quad (1)$$

where \oplus is the string concatenation, and H_t represents the dialogue history collected in a reversed order, and it is defined as follows:

$$H_t = U_{t-1} \oplus M_{t-1} \oplus \dots \oplus U_1 \oplus M_1 \text{ for } t > 1. \quad (2)$$

To utilise the question as a distinct feature for each slot, we propose the analogue question in the format of a domain-slot prompt. Here, additional special tokens are introduced to assist the model in recognising the domain-slot pair as distinct parts. Moreover, they provide clear signals for the start and end positions for each domain-slot pair. The equation for constructing the domain-slot prompt Q_t^n is defined below:

$$Q_t^n = \langle \text{dom.} \rangle \oplus d_t^n \oplus \langle / \text{dom.} \rangle \oplus \langle \text{slot} \rangle \oplus s_t^n \oplus \langle / \text{slot} \rangle, \quad (3)$$

where d_t^n refers to the name of the domain and s_t^n is the slot name for n -th question at turn t .

3.2 Shared Classification Gate

Our model contains a shared classification gate θ_{gate} for every domain-slot question as shown in Fig. 1. This shared gate provides shared knowledge among various domain-slot pairs, as it is neither domain-specific nor slot-specific.

For each input sentence S_t , this shared gate classifies it to one of six classes as described in three main categories. Special cases, *none/dontcare*, indicate that there is either no observable value from the input sequence S_t or any value that can become the answer for that slot question. Copy mechanism, *span*, indicates that the answer can be extracted from the current user utterance U_t by the span prediction module. Similarly, *Inform* is to copy from the system inform memory that tracks values mentioned in the preceding system utterance M_t . Boolean values *true/false* are used to deal with binary categorical values for Boolean slots where the value cannot be extracted from the utterance.

With these designed classes, it takes the output r_t^{CLS} from the encoder as its only input. It generates a probability distribution $p_t^{\text{gate}} \in \mathbb{R}^6$ over six classes as in the following equation:

$$p_t^{\text{gate}} = \text{softmax}(W_{\text{gate}} \cdot r_t^{\text{CLS}} + b_{\text{gate}}), \quad (4)$$

where W_{gate} represents the weights for our shared gate that is achieved by a linear classification layer, and b_{gate} is the corresponding bias term. The class is then determined by taking the maximal argument of $\text{argmax}(p_t^{\text{gate}})$.

3.3 Shared Span Prediction Layer

If the predicted class for the current input sequence S_t is *span*, the answer for that domain-slot question Q_t will be filled by predicting the start and end positions of the value from the input sequence. We implement the shared span prediction layer for every domain-slot question for the purpose of domain-adaptable design. This is achieved by constructing a linear layer that takes the entire token representations from r_t^1 to r_t^{seqmax} as inputs, and it generates two outputs with two parallel softmax layers for token positions, the start and end position distribution, p_t^{start} and p_t^{end} .

$$[p_t^{\text{start}}, p_t^{\text{end}}] = \text{softmax}(W_{\text{span}} \cdot r_t^i + b_{\text{span}}) \quad (5a)$$

$$\text{start}_t = \text{argmax}(p_t^{\text{start}}) \quad (5b)$$

$$\text{end}_t = \text{argmax}(p_t^{\text{end}}). \quad (5c)$$

The start and end positions of the predicted value are then determined by picking the largest probability from distributions p_t^{start} and p_t^{end} . Followed by that, we sequentially collect the tokens from the predicted start_t position to end_t position, treating any reversed sequence prediction as an empty value. We then detokenize them to form the final predicted value for that domain-slot question.

3.4 Turn-Domain Filtering

For a task-oriented dialogue, the user may shift the domain of conversation across turns so that a dialogue can have multiple domains. We introduce a novel turn-domain filtering strategy that puts a strict constraint and only allows the model to pay attention to currently active domains. Turn-domain filtering indicates that only the slots within the current domains D_t are used to prepare training features since slots are domain-specific. Hence, turn-domain filtering can reduce the potential noises introduced by unnecessary domains. Mathematically, this filtering strategy puts an additional constraint for slot domain d_t^m in Eq. 3:

$$d_t^m \in D_t. \quad (6)$$

3.5 Independent Multi-Domain Classifier

Turn-domain filtering allows the model to answer questions only within the interested domains. However, the domain information is no longer a given feature in the evaluation stage. Here, we propose a multi-domain sequence classifier as shown in Fig. 1. The input sequence is the complete dialogue context C_t without domain-slot questions. We then collect the entire sequence representation r_t^{CLS} by the context encoders as XLM-R(C_t). Followed by that, r_t^{CLS} is fed into $|D|$ softmax layers, thereby allowing a binary prediction that decides whether each domain d_t is present in the input context or not. Finally, we collect the domains that have been assigned to the ‘True’ class, which indicates the presence of that domain in the context.

$$p_t^d = \text{softmax}(W_{\text{MSC}}^d \cdot r_t^{\text{CLS}} + b_{\text{MSC}}^d) \quad (7a)$$

$$d_t = \text{argmax}(p_t^d) \quad (7b)$$

$$D_t = \{d_1, \dots, d_{|D|}\}. \quad (7c)$$

3.6 System Inform Memory and Update Rules

To further reduce the error of our span extractor, we have employed the same inform copy mechanism as Heck et al. (2020). This memory is a simple dictionary that records all values informed by the preceding system utterance M_t into a system inform memory $I_t = \{I_t^1, \dots, I_t^N\}$. Then, the value answer A_t^n for n th question Q_t^n asked at turn t can be predicted by the following copy mechanism, given that $\text{inform} = \text{argmax}(p_t^{\text{gate}})$:

$$A_t^n = I_t^n \text{ for } Q_t^n. \quad (8)$$

We implement a simple rule-based mechanism that is used to update dialogue states across turns

as same as [Chao and Lane \(2019\)](#). In each turn, if the model assigned class for the current input sequence S_t^n with Q_t^n is not *none*, the dialogue state will be updated by obtaining A_t^n from our value prediction modules. On the other hand, if the classification gate predicts that there is no value for S_t^n , the dialogue state will be kept unchanged.

3.7 Two-Stage Fine-Tuning

Our model is designed to be capable of not only DST tasks but also general question answering tasks. Therefore, the transfer learning ability of our base model can be enhanced by firstly fine-tuning it on mono-lingual and multi-lingual question answering datasets as the first-stage fine-tuning. Then, we initialise its weights on DST shared gates and further fine-tune the model on DST datasets as the second-stage fine-tuning. This two-stage fine-tuning strategy maximally brings domain-agnostic knowledge into the field of DST.

4 Experimental Setup

4.1 Dataset

The datasets that we carry out experiments on are WOZ 2.0 ([Wen et al., 2017](#)) and MultiWOZ 2.1 ([Eric et al., 2020](#)) for single-domain and multi-domain task-oriented dialogues, respectively. WOZ 2.0 is a restaurant reservation dataset, and it contains three slots: *area*, *food*, and *price range*. It provides the conversation in three languages: English, German, and Italian. MultiWOZ 2.1 contains multi-domain conversations for more than 10000 dialogues over seven domains. The dialogue domain can change across turns, thereby making MultiWOZ 2.1 the most challenging dataset for task-oriented dialogue systems. We exclude *hospital* and *police* domains with very few dialogues, and the remaining dataset contains five domains (*hotel*, *train*, *attraction*, *restaurant*, and *taxi*) with 30 domain-slot pairs in total. For domain adaptation experiments, we use an extractive QA dataset, SQuAD 2.0 ([Rajpurkar et al., 2018](#)), to provide the intermediate fine-tuning. In cross-lingual experiments, we further use the multilingual QA dataset, XQuAD ([Artetxe et al., 2020](#)), to study the effectiveness of multi-lingual intermediate fine-tuning.

4.2 Implementation Details

We employ the pretrained *XLM-RoBERTa-base* model from the Huggingface library of Transformers ([Wolf et al., 2020](#)), which consists of 12 hidden

layers of 768 units. We also employ the *BERT-base-uncased* model for ablation study and fine-tuned models on SQuAD 2.0 and XQuAD for adaptation experiments. For all implementations, we limit the maximal input sequence length to 180 tokens to save the cost while keeping a reasonable length for including dialogue history. We truncate from the earliest dialogue history when the input sequence length exceeds the limit. The training objective is to minimise the summations of individual loss functions for each module, where each loss is defined as the cross-entropy loss. The loss for each domain module in the multi-domain classifier is equally weighted, where the coefficient for each part of the joint loss of our main model is:

$$\mathcal{L}_{total} = 0.8 \cdot \mathcal{L}_{gate} + 0.2 \cdot \mathcal{L}_{span}. \quad (9)$$

During the training process, we implement the Adam optimiser ([Kingma and Ba, 2015](#)) with an initial learning rate of 10^{-5} . Then, we employ a linear scheduler with a warm-up proportion of 10% so that the learning rate will decay linearly until reaching zero after the warm-up steps. We put a dropout layer with a rate of 30% at the output of our context encoders. We use an early stopping strategy by monitoring the accuracy of the validation dataset until it stops increasing for at least 3 epochs. The batch size is fixed at 16. The multi-domain classifier is trained independently with the same experimental setting, and it is only involved in the evaluation stage. We report the mean of supervised DST and zero-shot experimental results for three runs with different random seeds.

5 Experimental Results

5.1 Zero-Shot Domain Adaptation

We rank our XQA-DST model with prior methods capable of zero-shot domain adaptation. The experiment is used to evaluate the transfer performance of models when tested with dialogues in a completely unseen domain. We train our model on the other four domains by excluding the target domains. We follow the experimental steps reported by [Kumar et al. \(2020\)](#). Since there is a single domain defined in the target domain, the domain classifier is not utilised here because the dialogue domain is given information. Table 1 shows a comparison of our XQA-DST model to baselines and recent approaches, where the JGA is defined as the ratio of dialogue turns that have been perfectly predicted over the number of turns for all dialogues. It is

Models	Type	Hotel	Train	Att.	Res.	Taxi	Avg.
MA-DST (Kumar et al., 2020)	G	16.3	22.8	22.5	13.6	59.3	26.9
SUMBT (Lee et al., 2019)	C	19.8	22.5	22.6	16.5	59.5	28.2
TRADE (Wu et al., 2019)	G	19.5	22.9	22.8	16.4	59.2	28.2
GPT2-m (Li et al., 2021)	G	24.4	29.1	31.3	26.2	59.6	34.1
T5DST* (Lin et al., 2021c)	G	21.2	35.4	33.1	21.7	64.6	35.2
TransferQA (Lin et al., 2021b)	G	22.7	36.7	31.3	26.3	61.9	35.8
XQA-DST w/o two-stage	S	22.9	37.0	24.0	25.7	62.2	34.4
XQA-DST w. SQuAD2	S	24.3	40.0	27.9	28.2	63.2	36.7

Table 1: The joint goal accuracy (%) of zero-shot domain adaptation experiments on each domain with recent models on MultiWOZ 2.1. The abbreviations for model types are: G: Generative; C: Classification; S: Span prediction. *Results from MultiWOZ 2.0 are reported by Lin et al. (2021c).

clear that our model has generated more accurate results than both MA-DST (Kumar et al., 2020) and SUMBT (Lee et al., 2019) baselines by at least 6.2% JGA on average in domain adaptation even without two-stage fine-tuning. SUMBT tracks the dialogue states by classifying every slot-value pair. Hence, it is a classification-based method, whereas our approach is mainly relying on the value filling by the span prediction module. It can be seen that our model has outperformed baselines by a significant (3-9%) margin in the *hotel*, *restaurant*, and *taxi* domains. This is because the classification-based method requires a predefined ontology for its enumeration of values, which inevitably makes it not robust to unseen values in new domains and results in relatively low performance for domain adaptation.

There is another class of methods that utilises generative value filling to handle the DST, including TRADE, GPT2-m, and TransferQA. Given GPT2-m as an example, it is in the framework of generative question answering, which also coincides with the underlying idea of our XQA-DST model but has a decoder to generate candidate values. With the two-stage fine-tuning strategy on the SQuAD 2.0 dataset, our model shows improvements in all domains of 2.3% on average. It shows the highest JGA in both *train* and *restaurant* domains (40.0% and 28.2%, respectively). It also outperforms the TransferQA approach that implements the cross-task transfer learning, which is similar to our two-stage fine-tuning that includes multi-task knowledge. Our results appear as the state-of-the-art results at 36.7% JGA on average for zero-shot domain adaptation experiments.

Furthermore, our approach is designed to be applicable for both domain adaptation and cross-lingual transfer learning, whereas all generative

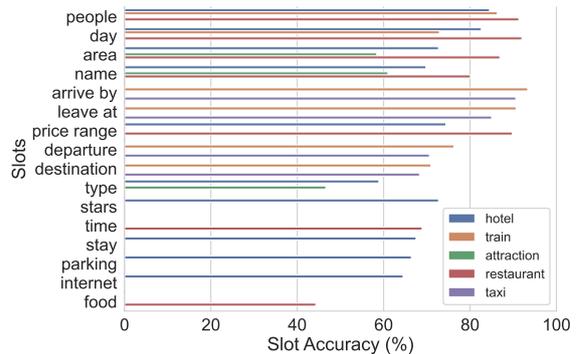


Figure 2: The categorical plot of slot accuracy (%) for each slot over 5 domains for the zero-shot domain adaptation experiment by XQA-DST.

methods listed above can only do mono-lingual learning. Therefore, our XQA-DST model has shown very competitive results in the zero-shot domain adaptation, and we can conclude that it is able to effectively generalise to task-oriented dialogues in new domains by understanding the linguistics behind our domain-slot questions.

5.2 Domain Adaptation Analysis

We analyse the individual slot accuracy for every domain-slot pair in 5 domains to study the impact of shared slots over domains on the performance of domain adaptation. The results are obtained by computing the slot accuracy on each target domain by XQA-DST. The slot accuracy is defined as the ratio of dialogue turns where the value for that slot is correctly predicted. Fig. 2 shows the slot accuracy for 16 slots over 5 domains, where multiple domain bars for the same slot indicate that the slot is shared across these domains.

It is observable that slots that have been shared among multiple domains lead to a relatively higher domain adaptation performance. By contrast, it is

Models	Joint Goal Accuracy (%)	
	GE	IT
XLM-R-DST	20.78	25.39
XL-NBT	30.80	41.20
MUSE + AMLT	36.51	39.35
XLM+CLCSA	48.70	-
mBERT+CLCSA	63.20	61.30
TLM+CLCSA	65.80	66.90
Ours <i>w/o</i> two-stage	64.88	68.63
<i>w.</i> XQuAD	66.16	72.84
<i>w.</i> SQuAD2	66.12	75.66

Table 2: The zero-shot cross-lingual DST results for target languages, German (GE) and Italian (IT), on WOZ 2.0. There are no results on Italian by XLM due to the absence of Italian in its pretraining (Liu et al., 2020).

also distinctive that slots that have not been shared among multiple domains have much lower accuracy. For instance, most slots in the *hotel* domain are not shared with other domains, so the slot accuracy for ‘*parking*’ and ‘*internet*’ slots (66.4% and 64.5%, respectively) are reasonably lower than others. The same rule applies to the ‘*time*’ and ‘*food*’ slots in the *restaurant* domain. Therefore, the number of shared domains for the slot is the foremost factor in achieving a good domain adaptation result. Secondly, we notice that slots with digital values such as ‘*people*’ and ‘*day*’ have very high slot accuracy (91.3% and 92.0% in the *restaurant* domain) even in the zero-shot setting. It validates the effectiveness of our model to domain adaptation for successfully extracting candidate values from the message. Last but not least, due to the wide surface form of location values, it is naturally hard to predict location slots, ‘*departure*’ and ‘*destination*’, that are not categorical with unseen values. Hence, even though they are shared in both *train* and *taxi* domains, they give relatively lower slot accuracy in the set of shared slots. Overall speaking, our XQA-DST model has generated reasonably well domain adaptation results on most domain-slot pairs and has shown a certain level of common knowledge across domains.

5.3 Zero-Shot Cross-Lingual DST

The zero-shot cross-lingual transfer learning is to train our XQA-DST on the source language, English. Then, it is sequentially evaluated on the test sets in German and Italian with labels that are kept in English. Since WOZ 2.0 is a single domain dataset with relatively short dialogues, the dialogue

history is not included as inputs, and the domain classifier is deactivated. To provide a fair comparison to the ground truth, we implement Google Translator (Wu et al., 2016) to translate the values filled by span prediction in the target language back to the source language.

In Table 2, our XQA-DST model with two-stage fine-tuning gives strong zero-shot results in both German and Italian languages (66.2% and 75.7% JGA, respectively). In comparison to recent approaches for cross-lingual DST, our XQA-DST model has generated results that significantly increase the margin by an absolute 8% on Italian. It is worth noting that both XLM+CLCSA and mBERT+CLCSA (Qin et al., 2020) are data augmentation-based approaches on multi-lingual models with the same model architecture as XL-NBT (Chen et al., 2018). TLM+CLCSA (Moghe et al., 2021) also implements two-stage fine-tuning with data augmentation. Even without two-stage fine-tuning, our model in extractive QA still outperforms most of them and appears as the state-of-the-art results in the zero-shot cross-lingual transfer learning on WOZ 2.0.

Besides the above approaches, we include XLM-R-DST as a baseline that we replace the context encoder of BERT-DST (Lai et al., 2020) with XLM-R. Then, we can study the effectiveness of different model architectures in cross-lingual transfer learning. We recall that XLM-R-DST fills the slot values by iterating through every candidate slot value with a relevance scorer. Table 2 shows a huge improvement in our approach by increasing the average JGA on target domains from 23.1% to 66.8% by more than 40%. It indicates that our specially designed extractive QA framework has a strong generalisation ability across languages, whereas the XLM-R-DST appears as only recognising each value as distinct features without understanding the deep semantics behind them. Lastly, we notice that the cross-lingual result on Italian has a higher joint goal accuracy than German in our experiments. We suppose that this is because of the declension in German, which leads to more diverse word forms with the same semantics and introduces noises to the translation process.

5.4 Supervised DST

We perform experiments on the supervised DST configuration and compare our XQA-DST model with prior methods capable of monolingual zero-

Models tested on MultiWOZ 2.1	JGA (%)
TRADE (Wu et al., 2019)	45.60
SUBMT (Lee et al., 2019)	46.70
STARC (Gao et al., 2020)	49.48
MA-DST (Kumar et al., 2020)	51.88
T5DST (Lin et al., 2021c)	52.21
GPT2-m (Li et al., 2021)	52.58
SDP-DST (Lee et al., 2021)	56.66
SOLOIST (Peng et al., 2021)	56.85
PPTOD (Su et al., 2022)	57.10
XQA-DST (our work)	53.21

Table 3: The performance of Supervised DST for our proposed XQA-DST model with prior methods capable of zero-shot inference on MultiWOZ 2.1.

shot domain adaptation on MultiWOZ 2.1. Table 3 comprises the JGA for each method, and we implement the same label mapping as TripPy (Heck et al., 2020) for a fair evaluation. In Table 3, our approach has outperformed most prior methods capable of zero-shot generalisation, including many generative approaches such as TRADE, T5DST, and GPT2-m. Though it is less competitive than the prompt-based SDP-DST model and end-to-end models in the supervised DST setting, its language transferability is still distinctive.

Based on the shared span prediction module, our model is able to extract values from the dialogue context directly, thereby being open-vocabulary and domain scalable. At the same time, it has successfully overcome the challenge of an unavailable ontology set. Moreover, it presents as the best-performed model in any framework with span prediction modules, where it has improved the margin of JGA by more than 3.5% from the STARC approach. None of the other approaches has ever studied their DST with multi-lingual pretrained models. By utilising the pretrained XLM-R model as the context encoder, our approach is the only method with cross-lingual transferability. Given its distinct advantages of being domain-adaptable and language transferable, a promising result in multi-domain DST at 53.2% is still competitive in the supervised setting.

To study the impact of essential designs in our model, we first analyse the performance of a mono-lingual model, BERT, and ablate it over different choices of domain classifiers. In Table 4, the vanilla model with undersampling of negative samples has the lowest JGA at 38.2%. This is because the shared span prediction layer lacks domain knowl-

Ablation	JGA (%)
BERT-base	
w. undersampling	38.23
w. joint domain classifier	41.10
w. independent domain classifier	49.04
+ dialogue history	51.11
XLM-RoBERTa-base	
w. independent domain classifier	51.67
+ dialogue history	53.21

Table 4: Ablation study of XQA-DST with different base models and training strategies on MultiWOZ 2.1.

edge and frequently generates false positive predictions for out-of-domain questions. Introducing a joint domain classifier at the output of the main model in parallel with θ_{gate} improves the JGA by about 3%, which convinces us about the effectiveness of domain classifiers. At the cost of the model size, the independent domain classifier significantly improves the JGA to 49.0% by removing the interference from asking out-of-domain questions. It encourages the model to learn to distinguish in-domain questions rather than additionally learning the relationship between the context and domains within a goal. We notice that implementing XLM-R instead of BERT further improves the performance to 51.7%. We suppose it is because of the well-trained RoBERTa model, and the multi-lingual pretraining does not greatly sacrifice the per-language performance. Lastly, due to the complexity of MultiWOZ dialogues, the history information is essential in accurately predicting current domains and extracting spans. Hence, appending the dialogue history has led our model to outperform most prior methods capable of zero-shot inference.

6 Conclusion

We introduce a new multi-domain and multi-lingual dialogue state tracker, XQA-DST, within an extractive question answering framework. It gives distinct advantages for avoiding relying on any predefined ontology and being open-vocabulary to new slots with unseen values. We have shown a strong domain and cross-lingual transferable ability of our model by outperforming famous baselines. We have demonstrated its competitive performance in multi-domain DST with a novel turn-domain filtering strategy and a multi-domain classifier in parallel. With the design of an XLM-R based multi-domain classifier, our approach is feasible for track-

ing states in multi-domain and multi-lingual scenarios. Therefore, it holds a strong potential to overcome the challenging data scarcity problem for either domains or languages in the real application of task-oriented dialogue systems.

Limitations

In the supervised DST experiments, our multi-domain classifier is effective when the range of domains is given. However, we have fixed weights for each domain projection layer, which inevitably makes the classifier not domain scalable. Though the shared span prediction layer is still scalable to all domains, the performance of our model will degrade if it encounters a dialogue in multiple unseen domains.

We recall that the independent multi-domain classifier provides a clearer training objective and significantly improves the JGA than the joint domain classifier. However, this is at the cost of model size and requires expensive computation resources. Therefore, we look forward to approaches that wisely incorporate the domain classifier.

In the cross-lingual experiments, we test the transfer performance for German and Italian, which have been used as the pretraining languages for XLM-R. Hence, we expect a degradation of cross-lingual performance for our model on low-resource languages that are not pretrained by XLM-R. In addition, our experiments rely on a back-translation from the target language to the source language. Though we have implemented a predefined label dictionary that collects vocabulary with similar semantics, it cannot perfectly handle the noise from an external translation system.

Acknowledgements Pasquale was partially funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 875160, ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence) EPSRC (grant no. EP/W002876/1), an industry grant from Cisco, and a donation from Accenture LLP.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. [On the cross-lingual transferability of monolingual representations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Guan-Lin Chao and Ian R. Lane. 2019. [BERT-DST: scalable end-to-end dialogue state tracking with bidirectional encoder representations from transformer](#). In *INTERSPEECH*, pages 1468–1472. ISCA.

Wenhu Chen, Jianshu Chen, Yu Su, Xin Wang, Dong Yu, Xifeng Yan, and William Yang Wang. 2018. [XL-NBT: A cross-lingual neural belief tracking framework](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 414–424, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. [Cross-lingual language model pretraining](#). In *NeurIPS*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. [MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Yue Feng, Aldo Lipani, Fanghua Ye, Qiang Zhang, and Emine Yilmaz. 2022. [Dynamic schema graph fusion network for multi-domain dialogue state tracking](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 115–126, Dublin, Ireland. Association for Computational Linguistics.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. 2020. [From machine reading comprehension to dialogue state tracking: Bridging the gap](#). In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89, Online. Association for Computational Linguistics.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. [TripPy: A triple copy strategy for value independent neural dialog state tracking](#). In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*,

- pages 35–44, 1st virtual meeting. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *ICLR (Poster)*.
- Adarsh Kumar, Peter Ku, Anuj Kumar Goyal, Angeliki Metallinou, and Dilek Hakkani-Tür. 2020. [MA-DST: multi-attention-based scalable dialog state tracking](#). In *AAAI*, pages 8107–8114. AAAI Press.
- Tuan Manh Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. 2020. [A simple but effective bert model for dialog state tracking on resource-limited systems](#). In *ICASSP*, pages 8034–8038. IEEE.
- Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. [Dialogue state tracking with a language model using schema-driven prompting](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. [SUMBT: slot-utterance matching for universal and scalable belief tracking](#). In *ACL (1)*, pages 5478–5483. Association for Computational Linguistics.
- Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021. [Zero-shot generalization in dialog state tracking through generative question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1063–1074, Online. Association for Computational Linguistics.
- Weizhe Lin, Bo-Hsiang Tseng, and Bill Byrne. 2021a. [Knowledge-aware graph-enhanced GPT-2 for dialogue state tracking](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7871–7881, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Zhenpeng Zhou, Paul Crook, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. 2021b. [Zero-shot dialogue state tracking via cross-task transfer](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7890–7900, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021c. [Leveraging slot descriptions for zero-shot cross-domain dialogue StateTracking](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5640–5648, Online. Association for Computational Linguistics.
- Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. [Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems](#). In *AAAI*, pages 8433–8440. AAAI Press.
- Nikita Moghe, Mark Steedman, and Alexandra Birch. 2021. [Cross-lingual intermediate fine-tuning improves dialogue state tracking](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1137–1150, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. [Neural belief tracker: Data-driven dialogue state tracking](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayan-deh, Lars Liden, and Jianfeng Gao. 2021. [Soloist: Building task bots at scale with transfer learning and machine teaching](#). *Transactions of the Association for Computational Linguistics*, 9:807–824.
- Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. [Cosda-ml: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP](#). In *IJCAI*, pages 3853–3860. ijcai.org.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. [Know what you don’t know: Unanswerable questions for SQuAD](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.
- Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. [Multi-task pre-training for plug-and-play task-oriented dialogue system](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. [A network-based end-to-end trainable task-oriented dialogue system](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame,

Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. [Transferable multi-domain state generator for task-oriented dialogue systems](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. [Google’s neural machine translation system: Bridging the gap between human and machine translation](#). *CoRR*, abs/1609.08144.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. [Global-locally self-attentive encoder for dialogue state tracking](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.

Li Zhou and Kevin Small. 2019. [Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering](#). In *The third Conversational AI workshop @ NeurIPS 2019, Vancouver, BC, Canada, 13 December 2019*.

A Reproducibility Details

Training details We use both *XLM-RoBERTa-base* (125M) and *BERT-base-uncased* (110M) with pretrained weights from the Huggingface library of Transformers. We run all experiments on a single RTX 3080Ti with 12 GB memory. We fix the batch size at 16 for all models during training and use the batch size at 1 for evaluations. During training, it takes about 40 minutes to run an epoch on MultiWOZ 2.1, and its inference time for all evaluation examples is about 7 minutes. For the WOZ 2.0 dataset, it takes roughly 20 minutes to train the model. In the cross-lingual setting, the inference time is about 10 minutes due to the back-translation procedure.

Hyperparameters For two-stage fine-tuning experiments, we implement QA fine-tuned models

from the Huggingface library of Transformers without tuning their hyperparameters. For the XQuAD experiment, it implements the batch size at 40 and a learning rate of 3×10^{-5} for the first-stage fine-tuning. For the SQuAD 2.0 experiment, we use the fine-tuned weights and hyperparameters from *deepset/xlm-roberta-base-squad2*.

Dataset details For the supervised DST experiments, we split the datasets into train/dev/test sets as same as Heck et al. (2020). In domain adaptation experiments, the MultiWOZ 2.1 datasets are divided into 5 domains in accordance with Lin et al. (2021c), where the *hospital* and *police* domains are excluded. Lastly, the multi-lingual WOZ 2.0 datasets have the same split as Moghe et al. (2021).

Improving Prediction Backward-Compatibility in NLP Model Upgrade with Gated Fusion

Yi-An Lai[♣] Elman Mansimov[♣] Yuqing Xie^{♡,*} Yi Zhang[♣]

[♣]AWS AI Labs [♡]University of Waterloo

{yianl,mansimov,yizhngn}@amazon.com

yuqing.xie@uwaterloo.ca

Abstract

When upgrading neural models to a newer version, new errors that were not encountered in the legacy version can be introduced, known as *regression*¹ errors. This inconsistent behavior during model upgrade often outweighs the benefits of accuracy gain and hinders the adoption of new models. To mitigate regression errors from model upgrade, distillation and ensemble have proven to be viable solutions without significant compromise in performance. Despite the progress, these approaches attained an incremental reduction in regression which is still far from achieving backward-compatible model upgrade. In this work, we propose a novel method, *Gated Fusion*, that promotes backward compatibility via learning to mix predictions between old and new models. Empirical results on two distinct model upgrade scenarios show that our method reduces the number of regression errors by 62% on average, outperforming the strongest baseline by an average of 25%.

1 Introduction

In order to achieve a smooth continuous improvement of NLP applications, it is critical to guarantee consistent operation of the system after an upgrade. New errors introduced during the model upgrade interfere with the existing user experience and are considered to be a *regression* in the quality. Due to the difficulty of modularizing or explaining the behavior of deep neural networks, traditional software regression tests are inapplicable to neural based systems. The cost of arduous error analysis and model patching often exceeds the benefits of model upgrades. Developing methods that ensure backward compatibility during model upgrades without compromise in performance becomes a valuable research direction (Yan et al., 2021; Xie et al., 2021; Träuble et al., 2021; Cai et al., 2022).

*Work done during author’s internship at AWS AI Labs.

¹Within this work, *regression* denotes performance degradation in software systems, instead of the statistical technique for estimating relationships among variables.

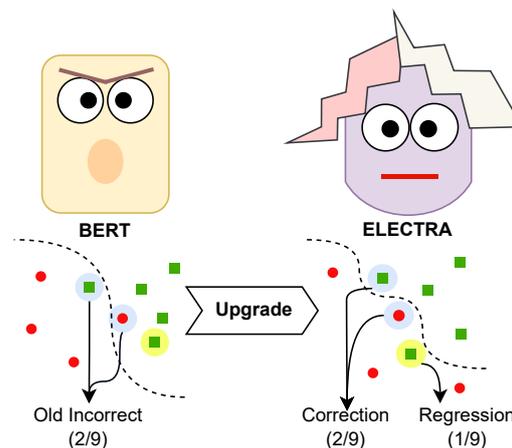


Figure 1: Illustration of regression errors when upgrading from BERT (Devlin et al., 2019) to ELECTRA (Clark et al., 2020) for classification. Red circles and green squares denote examples of different classes. Dashed lines represent decision boundaries.

The *prediction backward-compatible model upgrade* problem aims to improve consistency of correct classification predictions between legacy and upgraded models without accuracy loss. Yan et al. (2021) first studied backward compatibility during model upgrade on image classification tasks. They proposed to enforce the positive congruence of the new model with the old one by applying a knowledge distillation objective (Hinton et al., 2015) objective with re-weighting of training samples. Later, Xie et al. (2021) extended the work of Yan et al. (2021) by investigating the backward compatibility in NLP classification tasks. They found that their proposed distillation-based approach can help decrease the regression errors of specific linguistic phenomena in NLP classification tasks.

Despite progress with both distillation- and ensemble-based regression-mitigation approaches, there are limitations that prevent them from broad practical adoption in ML operations. Distillation-based methods attempt to transfer the prediction power of the old model to the new one on potential

regression instances (Hinton et al., 2015). However, given the huge complexity of current neural architectures and relatively scarce training data in downstream tasks, models could have insufficient data to reliably estimate the probable regression cases and carry out the transfer on them (Xie et al., 2021; Cai et al., 2022). On the other hand, model ensemble aggregates predictions from differently-trained new models but bears no connection with the legacy version (Yan et al., 2021). These limitations reveal the two major challenges when striving to ensure backward compatibility. First, the new model could have distinct inductive bias and prediction behavior than the old system, rooted from inherent differences such as architecture, model size, and pretraining procedure (Liu et al., 2021). Second, during new model training, a reliable mechanism is needed in place to bridge the gap between two models and mitigate potential inconsistencies.

Inspired by the strength and weakness of prior approaches, we propose *Gated Fusion* to integrate old and new models via gating mechanism (Hochreiter and Schmidhuber, 1997; Chung et al., 2014; Gu et al., 2016), essentially a light-weight ensemble of legacy and upgrade models connected via a learned fusion gate. Specifically, we add a *learned* gate on top of the new model. We combine the logits from old and new models according to the weight from the gate. We train our Gated Fusion model by minimizing the standard cross-entropy error. The intuition is that the gate could learn to put more weights on the old model when the new model cannot produce correct predictions, effectively doing fall-backs that optimizes backward compatibility.

Empirical results demonstrate that our proposed approach outperforms other competing methods significantly, where we can obtain on average 62% reduction of total negative flips, i.e. new errors caused by the model upgrade, without any degradation in accuracy performance. The effectiveness of Gated Fusion is validated across three diverse classification tasks and two distinct model upgrade scenarios (a) upgrade to larger model size (b) upgrade to advanced pretrained model, where consistent results are attained across the board.

Our main contributions are as follows:

- We propose Gated Fusion that integrates old and new models via gating mechanism for backward-compatible model upgrade;
- We evaluate competing methods on two distinct and challenging model upgrade scenarios

across three diverse classification tasks;

- Empirical results show that our proposed approach significantly outperforms competing methods and achieves regression reductions by a large margin across the board.

2 The Backward-Compatible Model Upgrade Problem

The goal of backward-compatible model upgrade is to minimize regression errors without compromising the accuracy performance during model upgrade (Yan et al., 2021; Xie et al., 2021). In this work, we aim to improve the backward compatibility of model predictions in the NLP classification tasks. Following Xie et al. (2021), we study the scenario where the underlying pretrained language model (LM) is being upgraded.

Let x be a natural language input with a class label $y \in \{1, 2, \dots, C\}$. $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ denotes a set of N examples with corresponding labels. A classifier f estimates the class probabilities given the input $\vec{f}(x) = (p(y=1|x), \dots, p(y=C|x))^\top$. When upgrading from an *old* model f_{old} to a *new* model f_{new} , normally with distinct architectures and trained on the same data, an improved model f^* is produced based on f_{old} and f_{new} . Our goal is for f^* to minimize *regression errors* as an additional objective, while still achieving comparable performance to f_{new}^o , the new model trained in the vanilla setting. Note that f^* could be multiple times larger than f_{new}^o , with model ensemble of f_{new}^o as one example (Yan et al., 2021).

Measuring Backward Compatibility. The backward compatibility is measured via quantifying regression errors on a given regression measurement set $\mathcal{D}_{reg} = \{x_i, y_i\}_{i=1}^M$. \mathcal{D}_{reg} could be a hidden customer test set comprising critical use cases, a set of behavioral testing examples for targeted evaluation (Ribeiro et al., 2020), or the development split from the dataset of interest. In this work, we take the development set as our \mathcal{D}_{reg} for evaluation.

For classification, regression errors are characterized by *negative flips*, denoted as \mathcal{R}_{NF} – the portion of samples in \mathcal{D}_{reg} that flip from correct prediction $f_{old}(x_i) = y_i$ to incorrect output $f_{new}(x_i) \neq y_i$ during model upgrade:

$$\mathcal{R}_{NF}(\mathcal{D}_{reg}, \vec{f}_{old}, \vec{f}_{new}) = \frac{|\{x | f_{old} = y, f_{new} \neq y\}|}{|\mathcal{D}_{reg}|}. \quad (1)$$

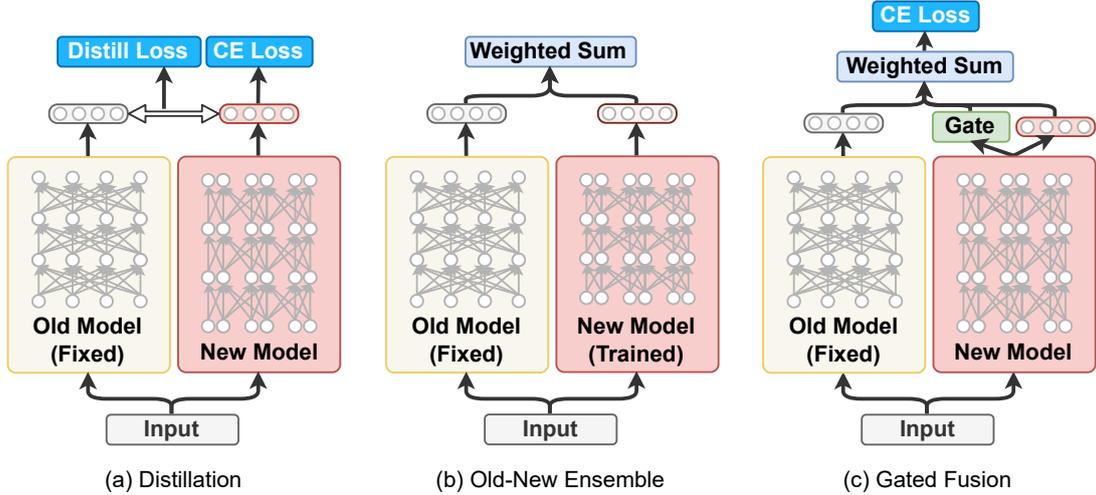


Figure 2: Methods to improve prediction backward compatibility during model upgrade. (a) Distillation-based approach to align predicted logits on potential regression instances (Xie et al., 2021). (b) Ensemble of old and new models via weighted sum of either predicted logits or probabilities. (c) Our proposed Gated Fusion that learns a gate as a soft switch to dynamically determine whether to fall back to previous predictions.

One thing to emphasize is that maximizing classifier performance does not necessarily help in minimizing \mathcal{R}_{NF} (Yan et al., 2021; Xie et al., 2021).

3 Gated Fusion: Methodology

3.1 Method Overview

To improve backward compatibility in model upgrade, it’s crucial to have a mechanism that detects potential regression errors and mitigates them when making predictions. We propose Gated Fusion (GF) to achieve this by learning a gate as a soft switch to choose between generating predictions by the new model or resorting to outputs of the old model. Gated Fusion is inspired by the gating mechanism widely used in other applications. For example, mixing word copying mode with word generation mode for language modeling (Merity et al., 2016) and summarization (See et al., 2017).

Our proposed Gated Fusion f_{GF}^* consists of the old model f_{old} , the new model f_{new} , and a gating network g_θ . The old model f_{old} is the legacy version before upgrade where the parameters are fixed. The new model f_{new} has the same architecture as f_{new}^o and is randomly initialized. The gating network g_θ is a multi-layer feed-forward network with sigmoid function. It produces a scalar weight α_{gate} in the range $[0, 1]$ from the output layer of f_{new} , denoted as \mathcal{E}_{new} :

$$\alpha_{gate}(x) = g_\theta(\mathcal{E}_{new}(x)). \quad (2)$$

We use α_{gate} to combine the logits of old and new

models as our final outputs:

$$l_{GF}^*(y|x) = (1 - \alpha_{gate}) \cdot \frac{l_{old}(y|x)}{T} + \alpha_{gate} \cdot l_{new}(y|x), \quad (3)$$

where $l(y|x)$ denotes predicted logits from models and T is the temperature scaling to regularize the magnitude of old model’s logits. f_{new} and g_θ are then jointly trained end-to-end with cross-entropy loss between our output logits $l_{GF}^*(y|x)$ and label distributions on downstream tasks.

The intuition behind Gated Fusion is that when f_{new} makes a mistake while f_{old} produces the correct output, the gate g_θ will learn to put more weight on f_{old} in order to minimize the final classification loss. This process effectively mitigates potential negative flips introduced by the model upgrade and thus improves the backward compatibility of final predictions.

3.2 Training and Inference

In practice, training Gated Fusion with randomly initialized f_{new} would make the shallow gating network quickly converge to favor the fully-trained f_{old} . To prevent this, we only train f_{new} for the first few epochs to ensure its competence before jointly training g_θ and f_{new} using $l_{GF}^*(x)$. In addition, we found that stopping gradient flow from g_θ to f_{new} can further prevent the performance decrease of the new model within Gated Fusion:

$$\alpha_{gate}(x) = g_\theta(\text{stop_grad}(\mathcal{E}_{new}(x))). \quad (4)$$

At inference time, Gated Fusion produces logits from f_{old} and f_{new} as well as the gate value α_{gate}

to make output predictions:

$$f_{GF}^*(x) = \text{Softmax}\left((1 - \alpha_{gate}) \cdot \frac{l_{old}}{T} + \alpha_{gate} \cdot l_{new}\right). \quad (5)$$

3.3 Inference with Cache

Our proposed Gated Fusion requires f_{old} to be hosted together with the new model. In reality, one could have a resource-constrained setting and request the old model to be discarded at inference. We note that in real applications, repetitive inputs are commonly seen in live traffic (Batrinca and Treleaven, 2015) and the backward compatibility of model upgrade entails that correct predictions can be preserved on the legacy instances already seen and predicted by the old model.

To simulate real scenarios, we randomly cache old model’s logits on a portion of test inputs. When getting out-of-cache instances, we use new model’s output embedding $\mathcal{E}_{new}(x)$ as key and euclidean distance as metric to search for the nearest cached instance. The cached old-model logits can then be used for Gated Fusion to make predictions without hosting f_{old} at inference.

4 Experiments Setup

4.1 Model Upgrade Scenarios

We conduct experiments on two representative model upgrade scenarios: (a) upgrade to a larger pretrained model of the same type, where we use BERT_{base} to BERT_{large}. (b) upgrade to a distinct pretrained model with the same size. We use BERT_{base} to ELECTRA_{base} (Clark et al., 2020) as this challenging model upgrade scenario for backward-compatibility, as they are pretrained under different self-supervised learning paradigms. The former uses masked language modeling (MLM) with reconstruction loss, while the latter is pretrained in generative-contrastive (adversarial) fashion with distributional divergence as the loss (Liu et al., 2021).

4.2 Datasets and Implementation

We evaluate our approach across three datasets. They represent different sentence-level classification tasks, from single-sentence to sentence-pair classification, with varying dataset sizes. We use: (a) Stanford Sentiment Treebank (SST-2), a single-sentence task to classify movie review sentiment, with 67k train and 0.9k dev set (Socher et al., 2013). (b) Microsoft Research Paraphrase Corpus (MRPC)

(Dolan and Brockett, 2005), a sentence-pair classification task for identifying paraphrases, with 3.7k train and 0.4k dev set. (c) Question Natural Language Inference (QNLI), a question-paragraph pair task to determine whether the paragraph contains the answer to the question, with 100k train and 5.5k dev set. Datasets are taken from GLUE Benchmark (Wang et al., 2018) and processed with scripts from Hugging Face².

For implementation, we use the sequence classification and pre-trained model parameters from Hugging Face Transformers³. Experiments are done in PyTorch (Paszke et al., 2019) with Tesla V100 GPUs and results are averaged over 5 random seeds. Learning rate, batch size, and train epoch are tuned during training new model alone on given tasks and then fixed for all backward-compatible solutions. In Gated Fusion, we first train f_{new} alone for first $(N - 1)$ epochs and then jointly train g_{θ} and f_{new} with Gated Fusion logits l_{GF}^* in the last epoch. Further implementation details can be found in the Appendix.

4.3 Baselines

We compare our approach with several strong baselines. (a) Train the new model directly on the target task without any adjustment, i.e. f_{new}^o . (b) The specialized distillation method proposed in Xie et al. (2021), where the KL-divergence of prediction probabilities between old and new models is applied when $p_{old}(y = y_i|x_i) > p_{new}(y = y_i|x_i)$. (c) Model ensemble via majority-voting that was shown to be very effective (Yan et al., 2021; Xie et al., 2021). Similarly, we use 5-seed new model ensemble as a strong baseline. (d) The ensemble of the old and new models probabilities, $p^*(y|x) = (1 - \alpha) \cdot p_{old}(y|x) + \alpha \cdot p_{new}(y|x)$, as well as ensemble of the old and new models logits, $l^*(y|x) = (1 - \alpha) \cdot l_{old}(y|x) + \alpha \cdot l_{new}(y|x)$, where α is searched among [0.5, 0.6, 0.7, 0.8, 0.9] to maximize backward-compatibility while achieving accuracy on par with the vanilla f_{new}^o .

5 Results and Analysis

5.1 Upgrade to a Larger Pretrained Model

Our first model upgrade scenario scales up the size of underlying pretrained language models. We experiment with BERT_{base} to BERT_{large}, where

²<https://huggingface.co/datasets/glue>

³<https://huggingface.co/docs/transformers/index>

$\text{BERT}_{base} \rightarrow \text{BERT}_{large}$	SST-2		MRPC		QNLI	
	\mathcal{R}_{NF}	Accuracy	\mathcal{R}_{NF}	Accuracy	\mathcal{R}_{NF}	Accuracy
Old Model	-	92.00 _{0.27}	-	85.69 _{0.90}	-	90.74 _{0.09}
New Model	2.18 _{0.21}	93.12 _{0.29}	4.12 _{1.04}	87.40 _{1.02}	2.72 _{0.13}	92.22 _{0.16}
Distillation (Xie et al., 2021)	1.97 _{0.22}	93.33 _{0.20}	3.53 _{0.77}	87.70 _{1.34}	2.31 _{0.14}	92.60 _{0.19}
New Model Ensemble	2.00 _{0.31}	93.30 _{0.24}	2.25 _{0.61}	88.87 _{0.77}	1.98 _{0.21}	92.97 _{0.22}
Old-New Probs Ensemble	1.06 _{0.27}	93.12 _{0.38}	1.67 _{0.78}	87.16 _{1.12}	1.04 _{0.26}	92.44 _{0.23}
Old-New Logits Ensemble	1.06 _{0.27}	93.12 _{0.38}	1.67 _{0.78}	87.16 _{1.12}	1.04 _{0.26}	92.44 _{0.23}
Gated Fusion	0.78 _{0.20}	93.05 _{0.09}	1.18 _{0.52}	87.45 _{0.52}	0.73 _{0.13}	92.24 _{0.24}

Table 1: Negative flip rate \mathcal{R}_{NF} and model accuracy (%) of competing methods to optimize backward-compatibility without performance degradation during $\text{BERT}_{base} \rightarrow \text{BERT}_{large}$ model upgrade.

$\text{BERT}_{base} \rightarrow \text{ELECTRA}_{base}$	SST-2		MRPC		QNLI	
	\mathcal{R}_{NF}	Accuracy	\mathcal{R}_{NF}	Accuracy	\mathcal{R}_{NF}	Accuracy
Old Model	-	92.00 _{0.27}	-	85.69 _{0.90}	-	90.74 _{0.09}
New Model	1.63 _{0.20}	95.00 _{0.06}	3.73 _{0.36}	88.58 _{0.57}	2.82 _{0.32}	92.90 _{0.26}
Distillation (Xie et al., 2021)	1.49 _{0.24}	95.02 _{0.21}	3.68 _{0.79}	88.82 _{0.94}	2.58 _{0.17}	93.03 _{0.16}
New Model Ensemble	1.12 _{0.09}	95.39 _{0.09}	3.24 _{0.24}	89.02 _{0.48}	2.26 _{0.08}	93.49 _{0.07}
Old-New Probs Ensemble	1.40 _{0.17}	95.07 _{0.15}	3.14 _{0.42}	88.53 _{0.48}	0.98 _{0.20}	93.04 _{0.21}
Old-New Logits Ensemble	0.89 _{0.17}	94.95 _{0.13}	3.28 _{0.43}	88.48 _{0.51}	0.98 _{0.20}	93.04 _{0.21}
Gated Fusion	0.71 _{0.18}	95.02 _{0.16}	2.40 _{0.50}	88.68 _{0.68}	0.81 _{0.16}	92.98 _{0.17}

Table 2: Negative flip rate \mathcal{R}_{NF} and model accuracy (%) of competing methods to optimize backward-compatibility without performance degradation during $\text{BERT}_{base} \rightarrow \text{ELECTRA}_{base}$ model upgrade.

the model size is tripled (110M vs 340M) and the model depth is doubled (12 vs 24 layers).

Table 1 shows the results. For f_{new}^o , we can observe that the negative flip rates \mathcal{R}_{NF} are usually much larger than the accuracy gains across tasks, which could be the reason to hinder new model adoptions in real-world applications. Besides, when dividing \mathcal{R}_{NF} over the error rate ($1 - accuracy$), we can observe that around 30% to 40% of all f_{new}^o prediction errors are in fact the *new* errors introduced during model upgrade. For improving prediction backward-compatibility, our proposed Gated Fusion outperforms other competing methods to considerably reduce \mathcal{R}_{NF} without degradation on accuracy. Note that best α values found for the two variants of old-new ensemble are both 0.5, hence producing identical results.

Compared to the vanilla new model, gated fusion obtains absolute \mathcal{R}_{NF} reductions of -1.40% on SST-2, -2.94% on MRPC, and -1.99% on QNLI. These translate to reducing the total negative flip cases by 64.2%, 71.4%, 73.2%, respectively. Compared to the strongest baseline (old-new ensemble), we obtain further absolute \mathcal{R}_{NF} reductions of -0.28% on SST-2, -0.49% on MRPC, and

-0.31% on QNLI, which translate to further reducing 12.8%, 11.9%, and 11.4% of negative flip cases. These results show the effectiveness of our method to mitigate a significant amount of regression errors during model upgrade.

5.2 Upgrade to a Different Pretrained Model

A more challenging upgrade scenario is when old and new models are pretrained under distinctive paradigms, producing two representation spaces of fairly different characteristics (Meng et al., 2021b). We experiment with BERT_{base} to ELECTRA_{base} in this scenario, where two models have the same size but are pretrained under utterly different schemes, i.e. generative versus adversarial.

Table 2 shows the results. For f_{new}^o , compared with upgrading to BERT_{large} , we observe larger accuracy gains and lower \mathcal{R}_{NF} on SST-2 and MRPC. However, on QNLI, upgrading to ELECTRA_{base} achieves a higher accuracy gain but an even a higher \mathcal{R}_{NF} . This implies that boosting accuracy and improving backward compatibility could be two related but different objectives.

For mitigation strategies, Gated Fusion achieves the lowest negative flip rates across datasets without any accuracy loss. We obtain absolute \mathcal{R}_{NF} re-

	SST-2		MRPC		QNLI	
	\mathcal{R}_{NF}	Accuracy	\mathcal{R}_{NF}	Accuracy	\mathcal{R}_{NF}	Accuracy
Old Model: BERT _{base}	-	92.00 _{0.27}	-	85.69 _{0.90}	-	90.74 _{0.09}
New Model: BERT _{large}	2.18 _{0.21}	93.12 _{0.29}	4.12 _{1.04}	87.40 _{1.02}	2.72 _{0.13}	92.22 _{0.16}
Model Ensemble: 5 seeds	2.00 _{0.31}	93.30 _{0.24}	2.25 _{0.61}	88.87 _{0.77}	1.98 _{0.21}	92.97 _{0.22}
Model Ensemble: 10 seeds	1.79 _{0.17}	93.69 _{0.15}	2.01 _{0.29}	89.46 _{0.51}	2.01 _{0.14}	92.97 _{0.19}
Model Ensemble: 20 seeds	1.79 _{0.25}	93.62 _{0.16}	1.76 _{0.50}	89.56 _{0.48}	1.82 _{0.15}	93.13 _{0.08}
Gated Fusion	0.78 _{0.20}	93.05 _{0.09}	1.18 _{0.52}	87.45 _{0.52}	0.73 _{0.13}	92.24 _{0.24}
New Model: ELECTRA _{base}	1.63 _{0.20}	95.00 _{0.06}	3.73 _{0.36}	88.58 _{0.57}	2.82 _{0.32}	92.90 _{0.26}
Model Ensemble: 5 seeds	1.12 _{0.09}	95.39 _{0.09}	3.24 _{0.24}	89.02 _{0.48}	2.26 _{0.08}	93.49 _{0.07}
Model Ensemble: 10 seeds	1.24 _{0.18}	95.30 _{0.16}	3.63 _{0.50}	88.58 _{0.20}	2.21 _{0.12}	93.57 _{0.15}
Model Ensemble: 20 seeds	1.19 _{0.16}	95.32 _{0.17}	3.43 _{0.51}	88.92 _{0.48}	2.15 _{0.17}	93.63 _{0.11}
Gated Fusion	0.71 _{0.18}	95.02 _{0.16}	2.40 _{0.50}	88.68 _{0.68}	0.81 _{0.16}	92.98 _{0.17}

Table 3: Negative flip rate \mathcal{R}_{NF} and model accuracy (%) when increasing number of seeds used in new model ensemble, comparing with our proposed method (Gated Fusion).

	SST-2	MRPC	QNLI
Old: BERT _{base}	92.00 _{0.27}	85.69 _{0.90}	90.74 _{0.09}
to BERT _{large}	93.12 _{0.29}	87.40 _{1.02}	92.22 _{0.16}
Gated Fusion	93.05 _{0.09}	87.45 _{0.52}	92.24 _{0.24}
- drop old model	93.17 _{0.61}	87.75 _{1.14}	92.22 _{0.44}
to ELECTRA _{base}	95.00 _{0.06}	88.58 _{0.57}	92.90 _{0.26}
Gated Fusion	95.02 _{0.16}	88.68 _{0.68}	92.98 _{0.17}
- drop old model	95.16 _{0.09}	88.63 _{0.94}	93.06 _{0.13}

Table 4: Accuracy (%) when dropping the old model within Gated Fusion at inference time.

ductions of -0.92% on SST-2, -1.33% on MRPC, and -2.01% on QNLI over the vanilla setup, reducing 56.4%, 35.7%, and 71.3% of overall negative flips, respectively. Compared with upgrading to BERT_{large}, we observe that upgrading to ELECTRA_{base} has much smaller relative negative flip reductions on SST-2 and MRPC, showing that it could be indeed harder to improve backward-compatibility when upgrading to a distinct pre-trained model. In contrast, similar relative negative flip reductions are observed on QNLI across two upgrade scenarios. This could be attributed to the abundant training data of the downstream task.

5.3 Drop Old Model at Inference Time

Our proposed method requires the old model to be hosted together with the new model. A natural question is whether we could train Gated Fusion with the old model and then discard it at inference time to host the new model only.

We first experiment with directly dropping the old model within Gated Fusion at inference time.

	SST-2	
	\mathcal{R}_{NF}	Accuracy
Old Model: BERT _{base}	-	92.00 _{0.27}
New Model: ELECTRA _{base}	1.63 _{0.20}	95.00 _{0.06}
Gated Fusion - 50% cache	1.26 _{0.10}	94.86 _{0.27}
Gated Fusion - 75% cache	0.99 _{0.25}	94.91 _{0.12}
Gated Fusion	0.71 _{0.18}	95.02 _{0.16}

Table 5: Negative flip rate \mathcal{R}_{NF} and model accuracy (%) of Gated Fusion with $X\%$ cache of old model logits at inference time.

Results in Table 4 show that dropping old model in Gated Fusion can still achieve comparable accuracy across the board, suggesting no performance degradation. Nonetheless, we observe that the negative flip rates also fall back to similar positions as training the new model in the vanilla setting.

However, in real application scenario, live inputs are often repetitively seen across time and ensuring backward-compatibility means that correct predictions on same instances can be preserved after model upgrade. We experiment with the caching method introduced in section 3.3 to store old model’s logits on random $X\%$ of test instances where Gated Fusion can later access them for inference. Results in Table 5 show that with higher percentage of cache, \mathcal{R}_{NF} is gradually reduced towards \mathcal{R}_{NF} of the original Gated Fusion, which is equivalent to 100% cache. Still, we observe a notable gap in \mathcal{R}_{NF} between the partial caching and full settings. We leave the examination of ways to achieve the upper bound in reduction in \mathcal{R}_{NF} with smaller cache to the future work.

	(Task, Label)	Examples
BERT _{base} → BERT _{large}	(SST-2, Positive)	[Sentence] A study in shades of gray, offering itself up in subtle plot maneuvers ...
	(SST-2, Negative)	[Sentence] Manages to be both repulsively sadistic and mundane.
	(MRPC, Not Equivalent)	[Sentence 1] Vivace was founded in 1999 and has raised over \$118 million in three rounds of venture financing. [Sentence 2] During difficult times for technology venture capital, Vivace raised over \$118 million in three rounds of venture financing.
	(QNLI, Entailment)	[Question] Why was there a depreciation of the industrialized nations dollars? [Sentence] Anticipating that currency values would fluctuate unpredictably for a time, the industrialized nations increased their reserves (by expanding their money supplies) in amounts far greater than before.
BERT _{base} → ELECTRA _{base}	(SST-2, Positive)	[Sentence] Aside from minor tinkering , this is the same movie you probably loved in 1994, except that it looks even better.
	(SST-2, Negative)	[Sentence] It showcases carvey’s talent for voices, but not nearly enough and not without taxing every drop of one’s patience to get to the good stuff .
	(MRPC, Equivalent)	[Sentence 1] Blair’s Foreign Secretary Jack Straw was to take his place on Monday to give a statement to parliament on the European Union. [Sentence 2] Blair’s office said his Foreign Secretary Jack Straw would take his place on Monday to give a statement to parliament on the EU meeting the prime minister attended last week.
	(QNLI, Not Entailment)	[Question] What is the main executive body of the EU? [Sentence] This means that the Commission has a monopoly on initiating the legislative procedure, although the Council is the "de facto catalyst of many legislative initiatives".

Table 6: Examples of regression errors present when upgrading to the vanilla new model f_{new}^o but fixed by our Gated Fusion approach, i.e. predictions of $(f_{old}, f_{new}^o, f_{GF}^*)$ are *(correct, incorrect, correct)*, respectively.

5.4 Limitations of New Model Ensemble

In previous works (Yan et al., 2021; Xie et al., 2021), new model ensemble via majority voting is shown to effectively reduce negative flips and posed as a difficult-to-beat baseline. Here, we increase the number of models in ensemble to examine its limitations. Results in Table 3 show that ensemble with more models generally help to obtain lower \mathcal{R}_{NF} . However, \mathcal{R}_{NF} converges quickly as number of models increased, where a notable gap remains between new model ensemble and Gated Fusion. Moreover, the results show once more that boosting accuracy does not necessarily improve the backward compatibility in model upgrade.

In principle, two sources could cause negative flips during model upgrade (a) the stochasticity during model training, including initializations, data loading order, and optimization process (Somepalli et al., 2022). (b) the distinctions between old and new model hypotheses, including architecture and pretraining data and procedure, leading to different representation space structures and prediction behaviors in terms of decision boundaries. Without an explicit connection to f_{old} , new model ensemble can only reduce negative flips primarily caused by the first factor, while our proposed Gated Fusion directly learns to mitigate regression errors regardless of their causes.

Besides, as large-scale generative models become more and more powerful and popular (Raffel

et al., 2020; Brown et al., 2020; Su et al., 2021), it would be difficult to fine-tune them multiple times on a target task for ensemble.

5.5 Analysis of Gated Fusion

Comparing f_{new}^o with f_{GF}^* , we can calculate the *fix rate* and *new fault rate* of our Gated Fusion method. During an upgrade, if there are 20 negative flips with f_{new}^o and 16 out of them can be mitigated by f_{GF}^* , we obtain the fix rate to be $16/20 = 80\%$. Similarly, if f_{GF}^* introduces another 4 new negative flips which are not present with f_{new}^o , the new fault rate is computed to be $4/20 = 20\%$. We calculate the 5-seed average of these two rates across different classification tasks and upgrade scenarios. In BERT_{base} to BERT_{large}, the averaged fix rates by Gated Fusion are 68.4% on SST-2, 83.8% on MRPC, and 82.9% on QNLI, with new fault rates being 4.1% on SST-2, 11.3% on MRPC, and 9.7% on QNLI. In BERT_{base} to ELECTRA_{base}, Gated Fusion achieves the averaged fix rates 58.0% on SST-2, 50.8% on MRPC, and 75.6% on QNLI, with new fault rates being 2.8% on SST-2, 15.2% on MRPC, and 4.0% on QNLI. These results show that, on average, Gated Fusion is able to eliminate 69.9% of total regression errors while adding only 7.9% new ones, comparing with doing model upgrade without any treatment, i.e. f_{new}^o .

Table 6 shows a few regression error cases fixed by our proposed approach. In general, Gated Fu-

sion can mitigate negative flips happened on different classes across diverse tasks as well as on inputs with variable lengths. With closer inspections of f_{GF}^* , we found that when f_{new} produces incorrect predictions and f_{old} gives correct outputs, g_θ is capable of putting larger weights on f_{old} to ensure the backward compatibility. We also observed that the gate g_θ is more prone to over-fitting when the downstream tasks have smaller training set, e.g. MRPC, or are more difficult in nature, e.g. single-sentence task SST-2 versus sentence-pair tasks, which causes Gated Fusion to introduce more new errors, i.e. higher new fault rates.

6 Discussion

Gated Fusion requires to host both old and new models at inference time, which could raise a concern regarding the increased computational burden. However, in practice, old model’s logits of previous inference instances can be cached in storage and later leveraged in our Gated Fusion. That is, we only need to host the new model with the gate at inference time and leverage old predictions from cache. And for the out-of-cache inputs, backward-compatibility would be less of an issue since users have not observed such examples to make conclusions on the underlying regression.

For real-world applications, there could be multiple model updates and thus multiple legacy versions. We note that in this scenario, user experience would be primarily grounded on predictions of the latest legacy version, which are also saved in cache. Our Gated Fusion can hence leverage them and make new model’s predictions compatible to those from the latest legacy version.

In addition, we emphasize that the main challenge in the regression reduction research problem is to find the best trade-off between model effectiveness and backward compatibility. In this work, we show that the weighted ensemble of old-new models with a learned gate, which we call Gated Fusion, achieves a better negative flip rate than previously explored methods for regression reduction, while straight-forward ensemble approaches cannot naturally weigh on this trade-off. We don’t claim to invent the gated ensemble of old and new models but rather that our main contribution is to show that by repurposing the classic gating mechanism, the gated ensemble can become the most competitive approach to the challenging model-upgrade regression reduction problem, with no overall per-

formance degradation on two realistic model update scenarios across three different datasets.

Recently, more and more NLP products have been deployed in the industry as this field matures. We would like to stress that as better NLP models are being developed, the backward-compatible model upgrade problem naturally emerges as the new research topic strongly motivated by the real-world challenges. While backward-compatibility is currently a niche research topic, we believe that there are many thrilling future directions worth to be investigated.

7 Related Work

[Yan et al. \(2021\)](#) first studied the backward compatibility of predictions during model upgrade on image classification tasks. Later, [Xie et al. \(2021\)](#) investigated the similar topic in natural language understanding and formulated it as a constrained optimization problem. They both show that customized variants of knowledge distillation ([Hinton et al., 2015](#)), which align the predictions of old and new models on potential regression errors, are effective approaches. A model ensemble has also shown to be surprisingly effective ([Yan et al., 2021](#); [Xie et al., 2021](#)), despite no explicit connection between old and new models. This was credited to variance reduction in model predictions, making it less prone to over-fitting and reducing regression errors indirectly. In this work, we leverage the gating mechanism to combine old and new models to further reduce model upgrade regression errors by a large margin across classification tasks.

[Cai et al. \(2022\)](#) analyzed and proposed backward congruent re-ranking to reduce regression in model upgrades for structured predictions tasks such as dependency parsing and conversational semantic parsing. [Träuble et al. \(2021\)](#) proposed an efficient probabilistic approach to locate data instances whose old predictions could be incorrect and update them with ones from the new model. [Zhou et al. \(2022\)](#) looked into forward compatibility, where new classes can be easily incorporated without negatively impacting existing prediction behavior. More recently, [Schumann et al. \(2023\)](#) inspected classification model regression during training data updates and mitigated the problem by interpolating between weights of the old and new models. On top of that, learning cross-model compatible embeddings has been extensively explored in visual search ([Chen et al., 2019](#); [Hu et al., 2019](#);

Wang et al., 2020). Several techniques have been proposed to optimize cross-model interoperability of embeddings, including metric space alignment (Shen et al., 2020), architecture search (Duggal et al., 2021), and aligning class centers between models Meng et al. (2021a). In this work, we focus on improving backward compatibility during model upgrade in terms of prediction behavior on classification tasks, i.e. old and new models should produce consistently correct predictions.

Reducing regression during model upgrade is also related to continual learning (Parisi et al., 2019; De Lange et al., 2019; Sun et al., 2019; Chuang et al., 2020; Sachidananda et al., 2021), incremental learning (Chaudhry et al., 2018; Shan et al., 2020) and concept drifting (Gama et al., 2014; Žliobaitė et al., 2016; Ganin et al., 2016; Zhuang et al., 2020; Lazaridou et al., 2021). In these problems, models are required to learn from and deal with continuously changing data (in terms of examples, classes or tasks), and also need to prevent the forgetting of previously learnt knowledge. This could be one potential cause of regression observed at inference. However, in backward-compatible model upgrade, a new model, usually with distinct network architecture, is trained from scratch to perform the same task and is expected to behave similarly wherever the previous model predicts correctly.

The gating mechanism is widely adopted by recurrent neural networks to effectively control information flows across networks (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Van Oord et al., 2016; Dauphin et al., 2017; Lai et al., 2019) and contextualize embeddings (Peters et al., 2018; Lai et al., 2020). It is then repurposed to act as a switch for the mixture of different prediction modes, notably to combine input word copying based on the pointer network (Vinyals et al., 2015) with the word generation from output vocabulary (Gu et al., 2016; Merity et al., 2016; See et al., 2017). Our proposed approach is inspired by these works and leverages the gating mechanism to effectively combine old and new models to improve backward compatibility during model upgrade.

8 Conclusion

Ensuring backward compatibility during model upgrade has become a critical topic in real-world NLP applications. In this work, we proposed a new approach, *Gated Fusion*, that achieves significantly better backward compatibility without

compromising accuracy performance on two challenging upgrade scenarios for NLP classification. Experiments demonstrated that our approach outperforms competing methods and achieves negative flip rate reductions by up to 73.2%. Our future research includes improving backward compatibility beyond classification to span detection, model upgrades with very large language models, and upgrades on training data or label schema. We hope that this work can inspire further research and make progress towards smoother transitions of prediction powers as NLP systems evolve.

Limitations

Our proposed method mostly works on the upgrades of underlying pretrained language models for NLP classification tasks. Potential limitations include applying our approach on distant tasks such as question answering or information retrieval, upgrade to models from different architecture families such as recurrent neural nets, and the inapplicability of our method to more recent learning formulation such as in-context learning via prompting.

Ethics Statement

Prediction backward compatibility during model upgrade is an emerging research topic to ensure positive congruency and smoother transitions from existing models towards more performant systems. With primary evaluation on accuracy and negative flips, we acknowledge that our method may also inherit social biases and other toxicity persisted in the legacy models. On the other hand, we have noted that fairness and safety have been one of principal criteria when developing system upgrades. Investigations of the inheritance of persistent toxicity and mitigation of it during backward-compatible upgrades merit interests of future research.

Acknowledgements

We would like to acknowledge AWS AI Labs for inspiring discussions, honest feedback, and full support. We are also very grateful to reviewers for judicious comments and valuable suggestions.

References

Bogdan Batrinca and Philip C Treleaven. 2015. Social media analytics: a survey of techniques, tools and platforms. *Ai & Society*, 30(1):89–116.

- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Deng Cai, Elman Mansimov, Yi-An Lai, Yixuan Su, Lei Shu, and Yi Zhang. 2022. Measuring and reducing model update regression in structured prediction for nlp. *arXiv preprint arXiv:2202.02976*.
- Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547.
- Ken Chen, Yichao Wu, Haoyu Qin, Ding Liang, Xuebo Liu, and Junjie Yan. 2019. R3 adversarial network for cross model face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9868–9876.
- Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.
- Yung-Sung Chuang, Shang-Yu Su, and Yun-Nung Chen. 2020. Lifelong language knowledge distillation. *arXiv preprint arXiv:2010.02123*.
- Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.
- Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.
- Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2019. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2(6).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.
- Rahul Duggal, Hao Zhou, Shuo Yang, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. 2021. Compatibility-aware heterogeneous visual search.
- João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37.
- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.
- Geoffrey E. Hinton, Oriol Vinyals, and J. Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Jie Hu, Rongrong Ji, Hong Liu, Shengchuan Zhang, Cheng Deng, and Qi Tian. 2019. Towards visual feature translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3004–3013.
- Yi-An Lai, Arshit Gupta, and Yi Zhang. 2019. Goal-embedded dual hierarchical model for task-oriented dialogue generation. *arXiv preprint arXiv:1909.09220*.
- Yi-An Lai, Garima Lalwani, and Yi Zhang. 2020. Context analysis for pre-trained masked language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3789–3804.
- Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d’Autume, Sebastian Ruder, Dani Yogatama, et al. 2021. Pitfalls of static language modelling. *arXiv preprint arXiv:2102.01951*.
- Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*.
- Qiang Meng, Chixiang Zhang, Xiaoqiang Xu, and Feng Zhou. 2021a. Learning compatible embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9939–9948.

- Yu Meng, Chenyan Xiong, Payal Bajaj, Paul Bennett, Jiawei Han, Xia Song, et al. 2021b. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *Advances in Neural Information Processing Systems*, 34:23102–23114.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.
- German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. **Beyond accuracy: Behavioral testing of NLP models with CheckList**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.
- Vin Sachidananda, Jason S Kessler, and Yi-An Lai. 2021. Efficient domain adaptation of language models via adaptive tokenization. *arXiv preprint arXiv:2109.07460*.
- Raphael Schumann, Elman Mansimov, Yi-An Lai, Nikolaos Pappas, Xibin Gao, and Yi Zhang. 2023. Backward compatibility during data updates by weight interpolation. *arXiv preprint arXiv:2301.10546*.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.
- Guangxu Shan, Shiyao Xu, Li Yang, Shengbin Jia, and Yang Xiang. 2020. Learn#: A novel incremental learning method for text classification. *Expert Systems with Applications*, 147:113198.
- Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. 2020. Towards backward-compatible representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. **Recursive deep models for semantic compositionality over a sentiment treebank**. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.
- Gowthami Somepalli, Liam Fowl, Arpit Bansal, Ping Yeh-Chiang, Yehuda Dar, Richard Baraniuk, Micah Goldblum, and Tom Goldstein. 2022. Can neural nets learn the same model twice? investigating reproducibility and double descent from the decision boundary perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13699–13708.
- Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. *arXiv preprint arXiv:2109.14739*.
- Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.
- Frederik Träuble, Julius Von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf, and Peter Gehler. 2021. Backward-compatible prediction updates: A probabilistic approach. *Advances in Neural Information Processing Systems*, 34:116–128.
- Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. **GLUE: A multi-task benchmark and analysis platform for natural language understanding**. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.
- Chien-Yi Wang, Ya-Liang Chang, Shang-Ta Yang, Dong Chen, and Shang-Hong Lai. 2020. Unified representation learning for cross model compatibility. *arXiv preprint arXiv:2008.04821*.
- Yuqing Xie, Yi an Lai, Yuanjun Xiong, Yi Zhang, and Stefano Soatto. 2021. Regression bugs are in your model! measuring, reducing and analyzing regressions in nlp model updates. *arXiv preprint arXiv:2105.03048*.

- Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqu Deng, Meng Wang, Wei Xia, and Stefano Soatto. 2021. Positive-congruent training: Towards regression-free model updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14299–14308.
- Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. 2022. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9046–9056.
- Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.
- Indrė Žliobaitė, Mykola Pechenizkiy, and Joao Gama. 2016. An overview of concept drift applications. *Big data analysis: new algorithms for a new society*, pages 91–114.

A Details on Experiment Settings

A.1 Model Training Hyper-parameters

We search among following hyper-parameter space for the training of the old model f_{old} and the new model in the vanilla setting f_{new}^o across all datasets:

- Learning Rate: $5e^{-6}$, $1e^{-5}$, $3e^{-5}$, $5e^{-5}$
- Batch Size: 16, 32
- Training Epochs: 3, 5, 8.

The selected hyper-parameters for each model with (*learning rate, batch size, training epoch*):

- BERT_{base}:
 - On SST-2: (lr $1e^{-5}$, batch 16, epoch 5)
 - On MRPC: (lr $3e^{-5}$, batch 16, epoch 5)
 - On QNLI: (lr $3e^{-5}$, batch 32, epoch 3)
- BERT_{large}:
 - On SST-2: (lr $1e^{-5}$, batch 16, epoch 5)
 - On MRPC: (lr $3e^{-5}$, batch 16, epoch 5)
 - On QNLI: (lr $3e^{-5}$, batch 32, epoch 3)
- ELECTRA_{base}:
 - On SST-2: (lr $1e^{-5}$, batch 16, epoch 5)
 - On MRPC: (lr $5e^{-5}$, batch 32, epoch 5)
 - On QNLI: (lr $3e^{-5}$, batch 32, epoch 3)

These model training hyper-parameters for a specific model on one specific dataset is then fixed and reused for all the competing methods to improve backward compatibility during model upgrade.

A.2 Distillation Hyper-parameters

The knowledge distillation method from [Xie et al. \(2021\)](#) imposed an additional loss $\lambda \cdot KL(l_{old}/T, l_{new}/T)$ on potential regression instances. We experimented the best possible hyper-parameters from the following:

- λ : 0.1, 1.0, 10.0
- Temperature T : 0.5, 1.0, 2.0

A.3 Details on Gated Fusion

We initialize the gate g_θ to be a two-layer feed-forward network with the architecture (*Dropout, Linear, LayerNorm, ReLU, Dropout, Linear, Sigmoid*) and fix the hidden size to be 64 across all our experiments.

During the training of Gated Fusion, we only train the f_{new} within f_{GF}^* for the first $(N - 1)$ epochs to ensure its competence, where N is the total training epochs. In the last training epoch, we jointly train g_θ and f_{new} using the Gated Fusion logits l_{GF}^* with the secondary learning rate $lr2$. To prevent the overfitting of the gate, we also apply

drop_gate where at each training step during the last epoch, there is $D\%$ to only train f_{new} within f_{GF}^* and $(1 - D)\%$ to train with l_{GF}^* .

The hyper-parameter space of Gated Fusion is listed as follows:

- Drop Gate (%): 40, 50, 60, 80
- Temperature T on old logits: 1.0, 1.2, 1.4, 1.6
- lr2: $5e^{-7}$, $1e^{-6}$, $3e^{-6}$, $1e^{-5}$, $3e^{-5}$

We found that to achieve good results, the gap in logit magnitude of f_{old} and f_{new} needs to be bridged by the temperature when upgrading from BERT_{base} to ELECTRA_{base}, with T being 1.6 on SST-2, 1.6 on MRPC, and 1.2 on QNLI. On the other hand, $T = 1$ gives good results across three datasets when upgrading from BERT_{base} to BERT_{large}. This could result from the distinct pretraining schemes between models where MLM seem to produce larger magnitude of output logits.

AMBICOREF: Evaluating Human and Model Sensitivity to Ambiguous Coreference

Yuewei Yuan and Chaitanya Malaviya and Mark Yatskar

University of Pennsylvania

{yuewei, cmalaviy, myatskar}@seas.upenn.edu

Abstract

Given a sentence “Abby told Brittney that she upset Courtney”, one would struggle to understand who “she” refers to, and ask for clarification. However, if the word “upset” were replaced with “hugged”, “she” unambiguously refers to Abby. We study if modern co-reference resolution models are sensitive to such pronominal ambiguity. To this end, we construct **AMBICOREF**, a diagnostic corpus of minimal sentence pairs with ambiguous and unambiguous referents. Our examples generalize psycholinguistic studies of human perception of ambiguity around particular arrangements of verbs and their arguments. Analysis shows that (1) humans are less sure of referents in ambiguous AmbiCoref examples than unambiguous ones, and (2) most coreference models show little difference in output between ambiguous and unambiguous pairs. We release **AMBICOREF** as a diagnostic corpus for testing whether models treat ambiguity similarly to humans.¹

1 Introduction

Ambiguity is a fundamental feature of language (Wasow et al., 2003) that some linguists believe arises because of a pressure for efficient communication (Haywood et al., 2005; Piantadosi et al., 2012). Recently, several works have highlighted the existence of ambiguity in tasks such as question answering (Min et al., 2020; Guo et al., 2021), frame disambiguation (Dumitrache et al., 2019), anaphora resolution (Poesio and Artstein, 2005) and language modeling (Aina and Linzen, 2021). Yet systematic evaluation of how models react to ambiguity across many types of language processing problems is missing. We contribute one such study about coreference resolution.

Coreference resolution is crucial to natural language understanding, especially in long contexts, such as dialog. Ambiguity may arise naturally

¹Our dataset and code is available at <https://github.com/LucyYYW/AmbiCoref>.

in dialog, but existing models do not have well-defined target behavior for such coreferences. In contrast, when people encounter coreferential ambiguity, they recognize it, and can ask for clarification. Existing resources, such as OntoNotes (Weischedel et al., 2013), do not provide fine-grained annotations of such instances to evaluate model behavior. This may result in models not being calibrated to handle the uncertainty in interpretations of ambiguous statements. In this work, we ask how sensitive to ambiguity are models trained on these resources?

To understand how existing coreference models react to ambiguity, we construct a diagnostic corpus, **AMBICOREF**. **AMBICOREF** is composed of minimal pairs with ambiguous and unambiguous referents, created from four types of templates. Ambiguity is achieved by reducing context sizes to one sentence, and creating sentences where participating verbs under-constrain the interpretation of their arguments. For example, in Table 1, line 2, our first template leverages ambiguity around verbs expressing subjective experiences.² The templates are designed by drawing on psycholinguistic studies (Springston, 1976; Caramazza et al., 1977; Rohde and Kehler, 2014) and a core contribution of our work is to generalize their observations to create thousands of instances. We achieve this by identifying VerbNet (Schuler, 2005) classes that are likely to contain appropriate verbs, and manually assigning them to templates. Combined with variability we introduce using noun lists, **AMBICOREF** contains over 96 thousand sentences.

We verify that humans perceive instances in **AMBICOREF** in intended ways by crowdsourcing judgements (§3). Annotators are asked to find the coreferent for a pronoun in a sentence, and rate their confidence, to account for the gradience in ambiguity judgements (Schutze, 1995). We find

²Such instances require specific syntactic arrangements: the ambiguous instance in line 2 is unambiguous if the pronoun is moved to the object position of bored.

	Type	Ambig.	Template	Count
1	Experiencer Obj (ECO-1)	✗	[<i>Emily</i>] _A told [<i>Jessica</i>] _B that [<i>she</i>] _A [saw] [Brian].	11336
2	Experiencer Obj (ECO-1)	✓	[<i>Emily</i>] _A told [<i>Jessica</i>] _B that [<i>she</i>] _? [bored] [Brian].	11336
3	Experiencer Obj (ECO-2)	✗	[<i>The mother</i>] _A told [<i>the sister</i>] _B that [<i>she</i>] _A [saw] the client.	11336
4	Experiencer Obj (ECO-2)	✓	[<i>The mother</i>] _A told [<i>the sister</i>] _B that [<i>she</i>] _? [bored] the client.	11336
5	Experiencer Sub (ECS-1)	✗	[<i>The aunt</i>] _A told [<i>Sarah</i>] _B that [the daughter] [met with] [<i>her</i>] _A .	4472
6	Experiencer Sub (ECS-1)	✓	[<i>The aunt</i>] _A told [<i>Sarah</i>] _B that [the daughter] [liked] [<i>her</i>] _? .	4472
7	Experiencer Sub (ECS-2)	✗	[<i>The father</i>] _A told [<i>the son</i>] _B that the client [met with] [<i>him</i>] _A .	4472
8	Experiencer Sub (ECS-2)	✓	[<i>The father</i>] _A told [<i>the son</i>] _B that the client [liked] [<i>him</i>] _? .	4472
9	Implicit Causality (IC)	✗	[<i>Abby</i>] _A [called] [<i>Jane</i>] _B because [<i>she</i>] _A [wanted to apologize].	8424
10	Implicit Causality (IC)	✓	[<i>Abby</i>] _A [called] [<i>Jane</i>] _B because [<i>she</i>] _? [is leaving soon].	8424
11	Transfer (TOP)	✗	[<i>Daniel</i>] _A [baked] [<i>the boy</i>] _B [a cake] [after] [<i>he</i>] _B [asked for one].	8424
12	Transfer (TOP)	✓	[<i>Daniel</i>] _A [baked] [<i>the boy</i>] _B [a cake] [before] [<i>he</i>] _? [had lunch].	8424

Table 1: Summary of the six template pairs that make up AMBICOREF. Template slot are indicated in square bracket, and clusters are marked with subscripts and color. All templates pair an unambiguous sentence with an ambiguous sentence, where they differ only in the choice of verb phrase.

that, for unambiguous instances, humans strongly associate the pronoun with the intended noun but for ambiguous ones, they show reduced confidence across all templates, where the majority of participants are either not confident or mark them as ambiguous. This suggests that humans process ambiguous and unambiguous sentences in AMBICOREF in qualitatively different ways.

AMBICOREF can be used to evaluate model behavior in the presence of ambiguity. We analyze five representative English models: three in CoreNLP (Manning et al., 2014), SpanBERT (Joshi et al., 2020), and NeuralCoref 4.0 (Wolf et al., 2020) (§4). Our main evaluation involves comparing coreference cluster assignments of the pronoun, between ambiguous and unambiguous samples. 4 out of the 5 models we analyze show almost no behavioral change. Unlike humans, coreference models largely do not alter their decisions in the presence of ambiguity. Our analysis implies models likely need to explicitly account for ambiguity to achieve human-like behavior in the face of ambiguous input.

2 Dataset Construction

To understand model sensitivity towards coreferential ambiguity, we build AMBICOREF using four types of templates, shown in Table 1. The templates are created in minimal pairs, and the only difference between the ambiguous and unambiguous counterparts lies in the choice of verb phrase. Note that while ambiguity is a graded phenomenon, we use the term “ambiguous” for instances that are *more likely* to elicit ambiguous human judgments and vice-versa. Verb phrases are extracted

from suitable verb classes in VerbNet (Schuler, 2005), identified by manual annotation of VerbNet clusters.³ Each template is instantiated with verbs, names, noun-phrases, and gender-appropriate pronouns, greatly expanding the variation in cases identified in previous studies.

2.1 Template Types

Experiencer Constraint for Objects (ECO) Springston (1976) propose the Experiencer Constraint for complement constructions which we operationalize in our templates. Verbs that mark their object as the experiencer of an emotion restrict the assignment of an object position pronoun to the subject of a declarative communication verb. Conversely, the assignment is unconstrained when the pronoun is the subject of an experiencer verb. For example, in row 2 of Table 1, a pronoun in the subject position of “bored” is ambiguous (but would not be so in the object position). If the main verb does not impose an experiencer constraint, row 1, then a pronoun in the subject position is unambiguous. We instantiate two variants with names (rows 1,2) and general entities (rows 3,4).

Experiencer Constraint for Subjects (ECS)

The Experiencer Constraint also suggests that verbs that mark their subjects as the experiencer of the emotion restrict the assignment of a subject position pronoun. The assignment of the pronoun is unconstrained when it is in the object position. For

³We consider verbs from verb classes 31: Psych-Verbs (Verbs of Psychological State), 13: Verbs of Change of Possession, 37: Verbs of Communication as they conceptually align well with conditions required for ambiguity. Verbs within clusters were individually evaluated for appropriateness for templates by the authors.

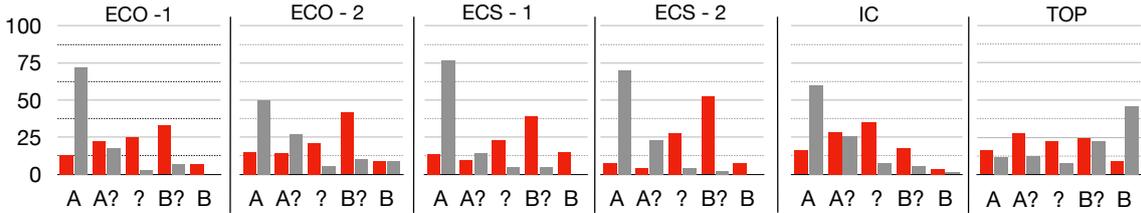


Figure 1: Human annotation of ambiguous (■) and unambiguous (■) sentences. We abbreviate human annotations by whether they identified noun A or B and whether they annotate definitely or likely (marked with ?). For example A? indicates, noun A, likely. The ground truth for unambiguous instances, from left to right, corresponds to A, A, A, A, A, B. Annotators read unambiguous examples as intended, and reduce their confidence on ambiguous examples.

example, in Table 1, row 6, “liked” is ambiguous when a pronoun is placed in the object position (but not in the subject position). We instantiate variants with names (rows 5,6) and entities (rows 7,8).

Implicit Causality (IC) Caramazza et al. (1977) hypothesize that implicit causality of a verb can determine the direction of pronoun assignment. For example, in Table 1 row 9, the phrase “wanted to apologize” establishes a cause for why “Emily called,” so the pronoun is constrained to the subject of “call”. Conversely, in row 10, the phrase “is leaving soon” fails to create such a relationship, leaving the pronoun ambiguous. For these templates (rows 9,10), we vary the names of the entities involved, and pair verbs (i.e. called) with constructed phrases that imply causality (i.e. apologizing), manually.

Transfer of Possession (TOP) Rohde and Kehler (2014) suggests that in transfer-of-possession contexts such as, “John passed the comic to Bill. *He...*”, the pronoun is equally likely to refer back to subject and non-subject. We draw upon this observation, and create a template around verbs that involve source-goal possession transfers. We distill the example to one sentence and pair the transfer event with a reason. For example, in Table 1 row 11, the phrase “asked for one” constrains the pronoun to be the receiver of “bake”. Conversely, before having lunch provides no such constraint, because either the receiver or giver could have “had lunch” before the event. Templates vary the names, verbs, objects, reasons, and preposition (rows 11,12).

2.2 Filling Template Slots

For each template, we construct a list of appropriate verb phrases, reasons (for IC and TOP templates), and shared list of gendered names and noun-phrases. Verb phrases were constructed by manually inspecting VerbNet classes. To control for name bias, we randomly sample names

from popular name lists⁴ from the last 50 years, and reuse gendered noun-phrase lists from Winobias (Zhao et al., 2018). Excluding name and noun-phrase variations, templates have 114, 45, 81, 82 instances for ECO, ECS, IC, and TOP, respectively.

3 Human Judgements

The templates used to create AMBICOREF generalize several psycholinguistic studies using lexical resources. Next, we verify that humans perceive ambiguity in these examples in the intended ways. We extract a subset of data for each template and ask Amazon Mechanical Turk workers which person a pronoun refers to (marked as A or B in Table 1) and assign confidence (*definitely*, or *likely*). Annotators were also allowed to mark the referent as entirely *ambiguous*. One sentence was sampled for each template and verb slot, uniformly at random. We collected 3 annotations per instance.⁵ See Appendix A for details on the collection of human judgements.

Figure 1 summarizes our results. Human judgements for unambiguous templates favor the intended coreference decision. For unambiguous ECO, ECS, IC, TOP instances, the intended reading is selected as likely or definitely, 83.2%, 91.9%, and 85.8%, 68.3% of the time, respectively. For ambiguous instances, annotations display a substantial shift toward ambiguity. As shown in previous work, humans display substantial disagreement on ambiguous instances (Poesio et al., 2019). This is reflected in many templates, such as TOP, where humans produce almost uniform responses.

⁴<https://www.ssa.gov/oact/babynames/decades/>

⁵In ambiguous cases, annotators do not reliably annotate a particular category, but often guess with low confidence. As such, we do not only report a majority opinion per instance, but instead simply report multiple annotations per sentence to see overall trends.

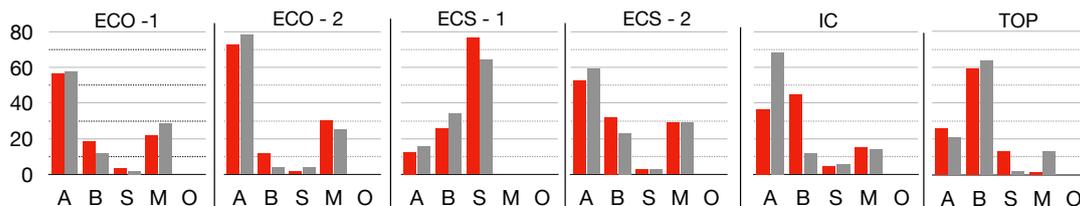


Figure 2: Percentage of ambiguous (■) and unambiguous (■) instances that fall into each of our five cases for the SpanBERT-based model across all templates. All other models show negligible shifts (red and grey distributions are almost identical). The ground truth for unambiguous instances, from left to right, corresponds to A, A, A, A, A, B.

4 Model Evaluation

We now examine if we can detect sensitivity to ambiguity in existing coreference resolution models by evaluating on AMBICOREF. We experiment⁶ with five representative models: NeuralCoref 4.0 model from Hugging Face⁷, SpanBERT (Joshi et al., 2020) representation within the independent framework for end-to-end coreference (Joshi et al., 2019), and the three models in Stanford CoreNLP (Manning et al., 2014): deterministic (Lee et al., 2013), statistical (Clark and Manning, 2015) and neural mention ranking (Clark and Manning, 2016). All models were trained on the CoNLL 2012 dataset (Pradhan et al., 2012).

Here, we evaluate the model’s final predictions, not their distribution over possible choices. The reason is two-fold: (1) not all models produce a distribution and (2) initial analysis revealed that the models are miscalibrated, as in other settings (De-sai and Durrett, 2020; Jiang et al., 2021), making it unreliable to interpret their output scores directly.

4.1 Setup

In this section, we ask, are there differences between how models process similar unambiguous and ambiguous examples? As our examples are synthetically generated, we use the unambiguous examples as a form of control. If a model is unable to link the pronoun with the correct noun on unambiguous examples for at least 40% of examples, we omit that template during evaluation.

We analyze model behavior by breaking it into cases that cover all possible cluster assignments for the pronoun in a single sentence. We compute the percentage of time a model outputs a cluster with:

- case **A**: the pronoun and noun A
- case **B**: the pronoun and noun B
- case **S**: the pronoun as a singleton

⁶Roughly one week of continuous Colab GPU compute.

⁷<https://github.com/huggingface/neuralcoref>

Model	Mean EMD %	Templates
SpanBERT	11.7	5
CoreNLP Neural	3.5	5
NeuralCoref 4.0	4.0	5
CoreNLP Statistical	1.2	3
CoreNLP Deterministic	0.6	5

Table 2: Mean Earth Mover’s Distance between matched ambiguous and unambiguous case distributions and the number of templates where models get at least 40% of unambiguous cases correct.

- case **M**: the pronoun, noun A, and noun B
- case **O**: the pronoun and any other span

For example, Figure 2 contains SpanBERT’s output distribution over these cases for each template. For each such distribution where the model’s performance is above threshold, we compare ambiguous (red bar) and unambiguous (grey bar) distributions using Earth Mover’s Distance (EMD) (Pele and Werman, 2009)⁸. Table 2 reports the number of templates above threshold, and their mean EMD.

4.2 Results

Overall, most models we evaluated show essentially no change in output distribution over cases between ambiguous and unambiguous templates, as evidenced by near zero EMD. Most models are evaluated on five of six templates, but TOP is often excluded, representing a hard unambiguous case for most systems in its own right.

Of the models we evaluated, only SpanBERT shows significant deviation in behavior with ambiguous inputs. Figure 2 breaks down SpanBERT’s performance on each template. While average EMD is higher than for other models, it still largely doesn’t change predictions. When deci-

⁸Earth Mover’s distances represent the amount of probability mass required to match two probability distributions. Hence, they help us compare distributions for ambiguous and unambiguous instances in a more interpretable way, than other possible measures like KL divergence.

sions change, often the pronoun is linked with the other noun. For example, in ambiguous cases of ECO-1, SpanBERT reduces merged outputs, and instead links the pronoun with noun B more frequently. In ambiguous cases, other models largely link the first noun-phrase (A) to the pronoun.

5 Discussion and Conclusion

Overall, our results suggest that model behavior significantly deviates from how humans treat ambiguous coreference. We lend more evidence that models miss aspects of how people understand language, especially in discourse (Upadhye et al., 2020). The reason is likely in part that models are trained on resources which do not account for distributions in judgments. As a result, models do not have well-defined behavior when ambiguity arises and are poorly calibrated.

Training models with finer-grained coreference judgments could allow models to better align with human behavior. Techniques to improve model calibration could also be effective, allowing models to abstain or seek clarification when ambiguity arises. We hope that AMBICOREF can serve as a diagnostic set for future modeling approaches in evaluating their sensitivity to instances of ambiguity in language.

6 Limitations

Our study focuses entirely on coreference in the English language with models trained in high-resource settings. Furthermore, the cases of ambiguity we identify are English-specific and the names we insert into templates are popular American names. It is an open question as to how our results generalize to low-resource non-American-English settings.

The language we use to evaluate models is templatic. While we make an effort to account for unnatural data, by only evaluating templates models do well at, models struggle to completely solve all our unambiguous examples. This presents a challenge for future model builders. On the other hand, our templates may not reflect a particular real world distribution that models will be tested on.

Acknowledgements

We thank Chris Callison-Burch and the PennNLP group for their helpful comments on this work.

References

- Laura Aina and Tal Linzen. 2021. [The language model understood the prompt was ambiguous: Probing syntactic uncertainty through generation](#). In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 42–57, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Alfonso Caramazza, Ellen Grober, Catherine Garvey, and Jack Yates. 1977. [Comprehension of anaphoric pronouns](#). *Journal of Verbal Learning and Verbal Behavior*, 16(5):601–609.
- Kevin Clark and Christopher D Manning. 2015. [Entity-centric coreference resolution with model stacking](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415.
- Kevin Clark and Christopher D. Manning. 2016. [Deep reinforcement learning for mention-ranking coreference models](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.
- Shrey Desai and Greg Durrett. 2020. [Calibration of pre-trained transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.
- Anca Dumitrache, Lora Aroyo, and Chris Welty. 2019. [A crowdsourced frame disambiguation corpus with ambiguity](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2164–2170, Minneapolis, Minnesota. Association for Computational Linguistics.
- Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. 2021. [Abg-coqa: Clarifying ambiguity in conversational question answering](#). In *3rd Conference on Automated Knowledge Base Construction*.
- Sarah L Haywood, Martin J Pickering, and Holly P Branigan. 2005. [Do speakers avoid ambiguities during dialogue?](#) *Psychological Science*, 16(5):362–366.
- Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. [How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering](#). *Transactions of the Association for Computational Linguistics*, 9:962–977.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. [SpanBERT: Improving pre-training by representing and predicting spans](#). *Transactions of the Association for Computational Linguistics*, 8:64–77.

- Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. [BERT for coreference resolution: Baselines and analysis](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.
- Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. [Deterministic coreference resolution based on entity-centric, precision-ranked rules](#). *Computational Linguistics*, 39(4):885–916.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. [The Stanford CoreNLP natural language processing toolkit](#). In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. [AmbigQA: Answering ambiguous open-domain questions](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.
- Ofir Pele and Michael Werman. 2009. [Fast and robust earth mover’s distances](#). In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467. IEEE.
- Steven T. Piantadosi, Harry Tily, and Edward Gibson. 2012. [The communicative function of ambiguity in language](#). *Cognition*, 122(3):280–291.
- Massimo Poesio and Ron Artstein. 2005. [The reliability of anaphoric annotation, reconsidered: Taking ambiguity into account](#). In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 76–83, Ann Arbor, Michigan. Association for Computational Linguistics.
- Massimo Poesio, Jon Chamberlain, Silviu Paun, Juntao Yu, Alexandra Uma, and Udo Kruschwitz. 2019. [A crowdsourced corpus of multiple judgments and disagreement on anaphoric interpretation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1778–1789, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. [CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes](#). In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.
- H. Rohde and A. Kehler. 2014. [Grammatical and information-structural influences on pronoun production](#). *Language, Cognition and Neuroscience*, 29(8):912–927.
- Karin Kipper Schuler. 2005. [VerbNet: A broad-coverage, comprehensive verb lexicon](#). University of Pennsylvania.
- Hinrich Schutze. 1995. [Ambiguity in language learning: computational and cognitive models](#). Stanford University.
- F. Springston. 1976. Verb-derived constraints in the comprehension of anaphoric pronouns. Paper presented at the Eastern Psychological Association (1976).
- Shiva Upadhye, Leon Bergen, and Andrew Kehler. 2020. [Predicting reference: What do language models learn about discourse models?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 977–982, Online. Association for Computational Linguistics.
- Thomas Wasow, Amy Perfors, and David Beaver. 2003. [The puzzle of ambiguity](#).
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. [Ontonotes release 5.0 ldc2013t19](#). *Linguistic Data Consortium, Philadelphia, PA*, 23.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. [Gender bias in coreference resolution: Evaluation and debiasing methods](#).

A Human Judgement Tests

In all our human judgement tests, we required annotators to be based primarily in English-speaking countries: the US, UK, Canada or Australia. Further, annotators needed to have at least 1000 approved HITs and a HIT acceptance rate of at least 98%. Each HIT contained 10 examples, and we estimated the completion time for each HIT to be ~5 minutes, so we paid \$1.25 per HIT, for a pay rate of \$15 per hour.

For our human judgement tests, we first ran a qualification round to ensure high-quality annotations. In this round, we asked annotators to complete a single HIT with 10 examples (5 unambiguous, 5 ambiguous randomly ordered). For each annotator who completed this round, we compute their accuracy by measuring how often they responded with the correct referent (or the ambiguous label), while ignoring their confidence. The top 100 annotators were qualified to work on the main task.

For our main task, we had 625 sentences labeled in total, with 3 assignments per sentence. Each annotator was asked to work on not more than 5 HITs, so that we get a diverse set of judgements. Similar to the qualification round, we asked each annotator to label the referent (or the ambiguous label) and their confidence. We group the annotations into 5 options: (Noun A, definitely), (Noun A, likely), Ambiguous, (Noun B, likely), and (Noun B, definitely). The human judgement labels for each template type were aggregated by computing the fraction of annotations in each of the five options. Our annotation interface for the main task is shown in Figure 3.

Instructions - [click to hide](#)

You have been qualified for the final task on coreference resolution.

Please work on **at most 5 HITs** for this task (ideally 3 to 4 HITs). We will not be able to compensate you for your work on more than 5 HITs.

You will be presented 10 sentences.

- Each sentence contains at least two nouns, and a nominative third-person singular pronoun (he/she).
- If three nouns appear in the sentence, please only consider the first two nouns as possible candidates.
- Following each sentence, there will be a question asking you which noun the specified pronoun refers to, or if you find the pronoun ambiguous.

For example, **Chloe told Emma that she was sad. Who was sad?**

- *Definitely Chloe* - I feel confident that "she" refers to "Chloe";
- *Probably Chloe* - I think "she" refers to "Chloe" as opposed to "Emma" but I feel unsure;
- *Mostly ambiguous* - I find it completely ambiguous whether "she" refers to "Chloe" or "Emma";
- *Probably Emma* - I think "she" refers to "Emma" as opposed to "Chloe" but I feel unsure;
- *Definitely Chloe* - I feel confident that "she" refers to "Emma";

Different people may have different judgments and tolerance for ambiguity, so please feel free to use your intuitive judgments.

Are you a native speaker of English?

Yes

No

1. Matthew bought Joshua a pizza after he asked for more food.
Who asked for more food?

Definitely Matthew

Probably Matthew

Mostly ambiguous

Probably Joshua

Definitely Joshua

2. Kimberly told the aunt that she chatted with the granddaughter.
Who chatted with the granddaughter?

Definitely Kimberly

Probably Kimberly

Mostly ambiguous

Probably the aunt

Definitely the aunt

Figure 3: Annotation interface for the human judgement tests, presented in section 2.

Improving Unsupervised Out-of-domain Detection through Pseudo Labeling and Learning

Byoungchan Lee¹ Jaesik Kim^{1,2} Junekyu Park^{1,3} Kyung-Ah Sohn^{1,*}

Ajou University¹, University of Pennsylvania², Superb AI³

{qudgks96, kasohn}@ajou.ac.kr

Jaesik.Kim@pennteam.upenn.edu

idbluefish@gmail.com

Abstract

Unsupervised out-of-domain (OOD) detection is a task aimed at discriminating whether given samples are from the in-domain or not, without the categorical labels of in-domain instances. Unlike supervised OOD, as there are no labels for training a classifier, previous works on unsupervised OOD detection adopted the one-class classification (OCC) approach, assuming that the training samples come from a single domain. However, in-domain instances in many real world applications can have a heterogeneous distribution (i.e., across multiple domains or multiple classes). In this case, OCC methods have difficulty in reflecting the categorical information of the domain properly. To tackle this issue, we propose a two-stage framework that leverages the latent categorical information to improve representation learning for textual OOD detection. In the first stage, we train a transformer-based sentence encoder for *pseudo labeling* by contrastive loss and cluster loss. The second stage is *pseudo label learning* in which the model is re-trained with pseudo-labels obtained in the first stage. The empirical results on the three datasets show that our two-stage framework significantly outperforms baseline models in more challenging scenarios.

1 Introduction

Deep neural networks show outstanding performance on benchmark datasets that have the same training and test domains. However, once the model is deployed to the real world, it can face out-of-domain (OOD) instances that make the model predict unreliable outcomes related to AI safety issues (Amodei et al., 2016; Hendrycks and Gimpel, 2017). For this reason, the OOD detection task aims to discriminate whether given instances are from in-domain (IND) or not. One of the main OOD detection approaches is to use a classifier that predicts the labels of IND samples, based on the

fact that the classifier has lower confidence in predicting the OOD samples than the IND (Hendrycks and Gimpel, 2017; Lee et al., 2018).

As this approach targets only supervised tasks that require IND labels to train the classifier, it has a limitation on unsupervised tasks. To overcome this problem, recent studies have proposed unsupervised OOD detection (or the without label scenario) that can be utilized in a more general use case (Xu et al., 2021; Jin et al., 2022). This setting can be regarded as one-class classification (OCC) because it uses only IND instances without labels and aims to distinguish novel samples from IND instances. Within this background, unsupervised OOD detection methods introduce OCC approaches such as OC-SVM and SVDD (Xu et al., 2021; Sohn et al., 2020). Meanwhile, self-supervision based models exploit a novel property named inlier priority (Wang et al., 2019) by using pseudo labels that are generated for surrogate supervision (e.g., rotation transformation (Hendrycks et al., 2019)).

In the field of natural language processing (NLP), this approach is adopted in combination with self-supervised methods of pretrained-language models (Manolache et al., 2021a). However, there are tasks where the categorical labels of training data are not available, while the IND has categorical distributions (e.g., summarization, topic modeling). OCC methods can suffer in this scenario (Jin et al., 2022; Park et al., 2021) due to the absence of IND labels, because it is difficult for the model to explicitly reflect the latent categorical distribution.

To tackle this problem, we propose a two-stage framework for textual out-of-domain detection that embeds similar INDs close together by considering latent categorical information of heterogeneous IND instances without labels, and then detects OOD instances based on the learned embedding space. To achieve this, in the first stage, we conduct *pseudo labeling* of training samples by using an unsupervised clustering method combined with contrastive

* indicates corresponding author.

loss. Next, the model from the first stage is refined by the given pseudo labels, which we call *pseudo label learning* (PLL). We find that this second stage of PLL greatly improves the representation learning for IND instances. After training is done, the inference step uses a confidence score function that measures the likelihood of whether an input is IND or OOD.

Our experimental results on three real-world datasets with the pre-trained RoBERTa (Liu et al., 2019) as a base architecture show that the proposed framework substantially outperforms the baseline models in various settings. In addition, we conduct embedding space analysis to confirm the effectiveness of PLL and show that it learns a more suitable representation for OOD detection by increasing inter-cluster distance significantly, which makes OOD samples more distinct from the clusters.

In summary, our main contributions are as follows:

- We propose a new framework for text OOD detection that effectively utilizes latent categorical information of IND through two successive steps of clustering for obtaining pseudo labels and then re-learning the pseudo labels for better representation learning.
- We provide a systematic analysis of the result by dividing OOD instances into near-OOD and far OOD depending on how close they are to IND samples. Our method works especially well on near-OOD, a more challenging scenario, in comparison with other methods. We also analyze the embedding space to confirm the effectiveness of our PLL approach.
- We empirically demonstrate that our proposed method is highly effective in multi-domain settings where the IND distribution has high variability, by increasing the inter-cluster distances and placing OOD out of detection boundaries of each cluster.

2 Related Work

Out of Domain Detection. OOD detection aims to distinguish OOD instances from IND to prevent a model trained for IND from making wrong predictions in the real applications. One of the main approaches is to rely on a classifier for IND labels, supposing that the softmax probability value of the IND will be larger than OOD (Hendrycks and

Gimpel, 2017). Furthermore, (Liang et al., 2018; Lee et al., 2018; Hsu et al., 2020) improve this method by adding perturbation to the inputs, which further increases the softmax probability of IND. In the NLP field, Hendrycks et al., 2020 find out that transformer-based models are more effective than convolutional neural networks (LeCun et al., 1998) or long short-term memory (Hochreiter and Schmidhuber, 1997) based models in detecting textual OODs. To improve OOD detection performance for the models, (Zhou et al., 2021) utilize supervised contrastive loss that creates a more compact representation. However, these approaches cannot be used without IND labels.

Unsupervised Out of Domain Detection. Self-supervised methods can handle this issue by using augmentation techniques (Sehwag et al., 2020; Wang et al., 2019). Manolache et al., 2021a adopt this approach by utilizing the training scheme introduced in ELECTRA (Clark et al., 2019). They use a generator to replace random masked tokens in the input and train a discriminator to predict whether each token is replaced by the generator or not. Xu et al., 2021 focus on the findings that different layers of BERT Devlin et al., 2019 can capture different linguistic information. They compute the Mahalanobis distance using the embeddings in each layer and construct a new vector consisting of the distance values across all the layers. This new feature vector is used as input to OCC-based OOD detection methods. However, these models are difficult to perform well when INDs are in heterogeneous domains (or multiple classes), because they do not explicitly reflect the multimodal IND distribution. Cluster-based approaches can help alleviate this problem since they assume that the IND has a latent class distribution in its feature space. Jin et al., 2022 introduce a clustering method for representation learning to reflect categorical distributions on the embedding space. Our approach is motivated by (Jin et al., 2022), but our method generates pseudo labels and uses them explicitly to reinforce this categorical information, which greatly improves the performance.

3 Proposed Framework

In this section, we describe our two-stage framework for unsupervised OOD detection. First, the purpose of stage 1 is to generate pseudo labels that include categorical information of IND samples. We train a sentence encoder based on a pre-trained

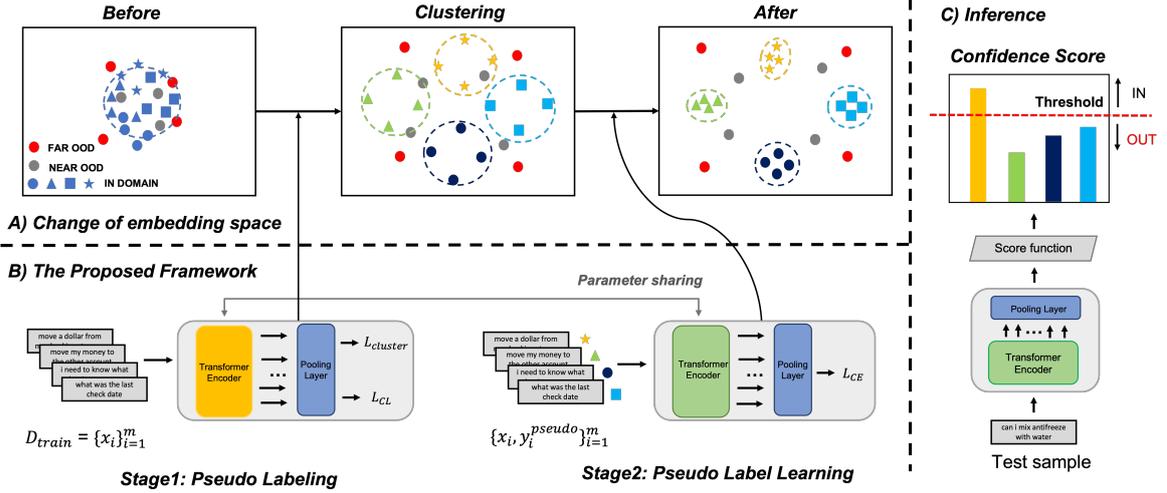


Figure 1: The overall framework of the proposed method. A) illustrates the change of representations in the embedding space during two stages of the training phase. B) shows the proposed framework. It consists of two stages of *pseudo labeling* (stage1) and *pseudo label learning* (stage2). C) shows the inference phase to detect OOD samples using a confidence score function. D_{train} is a training dataset that contains only IND samples x_i , without labels. y_i^{pseudo} is the pseudo label for x_i generated in stage 1.

transformer for *pseudo labeling* of IND training samples using contrastive loss and cluster loss. After then, in stage 2, we perform *pseudo label learning*, designed to explicitly utilize the pseudo labels for reinforcing the categorical information through a classification task. Finally, we use a scoring function that indicates the confidence of being IND to detect OOD samples at test time. Our proposed framework is illustrated in Figure 1.

3.1 Pseudo Labeling

The *pseudo labeling* stage is designed to generate pseudo labels y_i^{pseudo} for each x_i in $D_{train} = \{x_i\}_{i=1}^M$. To do that, we assume that IND data have K categories that are represented in the latent semantic space. Let μ_k denote the centroid of each cluster k and ψ be a transformer-based sentence encoder:

$$e_i = \psi(x_i).$$

For each sample x_i , we use the Student's t -distribution to compute a soft assignment probability q_{ik} , meaning the probability that the sample i belongs to the cluster k , by the following equation (Van der Maaten and Hinton, 2008):

$$q_{ik} = \frac{(1 + \|e_i - \mu_k\|_2^2)^{-\frac{\alpha+1}{2}}}{\sum_{k=1}^K (1 + \|e_i - \mu_k\|_2^2)^{-\frac{\alpha+1}{2}}},$$

Here, α represents the degrees of freedom of the Student's t -distribution. In this work, we set $\alpha =$

1. The cluster centroids and the soft assignment probability can be refined iteratively by using an auxiliary target distribution proposed by (Xie et al., 2016) as:

$$p_{ik} = \frac{q_{ik}^2 / f_k}{\sum_{k=1}^K q_{ik}^2 / f_k},$$

where $f_k = \sum_{j=1}^M q_{jk}$ is the soft cluster frequency to normalize q_{ik} raised to the second power. This target distribution first sharpens the soft assignment probability q_{jk} by raising it to the second power and then normalizes it by the associated cluster frequency. The soft assignment is optimized based on KL-divergence between $p_i = (p_{i1}, \dots, p_{iK})$ and $q_i = (q_{i1}, \dots, q_{iK})$:

$$l_i^C = KL(p_i || q_i) = \sum_{k=1}^K p_{ik} \log \frac{p_{ik}}{q_{ik}}$$

The clustering objective is then defined as follows:

$$L_{cluster} = \frac{1}{M} \sum_{i=1}^M l_i^C$$

This loss function encourages learning from cluster assignment with high confidence and debiasing imbalanced cluster assignment.

Following (Zhang et al., 2021), we also adopt contrastive learning to improve clustering performance. Contrastive loss scatters the samples while closely embedding samples sharing the same properties. For contrastive learning, we use dropout

mask augmentation which simply feeds the same input to the transformer-based encoder¹ twice (Gao et al., 2021). Using this augmentation method, we construct a positive pair (x_i^0, x_i^1) from the same x_i with different dropout masks. We try to minimize the following contrastive learning loss:

$$l_i^{CL} = -\log \frac{\exp(\text{sim}(z_{i^0}, z_{i^1})/\tau)}{\sum_{j=1}^{2M} \mathbb{I}_j \cdot (\exp(\text{sim}(z_{i^0}, z_j)/\tau))},$$

where $z_i = g(\psi(x_i))$ and g is a network of fully-connected layers. We choose $\text{sim}(\cdot)$ as the dot product between a pair of normalized outputs, i.e., $\text{sim}(z_i, z_j) = z_i^T z_j / (\|z_i\|_2 \|z_j\|_2)$. Then the overall contrastive learning objective is defined as:

$$L_{CL} = \frac{1}{2M} \sum_{i=1}^{2M} l_i^{CL}$$

In summary, the final objective for stage 1 is the following:

$$L_{stage1} = L_{cluster} + \lambda L_{CL} \quad (1)$$

After training the model for pseudo labeling by using the stage1 loss, we assign the pseudo label y_i^{pseudo} for each $x_i \in D_{train}$ using the soft assignment probability.

3.2 Pseudo Label Learning

Contrastive learning is useful for clustering and pseudo labeling because contrastive loss separates samples apart from each other to prevent overlap in the representation space. However, it is not sufficient for OOD detection because OOD samples can be located close to the cluster boundaries as illustrated in Figure 1. Therefore, we introduce *pseudo label learning* (PLL), which allows the text encoder to learn representations that are more suitable for OOD detection. PLL explicitly uses pseudo labels to further separate clusters in the embedding space. Therefore, we fine-tune the model by targeting pseudo labels y_i^{pseudo} using the cross-entropy loss. The loss function in stage 2 is as follows:

$$L_{stage2} = L_{CE} = - \sum_{i=1}^M y_i^{pseudo} \cdot \log(p_i)$$

where p_i is the predicted probability distribution for the pseudo label.

¹Transformer already has dropout mask in fully-connected layer and attention probabilities

3.3 Confidence score function

Next, we introduce the confidence score function s for OOD detection that uses a classifier in stage 2. The scoring function s aims to map the representations of instances to confidence scores, where higher scores indicate higher confidence for being IND. In the following, we present several options for this scoring function.

Maximum Softmax Probability (MSP). Hendrycks and Gimpel, 2017 suggest the maximum class probability among K training classes in the softmax layer as an OOD indicator. This method has been extensively used as a baseline for OOD detection (Hendrycks et al., 2020; Zhou et al., 2021), which defines the score as:

$$s = 1 - \max\{p_k \mid k = 1, \dots, K\}.$$

Energy Score (Energy). Liu et al., 2020 propose energy based score that theoretically outperforms the softmax based score, which is defined as:

$$s = -\log \sum_j^K (w_j^T h)$$

where w_j is the weight of the j^{th} class in the softmax layer, and h is the input to the softmax layer. A higher s means higher probability density in OOD classes and thus implies lower IND likelihood.

Mahalanobis Distance (Maha). Podolskiy et al., 2021 showed that the distance-based scoring function can outperform other methods in a supervised setting, which is defined as:

$$s = -\min_k (h - \mu_k)^T \Sigma_k^{-1} (h - \mu_k)$$

where μ_k is the mean vector and Σ_k is the covariance matrix of each latent class k . Then, given an instance x during inference, it calculates the confidence score as the minimum Mahalanobis distance among the K classes.

4 Experiments

In this section, we present the experimental setting for the evaluation of the proposed framework. We describe the used datasets and how to construct IND and OOD samples under unsupervised OOD scenarios.

4.1 Dataset

To evaluate the proposed model, we select the following three real world datasets.

Dataset	Ratio	Model	AUROC	AUIN	AUOUT	AUROC	AUIN	AUOUT
			Near OOD			Far OOD		
CLINC150	0.25	DATE	74.30	49.23	88.43	88.03	89.55	84.68
		MDF	79.51	59.43	91.60	91.19	90.54	88.61
		Ours	93.68	86.73	97.39	98.46	98.7	98.11
	0.5	DATE	69.67	67.53	68.23	86.71	93.09	72.37
		MDF	73.81	70.51	72.85	87.81	93.77	75.27
		Ours	89.72	89.71	88.9	97.00	98.56	94.04
	0.75	DATE	66.88	83.22	41.34	86.38	95.03	63.83
		MDF	69.42	85.78	45.83	83.18	93.83	59.05
		Ours	87.21	94.51	71.69	96.52	98.84	91.1
HWU64	0.25	DATE	69.85	44.43	85.77	79.36	59.23	91.00
		MDF	77.19	60.51	88.95	85.15	73.5	93.62
		Ours	85.25	72.25	92.84	91.69	82.97	96.69
	0.5	DATE	64.82	64.04	61.68	79.78	72.68	83.96
		MDF	68.60	70.22	65.37	82.32	77.22	86.55
		Ours	81.43	81.99	79.12	91.39	87.08	94.15
	0.75	DATE	63.03	82.97	34.58	81.55	80.19	79.74
		MDF	66.96	85.84	36.41	83.986	83.77	83.03
		Ours	78.63	91.11	52.88	90.46	89.33	90.76
BANKING77	0.25	DATE	75.36	44.24	90.34	98.41	97.7	98.9
		MDF	70.81	55.62	70.51	99.42	99.09	99.56
		Ours	88.72	77.04	95.31	99.83	99.72	99.82
	0.5	DATE	66.70	61.76	68.84	98.21	98.56	97.78
		MDF	64.73	63.46	63.48	99.14	99.11	98.73
		Ours	78.63	77.34	79.3	99.21	99.38	98.95
	0.75	DATE	60.65	79.25	38.31	97.94	98.83	96.64
		MDF	61.61	81.43	35.66	98.94	98.71	98.45
		Ours	70.34	85.87	46.51	98.57	99.27	97.46

Table 1: OOD detection performance with different IND class ratios (25%, 50%, and 75%) on three datasets, CLINC150, HWU64, and BANKING77. Scores in bold type are the best results. For all of our methods, we report the averaged results using Mahalanobis distance-based score and the number of clusters equal to the number of IND classes due to space limitations. We collected the results for other methods (Xu et al., 2021, Manolache et al., 2021b) by running their released codes.

CLINC150 (Larson et al., 2019) is a dataset designed for OOD detection. The training set contains 15,000 utterances with 10 domains and 150 classes (e.g., travel.timezone, home.reminder, and credit_cards.rewards_balance). This dataset also provides 1,000 OOD samples that are not within any of 150 classes. For evaluation, we use 4,500 IND and 1,000 OOD samples from the test set.

HWU64 (Xingkun Liu and Rieser, 2019) includes 8,954 utterances for 64 intents with 21 domains (e.g., alarm_set, cooking_recipe, and calendar_query). For evaluation, we use 1,076 IND samples from the test set.

BANKING77 (Casanueva et al., 2020) contains 8,622 utterances related to banking with 77 different fine-grained intents in the training set. Despite consisting of a single domain, this dataset is challenging, as it requires fine-grained differentiation between very similar intents. For evaluation, we use 3,080 IND samples from the test set.

4.2 Experimental setting

We carefully design experimental scenarios assuming that training data consist of instances distributed across multiple domains with any category given. Inspired by Zhang et al., 2022, we divide OOD samples into two types: near-OOD and far-OOD. We suppose that the near-OOD samples are distributed in the same domain with the training samples but labeled as different categories, whereas the far-OOD samples are distributed in distinct domains. The proposed scenarios are more challenging because OOD can share characteristics with IND.

For our scenarios, we randomly select a subset of classes in the training data as IND, with IND class ratios of 25%, 50%, and 75% and use the remaining classes as near-OOD. Following (Zhang et al., 2022), we use the OOD samples in the CLINC150 dataset as far-OOD. We split each dataset five times with different random seeds, which are shared

across all the models for a fair comparison.

4.3 Baselines

We compare our method with the following unsupervised OOD detection methods: MDF (Xu et al., 2021) and DATE (Manolache et al., 2021a). **MDF** utilizes full features from all the layers of a pretrain-transformer model and calculates the Mahalanobis distance vector from the layer representations, which is in turn used as input to OC-SVM. In addition, there are additional training stages such as IMLM (In-domain Masked Language Model) and BCAD (Binary Classification with Auxiliary Dataset) before feature extraction. **DATE** is a pseudo label based approach. It uses a self-supervised learning method of ELECTRA that distinguishes whether each token is replaced or not to generate anomaly scores from the loss obtained by pseudo-labeled tokens.

4.4 Evaluation Metric

To evaluate our proposed method, we report three different metrics following (Liang et al., 2018; Xu et al., 2021). The area under the receiver operating characteristic curve (AUROC) depicts the relationship between the true positive rate and the false positive rate. A higher score indicates improved distinction between IND and OOD by the model. The area under the precision-recall curve (AUPR) shows the precision and recall against each other, for IND and OOD testing sentences, denoted by AUIN and AUOUT, respectively.

4.5 Implement details

For a fair comparison, we also select *roberta-base* from Huggingface’s Transformers (Wolf et al., 2020) as a base architecture for the sentence encoder, the same as MDF. In stage 1, we choose $\tau = 0.5$, $\lambda = 10$, and $\alpha = 1$. We use a constant learning rate of $3e-6$ to optimize the sentence encoder and $3e-4$ to optimize $g(\cdot)$ and the liner layer for soft cluster assignment. In stage 2, we set the learning rate to $3e-5$. We use the Adam optimizer (Kingma and Ba, 2015) with a batch size of 128 for both stages. We used the same hyperparameters for all datasets and splits following Manolache et al., 2021a.

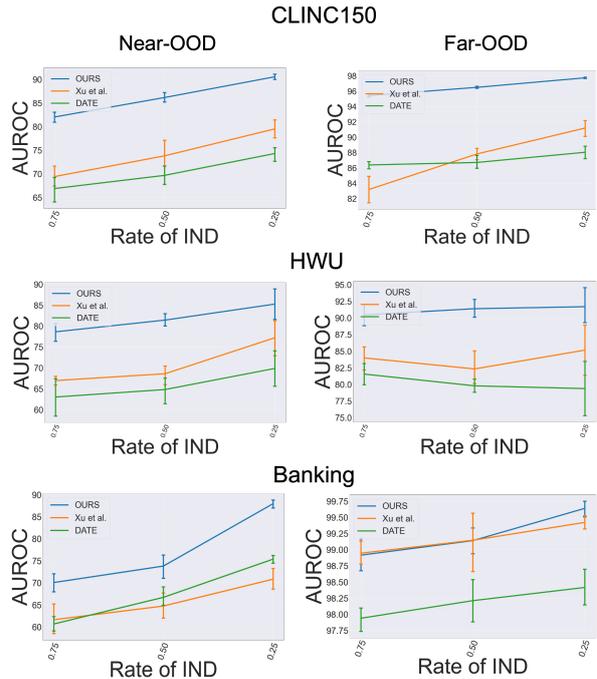


Figure 2: The OOD detection performance with respect to different ratios of IND classes (0.25%, 0.50%, and 0.75%).

5 Result

5.1 Comparisons with baseline methods

Table 1 presents the performance of each method on the three datasets with different IND class ratios (25%, 50%, and 75%). The proposed framework outperforms two baselines, DATE and MDF, by a large margin for the AUROC, AUIN, and AUOUT scores across all three datasets regardless of IND class ratios in the near-OOD and far-OOD setting, except just one case (BANKING77 with the ratio 0.75). In particular, our method greatly improves the performance over other methods on the near-OOD dataset, which represents a more challenging scenario. This shows that the proposed method is robust in multi-domain IND settings regardless of OOD types. In HWU64 dataset that contains more heterogeneous domains than the other two datasets, the OCC-based models, MDF and DATE, appear to have weaknesses in more heterogeneous domain settings, but our method shows good performance. In addition, in the BANKING77 that is the least heterogeneous setting, our method shows similar or higher performance than the other methods as well.

Figure 2 shows the performance with respect to the IND class ratio on three datasets. The performance of all models tend to increase when the ratio

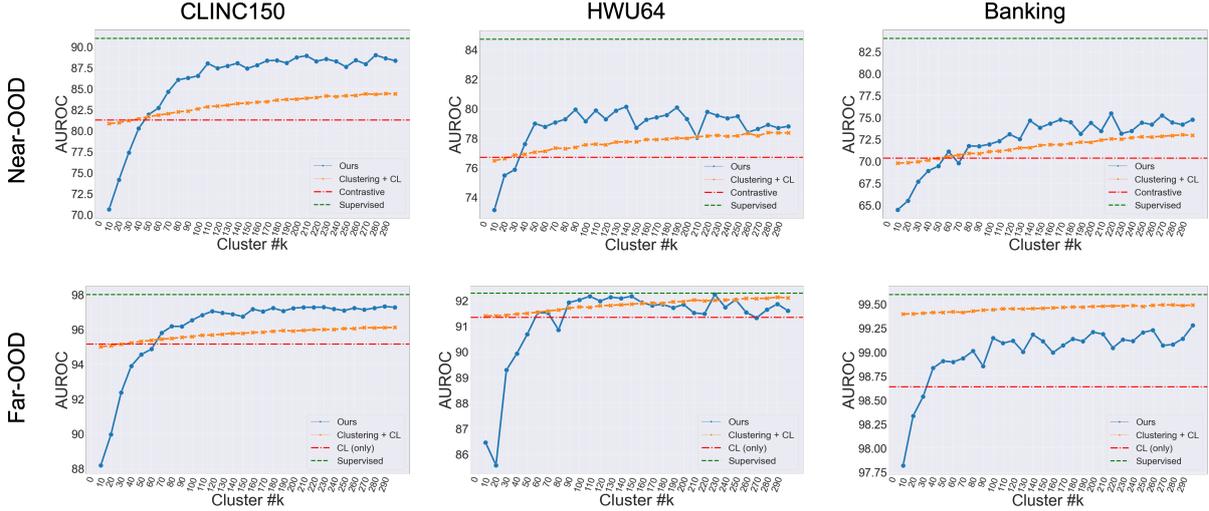


Figure 3: Effect of the number of clusters K with an ablation study. The first row is for the near-OOD setting, the second row is for the far-OOD setting. The columns sequentially correspond to the results on CLINC150, HWU64, and Banking77 datasets for the IND class ratio of 0.75. We compare our proposed method (shown in blue) with the result using only L_{stage1} (orange) and the result using only L_{CL} (red). The green line denoted as *supervised* shows the result when the ground truth labels of IND classes are used during training.

decreases, which is as expected, because fewer IND classes imply less heterogeneous IND distributions and hence easier scenarios. In addition, our method shows more accurate and robust performance with smaller variances (shown as vertical line segments). However, OCC-based methods are more susceptible to randomness during training because they need to bind one characteristic.

5.2 Number of clusters K

The selection of the number of clusters K is an open problem for unsupervised OOD detection since there is no validation OOD set to choose the hyperparameter value. For the results shown in Table 1 and Figure 2, we set K as the number of IND labels given. To measure the influence of K , we plot the change of performance as K is increased in Figure 3. The blue curve indicates our method and the orange line indicates clustering based method with clustering loss and contrastive loss. We find that a larger number of clusters K generally leads to better results for OOD detection. As K increases, the blue curve moves upward to the right, showing that the larger number of clusters allows more detailed consideration of IND samples. It allows more OOD samples to be pushed away from the clusters. In other words, OOD samples that are placed inside a cluster can be located in between as the clusters become more segregated. Therefore, choosing an appropriately large K is ad-

Dataset	Model	Near OOD	Far OOD
CLINC150	MSP	76.51	89
	Energy	78.55	91.53
	Mahalanobis	87.21	96.52
HWE	MSP	69.32	78.34
	Energy	71.32	83.31
	Mahalanobis	78.63	90.46
BANKING77	MSP	58.62	88.07
	Energy	58.87	92.23
	Mahalanobis	70.34	98.57

Table 2: Performance comparison using different confidence score functions. In this result, we set the number of clusters K equal to the number of IND classes in each dataset

vantageous for OOD detection. This is empirically demonstrated in Figure 3.

5.3 Ablation study

As shown in Figure 3, our two-stage approach combining clustering and PLL outperforms clustering-based approaches (shown in orange and red) especially on near-OOD setups. This result reveals that PLL at the second stage utilizes more categorical information than the clustering-based models in stage 1. In far-OOD, our method shows lower performance in only one case (BANKING77) with a very small margin (less than 0.5%). The green line indicates the performance of the oracle model that is supervised by ground truth labels of IND samples during training. In CLINC150, the perfor-

Dataset	Model	max	min	mean	median
CLINC150	Clustering(Only)	4.268	2.464	3.508	3.566
	PLL	18.776	4.46	9.92	8.968
HWU	Clustering(Only)	7.523	3.577	5.439	5.35
	PLL	16.997	5.588	10.082	9.153
BANKING	Clustering(Only)	5.054	3.706	4.428	4.386
	PLL	16.44	5.654	12.1	12.114

Table 3: Intra-cluster variance statistics

mance of our proposing model with high enough k can be almost close to the green line in the near-OOD setting. In addition, our methods show similar performance with the supervised model on the HWU64 dataset in far-OOD settings.

Regarding to the choice for a scoring function, Mahalanobis distance shows the best result regardless of datasets and OOD settings (Table 2). This is because MSP and energy-based methods are based on the predicted class probabilities while pseudo labels can contain errors. In contrast, Mahalanobis distance is based on representations, so it can be more robust to clustering results even when there are miss-labeled instances.

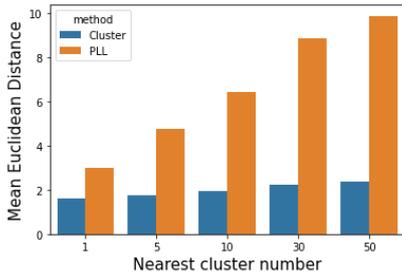


Figure 4: Inter-cluster distance statistics with different numbers of nearest cluster centers.

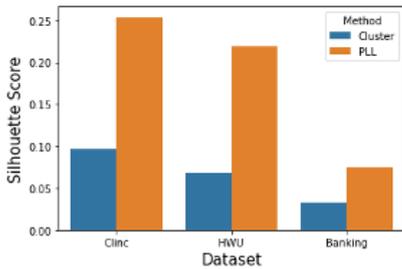


Figure 5: Average silhouette score before and after PLL.

5.4 Analysis of representation space

To investigate why our PLL approach improves OOD detection performance over clustering-based methods, we additionally examine three metrics: intra-cluster variance, inter-cluster distance, and

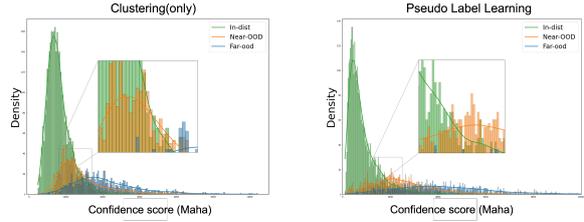


Figure 6: Distributions of the confidence scores before PLL (left) and after PLL (right) on CLINC150 dataset with IND class ratio of 0.75. The confidence score distribution is shown in green for IND, yellow for near-OOD, and blue for far-OOD.

the silhouette score. Table 3 shows the statistics of intra-cluster variance, which can indicate the degree of clustering of the data representations within a cluster. Specifically, we average the distances of the representations of samples with the same pseudo label to the cluster center in the test set as intra-cluster variance, then report min/max/mean/median values on all clusters. And Figure 4 shows the inter-class distances. We average dot product distances between each class center to its C nearest class centers, then average results from all classes as inter-class distance. The x -axis denotes the number of nearest centers C . We find that the intra-cluster variance becomes higher when the clustering is followed by PLL, which means PLL can ruin intra-cluster distribution. However, the inter-cluster distances are also significantly increased through PLL. To find out the balance of the two distances, we compare the silhouette scores before and after PLL in Figure 5, which shows that PLL improves the silhouette scores by a large margin. This implies that PLL can make the clusters far apart from each other and therefore OOD sample to be placed in between the clusters.

5.5 Visualizations

We visualize the confidence score distributions to confirm the effectiveness of our PLL scheme. Figure 6 shows the confidence score distribution on CLINC150 test set with the IND ratio of 0.75. Although the score distributions of near-OOD and IND still overlap when we apply clustering only, after performing PLL, the score distribution for IND shifted to the left, while the distributions of both OOD samples shifted to the other side. Therefore, the score distributions become more discriminable between IND and OOD samples through PLL.

6 Conclusion

In this work, we proposed a two-stage framework for unsupervised OOD detection that effectively utilizes the categorical information of IND instances by pseudo labeling and pseudo label learning. In addition, for a more systematic analysis of OOD performance, we introduced the near-OOD setting, which is a more challenging yet realistic scenario. In most of our experimental settings, our framework outperforms the baseline models with significant margins. We further justify the improvement of the proposed model’s OOD detection performance by analyzing the embedding space with inter or intra-cluster distances and silhouette scores. In future work, we will further investigate how to reduce intra-cluster variations while maintaining inter-cluster distances.

Limitations

The proposed methods show relatively stable performance with respect to the number of clusters (K), but it still has a limitation of choosing the optimal one. In particular, we conduct the experiment by setting the maximum value of K to 300. However, a too large K can degrade the model performance by reducing the number of samples per cluster for classification in stage 2. In addition, since the proposed framework depends on a clustering method, its performance can be limited by the clustering performance. Experiments are only conducted on three intent task datasets due to the near-OOD and the far-OOD settings in heterogeneous domains. We remain those limitations for future works.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No. NRF-2022R1A2C1007434), and also by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. 2019. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960.

Di Jin, Shuyang Gao, Seokhwan Kim, Yang Liu, and Dilek Hakkani-Tür. 2022. Towards textual out-of-domain detection without in-domain labels. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1386–1395.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

- Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.
- Shiyu Liang, Yixuan Li, and R Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*.
- Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Andrei Manolache, Florin Brad, and Elena Burceanu. 2021a. Date: Detecting anomalies in text via self-supervision of transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 267–277.
- Andrei Manolache, Florin Brad, and Elena Burceanu. 2021b. [DATE: Detecting anomalies in text via self-supervision of transformers](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 267–277, Online. Association for Computational Linguistics.
- Junekyu Park, Jeong-Hyeon Moon, Namhyuk Ahn, and Kyung-Ah Sohn. 2021. What is wrong with one-class anomaly detection?
- Alexander Podolskiy, Dmitry Lipin, Andrey Bout, Ekaterina Artemova, and Irina Piontkovskaya. 2021. Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13675–13682.
- Vikash Sehwal, Mung Chiang, and Prateek Mittal. 2020. Ssd: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*.
- Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minh Jin, and Tomas Pfister. 2020. Learning and evaluating representations for deep one-class classification. In *International Conference on Learning Representations*.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Siqi Wang, Yijie Zeng, Xinwang Liu, En Zhu, Jianping Yin, Chuanfu Xu, and Marius Kloft. 2019. Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. *Advances in neural information processing systems*, 32.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.
- Pawel Swietojanski Xingkun Liu, Arash Eshghi and Verena Rieser. 2019. [Benchmarking natural language understanding services for building conversational agents](#). In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, pages xxx–xxx, Ortigia, Siracusa (SR), Italy. Springer.
- Keyang Xu, Tongzheng Ren, Shikun Zhang, Yihao Feng, and Caiming Xiong. 2021. Unsupervised out-of-domain detection via pre-trained transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1052–1061.
- Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen Mckeown, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2021. Supporting clustering with contrastive learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5419–5430.
- Jianguo Zhang, Kazuma Hashimoto, Yao Wan, Zhiwei Liu, Ye Liu, Caiming Xiong, and S Yu Philip. 2022. Are pre-trained transformers robust in intent classification? a missing ingredient in evaluation of out-of-scope intent detection. In *Proceedings of the*

4th Workshop on NLP for Conversational AI, pages 12–20.

Wenxuan Zhou, Fangyu Liu, and Muhao Chen. 2021. Contrastive out-of-distribution detection for pre-trained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111.

How Many Data Samples is an Additional Instruction Worth?

Ravsehaj Singh Puri* Swaroop Mishra* Mihir Parmar Chitta Baral

Arizona State University, Tempe, USA

{rpuri8, srmishr1, mparmar3, chitta}@asu.edu

Abstract

Recently introduced *instruction-paradigm* empowers non-expert users to leverage NLP resources by defining a new task in natural language. Instruction-tuned models have significantly outperformed multitask learning models (without instruction); however they are far from state-of-the-art task-specific models. Conventional approaches to improve model performance via creating datasets with large number of task instances or architectural changes in the model may not be feasible for non-expert users. However, they can write alternate instructions to represent an instruction task. *Is Instruction-augmentation helpful?* We augment a subset of tasks in the expanded version of NATURAL INSTRUCTIONS with additional instructions and find that it significantly improves model performance (up to 35%), especially in the low-data regime. Our results indicate that an additional instruction can be equivalent to ~ 200 data samples on average across tasks.¹

1 Introduction

Large-scale benchmarks such as Imagenet (Russakovsky et al., 2015), SQuAD (Rajpurkar et al., 2018) and architectural development in models such as CNNs (Amari et al., 2003) and transformers (Vaswani et al., 2017) have propelled our progress in deep learning. However, creating high-quality benchmarks by controlling its artifacts (Gururangan et al., 2018; Mishra et al., 2020), developing new models, and training them is hard for non-expert users. Recently introduced *instruction-paradigm* empowers non-expert users, practitioners, and domain experts in other fields to leverage NLP resources (Weller et al., 2020) as they now can describe their tasks in natural language without requiring to create task-specific datasets or developing models². Even though the instruction paradigm

has led to the development of models that significantly outperform multitasking baselines, model performance has remained far behind the supervised learning model trained with task-specific data (Efrat and Levy, 2020; Mishra et al., 2021b).

Non-expert users can write multiple instructions per task each of which covers multiple perspectives spanning over a variety of linguistic features; many of these can be created automatically by replacing certain words with their synonyms without changing the overall semantics of instruction. Can the relatively inexpensive process of instruction augmentation improve the model’s performance in the *instruction-paradigm*, similar to the role data-augmentation has played conventionally in machine learning (Feng et al., 2021)? *Instruction-paradigm* is pivotal where it is expensive or infeasible to gather training data. How effective is instruction augmentation in low-data regimes?

Multi-variant instructions (original + augmented instructions) also can help evaluate the robustness of instruction-following models to respond to variant instructions. This is similar to the model robustness evaluation (Jia et al., 2019) that is done by creating variant data instances. Multi-variant instruction-based setup will also help gauge the true potential of instruction-following systems since in a real-world setting, users can write task instructions in many different ways.

The expanded version of NATURAL INSTRUCTIONS (Mishra et al., 2021b; Wang et al., 2022b)³ provides a rich collection of the diverse category of tasks that covers a variety of reasoning skills, domains, and languages. This constantly evolving benchmark is growing in size with respect to time. We take 426 tasks⁴ and creates variant instructions

³<https://github.com/allenai/natural-instructions>

⁴These were the accepted tasks in the expanded version of NATURAL INSTRUCTIONS in September 2021. The expanded dataset is also known as NATURAL INSTRUCTIONS v2 or SUPER-NATURALINSTRUCTIONS.

*Equal Contribution

¹Code and dataset is available at <https://github.com/Ravsehajsinghpuri/Multi-Variant-Instructions>

²Related work is presented in App. A

for each task. In NATURAL INSTRUCTIONS, the number of instances was limited to 6500 to reduce massive data imbalance, we leverage the remaining instances of source datasets in constructing instances of our variant instruction tasks. We experiment with 3 types of learning scenarios (i) task-specific (TS), (ii) multi-task (MT), and (iii) cross-task (CT) and observe that instruction augmented models outperform their single-instruction counterparts by 17%, 11%, and 11%, respectively when averaged over all experiments across the evaluation tasks. Interestingly, instruction augmentation is more effective on the low-data regime (average across 1%, 5%, and 10% data) as we see a performance gain of 26%, 16%, and 11% in TS, MT, and CT settings, respectively. We also quantify the contribution of each of the additional instructions and find that an additional instruction can be equivalent to ~ 200 data samples on average across tasks.

2 Multi-Variant Instruction Dataset

We construct a Multi-Variant Instruction dataset on top of various tasks in NATURAL INSTRUCTIONS. In total, our dataset has 426 different NLP tasks; each of which contains multi-variant instructions.

2.1 Variant Instruction Task

An instruction task in NATURAL INSTRUCTIONS contains the definition of the task, positive examples, negative examples, and instances. Figure 1 shows the schematic representation of variant instruction tasks where the blue boxes show the parts that differentiate variant instruction tasks from their original counterparts in NATURAL INSTRUCTIONS. While constructing a variant instruction task, we alter the definition and instances of the instruction task.

Parameter	Value
Avg. # of variants per task	4.59
Avg. # of instances per task	9510.64
Avg. # of positive examples per task	3.15
Avg. # of negative examples per task	2.30

Table 1: Multi-Variant Instructions dataset statistics

2.2 Dataset Creation Process

Computer Science graduate students who participated in the data creation process are asked to create as many variant instruction tasks as possible. They are instructed to change the definition

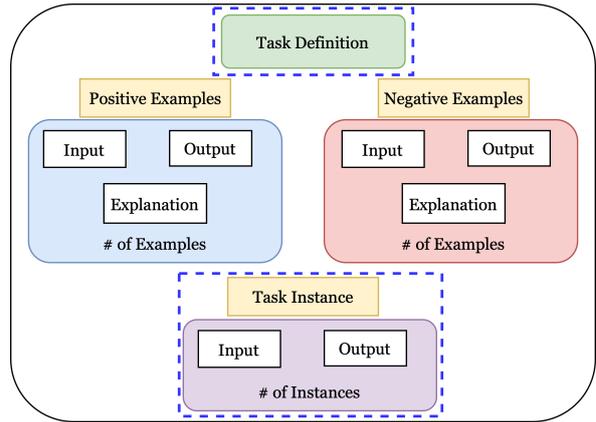


Figure 1: Schematic representation of instructional prompts (Wang et al., 2022b) - Dotted blue box represents entities that are changed in constructing variant instruction task.

(without changing the semantic meaning of the definition in the original task) and instances (by random sampling from the set of instances in the source dataset which is not part of instruction tasks in NATURAL INSTRUCTIONS). They are allowed to use automated tools such as Semantic Control (Ross et al., 2021), Text Style Transfer (Reif et al., 2021), NL-Augmenter (Dhole et al., 2021). Sometimes, the participants create variant instruction tasks manually. Table 5 and Table 6 in App. B illustrates examples of alternate definitions across variant instructions created for our dataset.

2.3 Dataset Properties and Statistics

Table 1 shows the statistics of our meta-dataset. Note that, variant instruction tasks contain all instances from NATURAL INSTRUCTIONS, so the average number of instances per task is higher than 6500 (which is a constraint in NATURAL INSTRUCTIONS). We describe various attributes of our dataset in the following.

2.3.1 Semantic Textual Similarity

Semantic Textual Similarity (STS) should be high between original instruction and augmented instructions as they represent the same task. We compute the pair-wise STS score between definitions of original instruction and variant instructions. Figure 2 shows the mean and SD of STS score between original instruction and its variants across 426 tasks. More detail is presented in App. C.

Analysis of dataset properties From all dataset properties, we can observe that STS score is higher for almost all the tasks. This indicates that all aug-

Task ID	Task Name	Task Category	# of Variants
task010	winogrande_answer_generation	Answer Generation	8
task011	winogrande_question_modification_object	Text Modification	8
task012	winogrande_question_modification_person	Text Modification	8
task017	qasc_question_generation	Question Generation	8
task018	qasc_answer_generation	Answer Generation	8
task020	essential_terms_answering_incomplete_questions	Classification	8
task028	multirc_correct_answer_single_sentence	Answer Generation	3
task058	babi_t1_single_supporting_fact_answer_generation	Answer Generation	5

Table 2: Number of variant instructions for 8 different tasks

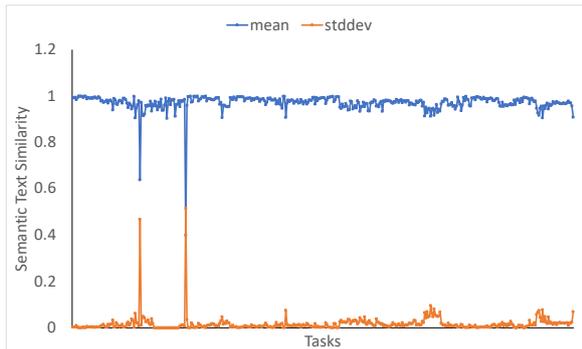


Figure 2: Semantic text similarity between original instruction and its variants.

mented variants are semantically similar to the original instruction. Moreover, we can see a significant variation in terms of word dissimilarity and length of definitions (see App. C). From this, we can conclude that the variants created in our meta-dataset for each task have sufficient variations in terms of words and length yet sustain semantic similarity with original instruction.

3 Experimental Setup

3.1 Models

BART-base (Lewis et al., 2019) and T5-base (Rafel et al., 2020) models are used with default hyperparameters from Huggingface (Wolf et al., 2019) to perform experiments. We use Single Instruction (SI) learning as baseline where only original instruction is used to fine-tune the model. We propose Multi-Variant Instruction (MVI) learning where variants are used to fine-tune models. We use the same number of instances for both original and variant instruction learning to accurately gauge the importance of additional instructions.

3.2 Experiments

We perform three experiments: (1) Task-Specific, (2) Multi-Task, and (3) Cross-Task. All experiments are performed using 1%, 5%, 10%, 50% and 100% instances from the task for fine-tuning. Here, we divide instances into train, test and dev splits by randomly sampling in the ratio 70%, 20% and 10%, respectively. Evaluation is performed on the test set of original instructions. As SI is dependent on NATURAL INSTRUCTIONS which has exactly one instruction per task, this limits our experiments to use only one instruction in the SI setting while comparing it with MVI which has multiple variant instructions.

Task-Specific Here, we fine-tune the baseline and our model on one task and evaluate on the same task. We have performed task-specific learning on 3 different tasks - winogrande_answer_generation, winogrande_question_modification_person, and qasc_answer_generation. In addition, we also analyze two different tasks in other task categories like tweetqa_question_generation and odd-man-out_classification_no_category for generation and classification tasks respectively.

Multi-Task To perform multi-task learning, we use 8 different tasks spanning across 4 different categories. Table 2 shows the different number of variant instructions for 8 tasks and their categories. In this setting, we fine-tune the baseline and our model on all 8 tasks combined and evaluate on each task. However, we use only two positive and two negative examples to satisfy the maximum token limit of the BART-base.

Cross-Task Here, we fine-tune the model on a set of tasks and evaluate on a different set of tasks. Here, we use 274 different tasks for training by sampling 10% instances from each task and evaluate on a set of 8 tasks which are the same as in the

multi-task setup. In addition to sampling instances, we also sampled number of tasks by taking 1%, 5%, 10%, 50%, and 100% tasks. We also investigate the extent of cross-task generalization in low-data regimes; we do this by randomly sampling 1%, 5%, and 10% instances for fine-tuning.

Metric We use the Rouge-L metric (Lin, 2004) for evaluation in all our experiments, following the evaluation in NATURAL INSTRUCTIONS.

4 Results and Analysis

4.1 Experimental Results

Task-Specific Figure 3 shows the comparison between SI and MVI across a different number of instances sampled for fine-tuning. From this, we can observe that MVI outperforms SI by 17% on average. The performance difference between MVI and SI increases to 26% in a low data regime (average performance with 1%, 5%, and 10% instances for fine-tuning). We observe similar results for the additional 2 tasks we have analyzed (present in App. D).

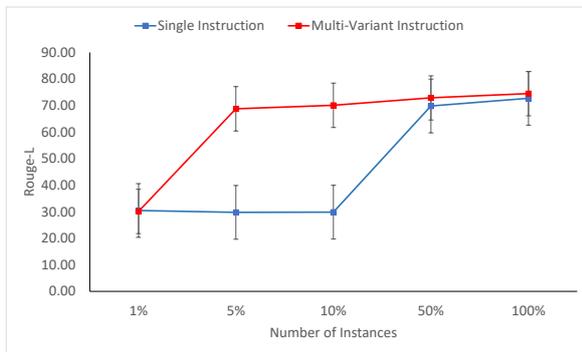


Figure 3: Comparison across SI and MVI learning in task-specific setting; Results are averaged over 3 tasks.

Multi-Task Figure 4 presents the comparison between SI and MVI for multi-task setting. We can observe that MVI outperforms SI by 11% on an average. Moreover, we can see higher improvement in low data regime (~16%). Our model achieves high performance boost (~35%) at 1% instances setting. App. E contains more details.

Cross-Task Figure 5 shows a comparison between SI and MVI for 100% tasks in cross-task setting (see Figure 9 in App. F for other settings). We can observe that MVI outperforms SI by 9% on an average. App. F contains more details.

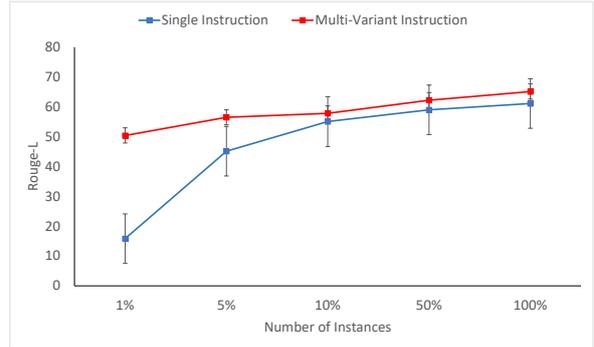


Figure 4: Comparison across SI and MVI learning in multi-task setting by varying number of instances.

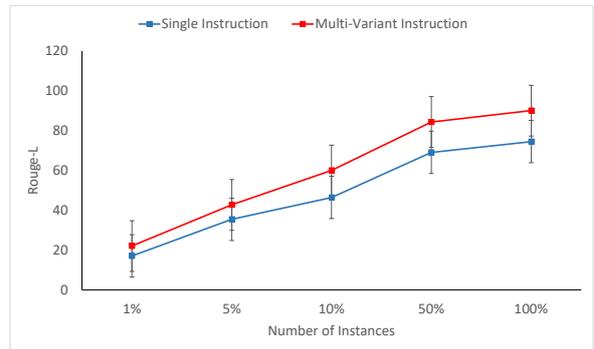


Figure 5: Comparison between SI and MVI learning in cross-task setting by varying number of instances and fixing number of tasks to 100%.

4.2 Analysis

How Many Data Samples is a Variant Instruction Worth? We calculate the contribution of an additional instruction with respect to data samples in the following way: we calculate model performance for BART-base in MVI with 5% instances. We interpolate the model performance plot in SI to find out the percentage of instances needed to match performance in MVI (with 5% instances). We divide the average number of instance difference by average number of instruction variants to get the number that indicates worth of an additional instruction in terms of data samples. Using the above described procedure, we calculate the contribution for additional instruction in all three settings and summarize the results in Table 3. We use MVI performance with 5% instances as the base because a typical instruction-paradigm is designed in a "low-data regime" where non-expert users can teach a task to a model without requiring to create a dataset. However, we also calculated the instruction-equivalence using MVI with 10% instances as the base and report the results in Table

3. On an average across TS, MT and CT, we conclude that an additional variant instruction alone is worth ~ 200 instances.

Base	Task-Specific	Multi-Task	Cross-Task	Average
5%	456.2	94.1	152.3	234.2
10%	460.4	58.2	279.6	266.1

Table 3: Weight of each additional instruction in terms of number of data samples across task-specific, multi-task and cross-task settings.

Equal Data Analysis We believe that each instruction variant is equivalent to ~ 200 data instances. To show this by experiment, we perform equal data analysis and observe that model trained using our approach shows competitive performance compared to single-instruction learning by using only N/V instances where N is the total number of instances in the original task and V is the number of instruction variants for this task. See App. G for more details.

Is Model Robust to Instruction Perturbations?

Here, we introduce 3 perturbations while testing SI and MVI: (1) we perturb the instruction by removing the task definition, (2) we perturb the instruction by changing the order of positive and negative examples by placing positive examples followed by negative different from training setup, and (3) we perturb the instruction by removing all positive and negative examples from the test set. We evaluate the model’s robustness across these perturbations (performance change while the change in instruction) which are excluded from the training data. Here, Table 4 for task-specific setting on T5-base (see Table 11 in App. H for multi-task results). We can clearly observe that our approach is robust to all three instruction perturbations whereas model trained with single-instruction learning is not able to perform equally well on perturbed test sets compared to its original test counterpart. A similar trend is observed in the multi-task setting as well (see App. H).

5 Conclusion

We introduced instruction augmentation to improve existing LMs in terms of improving performance and usability to non-expert users. To this extent, we created multi-variant instructions for 426 NLP tasks. Our experiment results show that instruction augmentation improves model performance in task-specific, multi-task and cross-task learning

# of Instances	SI		Perturbation 1		Perturbation 2		Perturbation 3	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours
1%	0.90	25.21	1.60	18.03	1.02	23.16	5.12	9.71
5%	0.98	75.72	2.18	75.32	1.36	75.50	5.52	74.26
10%	50.88	78.20	20.76	78.07	50.49	78.37	40.31	77.22
50%	76.55	82.16	68.88	82.15	76.50	82.16	75.34	81.92
100%	79.38	83.16	73.51	82.97	79.34	83.12	78.71	82.40

Table 4: Comparison of performance in task-specific setting across SI and MVI learning.

paradigms. We find that instruction augmentation is more effective in low-data regime. Our results further indicate that an additional instruction can be equivalent to ~ 200 instances on an average. We hope our work will bring more attention to developing unconventional techniques (beyond dataset creation and model training) to empower non-expert users to leverage NLP resources and teach a task without having domain knowledge.

Limitations

We use BART-base and T5-base for all our experiments, however, we wish to experiment with different language models in future to show the benefit of our approach. Our analysis includes only tasks in English language, hence, it is important to see if our approach can be extended to non-English tasks as well. We feel that developing diverse instruction augmentation techniques will be pivotal to achieving more improvements as future research.

References

- Shunichi Amari et al. 2003. *The handbook of brain theory and neural networks*. MIT press.
- Xiang Chen, Xin Xie, Ningyu Zhang, Jiahuan Yan, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Adaprompt: Adaptive prompt-based finetuning for relation extraction. *arXiv e-prints*, pages arXiv–2104.
- Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using bart. *arXiv preprint arXiv:2106.01760*.
- Kaustubh D Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, et al. 2021. NI-augmenter: A framework for task-sensitive natural language augmentation. *arXiv preprint arXiv:2112.02721*.
- Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.

- Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.
- Tanmay Gupta, Amita Kamath, Aniruddha Kembhavi, and Derek Hoiem. 2021. Towards general purpose vision systems. *arXiv preprint arXiv:2104.00743*.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.
- Peter Hase and Mohit Bansal. 2021. When can models learn from explanations? a formal framework for understanding the roles of explanation data. *arXiv preprint arXiv:2102.02201*.
- Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142.
- Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Kirby Kuznia, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. **Less is more: Summary of long instructions is better for program synthesis**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4532–4552, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.
- Tevan Le Scao and Alexander M Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2021. Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668*.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.
- Man Luo, Sharad Saxena, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Biotabqa: Instruction learning for biomedical table question answering. *arXiv preprint arXiv:2207.02419*.
- Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.
- Swaroop Mishra, Anjana Arunkumar, Bhavdeep Singh Sachdeva, Chris Bryan, and Chitta Baral. 2020. Dqi: Measuring data quality in nlp. *ArXiv*, abs/2005.00816.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021a. Reframing instructional prompts to gptk’s language. *arXiv preprint arXiv:2109.07830*.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021b. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.
- Swaroop Mishra and Elnaz Nouri. 2022. Help me think: A simple prompting strategy for non-experts to create customized content with models. *arXiv preprint arXiv:2208.08232*.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Preprint*.
- Mihir Parmar, Swaroop Mishra, Mirali Purohit, Man Luo, Murad Mohammad, and Chitta Baral. 2022. **In-BoXBART: Get instructions into biomedical multi-task learning**. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 112–128, Seattle, United States. Association for Computational Linguistics.
- Pruthvi Patel, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. **Is a question decomposition unit all we need?** In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4553–4569, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits

- of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.
- Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.
- Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E Peters, and Matt Gardner. 2021. Tailor: Generating and perturbing text with semantic controls. *arXiv preprint arXiv:2107.07150*.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.
- Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Liwen Wang, Rumei Li, Yang Yan, Yuanmeng Yan, Sirui Wang, Wei Wu, and Weiran Xu. 2022a. Instructioner: A multi-task instruction-based generative framework for few-shot ner. *arXiv preprint arXiv:2203.03903*.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. [Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
- Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E Peters. 2020. Learning from task descriptions. *arXiv preprint arXiv:2011.08115*.
- Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–22.
- Qinyuan Ye and Xiang Ren. 2021. Zero-shot learning by generating task-specific adapters. *arXiv e-prints*, pages arXiv–2101.
- Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.
- Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. *arXiv preprint arXiv:2104.04670*.

A Related Work

Prompt Learning Due to the success of large LMs, research paradigm in ML/DL has been shifted to prompt-based learning to achieve generalization and eliminate the need of creating task-specific models and large scale datasets (Liu et al., 2021). Past attempts have been made using prompt-based learning to solve various tasks including text classification (Yin et al., 2019), Natural Language Inference (NLI) (Schick and Schütze, 2020), Question Answering (QA) (Jiang et al., 2020), Information Extraction (IE) (Chen et al., 2021; Cui et al., 2021) and many more (Liu et al., 2021). Recently, T0 model (Sanh et al., 2021) is proposed which uses prompts to achieve zero-shot generalization across various NLP tasks. We were motivated by the work of Le Scao and Rush (2021) which shows that prompting is often worth 100s of data points on average. Our work instead focuses on instructions that are often different in terms of length, language, and capacity to represent a task (Wang et al., 2022b). Additionally, in contrast to prior works, we focus on the use of automatic methods for instruction augmentation and evaluate its efficacy across low-data to high-data regime in task-specific, multi-task, cross-task setups.

Instruction Learning Efrat and Levy (2020) studies whether existing LMs understands instructions. After that, many works have been proposed to show that models follow language instructions (Hase and Bansal, 2021; Ye and Ren, 2021; Gupta et al., 2021; Zhong et al., 2021). Furthermore, (Weller et al., 2020) has developed a framework that focuses on developing NLP systems that solve new tasks after reading their descriptions. Mishra et al. (2021b) has proposed natural language instructions for cross-task generalization of LMs. Along with that, PromptSource and FLAN (Wei et al., 2021; Sanh et al., 2021) were built for leveraging instructions and achieving zero-shot generalization on unseen tasks. Moreover, Parmar et al. (2022) shows the effectiveness of instructions in multi-task settings for the biomedical domain. Mishra et al. (2021a) discuss the impact of task instruction reframing on model response. Min et al. (2021) introduce a framework to better understand in-context learning. Ouyang et al. (2022) propose the InstructGPT model that is fine-tuned with human feedback to follow instructions. Wang et al. (2022a) has developed instruction-based multi-task

framework for few-shot Named Entity Recognition (NER) tasks. In addition, many approaches have been proposed to improve model performance using instructions (Wu et al., 2022; Lin et al., 2021; Wang et al., 2022b; Luo et al., 2022; Kuznia et al., 2022; Patel et al., 2022; Mishra and Nouri, 2022).

B Example of Variants

Table 5 and Table 6 show the examples of different variants created from the task117_afs_argument_similarity_gun_control and task018_qasc_answer_generation respectively.

C Multi-Variant Dataset Additional Details

C.1 Semantic Textual Similarity

We use en_core_web_md semantic similarity model of SpaCy to compute STS in our experiments. We also calculate STS score between definitions of variants of the same task. At the end, we calculate their mean and Standard Deviation (SD) for each task.

In the plot, the two exception points are task058 (Answer generation task based on babi dataset (Weston et al., 2015)) and task097 (Structured text generation task based on SCAN dataset (Lake and Baroni, 2018)) where the original instructions are very long and the variant task contains a short definition which causes the strong variation in STS. We also discuss the Word-Level Dissimilarity and Length Diversity properties of our dataset below.

C.2 Word-Level Dissimilarity

To show the quality and diversity of variant instructions, we calculate the pair-wise edit distance between the definition of the original instruction and its variant instructions. We also calculate distance between definitions of variant instructions of the same task, further normalize by the highest distance to obtain a dissimilarity score. We compute the mean and SD of these scores for each task and show it in Figure 6.

C.3 Length Diversity

It is necessary to see how task definition lengths vary between original instructions and their variants. To understand this, we compute the percentage difference between the length of the maximum instruction definition and the minimum instruction definition for each task and show it in Figure 7.

Original instruction along with its augmented variant instructions	
ORIGINAL INSTRUCTION	<p>Definition: We would like you to classify each of the following sets of argument pairs (discussing Gun Control) into either SIMILAR or NOT SIMILAR. A pair of arguments is considered SIMILAR if the arguments are about the same FACET (making the same argument), and is considered NOT SIMILAR if they do not have the same FACET. A FACET is a low level issue that often reoccurs in many arguments in support of the author's stance or in attacking the other author's position.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>
VARIANT INSTRUCTION 1	<p>Definition: Each of the following sets of argument pairs (on the topic of Gun Control) should be classified as SIMILAR or NOT SIMILAR. If the arguments are about the same FACET (making the same argument), they are deemed SIMILAR; otherwise, they are NOT SIMILAR. A FACET is a low-level problem that appears frequently in many arguments in favor of the author's position or in opposition to the position of the other author.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>
VARIANT INSTRUCTION 2	<p>Definition: Please classify the following sets of argument pairs (discussing the Gun Control) as SIMILAR or NOT SIMILAR. If the arguments are about the same FACET (making the same argument), they are regarded SIMILAR; if they are not, they are considered NOT SIMILAR. A FACET is a low-level problem that frequently recurs in numerous arguments in favor of the author's position or in opposition to the position of the other author.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>
VARIANT INSTRUCTION 3	<p>Definition: Two arguments are SIMILAR if they are making the same case related to author's position, else they are NOT SIMILAR. Your task is to classify any 2 arguments as SIMILAR or NOT SIMILAR.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>
VARIANT INSTRUCTION 4	<p>Definition: Each of the following sets of argument pairs (discussing the Gun Control) should be classified as SIMILAR or NOT SIMILAR. If the arguments are about the same FACET (making the same argument), they are regarded SIMILAR; otherwise, they are NOT SIMILAR. A FACET is a low-level issue that appears frequently in many arguments in support of the author's position or in opposition to the position of the other author.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>

Table 5: Example of an instruction for a classification task with its variant instructions; these belong to the task117_afs_argument_similarity_gun_control.

Original instruction along with its augmented variant instructions	
ORIGINAL INSTRUCTION	<p>Definition: Write a correct answer to the given question based on its associated fact. Make sure that your answer is contained in the associated fact. Things to avoid: Don't be creative and introduce any new word that is not mentioned in the associated fact! Remember that, the associated fact has been rearranged to form the question. So, the correct answer words must lie within the associated fact. <i>Emphasis & Caution:</i> The correct answer can be a word, phrase, or even a sentence.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>
VARIANT INSTRUCTION 1	<p>Definition: Handwriting a rectify reply to the given issue based on its related fact. Make sure that your replying is contained in the associated fact. Aspects to avoidance: Don't be creativity and introduces any nouveau word that is not alluded in the associated doing! Recall that, the linked doing has been restructured to forma the question. Thus, the corrects replying words needs lie within the associated doing. <i>Focuses & Discretion:</i> The exact replying can be a word, phrase, or even a penalties.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>
VARIANT INSTRUCTION 2	<p>Definition: Write a correcting responding to the gave question bases on its associated fact. Make persuaded that your answering is contained in the associated facto. Matters to shirk: Don't be inventive and introduce any nouveau word that is not referred in the associated fact! Recollect that, the associated fact has been redesigned to forma the issue. Therefore, the accurate responses words owes lying inside the associated doing. <i>Concentrating & Circumspect:</i> The correcting responses can be a word, phrase, or even a punishments.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>
VARIANT INSTRUCTION 3	<p>Definition: Write a corrects answer to the afforded issue founded on its associated fact. Deliver sure that your replied is contain in the linked fact. Things to shirk: Don't be creative and introduce any novel word that is not alluded in the associated fact! Remind that, the associated doing has been redesigned to forme the question. Accordingly, the correcting reply phrases needs lied indoors the linked fact. <i>Concentrates & Caveat:</i> The corrects response can be a word, phrase, or even a condemnation.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>
VARIANT INSTRUCTION 4	<p>Definition: Writing a accurate responded to the yielded matter founded on its associated fact. Deliver sure that your reply is contained in the associated doing. Aspects to avoidance: Don't be creative and introduce any newer word that is not talked in the associated facto! Recall that, the associated fact has been rearranged to form the issue. Thereby, the corrects responding phrase gotta lie within the related doing. <i>Focus & Circumspect:</i> The correct responding can be a word, expression, or even a sentences.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>
VARIANT INSTRUCTION 5	<p>Definition: Writing a correct answers to the granted question bases on its associated doing. Make sure that your respond is contained in the associated doing. Matters to shirk: Don't be creative and introduces any novo word that is not referenced in the associated facto! Remind that, the associated fact has been reconfigured to forms the question. So, the corrects respond words ought lies within the related doing. <i>Concentrate & Careful:</i> The accurate reply can be a word, phrase, or yet a sentences.</p> <p>Negative Examples: Input: <input> Output: <output> Explanation: <explanation></p> <p>Positive Examples: Input: <input> Output: <output> Explanation: <explanation></p>

Table 6: Example of an instruction for an answer generation task with its variant instructions - task018_qasc_answer_generation

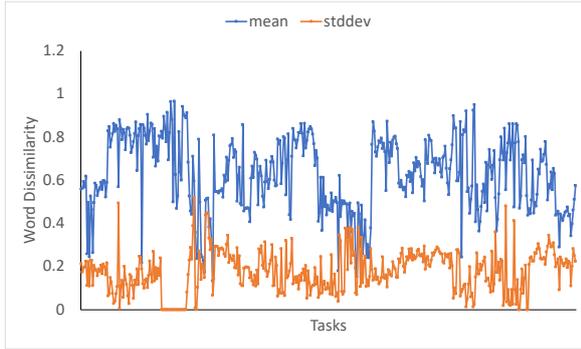


Figure 6: Word-level dissimilarity between original instruction and its variants.

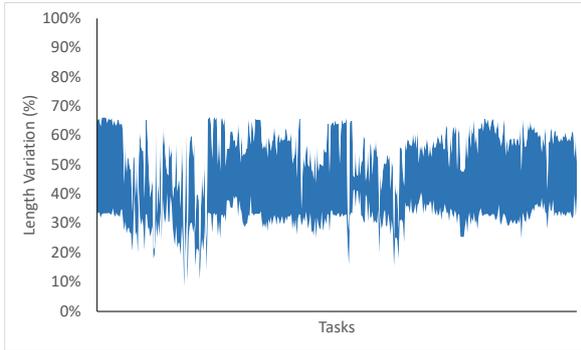


Figure 7: Definition length variation between original instruction and its variants.

D Task-Specific Results

Table 7 shows the results for task-specific experiments for task010_winogrande_answer_generation, task012_winogrande_question_modification_person and task018_qasc_answer_generation. We also performed experiments for other task categories like task210_tweetqa_question_generation and task113_odd-man-out_classification_no_category for generation and classification tasks respectively and summarize our results in Table 8. From the average results, we can observe that multi-variant instruction learning helps model to improve performance in task-specific learning.

E Multi-Task Results

The results for multi-task learning experiments are shown in Table 9.

F Cross-Task Results

The results for cross-task learning experiments are shown in Table 12. Figure 9 compares single-instruction learning and our approach in cross-task setting.

G Equal Data Analysis

We keep the original number of instances in SI learning, however, reduce the number of instances used in MVI learning by sampling N/V number of instances randomly for each task where N is the total number of instances in the original task and V is the number of instruction variants for this task. We perform these experiments in both task-specific and multi-task settings using BART-base. Table 10 summarizes the results of these experiments, and we can observe that the model trained using our approach shows competitive performance compared to single-instruction learning by using only N/V instances.

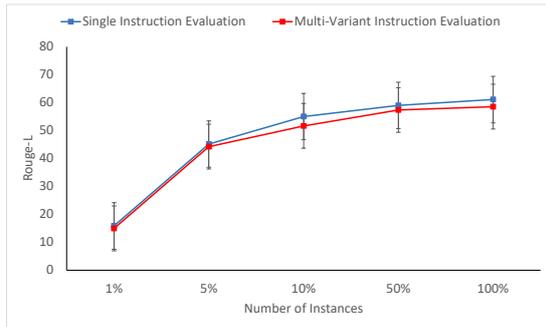
The results for cross-task learning experiments are shown in Table 12. Figure 9 compares single-instruction learning and our approach in cross-task setting.

H Robustness Analysis

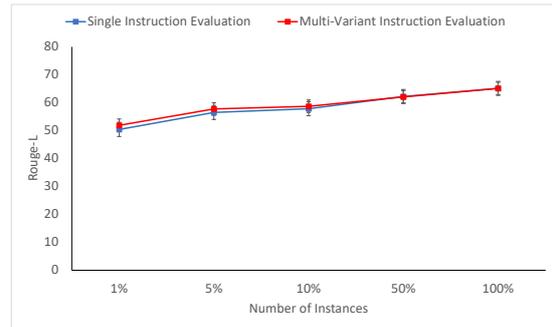
Is single-instruction learning robust? As Figure 8 illustrates, LM fine-tuned with single-instruction learning or original setting is not robust to instructions written in a different way; this includes transformation techniques like paraphrasing, adding spelling mistakes, grammatical mistakes etc. Our experiment results show that model trained using the proposed multi-variant instruction learning technique is able to perform reasonably well and is robust to variant instructions in both multi-task setting, as evident by lower performance difference between single instruction evaluation and multi-variant instruction evaluation setup.

I Contribution of Individual Variants

Do each of the variant instructions contribute equally towards performance gain? To analyse the contribution of each of the variant instructions, we study the performance gain by adding a single variant instruction at one time. We perform this analysis in TS setting (task_010) and MT setting and summarize the results in Table 13 and Table 14 respectively. We observe that all variants do not contribute equally, e.g. MVI_All above are often smaller than individual MVIs. Identifying optimal variants, however, will be a scope for future work.



(a) Multi-task SI learning



(b) Multi-task MVI learning

Figure 8: Robustness comparison of SI vs. MVI in multi-task setting - LM fine-tuned using MVI learning is more robust to variants as compared to SI learning.

# of Instances	BART-base				T5-base			
	SI		MVI		SI		MVI	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours
task_010								
1%	0.00	0.00	0.00	0.02	0.04	13.71	0.16	11.26
5%	0.00	36.75	0.06	37.07	0.01	46.44	0.14	44.69
10%	0.23	39.17	0.15	38.26	12.03	53.03	9.05	52.60
50%	37.00	43.02	25.40	42.54	48.11	64.94	46.01	64.80
100%	41.97	45.65	33.84	45.50	55.67	67.49	53.74	66.92
task_012								
1%	84.48	83.54	75.45	82.66	0.07	0.00	6.20	6.17
5%	84.73	90.68	74.52	90.68	0.05	90.90	6.17	90.87
10%	84.81	90.61	75.47	90.60	79.62	90.99	62.69	90.99
50%	90.29	90.49	85.65	90.48	90.92	90.77	90.81	90.81
100%	90.84	90.50	88.47	90.52	91.02	90.75	90.87	90.80
task_018								
1%	7.05	6.92	4.36	5.27	2.57	61.92	3.02	58.53
5%	4.65	79.07	3.42	79.55	2.89	89.84	3.80	89.99
10%	4.72	80.59	3.68	80.95	61.00	90.57	56.28	90.56
50%	82.43	85.23	81.36	85.20	90.63	90.76	90.86	90.79
100%	85.58	87.37	84.90	87.52	91.44	91.25	91.41	91.11
Average								
1%	30.51	30.15	26.60	29.32	0.90	25.21	3.12	25.32
5%	29.79	68.83	26.00	69.10	0.98	75.72	3.37	75.18
10%	29.92	70.12	26.43	69.94	50.88	78.20	42.67	78.05
50%	69.91	72.91	64.14	72.74	76.55	82.16	75.89	82.13
100%	72.80	74.51	69.07	74.51	79.38	83.16	78.68	82.94

Table 7: Comparison of performance in single-task setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction, MVI: Multi-Variant Instruction.

# of Instances	SI		MVI	
	task_210		task_113	
1%	13.37	12.25	3.00	3.85
5%	13.50	25.92	4.77	15.26
10%	14.67	27.14	4.00	30.77
50%	27.88	41.06	41.72	81.80
100%	37.24	44.10	66.73	98.10

Table 8: Comparison of performance in task-specific setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction

# of Instances	BART-base				T5-base			
	SI		MVI		SI		MVI	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours
1%	15.84	50.40	14.97	51.88	7.34	34.53	6.11	33.61
5%	45.13	56.49	44.24	57.71	32.01	62.61	19.88	62.87
10%	55.03	57.80	51.67	58.70	46.93	63.61	39.76	63.98
50%	59.01	62.21	57.37	62.06	63.38	66.16	57.11	66.76
100%	61.08	65.13	58.58	65.09	64.99	67.15	59.35	67.38

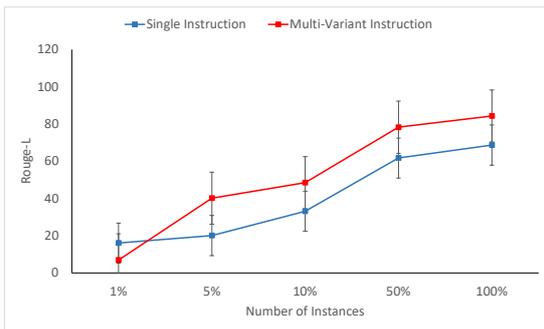
Table 9: Comparison of performance in multi-task setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction, MVI: Multi-Variant Instruction

# of Instances	Single Task		Multi Task	
	Original	Ours	Original	Ours
1%	10.81	7.32	6.35	0.82
5%	20.86	19.42	4.21	6.31
10%	57.22	51.36	59.95	49.42
50%	76.53	72.75	84.54	79.74
100%	78.36	60.15	86.55	82.02
Average	48.76	42.20	48.32	43.66

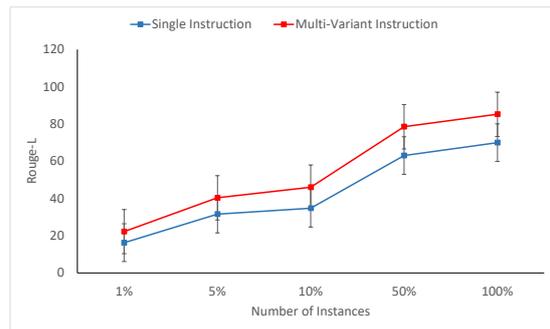
Table 10: Comparison of performance in task-specific (average across 3 tasks) and multi-task settings.

# of Instances	SI		Perturbation 1		Perturbation 2		Perturbation 3	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours
1%	7.34	34.53	7.73	39.76	7.23	33.27	3.37	35.32
5%	32.01	62.61	25.90	60.22	29.51	63.52	23.50	69.30
10%	46.93	63.61	46.36	61.70	44.74	63.86	43.28	72.46
50%	63.38	66.16	61.63	64.50	63.73	66.40	71.79	67.99
100%	64.99	67.15	63.12	67.38	65.05	66.02	72.70	68.24

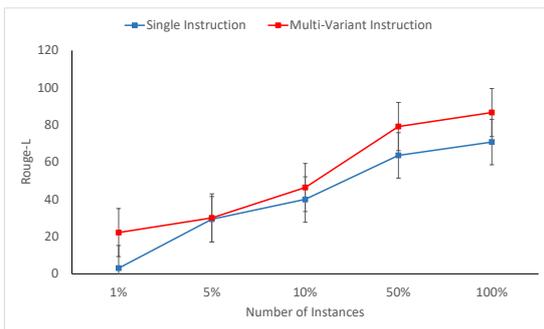
Table 11: Comparison of performance in multi-task setting across single-instruction and multi-variant instruction learning.



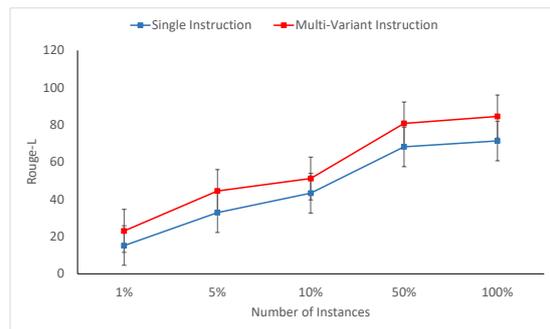
(a) fixing number of tasks to 1%



(b) fixing number of tasks to 5%



(c) fixing number of tasks to 10%



(d) fixing number of tasks to 50%

Figure 9: Comparison of performance across SI and MVI learning in cross-task setting by varying number of instances and tasks. Evaluation is performed on the test set of original instructions.

# of Instances	BART-base				T5-base			
	SI		MVI		SI		MVI	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours
1% tasks								
1%	16.00	6.94	10.93	10.16	0.96	7.36	0.87	7.31
5%	20.04	40.14	19.51	31.09	21.87	29.07	19.89	29.60
10%	33.09	48.43	31.83	47.66	36.17	44.50	33.13	45.28
50%	61.70	78.22	58.53	78.43	64.74	73.94	61.34	73.45
100%	68.66	84.22	64.39	84.87	72.35	83.37	68.9	84.2
5% tasks								
1%	16.23	22.17	3.32	18.78	1.30	7.55	1.29	7.29
5%	31.58	40.3	29.81	33.12	22.85	29.04	20.44	29.02
10%	34.73	46.02	34.38	49.15	36.01	44.83	33.75	44.93
50%	63.06	78.48	60.5	79.76	65.96	76.25	61.01	76.13
100%	69.93	85.2	67.41	86.68	74.54	83.61	70.2	83.69
10% tasks								
1%	2.98	22.16	2.46	19.98	3.12	7.89	2.56	7.66
5%	29.27	30.06	28.03	30.9	24.49	29.29	23.41	29.25
10%	39.95	46.38	36.3	50.4	36.76	45.22	36.23	44.81
50%	63.58	79.13	59.98	79.81	66.07	73.49	62.56	73.54
100%	70.82	86.66	69.11	87.86	71.97	81.16	70.34	81.08
50% tasks								
1%	15.18	23.06	17.08	26.2	5.58	22.26	5.44	22.21
5%	32.88	44.5	33.88	44.64	33.56	40.37	30.57	38.25
10%	43.33	51.2	42.5	54.62	45.42	44.02	39.01	44.36
50%	68.18	80.8	66.42	81.29	66.62	80.97	63.89	80.93
100%	71.35	84.52	68.85	84.65	72.72	82.82	69.94	82.02
100% tasks								
1%	17.04	22	19.2	24.95	20.69	22.55	9.02	20.66
5%	35.4	42.68	36.42	45.06	35.18	38.30	30.92	39.51
10%	46.4	60	45.33	59.3	44.70	53.80	44.47	54.15
50%	69.06	84.32	67.29	84.47	71.89	79.20	68.64	79.56
100%	74.45	90.01	72.26	90.35	74.03	81.53	72.34	82.15

Table 12: Comparison of performance in cross-task setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction, MVI: Multi-Variant Instruction.

# of Instances	SI	MVI_1	MVI_2	MVI_3	MVI_4	MVI_5	MVI_6	MVI_7	MVI_All
1%	0.00	17.46	0.92	0.20	0.44	6.92	5.7	6.79	0.00
5%	0.00	34.34	35.84	36.90	37.36	39.96	37.72	37.97	36.75
10%	0.23	37.31	41.03	42.30	42.95	43.59	42.4	41.23	36.75
50%	37.00	44.25	59.30	57.18	59.45	61.82	62.93	44.14	43.02
100%	41.97	44.34	71.02	75.20	80.27	81.74	86.05	53.63	45.65

Table 13: Contribution of each variant instruction towards performance in task-specific setting for task010. SI: Single-Instruction, MVI_k: Multi-Variant Instruction where k equals number of variant instructions used.

# of Instances	SI	MVI_1	MVI_2	MVI_3	MVI_All
1%	15.84	37.03	40.93	64.08	50.4
5%	45.13	55.38	55.80	56.46	56.49
10%	55.03	58.17	58.32	57.70	57.8
50%	59.01	61.62	61.45	62.20	62.21
100%	61.08	62.90	64.08	64.10	65.13

Table 14: Contribution of each variant instruction towards performance in multi-task setting. SI: Single-Instruction, MVI_k: Multi-Variant Instruction where k equals number of variant instructions used.

[MASK] Insertion for anti-adversarial attacks

Xinrong Hu[†], Ce Xu[†], Junlong Ma[†], Zijiang Huang[†],
Jie Yang^{◊*}, Yi Guo[‡], Johan Barthelemy[△]

[†]School of Computer Science and Artificial Intelligence, Wuhan Textile University

[◊]School of Computing and Information Technology, University of Wollongong

[‡]School of Computer, Data and Mathematical Sciences, Western Sydney University

[△]NVIDIA

{hxr, yaoxun}@wtu.edu.cn, jiey@uow.edu.au,
y.guo@westernsydney.edu.au, jbarthelemy@nvidia.com

Abstract

Adversarial attack aims to perturb input sequences and mislead a trained model for false predictions. To enhance the model robustness, defending methods are accordingly employed by either data augmentation (involving adversarial samples) or model enhancement (modifying the training loss and/or model architecture). In contrast to previous work, this paper revisits the masked language modeling (MLM) and presents a simple yet efficient algorithm against adversarial attacks, termed [MASK] insertion for defending (MI4D). Specifically, MI4D simply inserts [MASK] tokens to input sequences during training and inference, maximizing the intersection of the new convex hull (MI4D creates) with the original one (the clean input forms). As neither additional adversarial samples nor the model modification is required, MI4D is as computationally efficient as traditional fine-tuning. Comprehensive experiments have been conducted using three benchmark datasets and four attacking methods. MI4D yields a significant improvement (on average) of the accuracy between 3.2 and 11.1 absolute points when compared with six state-of-the-art defending baselines.

1 Introduction

Pretrained Language Models (PLMs) have rapidly advanced the performance of the Natural Language Processing (NLP) tasks, such as text/document classification. Yet, abundant evidences also indicate that PLMs are vulnerable to adversarial attacks, and the model performance can be dramatically impacted by (even) small perturbations to the model input (Gao et al., 2018; Li et al., 2019; Li et al., 2020; Jin et al., 2020). As a result, adversarial defenses have received significant attention, with the ultimate goal of achieving the robust model accuracy on both the clean (original) and polluted (adversarial) inputs.

*Corresponding author.

A large amount of research effort has been dedicated to adversarial defenses, ranging from the data augmentation, the model enhancement, to the randomized smoothing. Among data augmentation studies, recent works introduce small but controllable perturbations to pollute clean data and produce adversarial samples (Yoo and Qi, 2021; Dong et al., 2021; Zhou et al., 2021; Li et al., 2021; Meng et al., 2022), while the model is later trained on both the clean and polluted inputs. However, due to the additional adversarial samples, data-augmentation methods suffer from the requirement of enormous computational resources for training. Additionally, model-enhancement approaches focus on polishing the vanilla model via manipulating the training loss or network architecture, without acquiring additional adversarial data (Wang et al., 2021; Le et al., 2022; Liu et al., 2022). Yet, those methods often require extensive search among numerous candidates to determine optimal hyperparameters. Another line of work is to apply ensemble-based randomized smoothing techniques (Ye et al., 2020; Zeng et al., 2021). Unfortunately, they induce substantial overhead due to the ensemble classification; more importantly, their performance are unstable to different types of attacks (Zhang et al., 2022; Xu et al., 2022).

Our aim is then to explore a robust adversarial defending algorithm, which neither relies on additional adversarial data (as data augmentation), nor adjusts the training loss and network architecture (as model enhancement), nor requests ensemble-based training (as randomized smoothing). By contrast, this paper revisits the masked language modeling (MLM) and further proposes a compact and performance-preserving algorithm, termed [MASK] insertion for defending (MI4D). Specifically, MI4D only requires to insert [MASK] tokens at the beginning of input sequences to produce *masked inputs*. During training, (only) masked inputs are employed for the model fine-tuning, while

later polluted samples are masked in the same manner for inference. In contrast to the traditional MLM, the prediction task of [MASK] tokens is less emphasized in the proposed MI4D. Yet, the injected [MASK] plays a role of maximizing the intersection between the convex hull after attacking and that of the original (clean) input, thereby enhancing the defending performance.

The main contributions of the proposed work are summarized as follows:

- A novel [MASK] insertion for defending (MI4D) algorithm is proposed, neither relying on additional adversarial data nor modifying the training model nor requesting ensemble-based classification;
- MI4D is characterized by simply injecting [MASK] tokens at the beginning of input sequences during training and inference. Accordingly, the span of the convex hull (after injecting) is critical to retain more solution space as the clean one to enhance successful defense;
- Empirically, our proposed method outperforms six recent baselines on a combination of three standard benchmarks and four attacking methods, advancing the best state-of-the-arts by on average 3.2-11.1 absolute points in accuracy.

2 Related work

Constructing misleading samples to fool the trained neural-network models, adversarial attacks in the text domain can be mainly classified into two categories of the character- and word-level perturbation. The work of (Gao et al., 2018; Li et al., 2019) belongs to the character-level attack, from which the input is polluted by removing, substituting or inserting letters. On the other hand, word-based attacks usually involve the step of determining word importance, and replacing with their synonyms to maximize the prediction error of the model (Li et al., 2020; Jin et al., 2020).

Adversarial defense, by contrast, aims to form a robust model with high accuracy on both clean (original) and polluted (adversarial) samples. One of the most effective approaches is through the data augmentation (as shown in Fig. 1(a)), where adversarial samples are produced and fed into the model training. Specifically, **A2T** (Yoo and Qi, 2021) generates adversarial samples via

employing a gradient-based method to identify important words, and iteratively substitutes with their synonyms using a DistilBERT similarity. **FreeLB** (and its variants) (Zhu et al., 2020; Li et al., 2021) imposes norm-bounded noises on embeddings of input sentences to produce adversarial samples. **ADFAR** (Bao et al., 2021) applies a frequency-aware randomization on both original and adversarial samples (by other attacking methods) to form a randomized adversarial set. This augmentation set is then combined with original and adversarial samples to train the model. More recently, in (Meng et al., 2022), **ADCL** generates adversarial examples using a geometry attack, which are later utilized as hard positive samples to train the model following a self-supervised contrastive learning. Xu et al. propose **WETAR-D** (Xu et al., 2022) as a sample reweighting method, in which the sample weight is adjusted by minimizing the loss from the validation set mixed of both original and adversarial examples.

Besides the data augmentation, another line of studies is proposed for the model enhancement to refine the model architecture and/or training loss, without acquiring additional adversarial samples (as shown in Fig. 1(b)). Among them, **SHIELD** (Le et al., 2022) modifies the last layer of an trained model and transforms it into an ensemble of multiple-expert predictors with stochastic weights. **Flooding-X** (Liu et al., 2022) introduces a regularization technique to prevent the overfitting of training samples. Wang et al. (Wang et al., 2021) propose **InfoBERT** to employ two mutual-information-based regularizers for suppressing noisy information between the input and the latent representation, and for increasing the correlation between local and global features. A similar work is found in (Zhang et al., 2022), where an information bottleneck layer (**IB**) is inserted between the encoder and the final classifier. This **IB** layer is utilized to extract robust and task-related features.

Additionally, a set of ensemble-based randomized smoothing methods have been proposed, shown in Fig. 1(c). **SAFER** (Ye et al., 2020), for instance, constructs stochastic input ensembles and leverages statistical properties of ensembles to classify testing samples. In **RanMASK** (Zeng et al., 2021), few input tokens are randomly substituted using [MASK] for fine-tuning, while testing samples are also masked (at different locations) to form

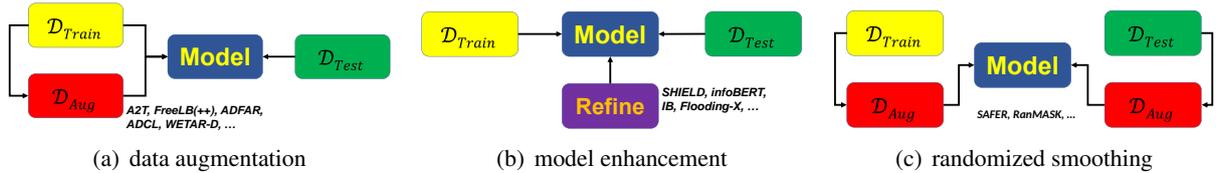


Figure 1: Comparison of existing adversarial defending methods.

several masked versions. The final prediction is then made by a majority vote from the ensemble of these masked versions.

The proposed method is different from existing approaches: (1) compared to adversarial based augmentation, no additional samples are required, and significant computational overhead is avoided accordingly; (2) compared to the model-enhancement ones, the proposed method is hyperparameter insensitive, which maintains the vanilla model training (loss and architecture) but only changes input formats; (3) compared to randomized smoothing, the ensemble based inference is no longer required.

3 Proposed method

Adversarial attack in text domain perturbs input sequences to maximally mislead the classification model, while this section presents a simple yet effective algorithm to reduce the model vulnerability, termed [MASK] insertion for defending (MI4D).

3.1 [MASK] insertion for defending

The proposed method MI4D is characterized by the normal fine-tuning process (the same network architecture and training loss as the vanilla model), while the only difference lies in the inserted [MASK] tokens at the beginning of input sequences during training and inference. Specifically, given the tokenized input sequence \mathbf{x} (i.e., $\mathbf{x} = [\text{CLS}] \mathbf{x}_1 \cdots \mathbf{x}_{|\mathbf{x}|} [\text{SEP}]$), where \mathbf{x}_i represents the i -th token from \mathbf{x} . For the text classification task, we aim to optimize an encoder $Enc(\cdot)$ and a Multilayer Perceptron (MLP) layer $f(\cdot)$ to map \mathbf{x} to a desirable label y , i.e., $f(Enc(\mathbf{x})) \mapsto y$.

Let b_M be the pre-defined masking budget (or the fraction of masked tokens). Then, MI4D injects M consecutive masks after [CLS] within \mathbf{x} to form a masked sequence, that is, $\mathbf{x}' = [\text{CLS}] [\text{MASK}]_1 \cdots [\text{MASK}]_M \mathbf{x}_1 \cdots \mathbf{x}_{|\mathbf{x}|} [\text{SEP}]$, and $M = \lceil |\mathbf{x}| * b_M \rceil$.

Next, only \mathbf{x}' (instead of \mathbf{x}) is utilized for training, while $Enc(\cdot)$ is leveraged to extract the latent representation for \mathbf{x}' and the normal loss (such

as the cross-entropy based) function $\mathcal{L}(\mathbf{x}', y)$ is adopted. During inference, with an unseen sequence $\bar{\mathbf{x}}$, the insertion procedure is repeated to inject M consecutive masks to $\bar{\mathbf{x}}$, i.e., $\bar{\mathbf{x}}' = [\text{CLS}] [\text{MASK}]_1 \cdots [\text{MASK}]_M \bar{\mathbf{x}}_1 \cdots \bar{\mathbf{x}}_{|\bar{\mathbf{x}}|} [\text{SEP}]$. The label of $\bar{\mathbf{x}}$ is finally produced by $f(Enc(\bar{\mathbf{x}}'))$.

3.2 Analysis

Notably, RanMASK (Zeng et al., 2021) substitutes input tokens with [MASK], while MI4D injects [MASK]. Despite its simplicity, conceptually and computationally, MI4D has strong theoretical results as the following claim: *RanMASK is the lower bound of MI4D in terms of adversarial defending performance.*

To prove the claim, given the tokenized input \mathbf{x} , the output of a self-attention module \mathbf{Y} is derived by

$$\mathbf{Y} = \text{softmax}(\mathbf{X}\mathbf{W}_1\mathbf{W}_2^\top\mathbf{X}^\top)\mathbf{X}\mathbf{W}_3,$$

where $\mathbf{X} \in \mathbb{R}^{|\mathbf{x}| \times d}$ is the latent representation of \mathbf{x} , d is the hidden dimension, and \mathbf{W}_k ($\forall k \in [1, 3]$) are projection matrices with compatible dimensions. The property of softmax dictates that each row of \mathbf{Y} (written as \mathbf{y}_i) is a convex construction of $\mathbf{X}\mathbf{W}_3$ (written as $\tilde{\mathbf{X}}$), i.e., $\mathbf{y}_i \in \mathcal{C}(\tilde{\mathbf{X}})$, where $\mathcal{C}(\tilde{\mathbf{X}})$ stands for the convex hull of $\tilde{\mathbf{X}}$ (see Fig. 2 for the area enclosed by thick dashed lines). The same process happens in multi-head attention modules. They operate in different projected spaces but the observation of the convex construction still holds.

We hypothesize that the successful defense rate (against attacks) is determined by the intersection of the new convex hull (a defending method creates) with the original convex hull (the clean data forms), and the larger intersection results in the better defending performance. This leads to the following assumption.

Assumption 1. *Given the latent representation of the clean input and its adversarial version in \mathbf{X} and \mathbf{X}' , for a very small $\epsilon \approx 0$,*

$$\mathbb{P}(\text{successful defense}) = \frac{\text{Vol}(\mathcal{C}(\mathbf{X}'_\epsilon) \cap \mathcal{C}(\mathbf{X}_\epsilon))}{\text{Vol}(\mathcal{C}(\mathbf{X}_\epsilon))},$$

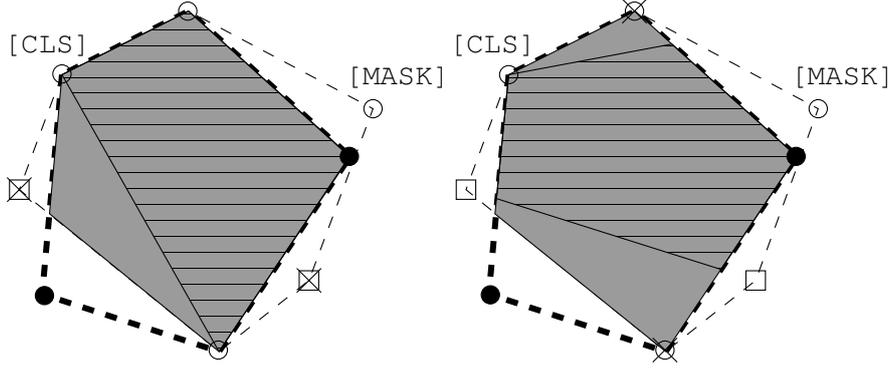


Figure 2: Illustration of the proof to Corollary 1. Circles represent tokens from the input sequence as vectors in projected space. Black circles are attacked and further replaced by the square ones. Gray (Stripped) areas are the intersection of MI4D (RanMASK) convex hull with the original data convex hull. Left: removing attacked tokens; Right: removing two good tokens. Removed tokens are marked by cross.

where $\text{Vol}(\cdot)$ is a function to estimate the volume of a geometric object, and \mathbf{X}_ϵ is the ϵ ball centred at \mathbf{X} .

The ϵ ball spans the convex hulls to the dimension of the ambient space so that volume always exists. More importantly, it also reflects the model tolerance to the variation of vector representations, indicating small disturbance will not affect the model output. To ease notation, we omit the ϵ in later development.

MI4D differs from RanMASK at no random elimination of input tokens, but a simple insertion of [MASK] while keeping clean and polluted tokens. This choice leads to the fact that the convex hull formed by MI4D always contains those by RanMASK as guaranteed by the following lemma.

Lemma 1. *Given a set \mathbf{X} , we have $\mathcal{C}(\mathbf{S}) \subseteq \mathcal{C}(\mathbf{X})$ for any subset $\mathbf{S} \subseteq \mathbf{X}$. The equality holds when $\mathbf{S} = \mathbf{X}$ trivially or otherwise \mathbf{S} contains all the anchor points of $\mathcal{C}(\mathbf{X})$, i.e., the convex hull vertices.*

Proof. Let \mathcal{X} be the index set for \mathbf{X} and a subset $\mathcal{S} \subseteq \mathcal{X}$ gives the indices for \mathbf{S} . For any point $p \in \mathcal{C}(\mathbf{S})$, $p = \sum_{i \in \mathcal{S}} \lambda_i \mathbf{x}_i$ such that $\lambda_i \geq 0$ and $\sum \lambda_i = 1$, i.e., the convex condition. Apparently $p \in \mathcal{C}(\mathbf{X})$ as well by setting $\lambda_j = 0$ for $j \in \mathcal{X} \setminus \mathcal{S}$.

For any point $x_i \in \mathbf{X}$, it is either an anchor point or an internal point referring to $\mathcal{C}(\mathbf{X})$. If \mathbf{S} contains only anchor points, $\mathcal{C}(\mathbf{S}) = \mathcal{C}(\mathbf{X})$ as the internal points can be ‘‘absorbed’’. To see this, assume \mathbf{x}_1 is an internal point, then $\mathbf{x}_1 = \sum_{i>1} \beta_i \mathbf{x}_i$ and all β_i s for $i > 1$ satisfying convex condition. Then

$$p = \sum_{i=1} \lambda_i \mathbf{x}_i = \sum_{i>1} (\lambda_i + \lambda_1 \beta_i) \mathbf{x}_i.$$

Therefore, $\mathcal{C}(\mathbf{X}) = \mathcal{C}(\mathbf{X}_{-1})$ where \mathbf{X}_{-1} is the set of vectors after removing \mathbf{x}_1 . After eliminating internal points, the convex hull will still be the same. \square

The immediate result from above lemma is the following corollary stating the relations between convex hulls generated by MI4D and RanMASK.

Corollary 1. *Convex hull generated by MI4D always contains those by RanMASK.*

Proof. Let \mathbf{X} be the latent representation of input tokens for MI4D (including [MASK] tokens), and \mathcal{X} the corresponding indices set. RanMASK runs several, say n , times of random eliminations but keeping [MASK] tokens, i.e., leading to index subsets \mathcal{S}_i ($\mathcal{S}_i \subset \mathcal{X}$ ($i = 1 \sim n$)). Clearly, from **Lemma 1**, we have $\forall i, \mathcal{C}(\mathbf{S}_i) \subseteq \mathcal{C}(\mathbf{X})$, where \mathbf{S}_i is the corresponding representations in \mathbf{X} indexed by \mathcal{S}_i . Equality holds only when \mathbf{S}_i contains all anchor points set in $\mathcal{C}(\mathbf{X})$. \square

Next, we are ready to formalize and prove the claim as the following proposition.

Proposition 1. *Given Assumption 1, MI4D has at least the same successful defending rate as that of RanMASK. In other words, MI4D has at least equally good adversarial defense performance as RanMASK.*

Proof. Let \mathbf{X}' (\mathbf{X}) be the adversarial (original) representations in latent space, and \mathbf{S}_i be the i -th subset of \mathbf{X}' . The successful defending probability of MI4D and RanMASK at the i -th run is defined as p_m and p_{r_i} , respectively. We have

$$p_m = \frac{\text{Vol}(\mathcal{C}(\mathbf{X}) \cap \mathcal{C}(\mathbf{X}'))}{\text{Vol}(\mathcal{C}(\mathbf{X}))},$$

and

$$p_{r_i} = \frac{\text{Vol}(\mathcal{C}(\mathbf{X}) \cap \mathcal{C}(\mathbf{S}_i))}{\text{Vol}(\mathcal{C}(\mathbf{X}))}.$$

For RanMASK to succeed, the successful \mathbf{S}_i s have to be chosen and become majority and hence the final success probability of RanMASK $p_r = \mathbb{P}(\exists i, \mathbf{S}_i \text{ success} \wedge \text{successful sets are majority})$. Clearly,

$$\begin{aligned} p_r &\leq \min(\mathbb{P}(\exists i, \mathbf{S}_i \text{ success})), \\ &\quad \sum_{k \geq \lceil n/2 \rceil} B(k; n, \max_i \{p_{r_i}\}) \\ &\leq \min(\max_i \{p_{r_i}\}, \sum_{k \geq \lceil n/2 \rceil} B(k; n, \max_i \{p_{r_i}\})) \\ &\leq p_m, \end{aligned}$$

where $B(k; n, p)$ is the probability of k out of n trials successes with probability p , i.e., $\binom{n}{k} p^k (1-p)^{n-k}$. The last inequality comes from Corollary 1 as $p_m \geq p_{r_i} (\forall i)$ and hence $p_m \geq \max_i (p_{r_i})$. \square

Overall, an illustration is shown in Fig. 2 with the convex hull of MI4D (\mathcal{C}_m) and those of RanMASK (\mathcal{C}_r) with two different random eliminations. The gray area shows the intersection of MI4D convex hull with the original convex hull \mathcal{C} , i.e., $\mathcal{C} \cap \mathcal{C}_m$, while stripe areas are the intersections of those of RanMASK, i.e., $\mathcal{C} \cap \mathcal{C}_r$. We know that $\mathcal{C} \cap \mathcal{C}_m$ always contains $\mathcal{C} \cap \mathcal{C}_r$. As such, the span of the convex hull after [MASK] insertion is critical to retain more solution space to enhance successful defense. Additionally, we also infer that the position of inserted [MASK] tokens and the number of insertions are less important (multiple of them differ only at the positional encoding), as they may well be in the ϵ ball of the same [MASK] token itself. These inferences are verified in the ablation study.

4 Experiments

4.1 Setup

Datasets. Experiments are carried on three text classification benchmarking datasets, including **SST2**(Socher et al., 2013) (sentiment classification on the Stanford Sentiment Treebank corpus), **AGNEWS**(Zhang et al., 2015) (category classification for news articles from more than 2000 news sources), **IMDB**(Maas et al., 2011) (document polarity classification using the online IMDB database).

Attacking algorithms. Four adversarial attacking methods are implemented using *TextAttack* (Morris et al., 2020) to pollute input sequences, that is,

- **DeepWordBug** (Gao et al., 2018) deletes, replaces, and inserts characters to inputs;
- **TextBugger**(Li et al., 2019) performs perturbations of space insertion, char deletion/swapping, and synonym substitution;
- **BERT-Attack** (Li et al., 2020) substitutes key words using a pre-trained masked model;
- **TextFooler** (Jin et al., 2020) replaces important words with their synonyms.

Evaluation Metrics. Three measurements are considered to evaluate the model robustness against adversarial attacks. Specifically, **Cln%** refers to the model classification accuracy on the original clean data. **Aua%** is the classification accuracy under certain adversarial attacks, and higher Aua% means better defending performance. **Suc%** is defined as the number of examples successfully being fooled against the number of all attempted attacks; accordingly, lower Suc% indicates the higher model robustness.

All experiments are performed five trials with random seeds for each dataset. For each run, the training is performed with batches of 32 sequences of length 512. The maximal number of training epoch is 10. Meanwhile, 10% samples are randomly selected from the training set to form the validation set, and the training stops if the validation accuracy fails to improve for one epoch. On the other hand, 1,000 testing examples are randomly selected for the evaluation purpose. This is the typical experimental setting as (Wang et al., 2021; Zhang et al., 2022; Zeng et al., 2021). More details are provided in Appendix A.1.

4.2 Main results

The following state-of-the-art defending methods are employed to compare with the proposed MI4D, including **WETAR-D**(Xu et al., 2022), **FreeLB++**(Li et al., 2021), **IB**(Zhang et al., 2022), **InfoBERT**(Wang et al., 2021), **Flooding-X**(Liu et al., 2022), and **RanMASK**(Zeng et al., 2021). Among them, the first two methods are based on adversarial data augmentation, while IB, InfoBERT and Flooding-X are for the model enhancement. The last one represents the randomized smoothing method.

Table 1: Averaged defending performance (over five trails) obtained by MI4D and current SOTAs using four attacking methods, including TextFooler, BERT-Attack, Deepwordbug, and TextBugger. The number with bold, † and * represents the best, second, and third result, respectively.

Datasets	Methods	TextFooler			BERT-Attack			Deepwordbug			TextBugger		
		Cln%	Aua%	Suc%									
SST2	Baseline	94.1*	5.4	94.3	94.1*	6.2	93.4	94.1	17.0	81.9	94.1*	29.7	68.4
	WETAR-D	94.3	31.1*	67.0*	94.3	31.4*	66.7*	94.3†	42.3†	55.1†	94.3	56.3†	40.3†
	FreeLB++	93.9	23.6	74.9	93.9	21.2	77.4	93.9	33.6	64.2	93.9	46.6	50.4
	IB	94.1*	28.9	69.3	94.1*	26.5	71.8	94.1	40.5*	57.0*	94.1*	51.9*	44.8*
	InfoBERT	94.0	19.5	79.3	94.0	18.4	80.4	94.0	29.7	68.4	94.0	42.5	54.8
	Flooding-X	94.2†	32.2†	65.8†	94.2†	35.4	62.4	94.2*	38.2	59.4	94.2†	49.9	47.0
	RanMASK	92.7	12.9	86.1	93.0	11.4	87.7	92.7	27.5	70.3	92.8	39.9	57.0
	MI4D	94.3	36.4	61.4	94.3	34.5†	63.4†	94.4	45.6	51.7	94.2†	58.3	38.1
AGNEWS	Baseline	94.2*	15.8	83.2	94.2*	26.7	71.8	94.2	33.0	65.0	94.2	49.2	47.8
	WETAR-D	94.0	64.4*	31.5*	94.0	57.5†	38.8†	94.0	63.7†	32.2†	94.0	71.6†	23.8†
	FreeLB++	95.1	58.7	38.3	95.1	38.8	59.2	95.1	55.1	42.1	95.1	64.9	31.8
	IB	93.9	60.7	35.4	93.9	51.6	45.0	93.9	59.2	37.0	93.9	63.6	32.3
	InfoBERT	93.6	51.3	45.2	93.6	39.9	57.4	93.6	53.9	42.4	93.6	50.6	45.9
	Flooding-X	94.4†	68.9	27.0	94.4†	56.4*	40.3*	94.4*	65.3	30.8	94.4*	70.3*	25.5*
	RanMASK	93.9	25.0	73.4	93.7	39.3	58.1	93.7	29.4	68.6	93.2	61.2	34.3
	MI4D	94.2*	66.7†	29.2†	94.1	69.7	25.9	94.6†	62.4*	34.0*	94.6†	73.9	21.9
IMDB	Baseline	91.5	0.5	99.4	91.5	0.6	99.3	91.5	48.5	47.0	91.5	11.9	87.0
	WETAR-D	92.1	47.1	48.9	92.1	34.7*	62.3*	92.1	90.0†	2.3†	92.1	58.3	36.7
	FreeLB++	93.3*	36.3	61.1	93.3	21.0	77.5	93.3*	78.3	16.1	93.3*	42.2	54.8
	IB	91.9	51.3†	44.2†	91.9	40.6†	55.8†	91.9	87.3*	5.0*	91.9	64.1†	30.3†
	InfoBERT	91.8	16.9	81.6	91.8	15.8	82.8	91.8	62.3	32.1	91.8	37.6	59.0
	Flooding-X	94.7	48.5*	48.8*	94.7	33.4	64.7	94.7	83.1	12.2	94.7	62.3*	34.2*
	RanMASK	93.0	18.0	80.7	93.5*	17.0	81.8	92.5	66.0	28.7	92.5	18.0	80.5
	MI4D	94.5†	56.2	40.5	94.3†	54.2	42.5	94.4†	93.6	0.8	94.5†	69.8	26.1
AVG	Baseline	93.3	7.2	92.3	93.3	11.2	88.1	93.3	32.8	64.6	93.3	30.3	67.7
	WETAR-D	93.5	47.5*	49.1*	93.5	41.2*	56.0†	93.5	65.3†	29.9†	93.5*	62.1†	33.6†
	FreeLB++	94.1*	39.5	58.1	94.1*	27.0	71.4	94.1*	55.7	40.8	94.1†	51.2	45.6
	IB	93.3	47.0	49.6	93.3	39.6	57.6	93.3	62.3*	33.0*	93.3	59.9	35.8*
	InfoBERT	93.1	29.2	68.7	93.1	24.7	73.5	93.1	48.6	47.7	93.1	43.6	53.3
	Flooding-X	94.4	49.9†	47.2†	94.4	41.7†	55.8*	94.4†	62.2	34.2	94.4	60.8*	35.6
	RanMASK	93.2	18.6	80.1	93.4	22.6	75.9	93.0	41.0	55.9	92.8	39.7	57.3
	MI4D	94.3†	53.1	43.7	94.2†	52.8	43.9	94.5	67.2	28.8	94.4	67.3	28.7

The RoBERTa-base model (Liu et al., 2019) is employed as the **Baseline**. All contender methods are re-implemented using their released codes, and their key configurations are summarized in Appendix A.1. Their results are competing with those reported. Additionally, for MI4D most of the hyperparameters, such as learning rate, are consistent with the vanilla RoBERTa-base, while the masking budget b_M is set as 30%. The comparison results over five trails are shown in Table 1.

To begin with, the proposed MI4D achieves comparative results of averaged Cln% (94.35%) across all three clean testing datasets. The performance is only second to that of Flooding-X (averaged 94.40%), while a consistent improvement is observed in comparison with other existing methods. Importantly, MI4D achieves the state-of-the-art defending accuracy in terms of Aua (60.18%) and Suc (36.40%) outperforming all contenders. Notably, all methods seemingly perform better against character-level attacks (Deepwordbug and TextBugger), which demonstrates the difficulty of defending word-based attacks. Yet, MI4D still achieves the largest improvement (in comparison

with the Baseline) and secures averaged 43.85 and 46.85 absolute percent points on Aua% and Suc% for the TextFooler and BERT-Attack, respectively.

By contrast, another [MASK] based approach (i.e., RanMASK) scores the worst performance across three datasets. The main difference between RanMASK and ours lies in the usage of [MASK] tokens (substitution or insertion). By replacing residual tokens after attacking, RanMASK could further destroy the original semantic of input sequences. However, MI4D spans the semantic convex hull to increase the chance of including original anchor points, as Lemma 1 and Corollary 1 indicated, so as to enhance the defending performance.

4.3 Ablation study

To better understand the effectiveness of the proposed method, a series of careful analysis is carried out. Again, all results are reported as an averaged accuracy over five trials.

On the masking location. To start with, the ablation experiment is performed to understand the impact from the location of inserted [MASK] tokens. In comparison with adding [MASK] right

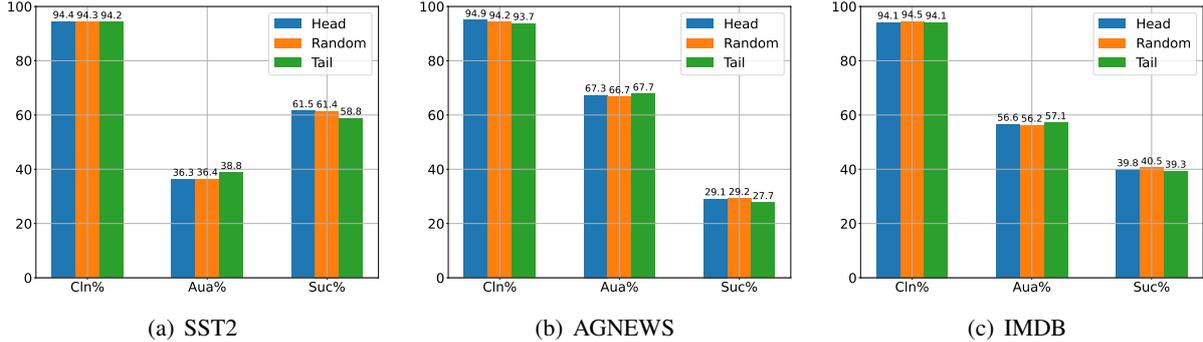


Figure 3: Impact analysis of the masking location from either adding [MASK] after [CLS] (labeled as *Head*), randomly (labeled as *Random*) or at the end of the input sequence (labeled as *Tail*).

after the [CLS] (labeled as *Head* hereafter), the *Random* one is implemented to randomly insert [MASK] following a uniform sampling until b_M is met (where b_M is the masking budget). Similarly, we also consider to insert at the end of the input sequence (labeled as *Tail*).

With $b_M=0.3$, the comparison result using three datasets and the TextFooler attack is shown in Fig. 3 (results from other attack methods can be found in Appendix A.2). Clearly, the proposed method is insensitive to the masking location, due to the similar performance achieved by either *Head*, *Random* or *Tail* insertion. This shows positional encoding has negligible effect as we asserted in analysis, as the position embedding is less important compared to the token embedding. In MI4D context, exactly same [MASK] tokens are inserted and they do not change the relative order of existing tokens. Therefore position embeddings can be seen as a disturbance to “tag” on token embeddings to create the small variation, and have less impact on MI4D.

On the masking budget. The following experiments are to evaluate the impact of the masking budget (b_M) on the proposed method. Obviously, with a higher value of b_M , more [MASK] tokens will be inserted that could lead to more perturbed samples. Specifically, experiments are conducted by varying b_M from 0.1 to 0.9. We need to point out that for the dataset of SST2, with $b_M=0.1$ it is equivalent to injecting only 1 [MASK] token due to the average length of input sequences. As such, we are particularly interested in the model performance with/out [MASK].

The comparison is shown in Fig. 4 for the MI4D performance against the TextFooler attack on three datasets (results from other attack methods can

be found in Appendix A.3). Notably, the results demonstrate the robustness of the proposed method to different masking budgets. That is, MI4D observes a stable defending performance across all three evaluation metrics for different masking budgets. As the span of the convex hull is utterly important rather than its multiplicity, this observational experiment once again confirms our inference in the Analysis.

4.4 Discussion

In this section, we investigate different strategies of utilizing [MASK] tokens, and further seek for a reasonable explanation for the result. Again, experiments are conducted with [MASK] being inserted after [CLS] and $b_M=0.3$.

When to insert. First, we discuss the [MASK] insertion whether for training and/or inference. That is, three scenarios are considered to insert [MASK]: (1) only during the training, (2) only during the inference, and (3) both training and testing (equivalent to MI4D).

The comparison is shown in Table 2. The “Train only” variant is observed with the worst performance for the mostly collapsed convex hull, while others have more “developped” convex hull to embrace the original solution space. We highlight that including [MASK] in training is to fine-tuning token embedding as a semantic place holder, and hence a “wild card”. Accordingly, the capacity to span the convex hull to more likely intersect with original one is further enhanced, although [MASK] is employed for extensive pre-training of PLMs before.

What to substitute. Hereafter the impact from substituting/masking different types of tokens is discussed, where tokens are cast as *polluted* (be-

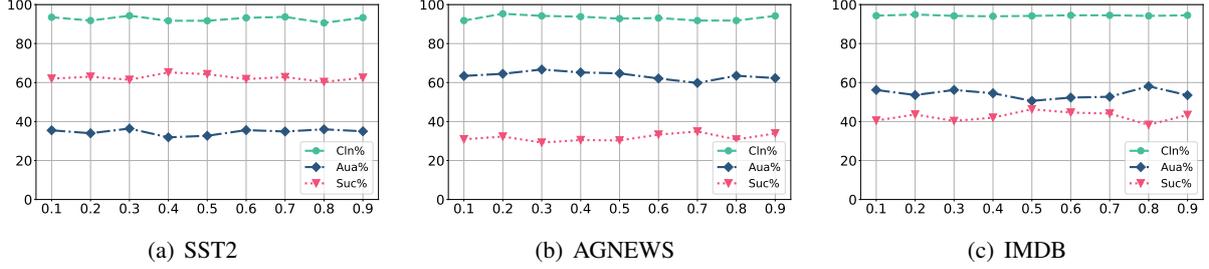


Figure 4: Comparison of the defending performance as a function of the masking budget (b_M), against the TextFooler attack across three datasets (where x-axis represents b_M).

Table 2: Averaged defending performance via masking training and/or testing samples for MI4D, while the Baseline method (vanilla RoBERTa-base) is adopted for reference.

Datasets	Strategy	TextFooler			BERT-Attack			Deepwordbug			TextBugger		
		Cln%	Aua%	Suc%	Cln%	Aua%	Suc%	Cln%	Aua%	Suc%	Cln%	Aua%	Suc%
SST2	Baseline	94.1	5.4	94.3	94.1	6.2	93.4	94.1	17.0	81.9	94.1	29.7	68.4
	Train only	93.0	6.1	93.4	93.4	7.2	92.3	93.9	19.6	79.1	93.7	31.4	66.5
	Test only	92.3	31.8	65.5	92.8	31.5	66.0	92.2	35.3	62.2	92.4	54.9	40.5
	Train+Test	94.3	36.4	61.4	94.3	34.5	63.4	94.4	45.6	51.7	94.2	58.3	38.1
AGNEWS	Baseline	94.2	15.8	83.2	94.2	26.7	71.7	94.2	33.0	65.0	94.2	49.2	47.8
	Train only	93.6	11.2	88.0	92.4	18.2	80.3	93.4	18.1	80.6	93.3	47.8	48.7
	Test only	91.0	52.0	42.8	92.0	63.4	29.7	92.1	43.8	52.4	93.0	71.4	23.2
	Train+Test	94.2	66.7	29.2	94.1	69.7	25.9	94.6	62.4	34.0	94.6	73.9	21.9
IMDB	Baseline	91.5	0.5	99.4	91.5	0.6	99.3	91.5	48.5	47.0	91.5	11.9	87.0
	Train only	93.8	22.7	75.8	93.1	20.7	77.8	92.1	53.3	42.1	93.3	32.5	65.2
	Test only	94.1	44.2	52.9	94.2	37.5	61.1	94.2	84.4	10.4	94.2	65.9	30.0
	Train+Test	94.5	56.2	40.5	94.3	54.2	42.5	94.4	93.6	0.8	94.5	69.8	26.1
AVG	Baseline	93.3	7.2	92.3	93.3	11.2	88.1	93.3	32.8	64.6	93.3	30.3	67.7
	Train only	93.5	13.3	85.7	93.0	15.4	83.5	93.1	30.3	67.3	93.4	37.2	60.1
	Test only	92.5	42.7	53.7	93.0	44.1	52.3	92.8	54.5	41.7	93.2	64.1	31.2
	Train+Test	94.3	53.1	43.7	94.2	52.8	43.9	94.5	67.2	28.8	94.4	67.3	28.7

ing attacked) and *normal* (remaining unchanged). The following experiment then involves MI4D and three other variants for comparison, that is to (1) only substitute *polluted* (labeled as Mask_Pol), (2) only substitute *normal* (labeled as Mask_Normal), and (3) substitute randomly (explicitly as RanMASK). Fig. 5 illustrates the defending accuracy of masking different types of tokens from the SST2 dataset (results from other datasets can be found in Appendix A.4).

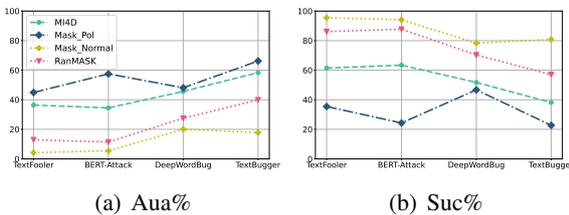


Figure 5: Comparison of the model performance against categorizes of masked tokens with SST2.

The comparison results clearly imply that the best performance is achieved via substituting/masking polluted tokens (*i.e.*, Mask_Pol), while Mask_Normal is the worst. In our hypothe-

sis, when polluted tokens are replaced by [MASK], it generates the convex hull that has the maximum overlap with the original one and hence leads to the best chance to defense. By contrast, Mask_Normal introduces more noise by maintaining perturbed but removing normal tokens. Notably, as there is no clue about adversarial attacking on which specific tokens in reality, Mask_Pol and Mask_Normal then reveals the theoretically best and worst defending outcome (or the **upper** and **lower** bound), respectively.

RanMASK is then a special combination of Mask_Pol and Mask_Normal, as tokens of either *polluted* or *normal* are randomly masked out with a predefined probability. On the other hand, the proposed MI4D becomes an effective solution for masking inputs (due to the uncertainty of which tokens being polluted during testing), that is consistently better than RanMASK (randomly mask tokens). Again, the reason is that MI4D includes all by exploiting the fact that polluted tokens are still informative, to some extent, when they are combined with residuals ones, to create a larger convex hull overlapping (compared to RanMASK)

with the original one. That is shown clearly in Proposition 1.

[MASK] or others. The last experiment aims to investigate the possibility of injecting different tokens, instead of [MASK]. Specifically, the [PAD] token is selected and further inserted into the original input sequence. Note that, in this regard, all other configurations (such as the masking budget and the random insertion) remain explicitly the same, but only to replace [MASK] with [PAD] for the injection. Table 3 reports the averaged performance using the SST2 dataset with four attacks. As observed, the performance using [PAD] is similar to that of [MASK], indicating we can insert [MASK] (or similar) as a “wild card” to increase the span of the convex hull.

Table 3: Averaged defending performance via injecting [PAD] (instead of [MASK]) tokens.

	TextFooler			BERT-Attack		
	Cln%	Aua%	Suc%	Cln%	Aua%	Suc%
[MASK]	94.3	36.4	61.4	94.3	34.5	63.4
[PAD]	94.1	36.4	61.4	93.5	32.8	64.9
	Deepwordbug			TextBugger		
	Cln%	Aua%	Suc%	Cln%	Aua%	Suc%
[MASK]	94.4	45.6	51.7	94.2	58.3	38.1
[PAD]	93.2	44.7	52.0	93.1	63.1	32.3

5 Conclusion

We propose a novel adversarial defending algorithm (MI4D), that is hyperparameter insensitive and structure free. The proposed method simply inserts [MASK] tokens at the beginning of input sequences, and follows the normal fine-tuning to train the model. Theoretically speaking, we have argued that adding additional [MASK], while remaining other residual tokens, creates a large convex hull overlapping with that of the clean one to increase the defending probability. Empirically, in comparison to existing state-of-the-arts, the proposed algorithm exhibits superior performance on three benchmark datasets with four attack methods. In future work, we could combine with external knowledge for more strategical masking. More importantly, MI4D is agnostic to downstream tasks, *i.e.*, we could incorporate it into other applications.

Limitations

Our theoretic analysis is constructed on a crucial assumption asserting that the successful defense probability is determined by the volume of the convex hull formed by the input. Although our empirical study results confirmed the inferences based on this

assumption repeatedly (shown in Section 4.4), we are still seeking direct dynamics of the convex hull to the prediction/classification probability where a more rigorous result may be derived. We envisage that the understanding of the current model behavior can lead to more robust models against adversarial attacks and hence further improvement to text classification.

Acknowledgments

This work was partially supported by the Australian Research Council Discovery Project (DP210101426) and the Australian Research Council Linkage Project (LP200201035).

References

- Rongzhou Bao, Jiayi Wang, and Hai Zhao. 2021. [Defending pre-trained language models from adversarial word substitution without performance sacrifice](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3248–3258, Online. Association for Computational Linguistics.
- Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. [Towards robustness against natural language word substitutions](#). In *International Conference on Learning Representations*.
- Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. [Black-box generation of adversarial text sequences to evade deep learning classifiers](#). In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. [Is BERT really robust? a strong baseline for natural language attack on text classification and entailment](#). volume 34, pages 8018–8025.
- Thai Le, Noseong Park, and Dongwon Lee. 2022. [SHIELD: Defending textual neural networks against multiple black-box adversarial attacks with stochastic multi-expert patcher](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6661–6674, Dublin, Ireland. Association for Computational Linguistics.
- Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. [TextBugger: Generating Adversarial Text Against Real-world Applications](#). In *Network and Distributed Systems Security (NDSS) Symposium*.
- Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. [BERT-ATTACK: Adversarial attack against BERT using BERT](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages

- 6193–6202, Online. Association for Computational Linguistics.
- Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. [Searching for an effective defender: Benchmarking defense against adversarial word substitution](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3137–3147, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Qin Liu, Rui Zheng, Bao Rong, Jingyi Liu, ZhiHua Liu, Zhazhan Cheng, Liang Qiao, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. [Flooding-X: Improving BERT’s resistance to adversarial attacks via loss-restricted fine-tuning](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5634–5644, Dublin, Ireland. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#). *arXiv Preprint*, abs/1907.11692.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. [Learning word vectors for sentiment analysis](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.
- Zhao Meng, Yihan Dong, Mrinmaya Sachan, and Roger Wattenhofer. 2022. [Self-supervised contrastive learning with adversarial perturbations for defending word substitution-based attacks](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 87–101, Seattle, United States. Association for Computational Linguistics.
- John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. [TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.
- Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. [InfoBERT: Improving robustness of language models from an information theoretic perspective](#). In *International Conference on Learning Representations*.
- Jianhan Xu, Cenyuan Zhang, Xiaoqing Zheng, Linyang Li, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2022. [Towards adversarially robust text classifiers by learning to reweight clean examples](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1694–1707, Dublin, Ireland. Association for Computational Linguistics.
- Mao Ye, Chengyue Gong, and Qiang Liu. 2020. [SAFER: A structure-free approach for certified robustness to adversarial word substitutions](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3465–3475, Online. Association for Computational Linguistics.
- Jin Yong Yoo and Yanjun Qi. 2021. [Towards improving adversarial training of NLP models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jiehang Zeng, Xiaoqing Zheng, Jianhan Xu, Linyang Li, Liping Yuan, and Xuanjing Huang. 2021. [Certified robustness to text adversarial attacks by randomized \[Mask\]](#). *arXiv preprint*, abs/2105.03743.
- Cenyuan Zhang, Xiang Zhou, Yixin Wan, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. 2022. [Improving the adversarial robustness of NLP models by information bottleneck](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3588–3598, Dublin, Ireland. Association for Computational Linguistics.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. [Character-level convolutional networks for text classification](#). In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15*, page 649–657, Cambridge, MA, USA. MIT Press.
- Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2021. [Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5482–5492, Online. Association for Computational Linguistics.
- Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. [FreeLB: Enhanced adversarial training for Natural Language Understanding](#). In *International Conference on Learning Representations*, pages 26–30, Addis Ababa, Ethiopia.

A Appendix

A.1 Training Details

The RoBERTa-base model (Liu et al., 2019) is employed as the contextual encoder. The dropout rate across all layers is set as 0.1. The Adam optimizer with a dynamic learning rate is adopted, for which the learning rate is warmed up for 10 thousand steps to a maximum value of $2e^{-5}$ before decaying linearly to a minimum value of $1e^{-6}$ (by the cosine annealing) and a gradient clip of $(-1, 1)$. Additionally, for WETAR-D, 50% of samples are polluted in the validation set (the size of 256); for FreeLB++, the number of search steps (for adversarial samples) is 30; for IB, the hidden dimension for the IB layer is set as 150 and the penalty of the IB loss is 0.1; for InfoBERT, the penalty of the mutual-information loss is 5×10^{-2} ; for RanMASK, the masking budget is set as 30%, while the majority vote is adopted for the final classification stage. At last, all models are performed using a machine with NVIDIA Tesla V100 PCIe of 32G GPU memory.

A.2 Impact from the location of inserting [MASK]

The model accuracy is evaluated by adding the [MASK] token in different locations, *i.e.*, either after [CLS] (termed *Head*), randomly (termed *Random*) across the input sequence, or at the end (termed *Tail*). The comparison is shown in Fig 6, and the result illustrates that the proposed method achieves a similar performance regardless of the inserted [MASK] location.

A.3 Impact from the [MASK] budget

The model accuracy is also evaluated as a function of the masking budget. The comparison is shown in Fig 7, and the result illustrates that the proposed method achieves a stable performance regardless of different budgets.

A.4 Result from masking different types of tokens

Fig. 8 shows the comparison of the defending results with different types of tokens being masked. Clearly, masking all polluted but retaining normal tokens leads to the best performance, while masking normal tokens is the worst. The proposed MI4D achieves the competitive outcome by injecting additional [MASK] tokens while keeping others. The result indicates that the larger the insertion between

new convex hull (after masking) with the original one, the better defending performance.

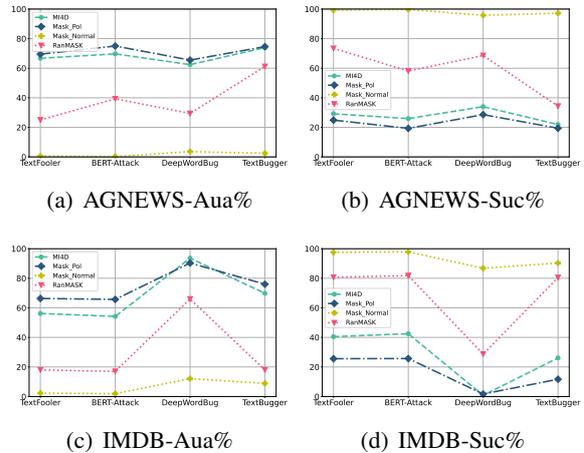


Figure 8: Comparison of the model performance against categories of masked tokens.

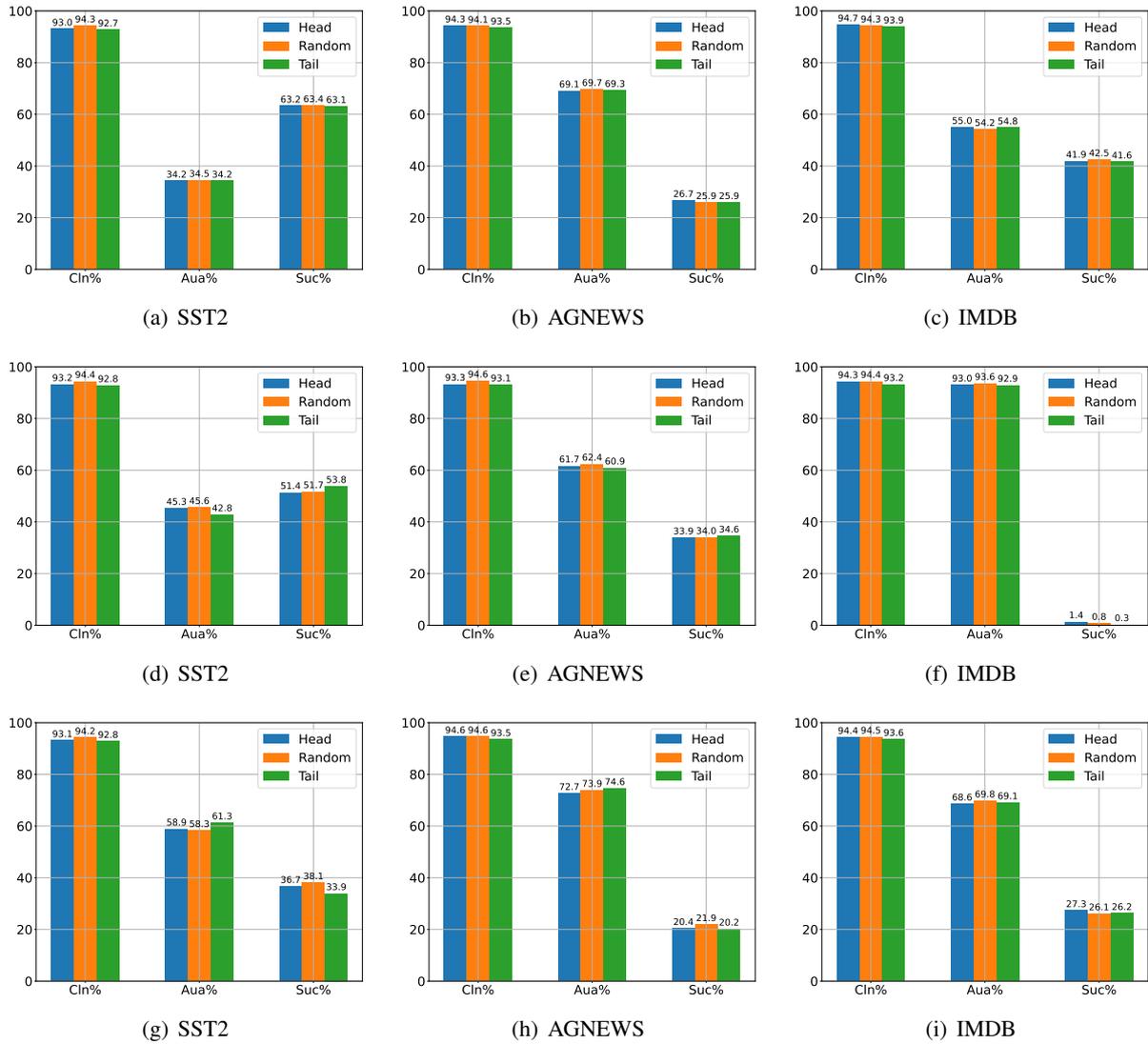


Figure 6: Comparison of the defending performance as a function of inserting [MASK] after [CLS] (labeled as *Head*), randomly (labeled as *Random*), or at the end of the input sequence (labeled as *Tail*). Among them, (a)-(c) is for the BERT-Attack, (d)-(f) is for DeepWordBug, and (g)-(i) is for TextBugger, respectively.

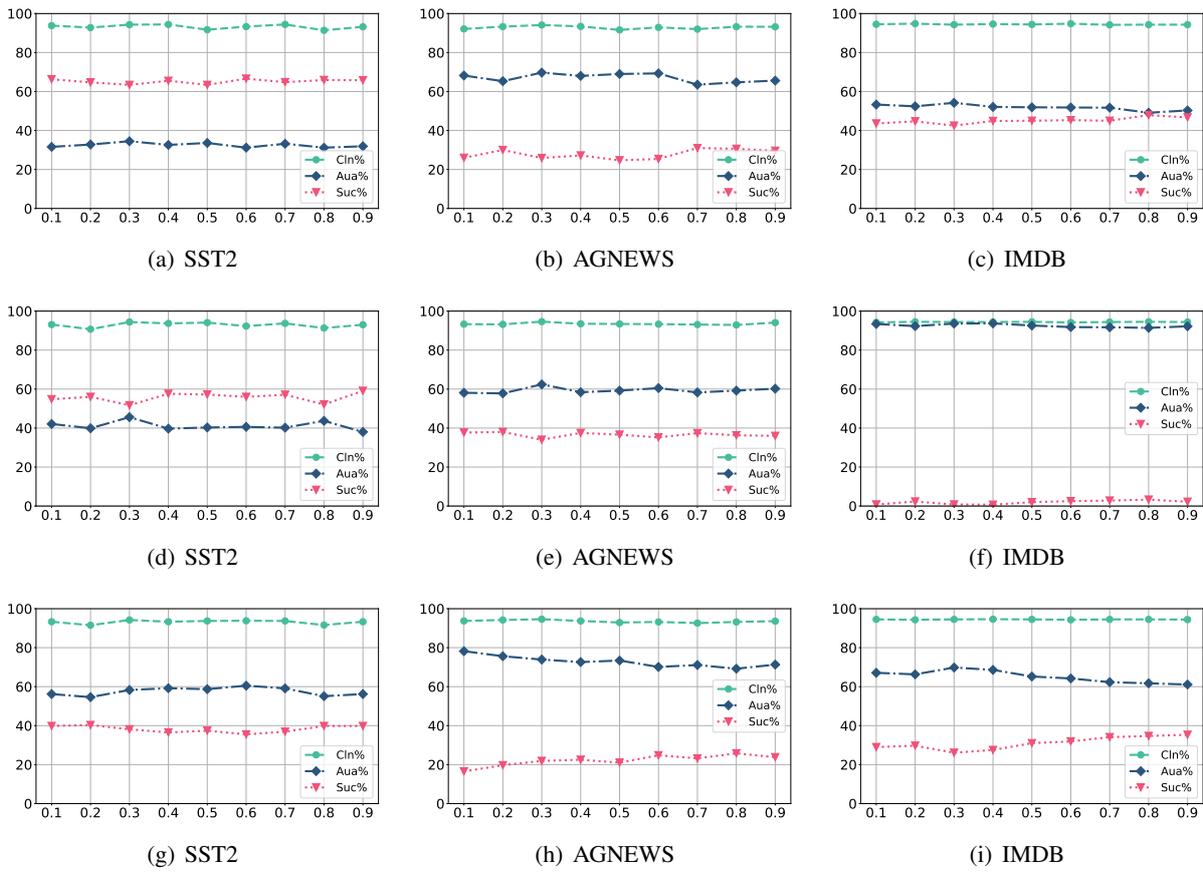


Figure 7: Comparison of the defending performance as a function of masking budget. Among them, (a)-(c) is for the BERT-Attack, (d)-(f) is for DeepWordBug, and (g)-(i) is for TextBugger, respectively.

ViDeBERTa: A powerful pre-trained language model for Vietnamese

Cong Dao Tran *
FPT Software AI Center
daotc2@fsoft.com.vn

Nhut Huy Pham *
FPT Software AI Center
huypn10@fsoft.com.vn

Anh Nguyen
Microsoft
anhnguyen@microsoft.com

Truong Son Hy †
University of California San Diego
tshy@ucsd.edu

Tu Vu
University of Massachusetts Amherst
tuvu@cs.umass.edu

Abstract

This paper presents ViDeBERTa, a new pre-trained monolingual language model for Vietnamese, with three versions - ViDeBERTa_{xsmall}, ViDeBERTa_{base}, and ViDeBERTa_{large}, which are pre-trained on a large-scale corpus of high-quality and diverse Vietnamese texts using DeBERTa architecture. Although many successful pre-trained language models based on Transformer have been widely proposed for the English language, there are still few pre-trained models for Vietnamese, a low-resource language, that perform good results on downstream tasks, especially Question answering. We fine-tune and evaluate our model on three important natural language downstream tasks, Part-of-speech tagging, Named-entity recognition, and Question answering. The empirical results demonstrate that ViDeBERTa with far fewer parameters surpasses the previous state-of-the-art models on multiple Vietnamese-specific natural language understanding tasks. Notably, ViDeBERTa_{base} with 86M parameters, which is only about 23% of PhoBERT_{large} with 370M parameters, still performs the same or better results than the previous state-of-the-art model. Our ViDeBERTa models are available at: <https://github.com/HySonLab/ViDeBERTa>.

1 Introduction

In recent years, pre-trained language models (PLMs) and Transformer-based architecture models have been essential in the advancement of Natural Language Processing (NLP). Large-scale Transformer-based pre-trained models with the capacity to derive a contextual representation of the languages in the training data include GPT (Radford et al., 2019; Brown et al., 2020), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019b), ELECTRA (Clark et al., 2020), T5 (Raffel et al., 2020), and DeBERTa (He

et al., 2020, 2021). Following pre-training, these models performed at the cutting edge on various downstream NLP tasks (Devlin et al., 2019). The development of pre-trained models in other languages, including Vietnamese (PhoBERT (Nguyen and Nguyen, 2020); ViBERT (Tran et al., 2020); ViT5 (Phan et al., 2022)), and Arabic (Antoun et al., 2021), has been spurred on by the success of pre-trained models in English. In order to enhance performance across several languages by learning both general and language-specific representations, multilingual pre-trained models (XLM-R (Conneau et al., 2020), mT5 (Xue et al., 2021), and mBART (Liu et al., 2020) are also being developed.

Most recently, PhoBERT (Nguyen and Nguyen, 2020), the first large pre-trained model for Vietnamese that inherits the RoBERTa (Liu et al., 2019) architecture, has demonstrated the effectiveness of the trained language model compared with current methods modernized in four Vietnamese-specific tasks, including Part of Speech Tagging (POS), Dependency Parsing, Named Entity Recognition (NER), and Natural Language Inference (NLI). Nevertheless, there are still rooms to build an improved pre-trained language model for Vietnamese. Firstly, PhoBERT was pre-trained on a relatively small Vietnamese dataset of 20GB of uncompressed texts, while pre-trained language models can be significantly improved by using more pre-training data (Liu et al., 2019). Secondly, Question answering (QA) is one of the most impactful tasks that has mainly focused on the computational linguistics and artificial intelligence research community within information retrieval and information extraction in recent years. However, there are a few pre-trained models for Vietnamese that produce efficient results in the QA tasks, especially PhoBERT (Nguyen and Nguyen, 2020) and ViT5 (Phan et al., 2022). Last but not least, some previous works point to DeBERTa architecture (He et al., 2020, 2021) using several novel techniques

*: Co-first authors. †: Correspondent author.

that can significantly outperform RoBERTa and improve the efficiency of model pre-training and the performance of downstream tasks in some respects.

Inspired by that, we introduce an improved large-scale pre-trained language model, ViDeBERTa, trained on CC100 Vietnamese monolingual, following the architecture and pre-training methods of DeBERTaV3 (He et al., 2021). We comprehensively evaluate and compare our model with competitive baselines, i.e., the previous SOTA models PhoBERT, ViT5, and the multilingual model XLM-R on three Vietnamese downstream tasks, including POS tagging, NER, and QA. In this work, we focus on two main categories of QA: Machine Reading Comprehension (MRC) and Open-domain Question Answering (ODQA). The experiment results show the performance of our model surpasses all baselines on all tasks. Our main contributions are summarized as follows:

- We present and implement ViDeBERTa with three versions: ViDeBERTa_{small}, ViDeBERTa_{base}, and ViDeBERTa_{large} which are the improved large-scale monolingual language models pre-trained for Vietnamese based on the DeBERTa architecture and pre-training procedure.
- We also conduct extensive experiments to verify the performance of our pre-trained models compared to previous strong models in terms of Vietnamese language modeling. Our empirical results demonstrated the state-of-the-art (SOTA) results on Vietnamese downstream tasks: POS tagging, NER, and QA, thus confirming the effectiveness of our improved pre-trained language model for Vietnamese.
- Our model, ViDeBERTa, which works with *huggingface* and *transformers*, is available to the public. We expect that ViDeBERTa will be an effective pre-trained model for many NLP applications and research in Vietnamese and other low-resource languages.

2 Related work

Pre-trained language models for Vietnamese. PhoBERT (Nguyen and Nguyen, 2020) is the first large-scale PLM for Vietnamese, which has the same architecture as BERT (Devlin et al., 2019) and the same pre-training approach as RoBERTa (Liu et al., 2019) for more robust performance. This

model was trained on a Vietnamese Wikipedia corpus of 20GB word-level texts and produced SOTA results on Vietnamese understanding tasks such as POS, NER, Dependency parsing, and NLI. Following PhoBERT, ViBERT (Tran et al., 2020) and ViELECTRA are public monolingual language models for Vietnamese based on BERT and ELECTRA pre-training techniques (Clark et al., 2020) that are pre-trained on syllable-level Vietnamese textual data. Recent works such as BARTpho (Tran et al., 2021) and ViT5 (Tran et al., 2020) are pre-trained for Vietnamese text summarization.

Fine-tuning tasks. This work utilizes three Vietnamese natural language understanding (NLU) tasks, including POS tagging, NER, and QA, for fine-tuning and evaluating our model’s performance. For POS tagging and NER, PhoBERT still produces better results than ViELECTRA, PhoNLP, and ViT5 (Nguyen and Nguyen, 2020, 2021; Phan et al., 2022). While early QA (Voorhees et al., 1999; Brill et al., 2002; Ferrucci et al., 2010) systems were commonly complex and had many parts, MRC models have evolved and now suggest a simpler two-stage retriever-reader framework (Chen et al., 2017). A context retriever first selects a small subset of passages where some of them contain the answer to the question then a machine reader can carefully review the retrieved contexts and determine the correct answer. The tasks based on QA have gained much attention in recent years in the Vietnamese natural language processing and computational linguistics community. However, to the best of our knowledge, there is only the work (Van Nguyen et al., 2022) that proposes the first Vietnamese retriever-reader QA system employing a transformer-based model (XLM-R) evaluated on the ViQuAD corpus (Nguyen et al., 2020).

3 ViDeBERTa

3.1 Pre-training data

In this work, we use a large corpus CC100 Dataset of 138GB uncompressed texts (Monolingual Datasets from Web Crawl Data) (Conneau et al., 2020) as a pre-training dataset. This corpus includes data for romanized languages and monolingual data for more than 100 languages.

According to Nguyen and Nguyen (2020); Tran et al. (2021), pre-trained language models trained on word-level data can perform better than those trained on syllable-level data for word-level Vietnamese NLP tasks. As a result, we perform word

and sentence segmentation using a Vietnamese toolkit PyVi¹ on the pre-training dataset. After that, we use a pre-trained SentencePiece tokenizer from DeBERTaV3 (He et al., 2021) to segment these sentences with sub-word units, which have a vocabulary of 128K sub-word types.

3.2 Model Architecture

Our model, ViDeBERTa, follows the DeBERTaV3 architecture by He et al. (2021), which is trained using the self-supervise learning objectives of MLM and RTD task and a new weight-sharing Gradient-Disentangled Embedding Sharing (GDES) to enhance the performance of the model. We present three versions of our model, ViDeBERTa_{xsmall}, ViDeBERTa_{base}, and ViDeBERTa_{large} with 22M, 86M, and 304M backbone parameters, respectively.

The details of our model architecture hyper-parameters are listed in Table 1.

Table 1: Statistic of our model hyper-parameters. #layer and #heads denote the numbers of layers and attention heads of ViDeBERTa model versions, respectively.

Model	#layers	#heads	hidden size
ViDeBERTa _{xsmall}	6	12	768
ViDeBERTa _{base}	12	12	768
ViDeBERTa _{large}	24	12	1024

3.3 Optimization

We employ our model based on the DeBERTaV3 implementation from (He et al., 2021). We use Adam (Kingma and Ba, 2015) as the optimizer with weight decay (Loshchilov and Hutter, 2018) and use a global batch size of 8,192 across 32 A100 GPUs (80GB each) and a peak learning rate of 6e-4 for both ViDeBERTa_{xsmall} and ViDeBERTa_{base}, while peak learning rate of 3e-4 was used for ViDeBERTa_{large}. We pre-train ViDeBERTa_{xsmall} and ViDeBERTa_{base} for 500k training iterations and ViDeBERTa_{large} for 250k training iterations.

4 Experiments and Results

4.1 POS tagging and NER

4.1.1 Experimental setup

For POS tagging and NER tasks, we use standard benchmarks of the VLSP POS tagging dataset² and the PhoNER dataset (Truong et al., 2021).

¹<https://pypi.org/project/pyvi/>

²<https://vlsp.org.vn/vlsp2013/eval/ws-pos>

We follow the procedure in Devlin et al. 2019; Nguyen and Nguyen 2020 to fine-tune our pre-trained model for POS tagging and NER tasks. In particular, a linear layer for prediction is appended on top of our model architecture (the last Transformer layer). We then use Adam (Kingma and Ba, 2015) to optimize our model for fine-tuning with a fixed learning rate of 1e-5 and batch size of 16 (He et al., 2021). The final results for each task and each dataset are averaged and reported over five independent runs with different random seeds.

We compare the performance of ViDeBERTa models with the solid baselines, including PhoBERT, XLM-R, and ViT5, for these tasks. Here, XLM-R is a multilingual masked language model pre-trained on 2.5 TB of CommonCrawl dataset of 100 languages, which includes 137GB of Vietnamese texts.

4.1.2 Main results

Model	POS	NER	MRC
	Acc.	F ₁	F ₁
XLM-R _{base}	96.2 [†]	—	82.0 [‡]
XLM-R _{large}	96.3 [†]	93.8*	87.0 [‡]
PhoBERT _{base}	96.7 [†]	94.2*	80.1
PhoBERT _{large}	96.8 [†]	94.5*	83.5
ViT5 _{base1024-length}	—	94.5*	—
ViT5 _{large1024-length}	—	93.8*	—
ViDeBERTa _{xsmall}	96.4	93.6	81.3
ViDeBERTa _{base}	96.8	94.5	85.7
ViDeBERTa _{large}	97.2	95.3	89.9

Table 2: Test results (%) for three tasks POS tagging (POS for short), NER, and MRC on test sets. Note that “Acc.” abbreviates the accuracy. †, *, and ‡ denote scores taken from the PhoBERT paper (Nguyen and Nguyen, 2020), the ViT5 paper (Phan et al., 2022), and the ViQuAD paper (Nguyen et al., 2020), respectively.

Table 2 shows the obtained scores of ViDeBERTa compared to the baselines with the highest reported results. It can be seen clearly that our model produces significantly better results than the baselines and achieves new SOTA performance on both POS tagging and NER tasks.

For POS tagging, ViDeBERTa obtains 0.9% and 0.4% absolute higher accuracy than the large-scale multilingual model XLM-R (Nguyen et al., 2020) and the previous SOTA model PhoBERT (Nguyen and Nguyen, 2020), respectively. Table 2 also shows our ViDeBERTa_{xsmall} obtains 96.4% accuracy that are better than the baseline XLM-R_{large}

and ViDeBERTa_{base} obtains 96.8% that are competitively the same as the PhoBERT_{large}.

For NER, our ViDeBERTa_{large} achieves F₁ score at 95.3% and improves 0.8% absolute higher score than the previous SOTA models ViT5_{base1024-length} and PhoBERT_{large}. Furthermore, ViDeBERTa_{large} and ViDeBERTa_{base} perform 1.5% and 0.7% absolute higher scores than the baseline XLM-R_{large} on the PhoNER corpus.

4.2 Question Answering

4.2.1 Experimental setup

For QA, we evaluate our model on two main tasks: MRC and ODQA. For ODQA, we propose a new framework ViDeBERTa-QA, that uses a BM25 (Robertson et al., 2009) as a retriever and ViDeBERTa as a text reader.

Figure 1 depicts an overview of our ViDeBERTa framework for the Vietnamese Open-domain Question answering task. The statistics of the ViQuAD dataset used for the task, which is introduced by Nguyen et al. (2020), are summarized in Table 3.

Corpus	#article	#passage	#question
Train	138	4,101	18,579
Dev	18	515	2,285
Test	18	493	2,21
Full	174	5,109	23,074

Table 3: Statistics of the ViQuAD dataset for QA. “#article”, “#valid”, and “#test” denote the number of articles, passages, and questions in the ViQuAD, respectively.

We compare ViDeBERTa to the best model XLM-R (Nguyen et al., 2020) and PhoBERT³ for Vietnamese MRC. We also compare our framework, ViDeBERTa-QA, to strong baselines DrQA (Chen et al., 2017), BERTserini (Yang et al., 2019a), and the first Vietnamese ODQA system XLMRQA (Van Nguyen et al., 2022) that uses XLM-R_{large} as a reader. We use the ViQuAD corpus introduced by Nguyen et al. (2020) for assessing these tasks. ViQuAD is a Vietnamese corpus that comprises over 23k triples and each triple includes a question, its answer, and a passage containing the answer.

Similar to POS tagging and NER, we use Adam (Kingma and Ba, 2015) as an optimizer with a learning rate of 2e-5 and a batch size of 16. We report

³We carefully fine-tune PhoBERT for the MRC task following the fine-tuning approach that we use for ViDeBERTa.

the final results as an average over five independent runs with different random seeds.

4.2.2 Main results

Table 2 presents the results obtained by ViDeBERTa and two baselines XLM-R (reported by Nguyen et al. (2020)) and PhoBERT for MRC on ViQuAD corpus. We find that our ViDeBERTa performance outperforms both XLM-R and PhoBERT in terms of F₁ score.

In particular, the previous SOTA model XLM-R_{large} for Vietnamese MRC obtains 87%. Clearly, ViDeBERTa helps boost the XLM-R with about 2.9% absolute improvement, obtaining a new SOTA result at 89.9%. In addition, both versions ViDeBERTa_{base} and ViDeBERTa_{large} also outperform PhoBERT_{base} and PhoBERT_{large} by large margins, respectively. Especially, ViDeBERTa_{xsmall} (22M parameters) produces 1.2% absolute higher score than PhoBERT_{base} (135M parameters) and ViDeBERTa_{base} (86M parameters) produces 2.2% absolute higher score than PhoBERT_{large} (370M parameters) but uses far fewer parameters than PhoBERT.

For ODQA, Table 4 shows the obtained F₁ scores for ViDeBERTa-QA and its baselines on the test set. Obviously, ViDeBERTa-QA achieves better scores than the previous SOTA XLMRQA, BERTserini, and DrQA at the top k passages, selected by retrievers, is 10 and 20. In particular, ViDeBERTa-QA performs 0.85% (at $k = 20$) and 0.4% (at $k = 10$) absolute higher scores than the previous SOTA system. At smaller k ($= 1, 5$), ViDeBERTa performs better BERTserini and DrQA by a large margin; however, XLMRQA does better than ViDeBERTa-QA.

Model	Top k selected passages			
	1	5	10	20
DrQA [*]	37.86	37.86	37.86	37.86
BERTserini [*]	55.55	58.30	57.98	58.09
XLMRQA [*]	61.83	64.99	64.49	64.49
ViDeBERTa _{xsmall}	52.76	56.24	56.93	57.40
ViDeBERTa _{base}	58.55	61.37	61.89	62.43
ViDeBERTa _{large}	61.23	63.57	64.89	65.34

Table 4: Test scores (F₁ in %) for ODQA on ViQuAD corpus with different k values. Note that [*] indicates the results reported following Van Nguyen et al. (2022).

4.3 Discussion

According to the results on both downstream tasks of POS tagging and NER in Table 2, we find that

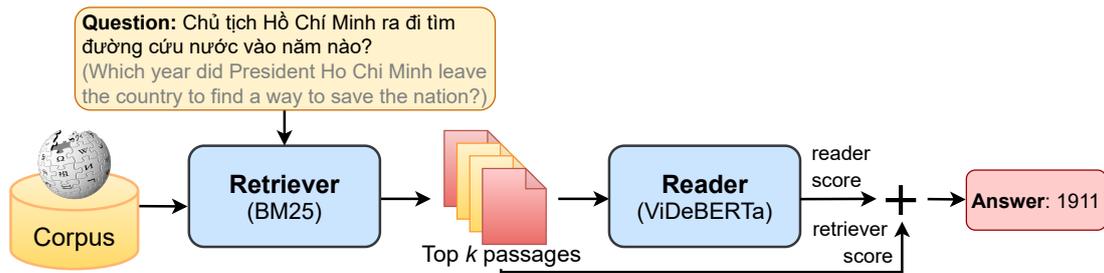


Figure 1: An overview of ViDeBERTa-QA framework for Vietnamese Open-domain Question Answering task.

ViDeBERTa_{small} (86M) with fewer parameters (i.e. only about 15% of XLM-R_{large} 560M and 25% of PhoBERT_{large} 370M) but still performs slightly better than XLM-R_{large} and competitively the same as the previous SOTA PhoBERT_{large}. One possible reason is that our model inherits the robustness of DeBERTaV3 architecture and pre-training techniques, which are demonstrated superior performance by He et al. (2020, 2021). Moreover, using more high-quality pre-training data (138GB) can help ViDeBERTa significantly improve its performance compared to PhoBERT (using 20GB).

For Vietnamese QA, the results on the MRC task show that ViDeBERTa outperforms PhoBERT by a large margin. It is worth noting that PhoBERT set a maximum length of 256 subword tokens for both versions while ViDeBERTa set a larger one of 512. As a result, our models are more scalable than PhoBERT for long contexts. The results obtained by ViDeBERTa-QA on ODQA also suggest that our framework achieves the best performance with large top k passages selected by the retriever (i.e. $k = 10, 20$).

5 Conclusion

In this paper, we have introduced ViDeBERTa, a new pre-trained large-scale monolingual language model for Vietnamese. We demonstrate the effectiveness of our ViDeBERTa by showing that ViDeBERTa with fewer parameters performs better than the recent strong pre-trained language models as XLM-R, PhoBERT, and ViT5, and achieves SOTA performances for three downstream Vietnamese language understanding tasks, including POS tagging, NER, and especially QA. We hope that our public ViDeBERTa model will boost ongoing NLP research and applications for Vietnamese and other low-resource languages.

Limitations

While we have shown that ViDeBERTa can achieve state-of-the-art performance on a variety of NLP tasks for Vietnamese, we believe that more analyses and ablations are required to better understand what facets of ViDeBERTa contributed to its success and what knowledge of Vietnamese that ViDeBERTa captures. We leave these further explorations to future work.

References

- Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Aragpt2: Pre-trained transformer for arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207.
- Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the askmsr question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 257–264.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Asso-*

- ciation for Computational Linguistics*, pages 8440–8451.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Dat Quoc Nguyen and Anh-Tuan Nguyen. 2020. Phobert: Pre-trained language models for vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042.
- Kiet Nguyen, Vu Nguyen, Anh Nguyen, and Ngan Nguyen. 2020. A vietnamese dataset for evaluating machine reading comprehension. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2595–2605.
- Linh The Nguyen and Dat Quoc Nguyen. 2021. Phonlp: A joint multi-task learning model for vietnamese part-of-speech tagging, named entity recognition and dependency parsing. *arXiv preprint arXiv:2101.01476*.
- Long Phan, Hieu Tran, Hieu Nguyen, and Trieu H Trinh. 2022. Vit5: Pretrained text-to-text transformer for vietnamese language generation. *arXiv preprint arXiv:2205.06457*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.
- Nguyen Luong Tran, Duong Minh Le, and Dat Quoc Nguyen. 2021. Bartpho: Pre-trained sequence-to-sequence models for vietnamese. *arXiv preprint arXiv:2109.09701*.
- Thi Oanh Tran, Phuong Le Hong, et al. 2020. Improving sequence tagging for vietnamese text using transformer-based neural models. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 13–20.
- Thinh Hung Truong, Mai Hoang Dao, and Dat Quoc Nguyen. 2021. Covid-19 named entity recognition for vietnamese. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2146–2153.
- Kiet Van Nguyen, Phong Nguyen-Thuan Do, Nhat Duy Nguyen, Tin Van Huynh, Anh Gia-Tuan Nguyen, and Ngan Luu-Thuy Nguyen. 2022. Xlmrqa: Open-domain question answering on vietnamese wikipedia-based textual knowledge source. *arXiv preprint arXiv:2204.07002*.
- Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.
- Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

A Background of DeBERTa

DeBERTa enhances BERT with disentangled attention and a more powerful mask decoder. The disentangled attention mechanism is distinct from prior methods in that it uses two distinct vectors to represent each input word: one for the content and one for the location. The words’ attention weights are calculated using disentangled matrices based on both their relative placements and contents. Similar to BERT, DeBERTa has been pre-trained using masked language modeling. The disentangled attention process already accounts for the relative locations and contents of the context words but not for their absolute positions, which are usually crucial for prediction. DeBERTa improves MLM by utilizing a better mask decoder at the MLM decoding layer and absolute position information of the context words.

A.1 Masked Language model

Large-scale Transformer-based PLMs are often pre-trained using a self-supervision aim called Masked Language Model (MLM) (Devlin et al., 2019) to learn contextual word representations in enormous volumes of text. In further detail, we corrupt a given sequence $X = \{x_i\}$ into \tilde{X} by randomly masking 15% of its tokens and train a language model parameterized by θ to reconstruct X by anticipating the masked tokens \tilde{x} conditioned on \tilde{X} :

$$\max_{\theta} \log p_{\theta}(X|\tilde{X}) = \max_{\theta} \sum_{i \in C} \log p_{\theta}(\tilde{x}_i = x_i|\tilde{X}), \quad (1)$$

where C is the sequence’s index set for the masked tokens. The authors of BERT suggest keeping 10% of the masked tokens unchanged, replacing another 10% with tokens chosen at random, and replacing the remaining tokens with the [MASK] token.

A.2 Replaced token detection

Like ELECTRA, which was trained with two transformer encoders in GAN style, DeBERTaV3 (He et al., 2021) improves DeBERTa by using the training loss in the generator is MLM and discriminator is Replaced Token Detection (RTD). The loss function of the generator can be written as follows:

$$L_{MLM} = \mathbb{E} \left(- \sum_{i \in C} \log p_{\theta_G}(\tilde{x}_{i,G} = x_i|\tilde{X}_G) \right), \quad (2)$$

where θ_G and \tilde{X}_G are the parameter and the input of the generator by masking 15% tokens in X , respectively.

The discriminator’s input sequence is constructed by replacing masked tokens with new tokens sampled according to the generator’s output probability:

$$\tilde{x}_{i,D} = \begin{cases} \tilde{x}_i \sim p_{\theta_G}(\tilde{x}_{i,G} = x_i|\tilde{X}_G), & i \in C \\ x_i, & i \notin C \end{cases} \quad (3)$$

The loss function of the discriminator is written as follows:

$$L_{RTD} = \mathbb{E} \left(- \sum_i \log p_{\theta_D}(\mathbb{1}(\tilde{x}_{i,D} = x_i)|\tilde{X}_D) \right), \quad (4)$$

where θ_D is the parameter of the discriminator, $\mathbb{1}(\cdot)$ is the indicator function, and \tilde{X}_D is the input to the discriminator constructed by Equation 4. Then L_{MLM} and L_{RTD} are optimized jointly by the final loss $L = L_{MLM} + \lambda L_{RTD}$, where λ is the weight of the discriminator loss.

Besides using the RTD training loss like ELECTRA (Clark et al., 2020), DeBERTaV3 improves DeBERTa by using a new weight-sharing method called Gradient-Disentangled Embedding Sharing (GDES) (He et al., 2021). The experimental results conducted by He et al. indicate that GDES is an effective weight-sharing method for language model pre-trained with MLM and RTD tasks.

NapSS: Paragraph-level Medical Text Simplification via Narrative Prompting and Sentence-matching Summarization

Junru Lu¹, Jiazheng Li², Byron C. Wallace³, Yulan He^{1,2,4} and Gabriele Pergola¹

¹Department of Computer Science, University of Warwick, UK

²Department of Informatics, King's College London, UK

³Northeastern University, USA ⁴The Alan Turing Institute, UK

{junru.lu, gabriele.pergola}@warwick.ac.uk

b.wallace@northeastern.edu, {jiazheng.li, yulan.he}@kcl.ac.uk

Abstract

Accessing medical literature is difficult for laypeople as the content is written for specialists and contains medical jargon. Automated text simplification methods offer a potential means to address this issue. In this work, we propose a *summarize-then-simplify* two-stage strategy, which we call NapSS, identifying the relevant content to simplify while ensuring that the original narrative flow is preserved. In this approach, we first generate reference summaries via sentence matching between the original and the simplified abstracts. These summaries are then used to train an extractive summarizer, learning the most relevant content to be simplified. Then, to ensure the narrative consistency of the simplified text, we synthesize auxiliary *narrative prompts* combining key phrases derived from the syntactical analyses of the original text. Our model achieves results significantly better than the seq2seq baseline on an English medical corpus, yielding 3%~4% absolute improvements in terms of lexical similarity, and providing a further 1.1% improvement of SARI score when combined with the baseline. We also highlight shortcomings of existing evaluation methods, and introduce new metrics that take into account both lexical and high-level semantic similarity. A human evaluation conducted on a random sample of the test set further establishes the effectiveness of the proposed approach. Codes and models are released here: <https://github.com/LuJunru/NapSS>.

1 Introduction

The medical literature is vast, and continues to expand quickly. Most patients (laypeople), however, are unable to access this information because it is written for specialists and so dense and laden with jargon. As the recent ‘infodemic’ has shown, access to reliable and comprehensible information about citizens’ health is a fundamental need: for example, a European Health Literacy Survey (HLS-EU) reports that “at least 1 in 10 (12%) respondents

Complex Medical Abstract – ABS
S1. We included two trials involving 754 participants.
S2. One new trial of 660 participants showed the same success rate of vacuum procedure of 98.2% by both methods (risk ratio (RR) 1.00, 95% confidence interval (CI) 0.98 to 1.02).
S3. The two included trials showed significant reductions in the time between applying the vacuum cup and delivery, (one trial (74 women): mean difference (MD) -6.10 minutes, 95% CI -8.83 to -3.37 and the other trial (660 women): with median difference -4.4 minutes, 95% CI -4.8 to -4.0).
S4. The two included trials showed no significant difference in detachment rate (RR 0.85, 95% CI 0.38 to 1.86, 2 studies, 754 women), no significant difference in Apgar score below seven at one minute (RR 1.04, 95% CI 0.51 to 2.09) and five minutes (RR 1.0, 95% CI 0.29 to 3.42), no significant differences in scalp abrasions or lacerations, cephalhematoma, subgaleal hemorrhage and hyperbilirubinemia.
S5. There were no significant differences between the two methods in all secondary outcomes.
S6. The rapid negative pressure application for vacuum assisted vaginal birth reduces duration of the procedure whilst there is no evidence of differences in maternal and neonatal outcomes.
S7. Rapid method of negative application should be recommended for vacuum extraction assisted vaginal delivery.

Plain-Language Summary - PLS
S1. Two good quality randomized controlled trials involving 754 women were identified.
S2. Rapid negative pressure application reduced the duration of the procedure without any evidence of differences in outcomes for the mother or infant.
S3. Rapid method of negative pressure application should be recommended for vacuum extraction assisted vaginal delivery.

Figure 1: A typical sample of Medical Text Simplification task. The abstract and plain-language summary are split into sentences for easy inspection. Key phrases in each sentence, and marks of chosen sentences in reference summary are in bold.

show insufficient health literacy and almost 1 in 2 (47%) has insufficient or problematic health literacy” (Sørensen et al., 2015). Automated text simplification methods offer a potential means to address this issue, and make evidence available to a wide audience as it is published. However, performing paragraph-level simplification of medical texts is a challenging NLP task.

Online medical libraries such as Cochrane library,¹ provide synopses of the medical literature across diverse topics, and manually-written plain language summaries. We are interested in developing accurate automated medical text simplification systems upon those libraries to help timely popularization of medical information to lay audience.

¹<https://www.cochranelibrary.com/>

We show a typical example of a technical abstract and associated simplified summary from a recently introduced paragraph-level medical simplification corpus (Devaraj et al., 2021a) in Figure 1. The sample consists of a technical abstract (ABS) written for experts, and a manually authored Plain-Language Summary (PLS) of the same publication collected from the Cochrane website. The dataset only provides raw abstract-PLS pairs. For easy inspection, we further add sentence splitting and highlight key phrases.

As this example illustrates, a text simplification system needs to first have an overview of the key details reported in the abstract (e.g., that the review synthesizes ‘two trials’) and must also infer that there ‘were no significant differences’ when ‘rapid negative pressure application’ was applied to all participants, and thus that the ‘rapid method should be recommended’. This entails an overall understanding of the key concepts to simplify, while preserving a consistent narrative flow. Built upon this general framing, the system should identify that the most representing sentences in the abstract are sentences 1, 6, 5 and 7. The key challenges here for a model include: (i) identifying the most important content to simplify within the synopsis; (ii) preserving the original narrative flow from a linguistic and medical point of view; (iii) synthesising the findings in a simple and consistent language.

To address these challenges, we propose a *summarize-then-simplify* two-stage framework NapSS—**N**arrative **P**rompting and **S**entence-matching **S**ummarization—for paragraph-level medical text simplification. The *narrative prompt* is designed to promote the factual and logical consistency between abstracts (ABSs) and PLSs, while the *simplification-oriented summarizer* identifies and preserves the relevant content to convey and simplify.

In the first stage, we construct intermediate summaries via sentence matching between the abstract and the PLS sentences based on their Jaccard Distance. This preliminary set of summaries is used to fine-tune a simplification-oriented summarizer which at inference time identifies and extracts the most relevant content to be simplified from the technical abstracts. This extractive summarizer is simplification-aware in that the reference summary is built with PLS ground truth.

In the second stage of simplification, the intermediate summary is concatenated to a narrative

prompt generated by synthesising the main concepts, entities, or events mentioned in text resulting from the syntactic analysis of the PLSs. The prepared input is passed to a seq2seq model (e.g., BART (Lewis et al., 2020)) to produce a plain-language output.

Our contributions can be summarized as follows:

- We introduce NapSS, a two-stage *summarize-then-simplify* approach for paragraph-level medical text simplification, leveraging extractive summarization and narrative prompting.
- We design a *simplification-aware summarizer* and a narrative prompt mechanism. The former is based on a Pre-trained Language Model (PLM) fine-tuned for extractive summarization on an intermediate set of summaries built via sentence matching between the technical and simplified text. The latter synthesises key concepts from the medical text by syntactic dependency parsing analyses, promoting the overall consistency with the narrative flow.
- We conduct a thorough experimental assessment on the Cochrane dataset for paragraph-level medical simplification, evaluating the different features of the generated text (i.e., simplicity and semantic consistency) using several automatic metrics, and the model generalization on sentence-level simplification. Additionally, to mitigate the limitations of the automatic metrics, we designed and conducted a human evaluation assessment, involving “layperson” readers and medical specialists. The results demonstrated the state-of-the-art performance on quality and consistency of the simplified text.

2 Related work

We review three lines of work relevant to this effort: text simplification, extractive summarization, and prompting.

2.1 Text Simplification

Work on text simplification has mainly focused on sentence-level simplification, using the Wikipedia-Simple Wikipedia aligned corpus (Zhu et al., 2010; Woodsend and Lapata, 2011) and the Newsela simplification corpus (Xu et al., 2015). There has been less work on document-level simplification, perhaps owing to a lack of resources (Sun et al., 2021; Alva-Manchego et al., 2019).

The medical domain stands to benefit considerably from automated simplification: The medical literature is vast and technical, and there is a need to make this accessible to non-specialists (Kickbusch et al., 2013). Some research uses those medical documents and deploys various simplification methods based on lexical and syntactic simplification (Damay et al., 2006; Kandula et al., 2010; Llanos et al., 2016). The recent release of the Cochrane dataset provided a new parallel corpus of technical and lay overview of published medical evidence (Devaraj et al., 2021a).

2.2 Extractive Summarization

Extractive summarization aims to select the most important words, sentences, or phrases from input texts and combine them into a summary. Many approaches have been proposed: ranking and selecting sentences based on their graph overlap (Mihalcea and Tarau, 2004), deriving the relevance of the sentences within the text using WordNet (Pal and Saha, 2014), extracting information by named entity recognition (Maddela et al., 2022), and using continuous vector representations to perform semantic matching and sentence selection (Liu and Lapata, 2019; Narayan et al., 2018b; Gui et al., 2019; Lu et al., 2020; Pergola et al., 2021a).

There are some works that focus on extractive summarization of biomedical texts (Mishra et al., 2014; Sun et al., 2022). These have either aimed to provide a summary via graph-based methods or via sequence extraction to present key information in structured (tabular) form (Gulden et al., 2019; Aramaki et al., 2009). In this work we follow a standard *sentence matching* extractive summarization method (Goldstein et al., 1999; Zhong et al., 2020) and fine-tune a pre-trained language model to perform sentence classification. We use extractive summaries as an intermediate step.

2.3 Prompting

Recent work has shown that language models can be *prompted* to perform tasks without supervision (i.e., “zero-shot”) (Radford et al., 2018; Brown et al., 2020). Prompts have been shown to work across a wide range of NLP tasks, e.g., sentiment classification, “reading comprehension”, and “commonsense reasoning” (Seoh et al., 2021; Petroni et al., 2019; Pergola et al., 2021b; Jiang et al., 2019; Lu et al., 2022; Zhu et al., 2022; Wei et al., 2022). Recent work has shown that prompt-based methods can be used even with smaller language models

(Schick and Schütze, 2020; Gao et al., 2020). In this work we focus on a novel use of prompts: Assisting generation of simplified text.

3 Methods

We first define the Paragraph-level Text Simplification task, introducing the relevant notations, and then present the NapSS model.

3.1 Task Formulation

In many cases, text simplification can be viewed as a generative task with additional constraints regarding the simplicity of the generated text. Analogously to text summarization, paragraph-level text simplification can be formulated as follows: for a given *complex* paragraph with M sentences, $\mathbf{x} = \{\{x_1^1, x_2^1, \dots, x_{N_{x_1}}^1\} \dots \{x_1^M, x_2^M, \dots, x_{N_{x_M}}^M\}\}$, the aim is to generate a *plain-language summary* (PLS) $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{N_s}\}$, summarizing and simplifying the original paragraph, with N_{x^m} denoting the length of the m -th sentence \mathbf{x}^m .

3.2 NapSS

We now describe NapSS, a text simplification approach based on a *summarize-then-simplify* two-stage pipeline with the aims of (i) identifying the relevant content to simplify while (ii) ensuring that the original narrative flow is preserved. First, we generate a preliminary summary by using a *simplification-oriented BERT summarizer*, an extractive model fine-tuned beforehand to identify the most relevant content to attend and simplify (§3.2.1). These preliminary summaries are then combined with a *narrative prompt*, a synthetic set of key phrases describing the main concepts, entities, or events discussed in the original text and derived from its syntactic analysis (§3.2.2). The overall working flow of our proposed NapSS model is illustrated in Figure 2. We next provide the details of each of these modules.

3.2.1 Sentence-matching Summarization

The idea behind the summarization stage is to identify the most important content within a given technical abstract (with respect to target simplifications). We automatically construct an intermediate “reference” summary dataset using the simplification training set with which to fit a simplification-oriented summarizer. Specifically, we train the latter as a binary sentence classifier, which provides a simple extractive summarization approach.

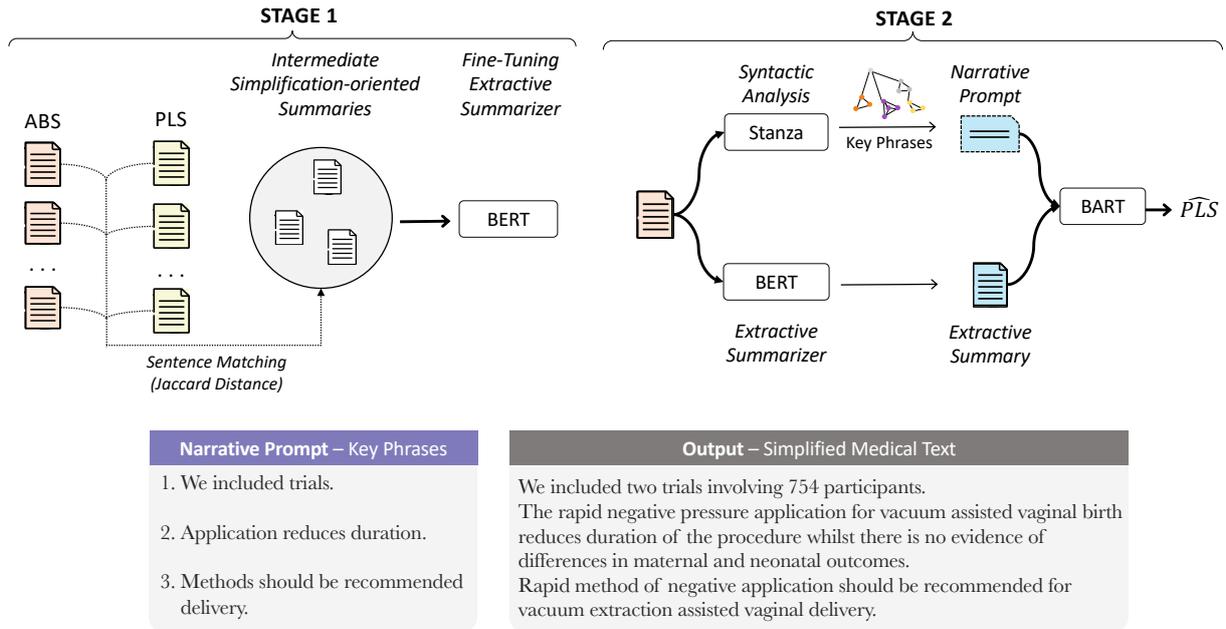


Figure 2: Overview of the two-stage pipeline in the NapSS model. In the first stage, we perform sentence “labelling” using Jaccard Distances (Jaccard, 1912) over abstract (ABS) sentences in reference to PLS sentences, generating a set of intermediate summaries. A binary BERT-based (Devlin et al., 2018) classifier is fine-tuned over these summaries and used, at test time, to generate an extractive summary x' . During the second stage (right side), we perform syntactic dependency parsing over the PLS sentences to extract key phrases k . These are concatenated to form a narrative prompt and combined with the extractive ABS summary to serve as input of the simplification module for the generation of plain-language outputs \hat{PLS} . In the bottom part, we reported an example of narrative prompt and simplified text generated by NapSS on the ABS introduced in Figure 1.

Algorithm 1 details the process of building this pseudo reference summary dataset. The input to the algorithm are the sets of sentences from the technical abstract (ABS) and the corresponding simplified text (PLS). For each PLS sentence, we calculate the Jaccard Distance to every ABS sentence, and select the one with the lowest score. The set of selected ABS sentences constitute an intermediate extractive summary of the technical abstract. The complexity of Algorithm 1 is $O(N_x \cdot N_y \cdot D)$, where D denotes the size of entire corpus.

Based on the intermediate summary dataset, we fine-tune a BERT model to perform binary classification over sentences. At inference time, the resultant trained *simplification-oriented summarizer* is used to select sentences from the technical abstract which will be simplified. These are concatenated and then passed to a BART model (Lewis et al., 2020) along with the narrative prompt.

As an example, the bottom left of Figure 2 shows 3 PLS sentences guiding the automatic labelling (0/1) of 7 ABS sentences. The intermediate extracted summary x' derived via Jaccard matching is used at training time, while at inference time we extract this using the trained model.

3.2.2 Narrative Prompting

Intuitively, the simplification-oriented summarizer should identify the most important content in ABS which should be simplified. However, the similarity matching with which we train the sentence classifier may be noisy and miss relevant information constituting the narrative flow, resulting in errors that lead to omissions in outputs. Therefore, in our NapSS model, we incorporate another simple mechanism, *narrative prompting*, to encourage factual consistency between the input and output.

Inspired by recent work on *chain-of-thought* “reasoning” (Wei et al., 2022), we assume a logical narrative chain can be explicitly constructed with key phrases extracted via syntactic dependency parsing, and then used as a prompt. Specifically, we use a light natural language processing tool Stanza² for dependency parsing on every abstract sentence to extract key phrases. Algorithm 2 details the algorithmic process of our narrative prompting. The algorithm takes abstract sentences as input, runs a dependency parse on each, collects the root token and its closest child tokens to form key phrases in

²<https://stanfordnlp.github.io/stanza/>

Algorithm 1 Build reference summary dataset

- 1: Input require: abstract sentence sets $\{\mathbf{x}_{1\sim M}^m\}$,
- 2: PLS sentence sets $\{\mathbf{y}_{1\sim Q}^q\}$
- 3: Initialization: Empty positive sentence set x_{pos}
- 4: **for** PLS sentence $\mathbf{y}^q \in \{\mathbf{y}_{1\sim Q}^q\}$ **do**
- 5: Initialization: Minimum Jaccard Distance
 $\text{Dist}_q \leftarrow 10.0$,
- 6: corresponding sentence index $\text{Ind}_q \leftarrow 0$
- 7: **for** abstract sentence $\mathbf{x}^m \in \{\mathbf{x}_{1\sim M}^m\}$ **do**
- 8: $\text{Dist}_{qm} = \text{JaccardDistance}(\mathbf{y}^q, \mathbf{x}^m)$
- 9: **if** $\text{Dist}_{qm} < \text{Dist}_q$ **then**
- 10: $\text{Dist}_q \leftarrow \text{Dist}_{qm}$
- 11: $\text{Ind}_q \leftarrow m$
- 12: **end if**
- 13: **end for**
- 14: **if** $\mathbf{x}^{\text{Ind}_q} \notin x_{pos}$ **then**
- 15: add $\mathbf{x}^{\text{Ind}_q}$ in x_{pos}
- 16: **end if**
- 17: **end for**
- 18: Negative sentence set $x_{neg} = \{\mathbf{x}_{1\sim M}^m\} - x_{pos}$

Algorithm 2 Build narrative prompt

- 1: Input require: abstract sentence sets $\{\mathbf{x}_{1\sim M}^m\}$
- 2: Initialization: Empty key phrases queue x_{que}
- 3: **for** abstract sentence $\mathbf{x}^m \in \{\mathbf{x}_{1\sim M}^m\}$ **do**
- 4: $\text{DTree} = \text{DependencyParsing}(\mathbf{x}^m)$
- 5: $\mathbf{x}_{root}^m = \text{DTree.Root}()$
- 6: $\mathbf{x}_{root_l}^m, \mathbf{x}_{root_r}^m = \text{DTree.Children}(\mathbf{x}_{root}^m)$
- 7: $k^m = \mathbf{x}_{root_l}^m \mathbf{x}_{root}^m \mathbf{x}_{root_r}^m$
- 8: add k^m in x_{que}
- 9: **end for**
- 10: Prompt $k^M = k^1 \langle /s \rangle k^2 \langle /s \rangle \dots \langle /s \rangle k^m$

natural linguistic orders, and assembles these as the narrative prompt. Let k^m denotes the key phrase of sentence \mathbf{x}^m , the narrative prompt k^M equals to $[k^1 \langle /s \rangle k^2 \langle /s \rangle \dots \langle /s \rangle k^m]$, in which “ $\langle /s \rangle$ ” is a special separation token. The complexity of this building algorithm is $O(N_x \cdot D)$. As shown in Figure 2, key tokens are shown with bold fonts in every abstract sentences.

3.2.3 Text Simplification

The resulting input of the second text simplification stage is composed by $[k^M \langle /s \rangle \mathbf{x}']$, as depicted in the bottom right part of Figure 2. NapSS adopts encoder-decoder PLM models as the backbone for generative text simplification. Let $L_{gen_{TS}}$ be the

loss of the generative text simplification task:

$$L_{gen_{TS}} = -\frac{1}{N_k + N_{x'}} \sum_{t=1}^{N_k + N_{x'}} y_t \log \hat{y}_t \quad (1)$$

where N_k , $N_{x'}$ are the lengths of the narrative prompt k^M and of the extractive summary \mathbf{x}' , respectively.

4 Experimental Assessment

4.1 Experimental Setup

Dataset We build and evaluate NapSS on the first published paragraph-level medical text simplification dataset (Devaraj et al., 2021a). The dataset is derived from the Cochrane library of systematic reviews and contains 4,459 parallel pairs of technical (ABS) and simplified (PLS) medical abstracts curated by domain experts. The average length of abstract is around 300 to 700 tokens, while the average length of PLS is around 130 to 390 tokens (Devaraj et al., 2021a). All abstract and PLS text are preprocessed to have a total token length lower than 1,024, which is a typical input upper bound of large PLM models. The dataset was split into 3,568 training, 411 development and 480 testing instances. To our knowledge, this is the only accessible paragraph-level text simplification dataset.

For the summarization model, the derived summary dataset contains 51,635 training, 5,856 development, and 7,009 testing sentences (constructed from the respective dataset splits). This dataset contains around 53% positive sentences and 47% negative sentences, which is relatively balanced, and consistent with the proportion of average amount of PLS sentences and average amount of paired abstract sentences. We describe hyperparameter selection in the Appendix Section A.1.

Evaluation Metrics For evaluation we largely adopt the metrics used in prior work on this task and dataset (Devaraj et al., 2021a). These can be placed into three groups: readability metrics, lexical similarity metrics, and simplification metrics. The readability metrics include the Flesch–Kincaid grade level score (FK) (Kincaid et al., 1975) and the automated readability index (ARI) (Senter and Smith, 1967). Lexical similarity metrics are widely adopted to evaluate text generation, including ROUGE-1, ROUGE-2, ROUGE-L (Lin, 2004) and BLEU (Papineni et al., 2002). The simplification metrics include SARI (Xu et al., 2016), which is an editing-base metric especially designed for

Models	Readability		Lexical Similarity				Simplification	Semantic Similarity	Comprehensive
	FK	ARI	Rouge-1	Rouge-2	Rouge-L	BLEU	SARI	BertScore	BLEURT
Vanilla BART	10.89	14.32	46.79	19.23	43.55	11.5	38.72	23.94	-0.194
UL-BART (Devaraj et al., 2021a)	11.97	13.73	38.00	14.00	36.00	39.0	40.00	/	/
UL-BART (by us)	9.30	12.40	43.25	16.36	40.22	7.9	40.08	24.64	-0.309
NapSS (our)	10.97	14.27	48.05	19.94	44.76	12.3	40.37	25.73	-0.155
NapSS BioBART	10.98	14.24	47.66	19.77	44.39	11.9	40.21	25.61	-0.166
NapSS (+UL)	8.67	11.80	45.39	16.77	42.53	9.1	41.12	23.13	-0.219
NapSS (-Prompt)	9.86	13.06	45.62	20.01	44.83	12.1	39.68	25.57	-0.158
NapSS (-Summary)	10.62	13.99	46.91	19.51	44.18	11.8	39.62	25.29	-0.167

Table 1: Overall results on the testing set. UL BART is the previous SOTA, and we report results from our re-implementation of this. The inconsistency between (Devaraj et al., 2021a) and our re-implementation is due to the inavailability of evaluation code. For NapSS, we provide 2 groups of results by changing backbone model of text simplification module. The robustness verification of proposed NapSS is provided in appendix B. We further provide fusion and ablation results based on BART version of NapSS. NapSS (-Prompt) refers to remove the narrative prompt, while NapSS (-Summary) is to replace the abstract summary with full abstract.

text simplification task. In our setting, SARI would reward the *generation* of words occurring only in the paired PLSs, and avoidance of ABS words not occurring in the corresponding PLS.

Simple automated metrics fail to capture semantic agreement between outputs and references. We therefore consider two additional metrics: BertScore (Zhang et al., 2019) and BLEURT (Selam et al., 2020). BertScore was originally designed to evaluate semantic similarity via BERT (Devlin et al., 2018) embeddings. Alva-Manchego et al. (2021) and Devaraj et al. (2022) recently assessed and verified its effectiveness on the text simplification task. BLEURT is a metric finetuned on both lexical BLEU metric and semantic BertScore metric. Along with the automatic assessment, we also conduct a manual (human) evaluation of the simplicity, fluency and factuality whose evaluation criteria are detailed Section §4.2.3.

Prior work did not publicly provide code to perform evaluations beyond computing ROUGE.³ Therefore, we mainly compare results according to our re-implementation of evaluation metrics.

Baseline “Vanilla” BART is a pretrained encoder-decoder architecture, based on transformers, whose auto-regressive decoder made it a suitable a strong baseline for text generation. In ours setting, we adopted a specific checkpoint version⁴ additionally fine-tuned on the XSUM dataset (Narayan et al., 2018a; Devaraj et al., 2021b), providing higher performance on text summarization. The only other model developed for paragraph-level

medical text simplification is *UL-BART* (Devaraj et al., 2022), is also based on BART but integrates an auxiliary “unlikelihood” (UL) penalty to demote generation of technical jargon, which improved the readability and simplicity of outputs compared to the base BART model.

4.2 Results

4.2.1 Automatic Metrics

We report quantitative results in Table 1 comparing the main models and the ablation studies. We notice that UL-BART can generate text which is more readable (lower FK and ARI) and simpler (higher SARI) than the “Vanilla” BART. However, the model struggles to maintain lexical and semantic similarity (lower ROUGE, BLUE, and higher BLEURT) to the human references, perhaps because omitting jargon terms as the modified objective degrades coherence.

By contrast, NapSS improves lexical similarity by 3% to 4% in terms of ROUGE and BLEU scores while maintaining a comparable SARI score. NapSS additionally improves the semantic similarity between the model outputs and the human references at the cost of a slightly higher FK and ARI scores, demonstrating an higher semantic consistency while simplifying the medical text. For the sake of completeness, we also tested whether replacing the “Vailla” BART backbone with a specialised medical PLM, such as BioBART (Yuan et al., 2022), would lead to better performance. Surprisingly, the replacement did not lead to any significant change in any of the adopted metrics.

We further explored the integration of the auxiliary “unlikelihood” (UL) loss in NapSS (+UL), aiming at increasing the degree of simplification

³<https://github.com/Ash0logn/Paragraph-level-Simplification-of-Medical-Texts>

⁴<https://huggingface.co/facebook/bart-large-xsum>

Models	FK	ARI	BLEU	SARI	BLEURT
Vanilla BART	4.91	6.83	9.71	43.47	-0.663
UL-BART (by us)	4.76	7.61	8.75	40.83	-0.654
NapSS (our)	6.32	7.99	10.1	45.78	-0.648
NapSS (+UL)	5.49	8.25	12.8	44.46	-0.553

Table 2: Zero-shot inference results. All above models are only fine-tuned on the Cochrane dataset (2021a), then run zero-shot inference on the TICO-19 testing set.

while preserving semantic consistency. The resulting model yielded further state-of-the-art performance on the overall text simplicity with an increase of $\sim 0.8\%$ in readability and 1.1% in SARI score. NapSS (-Prompt) and (-Summary) refer to two ablation models. The first one removes the narrative prompt, leading to improved readability but decreased simplification (lower SARI). The second one show that the full abstract is necessary for improving the lexical similarity.

We report and discuss in Appendix C the binary classification performance of the extractive summarization module used in stage one.

4.2.2 Out-of-Domain Evaluation

To evaluate the generalization ability of NapSS, we evaluate the model on a *different* medical text simplification dataset: TICO-19 (Shardlow and Alva-Manchego, 2022). Unlike the Cochrane dataset, this is designed for sentence-level simplification and contains over 6k parallel technical and simplified sentences related to COVID-19.

Table 2 reports results. The ‘‘Vanilla’’ BART and UL-BART have the best performance on readability while NapSS yields over $\sim 2\%$ improvement in terms of simplicity. Integrating NapSS with the ‘‘unlikelihood’’ (UL) penalty (NapSS (+UL)) achieves around $\sim 1\text{-}3\%$ boost on lexical and semantic evaluation. The overall results highlight that our approach can preserve a high level of semantic consistency for simplification at the sentence level, yet with slightly reduced readability.

4.2.3 Human Evaluation

We designed and conducted a manual evaluation of the outputs generated by the simplification models to provide additional insights into *fluency* and *factuality*; the latter is especially difficult to assess with existing automatic metrics.

Evaluation Procedure We randomly sampled 100 unsimplified instances (ABSs) from the test set and paired each with simplified outputs gener-

Models	Simplicity	Fluency	Factuality	(Experts)	Overall
UL BART (by us)	1.43	1.53	1.17	0.99	4.13
NapSS	1.12	1.54	1.66	1.28	4.32

Table 3: Human evaluation result by each category.

ated by two models, one from UL-BART (Devaraj et al., 2021a) and one from the proposed NapSS. Each simplified text was assessed by three different annotators. We hired 6 annotators to participate in this evaluation, who are postdoctoral researchers and PhD students in computer science. Each was assigned 100 instances; this took nearly 8 hours to complete. Additionally, we hired two expert annotators who have professional background in the medical domain to obtain a reliable evaluation on the factual consistency between the complex and the simplified text. Annotators were paid \$19 per hour. To ensure that annotators shared a common understanding of our evaluation criteria, we held a tutorial session with detailed instructions and provided 20 instances as a trial run. We then resolved any annotation inconsistencies afterwards.

Evaluation Criteria We followed a previous approach to ask annotators to give numerical scores for each instance (Alva-Manchego et al., 2021). Considering the requirement for the simplification tasks and text styles characterizing medical documents (Devaraj et al., 2022), we separated numerical scores into three aspects: simplicity, fluency and factuality. Annotators can select a numerical rating (from 0, 1, and 2) for each aspect. Appendix A.2 provides details for each category.

Results In Table 3, we present average annotator scores assigned to all aspects. Our model achieves higher overall and average scores on Fluency and Factuality, respectively. UL-BART model got higher score on Simplicity because this model sometimes generates too simple outputs. Simplicity from our evaluation schema only focuses on evaluating the length of the text and the vocabulary. It does not involve the evaluation of the content. Therefore, if the generated text only contains a conclusion from the paragraph, our evaluator would give a higher score on Simplicity. On the contrary, the fluency and factuality aspects focus on evaluation at the context and semantic level, where our model got a higher score in the assessment. As Factuality is an aspect that the evaluation is subject to evaluators’ background knowledge, therefore we selected those instance been given three different

Complex Medical Abstract – ABS	
<p>S1. Five randomized studies involving 1382 patients were included in this review.</p> <p>S2. All the included studies involved advanced (T3 or T4) prostate cancer, had relatively small populations, and were of short duration.</p> <p>S3. Few events were reported and did not assess disease-specific survival or metastatic disease. Only one study (N = 77) evaluated biochemical outcomes.</p> <p>S4. A subgroup analysis found no significant differences in biochemical progression (defined by the authors as PSA \geq 10 ng/mL) between IAS and CAS for Gleason scores 4 - 6, 7, and 8 - 10.</p> <p>S5. For patients with a Gleason score > 6, reduction in biochemical progression favoured the IAS group (RR 0.10, 95% CI 0.01 to 0.67, P = 0.02).</p> <p>S6. Studies primarily reported on adverse events.</p> <p>S7. One trial (N = 43) found no difference in adverse effects (gastrointestinal, gynecomastia and asthenia) between IAS (two events) and CAS (five events), with the exception of impotence, which was significantly lower in the IAS group (RR 0.72, 95% CI 0.56 to 0.92, P = 0.008).</p> <p>S8. Data from RCTs comparing IAS to CAS are limited by small sample size and short duration.</p> <p>S9. There are no data for the relative effectiveness of IAS versus CAS for overall survival, prostate cancer-specific survival, or disease progression.</p> <p>S10. Limited information suggests IAS may have slightly reduced adverse events.</p> <p>S11. Overall, IAS was also as effective as CAS for potency, but was superior during the interval of cycles (96%).</p>	
Plain-Language Summary – PLS – Reference	
<p>S1. Five studies involving 1382 patients were included in this review.</p> <p>S2. All the included studies involved advanced (T3 or T4) prostate cancer. No study was of adequate size and duration.</p> <p>S3. Few events were reported and they did not assess disease-specific survival or metastatic disease.</p> <p>S4. Only one study evaluated biochemical outcomes. Studies primarily reported on adverse events.</p> <p>S5. There are no data for the relative effectiveness of IAS versus CAS for overall survival, prostate cancer specific survival, disease progression, or quality of life.</p> <p>S6. Limited information suggests IAS may have slightly reduced adverse events.</p> <p>S7. <u>In Hering 2000, IAS (18/25 versus 18/18) appears to be slightly more favorable than CAS in controlling impotence.</u></p> <p>S8. Overall, IAS was also as effective as CAS for potency, but was superior during the interval of cycles (96%).</p> <p>S9. More research is needed.</p>	
Plain-Language Summary – PLS – UL+BART	Plain-Language Summary – PLS - NapSS
<p>S1. This review included five randomized controlled trials involving 1382 patients.</p> <p>S2. All the included studies involved advanced prostate cancer, had relatively small numbers of patients, and were of short duration.</p> <p>S3. Few side effects were reported and did not assess disease-specific survival or metastatic disease.</p> <p>S4. Only one of the studies evaluated biochemical progression, and found that IAS was as effective as CAS for potency, but was better during the interval of cycles.</p> <p>S5. Limited information suggests IAS may have slightly reduced side effects, <u>with the exception of impotence, which was lower in the IAS group.</u></p>	<p>S1. Five randomised controlled trials involving 1382 patients were included in this review.</p> <p>S2. All the included studies involved advanced (T3 or T4) prostate cancer, had relatively small populations, and were of short duration.</p> <p>S3. There are no data for the relative effectiveness of IAS versus CAS for overall survival, prostate cancer-specific survival, or disease progression.</p> <p>S4. Overall IAS was also as effective as CAS in reducing the risk of adverse events.</p>

Figure 3: Case study and error analysis on a typical example from the testing set. Smearred sentences illustrate factual improvement by NapSS, while underlined parts reveal information omission of our model outputs.

scores from basic evaluators to create an experts set. We can see experts’ evaluation also shows the same trend. We believe the narrative prompt benefits this improving. Our model tends to produce a reasonable reduction in the context while keeping the majority of critical points. It is also useful for the model to calibrate grammar and plausibility with prompts. Combined with narrative prompt, NapSS generates simplification more consistent with the original text than the UL-BART. We can observe the better performance on human evaluation results also correlated with the improvement in semantic and comprehensive metrics, which proves the necessity of semantic level simplification evaluation.

4.2.4 Case Study and Error Analysis

We present a case study and error analysis based on the examples reported in Figure 3.⁵ The Abstract (ABS) mentions the the analysis of 5 studies on the effects of the continuous (CAS) or intermittent (IAS) androgen suppression therapy on advanced prostate cancer. The UL-BART model generated a slightly longer simplified text than NapSS. Specifically, sentence 4 from the UL-BART output mixed and linked the biochemical progression assessment with the IAS and CAS side-effect for potency. In contrast, sentence 3 generated by NapSS is more

⁵Better with colors

relevant to the findings of all studies considered.

On the other hand, the last sentence 5 from the UL output reported a meaningful finding in consistency with reference sentence 7 from the PLS and reference sentence 7 from the ABS. NapSS instead omitted this information, probably because the related PLS sentences were not considered sufficiently relevant by the model.

5 Conclusions

We proposed a *summarize-then-simplify* two-stage model—NapSS—for paragraph-level medical text simplification. The first component is a “simplification-oriented” summarizer, which we trained over a heuristically derived set of “psuedo” references derived via sentence matching. At inference time, the summarizer extracts the most relevant content to be simplified. This is combined with an additional “narrative prompt” intended to promote consistency, and then passed to an encoder-decoder model to produce the simplified text. Experiments on a paragraph-level medical text simplification showed that, under several automatic metrics and human evaluation (involving “laypeople” and medical specialists), this method realized significant improvements with respect to both simplification quality and consistency.

Limitations

Our study is primarily based on the Cochrane paragraph-level medical text simplification dataset (Devaraj et al., 2021a). While this dataset provides richer and more elaborated text than previous sentence-level medical datasets, such as TICO-19 (Shardlow and Alva-Manchego, 2022), it is worth noting that experimental documents tend to share a common pattern whose structure consists of: (i) discussing statistics about the clinical trials considered, (ii) list the experimental assessments, (iii) summarize the conclusions of the related findings.

Despite the already significant difficulty of the task, a limited variety of documents would inevitably introduce linguistic bias, hindering the model generalization and our current ability to conduct thorough assessment of the methodologies.

Moreover, although we made effort to examine the factuality aspect with expert annotators, we acknowledge that factuality is a subjective aspect and existing methods may not be sufficient to verify.

Ethics Statement

This work is based on publicly available medical datasets (Devaraj et al., 2021a; Shardlow and Alva-Manchego, 2022). As stated by the authors of datasets, no personal identification information were released. Current language technologies generally—and automated simplification models such as the one proposed in this work—still introduce “hallucinations” and factual inaccuracies into outputs; at present we would therefore recommend against deploying fully automated generative models for medical texts.

Acknowledgment

This work was supported in part by the UK Engineering and Physical Sciences Research Council (EP/T017112/1, EP/V048597/1, EP/X019063/1), and the National Science Foundation (NSF) grant 1750978. YH is supported by a Turing AI Fellowship funded by the UK Research and Innovation (EP/V020579/1). This work was conducted on the UKRI/EP SRC HPC platform, Avon, hosted in the University of Warwick’s Scientific Computing Group. BCW was supported in this work by the National Institutes of Health (NIH), grant R01-LM012086. GP, JL, and JL, were supported by the National AI Strategy award (Warwick/ATI): ‘METU: An Inclusive AI-Powered Framework Making Text Easier to Understand’.

References

- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2019. [Cross-sentence transformations in text simplification](#). In *Proceedings of the 2019 Workshop on Widening NLP*, pages 181–184, Florence, Italy. Association for Computational Linguistics.
- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (Un)Suitability of Automatic Evaluation Metrics for Text Simplification. *Computational Linguistics*, 47(4):861–889.
- Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Masuichi, and Kazuhiko Ohe. 2009. Text2table: Medical text summarization system based on named entity recognition and modality identification. In *Proceedings of the BioNLP 2009 Workshop*, pages 185–192.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Jerwin Jan S Damay, Gerard Jaime D Lojico, Kimberly Amanda L Lu, and Dex B Tarantan. 2006. Simtext: text simplification of medical literature.
- A. Devaraj, I. Marshall, B. Wallace, and J. J. Li. 2021a. Paragraph-level simplification of medical texts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Ashwin Devaraj, Iain Marshall, Byron Wallace, and Junyi Jessy Li. 2021b. [Paragraph-level simplification of medical texts](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4972–4984, Online. Association for Computational Linguistics.
- Ashwin Devaraj, William Sheffield, Byron C Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. *arXiv preprint arXiv:2204.07562*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. [Making pre-trained language models better few-shot learners](#). *CoRR*, abs/2012.15723.

- Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 121–128.
- Lin Gui, Jia Leng, Gabriele Pergola, Yu Zhou, Ruifeng Xu, and Yulan He. 2019. [Neural topic model with reinforcement learning](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3478–3483, Hong Kong, China. Association for Computational Linguistics.
- Christian Gulden, Melanie Kirchner, Christina Schütler, Marc Hinderer, Marvin Kampf, Hans-Ulrich Prokosch, and Dennis Toddenroth. 2019. Extractive summarization of clinical trial descriptions. *International journal of medical informatics*, 129:114–121.
- Paul Jaccard. 1912. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2019. [How can we know what language models know?](#) *CoRR*, abs/1911.12543.
- Sasikiran Kandula, Dorothy Curtis, and Qing Zeng-Treitler. 2010. A semantic and syntactic text simplification tool for health content. In *AMIA annual symposium proceedings*, volume 2010, page 366. American Medical Informatics Association.
- Ilona Kickbusch, Jürgen M. Pelikan, and Franklin Apfel Agis D. Tsouros. 2013. Health literacy : the solid facts.
- J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Yang Liu and Mirella Lapata. 2019. [Text summarization with pretrained encoders](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.
- Leonardo Campillos Llanos, Dhouha Bouamor, Pierre Zweigenbaum, and Sophie Rosset. 2016. Managing linguistic and terminological variation in a medical dialogue system. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 3167–3173.
- Junru Lu, Gabriele Pergola, Lin Gui, Binyang Li, and Yulan He. 2020. CHIME: Cross-passage hierarchical memory network for generative review question answering. pages 2547–2560.
- Junru Lu, Xingwei Tan, Gabriele Pergola, Lin Gui, and Yulan He. 2022. Event-centric question answering via contrastive learning and invertible event transformation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2377–2389, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Mounica Maddela, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2022. [EntSUM: A data set for entity-centric extractive summarization](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3355–3366, Dublin, Ireland. Association for Computational Linguistics.
- Rada Mihalcea and Paul Tarau. 2004. [TextRank: Bringing order into text](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.
- Rashmi Mishra, Jiantao Bian, Marcelo Fiszman, Charlene R Weir, Siddhartha Jonnalagadda, Javed Mostafa, and Guilherme Del Fiol. 2014. Text summarization in the biomedical domain: a systematic review of recent research. *Journal of biomedical informatics*, 52:457–467.
- Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018a. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. [Ranking sentences for extractive summarization with reinforcement learning](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.
- Alok Ranjan Pal and Diganta Saha. 2014. [An approach to automatic text summarization using wordnet](#). In *2014 IEEE International Advance Computing Conference (IACC)*, pages 1169–1173.

- Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Gabriele Pergola, Lin Gui, and Yulan He. 2021a. A disentangled adversarial neural topic model for separating opinions from plots in user reviews. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2870–2883, Online. Association for Computational Linguistics.
- Gabriele Pergola, Elena Kochkina, Lin Gui, Maria Liakata, and Yulan He. 2021b. Boosting low-resource biomedical QA via entity-aware masking strategies. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1977–1985, Online. Association for Computational Linguistics.
- Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.
- Timo Schick and Hinrich Schütze. 2020. [Exploiting cloze questions for few-shot text classification and natural language inference](#). *CoRR*, abs/2001.07676.
- Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.
- RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, Cincinnati Univ OH.
- Ronald Seoh, Ian Birle, Mrinal Tak, Haw-Shiuan Chang, Brian Pinette, and Alfred Hough. 2021. Open aspect target sentiment classification with natural language prompts. *arXiv preprint arXiv:2109.03685*.
- Matthew Shardlow and Fernando Alva-Manchego. 2022. Simple tico-19: A dataset for joint translation and simplification of covid-19 texts. In *Proceedings of the 13th Language Resources and Evaluation Conference, Marseille, France, June. European Language Resources Association*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.
- Kristine Sørensen, Jürgen M Pelikan, Florian Röhlin, Kristin Ganahl, Zofia Slonska, Gerardine Doyle, James Fullam, Barbara Kondilis, Demosthenes Agrafiotis, Ellen Uiters, et al. 2015. Health literacy in europe: comparative results of the european health literacy survey (hls-eu). *European journal of public health*, 25(6):1053–1058.
- Renliang Sun, Hanqi Jin, and Xiaojun Wan. 2021. [Document-level text simplification: Dataset, criteria and baseline](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7997–8013, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhaoyue Sun, Jiazheng Li, Gabriele Pergola, Byron Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. PHEE: A dataset for pharmacovigilance event extraction from text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5571–5587, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Kristian Woodsend and Mirella Lapata. 2011. [Learning to simplify sentences with quasi-synchronous grammar and integer programming](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. [Problems in current text simplification research: New data can help](#). *Transactions of the Association for Computational Linguistics*, 3:283–297.
- Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.
- Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.
- Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaying Zhang, Yutao Xie, and Sheng Yu. 2022. Biobart: Pretraining and evaluation of a biomedical generative language model. *arXiv preprint arXiv:2204.03905*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. [Extractive summarization as text matching](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Lixing Zhu, Zheng Fang, Gabriele Pergola, Robert Procter, and Yulan He. 2022. [Disentangled learning of stance and aspect topics for vaccine attitude detection in social media](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1566–1580, Seattle, United States. Association for Computational Linguistics.

Zheming Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. [A monolingual tree-based translation model for sentence simplification](#). In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.

A Experimental Setup

A.1 Hyperparameters

For the summarization stage, we adopt NLTK⁶ for the building of reference summary dataset, and fine-tune a distilbert-base-uncased-finetuned-sst-2-english⁷ PLM as the classifier. The chosen PLM is a distilbert-base-uncased(Sanh et al., 2019) checkpoint additionally fine-tuned on SST-2 dataset(Socher et al., 2013), which is a sentiment binary classification corpus. The hidden size of the checkpoint is 768 and the corresponding vocabulary size is 30,522. The random seed is 42. The batch size is set to 16 and the accumulation steps is set to 1 on 2 quadro_rtx_6000 GPUs. The optimizer is BertAdam⁸ with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-6$. The weight of decay is 0.01. The learning rate is $2e-5$ without warmup. It takes 0.5 hour in total to fine-tune the checkpoint on the training set, and predict over development and testing sets.

And for the simplification stage, except for possible replacement of backbone encoder-decoder PLM, we adopt exact same settings with the SOTA baseline (Devaraj et al., 2021a), including training strategy and sampling method during the predictive generation. It takes less than 20 mins to fine-tune the PLM, while requires 2 hours to generate simplified text over entire testing set on same GPUs.

⁶<https://www.nltk.org/>

⁷<https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english>

⁸<https://github.com/google-research/bert/blob/master/optimization.py>

A.2 Annotation Schema

To overcome the aforementioned limitations on evaluation metrics, we followed a previous approach to ask our annotators give numerical scores for each instances (Alva-Manchego et al., 2021). Considering the requirement on simplification task and feature of text in medical domains (Devaraj et al., 2022), we designed our numerical scores into three aspects: Simplicity, Fluency and Factuality. Annotator can select one numerical score under each aspect, which include three options 0,1 and 2. Higher score stands for annotator consider the paragraph level performance under that aspect is excellent, vice versa. In here, we provide detail explanation of each aspect.

Simplicity aspect considers how simple that text is to read. This category assess the generated text by annotator’s impression of simplicity, in terms of length of the texts and use of vocabulary. A good simplified text is expected to omit unnecessary numerical descriptions and explain jargons that are hard to be understood by layman readers.

Fluency aspect considers the how fluent the text is. That is, to assess the simplified text by annotator’s impression on connectivity and fluency. A good simplified paragraph should consider the fluency among sentences, such as use of conjunction words or adversative words for sentences. This category also includes the evaluation on overall grammar correctness of each sentences, and penalty on duplicate sentences generated by the model.

Factuality considers how consistent is the simplified text with the original text. This category requires annotators to assess the generated text by compare the facts that mentioned from the original text and those included in the generated text. A good simplified text should includes all the important information appears in the original text. Any paraphrase on the simplified text that lead to different meaning and against the original texts, or any omits on important information should consider to give penalize under this category.

B Robustness of NapSS

We finetune our NapSS model with another two random seeds 123 and 2023. The results of three experiments in 4 share high similarity, confirming the robustness of our proposed pipeline.

Models	Readability		Lexical Similarity				Simplification	Semantic Similarity	Comprehensive
	FK	ARI	Rouge-1	Rouge-2	Rouge-L	BLEU	SARI	BertScore	BLEURT
NapSS (seed=42)	10.97	14.27	48.05	19.94	44.76	12.3	40.37	25.73	-0.155
NapSS (seed=123)	10.89	14.17	48.38	20.24	45.11	12.5	40.36	25.67	-0.149
NapSS (seed=2023)	10.85	14.09	48.29	20.09	45.02	12.4	40.31	25.60	-0.148

Table 4: Robustness checking of our NapSS.

C Summarizer Results

We fine-tuned two different bert-based classifiers, the aforementioned distillbert one, and another BioLinkBERT-base⁹, which is a bert-based model pretrained on PubMed abstracts concerning citation links (Yasunaga et al., 2022). Although the BioLink backbone was pretrained on medical corpus, the general Distillbert fine-tuned on similar binary classification dataset performed better.

Models	Accuracy	F1
BioLinkBERT-base	61.91	67.04
Distilbert-base-uncased-finetuned-sst-2-english	62.50	68.91

Table 5: Performance on the constructed testing set.

⁹<https://huggingface.co/michiyasunaga/BioLinkBERT-base>

Long-tailed Extreme Multi-label Text Classification by the Retrieval of Generated Pseudo Label Descriptions

Ruohong Zhang and Yau-Shian Wang
ruohongz,yaushiaw@andrew.cmu.edu

Yiming Yang
yiming@cs.cmu.edu

Donghan Yu
dyu2@cs.cmu.edu

Tom Vu
tom.m.vu@gmail.com

Likun Lei
llei@flexport.com

Abstract

Extreme Multi-label Text Classification (XMTC) has been a tough challenge in machine learning research and applications due to the sheer sizes of the label spaces and the severe data scarcity problem associated with the long tail of rare labels in highly skewed distributions. This paper addresses the challenge of tail label prediction by leveraging the power of dense neural retrieval model in mapping input documents (as queries) to relevant label descriptions. To further enhance the quality of label descriptions, we propose to generate pseudo label descriptions from a trained bag-of-words (BoW) classifier, which demonstrates better classification performance under severe scarce data conditions. The proposed approach achieves the state-of-the-art (SOTA) performance of overall label prediction on XMTC benchmark datasets and especially outperforms the SOTA models in the tail label prediction. We also provide a theoretical analysis for relating the BoW and neural models w.r.t. performance lower bound.

1 Introduction

Extreme multi-label text classification (XMTC) is the task of tagging documents with relevant labels in a very large and often skewed candidate space. It has a wide range of applications, such as assigning subject topics to news or Wikipedia articles, tagging keywords for online shopping items, classifying industrial products for tax purposes, etc.

The most difficult part in solving the XMTC problem is to train classification models effectively for the rare labels in the long tail of highly skewed distributions, which suffers severely from the lack of sufficient training instances. Efforts addressing this challenge by the text classification community include Bayesian modeling of graphical dependencies among labels (Gopal and Yang, 2010; Gopal et al., 2012), novel loss or regularization of label embeddings (Babbar and Schölkopf, 2019a; Wei

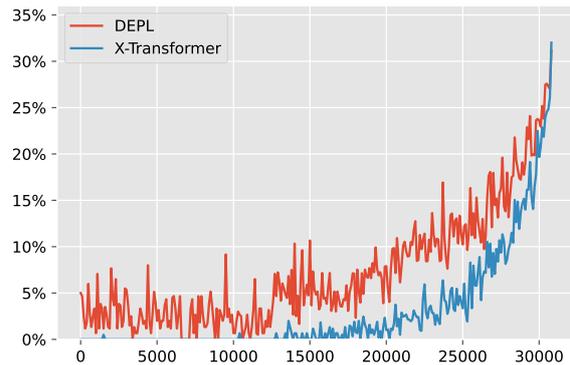


Figure 1: The classification performance of X-Transformer and DEPL (ours) measured in macro-averaged $F1@19$ on the Wiki10-31K dataset.

et al., 2021), clustering-based algorithms (Chang et al., 2020; Khandagale et al., 2019; Prabhu et al., 2018), and so on. Despite the remarkable progresses made so far, the problem is still very far from being well solved. Figure 1 shows the performance of X-Transformer (Chang et al., 2020), one of the state-of-the-art (SOTA) XMTC models, on the Wiki10-31K benchmark dataset (with over 31k labels). The horizontal axis is the ranks of the labels sorted from rare to common and the vertical axis is the text classification performance measured in macro-averaged $F1@19$ (higher the better) for binned labels (100 labels per bin). The blue curve is the result of X-Transformer, which has the scores close to 0 (worst possible score) for nearly half of the total labels. In other words, SOTA methods in XMTC still perform poorly in tail label prediction.

In this paper, we seek solutions for tail label prediction from a new angle: we introduce a novel framework, namely the Dual Encoder with Pseudo Label (DEPL). It treats each input document as a query and uses a neural network model to retrieve relevant labels from the candidate space based on the textual descriptions of the labels. The underlying assumption is, if the label descriptions are

highly informative for text-based matching, then the retrieval system should be able to find relevant labels. The system would be particularly helpful for tail label prediction as the retrieval effectiveness does not necessarily rely on the availability of a large number of training instances, which is what the tail labels are lacking.

The next research question that we tackle is how to obtain highly informative descriptions for each label without human annotation. In reality, class names are often available but they are typically one or two words, which cannot be sufficient for retrieval-based label prediction. Therefore, we propose to augment the label description with statistical learning algorithms. Specifically, we train linear support vector machine (SVM) model with the bag-of-words (BoW) features, such as tf-idf, to automatically generate informative keywords for each label, which we call the *pseudo description* of the label. Since the learned label embeddings of the BoW classifier encode token importance information, it is natural and efficient to leverage them for keywords extraction. In sections 4 and 6, we further provide theoretical motivations and empirical evidence to show the advantage of unsupervised statistical features for classification under extreme scarce data conditions.

The result of our approach (DEPL) is shown as the red curve in Figure 1, which significantly outperforms the blue curve of X-Transformer not only in the tail-label region but also in all other regions. We also observed similar improvements by DEPL over strong baselines on other benchmark datasets (see section 6). Our main contributions are summarized as the following:

1. We propose DEPL, a retrieval-based model to alleviate the difficulty in tail label prediction by matching the semantics between documents and augmented label descriptions which are generated automatically by a statistical model with BoW features.
2. We provide theoretical analyses to motivate the usage of BoW feature for classification under scarce data setting, and prove a performance lower bound of the neural model.
3. We did extensive experiments with different tail label evaluation metrics to show that our method significantly and consistently outperforms strong baselines on multiple challenging benchmark datasets.

2 Related Work

XMTC Classifier Traditional BoW classifiers rely on the bag-of-words features such as one-hot vector with tf-idf weights, which capture the word importance in a document. Examples include one-vs-all SVM models such as DiSMEC (Babbar and Schölkopf, 2017), ProXML (Babbar and Schölkopf, 2019b), PPDSparse (Yen et al., 2017), tree-based models such as Parabel (Prabhu et al., 2018) and Bonsai (Khandagale et al., 2019).

To compensate for the lack of semantics in BoW features, deep learning models were proposed for XMTC. Examples include CNN-based models such as XML-CNN (Liu et al., 2017) and SLICE (Jain et al., 2019), RNN-based models such as AttentionXML (You et al., 2018) and Transformer-based models such as X-Transformer (Chang et al., 2020), LightXML (Jiang et al., 2021) and APLC-XLNet (Ye et al., 2020).

Label Description The SiameseXML (Dahiya et al., 2021) for XMTC encodes both input documents and label descriptions with pretrained word embeddings with shallow networks and leverages the embedding matching. The SOTA pretrained Transformer-based models (Chang et al., 2020; Jiang et al., 2021) leverage the label descriptions to build label clusters. To generate label descriptions, Chai et al. (2020) adopt reinforcement learning to produce extended label descriptions from predefined label descriptions. However, the algorithm can not scale to the extreme label space and relies on the availability of sufficient training data.

3 Proposed Method

3.1 Preliminaries

Let $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^{N_{\text{train}}}\}$ be the training data where \mathbf{x}_i is the input text and $\mathbf{y}_i \in \{0, 1\}^L$ are the binary ground truth labels of size L . Given an instance \mathbf{x} and a label l , a classification system produces a matching score of the text and label:

$$f(\mathbf{x}, l) = \langle \phi(\mathbf{x}), \mathbf{w}_l \rangle$$

where $\phi(\mathbf{x})$ represent the document feature vector and \mathbf{w}_l represents the label embedding of l . The dot product $\langle \cdot, \cdot \rangle$ is used as the similarity function.

Typically, the label embedding \mathbf{w}_l is randomly initialized and trained from the supervised signal. While learning the embedding as free parameters is expressive when data is abundant, it could be difficult to be optimized under the scarce data situation.

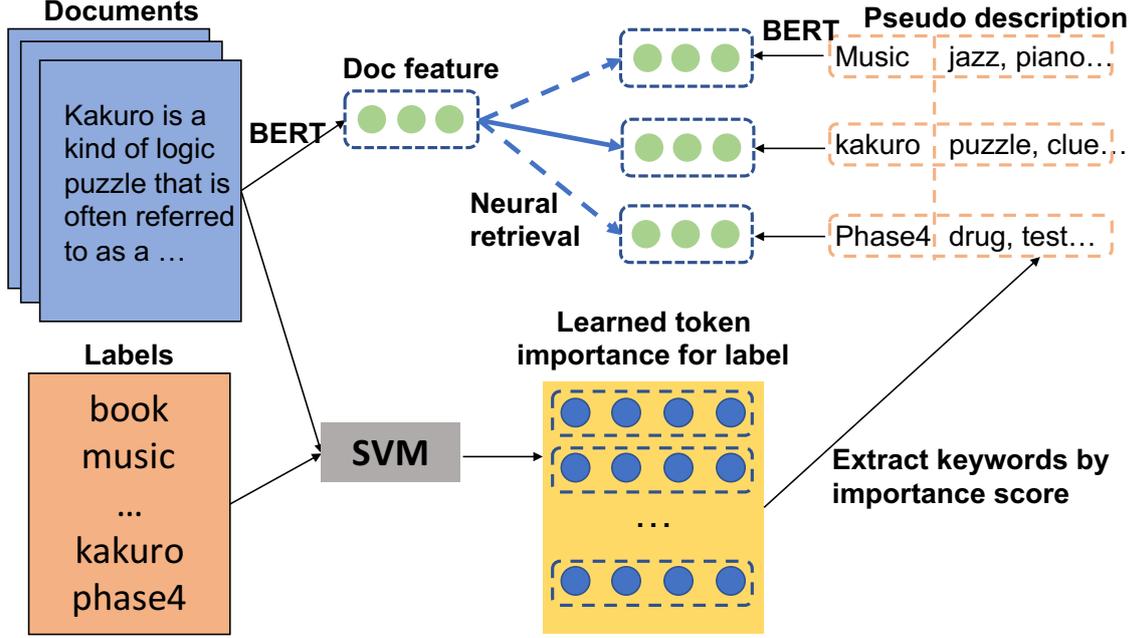


Figure 2: The proposed DEPL framework. First, we train a BoW classifier (SVM) and extract the top keywords from the label embeddings according to the learned token importance. Then, we concatenate the keywords with the original label names to form pseudo descriptions. Finally, we leverage the neural retrieval model to rank the labels according to semantic matching between document text and label descriptions.

Sketch of Method DEPL tackles the long-tailed XMTTC by neural retrieval with generated pseudo label descriptions, as shown in figure 2. Instead of learning the label embedding from scratch, the retrieval module directly leverages the **semantic matching** between the document and label text, providing a strong inductive bias on tail label prediction. Next, we introduce the components of our system in details.

3.2 Generated Pseudo Label Description

As the provided label names are usually short and noisy, we augment it with generated pseudo label description from a SVM model. As the tf-idf features $\phi_t(\mathbf{x})$ used by SVM are sparse, we also call the statistical model a *sparse* model:

$$f_{\text{sparse}}(\mathbf{x}, l) = \langle \phi_t(\mathbf{x}), \mathbf{w}_l^{\text{svm}} \rangle$$

The label embedding weight $\mathbf{w}_l^{\text{svm}}$ is optimized with the hinge loss:

$$\mathcal{L}_{\text{hinge}} = \frac{1}{LB} \sum_{i=1}^B \sum_{l=1}^L \max(0, 1 - \tilde{y}_l \cdot f_{\text{sparse}}(\mathbf{x}_i, l))$$

where $\tilde{y}_l = 2y_l - 1 \in \{-1, 1\}$, B is the batch size.

For a trained SVM model, $\mathbf{w}_l^{\text{svm}}$ has the dimension equal to the vocabulary size and each value

w_i^{svm} of the label embedding denotes the learned importance of the token i w.r.t label l . We select the top k most important tokens (ranked according to the importance score) as keywords, which are appended to the original label name to form the pseudo label description:

$$\text{pseudo_label}(l) = \text{label_name}(l) \oplus \text{keywords}(l)$$

where \oplus is the append operation.

3.3 Retrieval Model with Label Text

DEPL leverages the semantic matching of document and label texts via a dual encoder model (Gao and Callan, 2021; Xiong et al., 2020; Luan et al., 2020; Karpukhin et al., 2020). We use the BERT (Devlin et al., 2018) model as the backbone of our neural encoder, which is shared for both the document and label text encoding. Since a neural model encodes textual inputs into condensed vector representations, we call them *dense* models.

The similarity between text and label representation is measured by:

$$f_{\text{dual}}(\mathbf{x}, l) = \langle \phi_{\text{doc}}(\mathbf{x}), \phi_{\text{label}}(\text{text}(l)) \rangle$$

where $\text{text}(l)$ is the textual information of the label l . When the textual information only includes the label name given in the dataset, we call the model

DE-ret. Otherwise, when the textual information includes the generated pseudo label description, we call the model **DEPL**.

The document embedding $\phi_{\text{doc}}(\mathbf{x})$ is obtained from the CLS embedding of the BERT model followed by a linear pooling layer:

$$\phi_{\text{doc}}(\mathbf{x}) = \mathbf{W}_{\text{doc}} \cdot \text{BERT}(\mathbf{x}, \text{CLS}) + \mathbf{b}_{\text{doc}}$$

where $\text{BERT}(\mathbf{x}, \text{CLS})$ represents the contextualized embedding of the special CLS token. \mathbf{W}_{doc} and \mathbf{b}_{doc} are the weights and biases for the document pooler layer.

For the label embedding $\phi_{\text{label}}(\text{text}(l))$, we take an average of the last hidden layer of BERT followed by a linear pooler layer:

$$\phi_{\text{label}}(\text{text}(l)) = \mathbf{W}_{\text{label}} \cdot \psi_{\text{bert}}(\text{text}(l)) + \mathbf{b}_{\text{label}} \quad (1)$$

$$\psi_{\text{bert}}(\text{text}(l)) = \frac{1}{|\text{text}(l)|} \sum_{j=1}^{|\text{text}(l)|} \text{BERT}(\text{text}(l), j) \quad (2)$$

where $\text{BERT}(\text{text}(l), j)$ represents the contextualized embedding of the j -th token in $\text{text}(l)$ obtained from the last hidden layer of the BERT model. $\mathbf{W}_{\text{label}}$ and $\mathbf{b}_{\text{label}}$ are the weights and biases for the label pooler layer. In the equation 2, the average embedding of label tokens yields better performance empirically than the CLS embedding possibly because the keywords are not natural language, and BERT may not effectively aggregate such type of information into CLS.

Learning with Negative Sampling Since calculating all the label embeddings for each batch is both expensive and prohibitive by the memory limit, we resort to negative sampling strategies for in-batch optimization. Specifically, we sample a fixed-sized subset of labels for each batch containing: 1) all the positive labels of the instances in the batch, 2) the top negative predictions by the sparse classifier as the hard negatives, and 3) the rest of the batch is filled with uniformly random sampled negatives labels.

Let \mathcal{S}_b be the subset of labels sampled for a batch. The objective for the dual encoder is:

$$\mathcal{L}_{\text{dual}} = -\frac{1}{B|\mathcal{S}_b|} \sum_{i=1}^B \left(\sum_{p \in \mathbf{y}_i^+} \log \sigma(f_{\text{dual}}(\mathbf{x}_i, p)) + \sum_{n \in \mathcal{S}_b \setminus \mathbf{y}_i^+} \log \sigma((1 - f_{\text{dual}}(\mathbf{x}_i, n))) \right)$$

where B is the batch size, \mathbf{y}_i^+ is the positive labels for instance i , and σ is the sigmoid function.

3.4 Connection of Sparse and Dense Model

Complementary features: the sparse model uses the tf-idf feature based on corpus-level token statistics, while the dense model relies on the knowledge of the language learned during pretraining. The two types of features focus on different aspects of the text corpus and the combination of the two brings gains in performance.

Difference from ensemble: utilizing the augmented text for retrieval is better than a pure ensemble of sparse and dense methods such as in X-Transformer. In the ensemble method, the semantic meaning of important tokens in a label embedding learned from sparse classifier is not leveraged. By extracting the keywords from the sparse label embedding and presenting them as pseudo label descriptions, our model can additionally exploit the value of those key token semantics.

3.5 Enhance Classification with Retrieval

Our introduced retrieval model can be combined with a neural classifier for a performance boost on overall label classification (since our retrieval model is primarily targeted on improving tail label performance). In a neural classification system, the label embedding is treated as free parameters to be learned from supervised data, which is more expressive for labels with abundant training instances. The neural classifier learns the function:

$$f_{\text{cls}}(\mathbf{x}, l) = \langle \phi_{\text{doc}}(\mathbf{x}), \mathbf{w}_l^{\text{cls}} \rangle \quad (3)$$

We propose to enhance the classification model with the retrieval mechanism by jointly fine-tuning:

$$f_{\text{dual-cls}}(\mathbf{x}, l) = \frac{\sigma(f_{\text{dual}}(\mathbf{x}, l)) + \sigma(f_{\text{cls}}(\mathbf{x}, l))}{2} \quad (4)$$

The classification and retrieval modules share the same BERT encoder. We refer to the system as **DEPL+cls**. The object function $\mathcal{L}_{\text{dual-cls}}$ is similar to $\mathcal{L}_{\text{dual}}$ except for replacing f_{dual} with $f_{\text{dual-cls}}$.

The **DEPL+cls** model looks like an ensemble of the two systems at first sight, but there are two major differences: 1) As the BERT encoder is shared between the classification and retrieval modules, it doesn't significantly increase the number of parameters as in (Chang et al., 2020; Jiang et al., 2021); and 2) when the two modules are optimized together, the system can take advantages of both units according to the situation of head or tail label predictions.

4 Theoretical Analyses of DEPL

4.1 Rethinking Dense and Sparse Model for Imbalanced Text Classification

We analyze dense and sparse models from a gradient perspective for classification problems with skewed label distribution.

Preliminary: The predicted probability optimized by the binary cross entropy (BCE) loss is:

$$\mathcal{L}_{\text{BCE}} = - \sum_{l=1}^L y_l \log p_l + (1 - y_l) \log(1 - p_l)$$

The derivative of \mathcal{L}_{BCE} w.r.t the logits s_l is:

$$\frac{\partial \mathcal{L}_{\text{BCE}}}{\partial s_l} = \begin{cases} p_l - 1 & \text{if } y_l = 1 \\ p_l & \text{otherwise} \end{cases} \quad (5)$$

Q1: *Why would sparse model with BOW feature benefit tail label prediction?*

Applying the chain rule to equation 5, the gradient of \mathcal{L}_{BCE} w.r.t the document feature $\phi_n(\mathbf{x})$ is:

$$\frac{\partial \mathcal{L}_{\text{BCE}}(y_l, p_l)}{\partial \phi_n(\mathbf{x})} = \begin{cases} (p_l - 1)\mathbf{w}_l & \text{if } y_l = 1 \\ p_l \mathbf{w}_l & \text{otherwise} \end{cases}$$

By optimizing parameters θ of feature extractor, the document representation is encourage to move away from the negative label representation, that is:

$$\phi_n(\mathbf{x}; \theta') \leftarrow \phi_n(\mathbf{x}; \theta) - \eta p_l \mathbf{w}_l$$

where η is the learning rate.

For a dense model, the parameter θ of the feature extractor (such as BERT) is shared for all the data, so the optimization of the feature extractor is affected by the distribution of labels in the training data. Since a tail label appears more often as a negative target, the feature extractor is likely to under-represent the tail label information, making a tail label more difficult to be predicted. In comparison, the sparse feature like tf-idf is derived in an unsupervised manner from corpus statistics, which is independent of training label distribution. Therefore, the sparse feature may maintain better representation power to separate the tail labels.

Q2: *What is the advantage of a retrieval system on tail label prediction?*

In a typical classification system, labels are treated as indices whose embeddings are randomly

initialized and learned from supervised signals. The gradients of \mathcal{L}_{BCE} w.r.t the label feature is:

$$\frac{\partial \mathcal{L}_{\text{BCE}}(y_l, p_l)}{\partial \mathbf{w}_l} = \begin{cases} (p_l - 1)\phi_n(\mathbf{x}) & \text{if } y_l = 1 \\ p_l \phi_n(\mathbf{x}) & \text{otherwise} \end{cases}$$

The label embedding is updated by:

$$\begin{aligned} \mathbf{w}'_l &= \mathbf{w}_l + \frac{\eta}{N_{\text{train}}} \sum_{i: y_{il}=1} (1 - p_{il}) \phi_n(\mathbf{x}_i) \\ &\quad - \frac{\eta}{N_{\text{train}}} \sum_{i: y_{il}=0} p_{il} \phi_n(\mathbf{x}_i) \end{aligned}$$

As most of the instances are negative for a tail label, the update of tail label embedding is inundated with the aggregation of negative features, making it hard to encode distinctive feature reflecting its identity. Therefore, learning the tail label embedding from supervised signals alone can be distracting. Although previous works leverage negative sampling to alleviate the problem (Jiang et al., 2021; Chang et al., 2020), we argue that a fundamental solution is to inject the label information into the embedding. Our proposed retrieval system presents a natural way to incorporate label text for enhanced performance of tail label prediction.

4.2 Analysis on Performance Lower Bound

We will show the connection between DEPL and a sparse SVM classifier (for pseudo label extraction) by a performance lower bound. Specifically, DEPL outperforms a sparse model with high probability given that the selected keywords are important and the sparse classifier can separate the positive from the negative instances with non-trivial margin.

Notation: Let $\phi_t(\mathbf{x})$ be the normalized tf-idf feature vector of text with $\|\phi_t(\mathbf{x})\|_2 = 1$. The sparse label embeddings $\{\mathbf{w}_1, \dots, \mathbf{w}_L\}$ satisfies $\|\mathbf{w}_l\|_2 \leq 1, w_{li} > 0$. In fact, label embeddings can be transformed to satisfy the condition without affecting the prediction rank. Let z_l be the top selected keywords from the sparse classifier, which is treated as the pseudo label. Define the sparse keyword embedding \mathbf{v}_l with $v_{li} = w_{li}$ if i is an index of selected keywords and 0 otherwise.

In the following, we define the keyword importance and the classification error margin.

Definition 1. For label l and $\delta \geq 0$, the sparse keyword embedding \mathbf{v}_l is δ -bounded if $\langle \phi_t(\mathbf{x}), \mathbf{v}_l \rangle \geq \langle \phi_t(\mathbf{x}), \mathbf{w}_l \rangle - \delta$.

Definition 2. For two labels p and n , the error margin μ is the difference between the predicted scores $\mu(\phi(\mathbf{x}), \mathbf{w}_p, \mathbf{w}_n) = \langle \phi(\mathbf{x}), \mathbf{w}_p - \mathbf{w}_n \rangle$.

The main theorem is stated as below:

Theorem 3. Let $\phi_t(\mathbf{x})$ and $\phi_n(\mathbf{x})$ be the sparse and dense (dimension d) document feature, \mathbf{w}_l be the label embedding and \mathbf{z}_l be the δ -bounded keywords. For a positive label p , let $\mathbb{N}_p = \{n_1, \dots, n_{M_p}\}$ be a set of negative labels ranked lower than p . The error margin $\epsilon_i = \mu(\phi_t(\mathbf{x}), \mathbf{w}_p, \mathbf{w}_{n_i})$ and $\epsilon = \min(\{\epsilon_1, \dots, \epsilon_{M_p}\})$. An error \mathcal{E}_i of the neural classifier occurs when

$$\mu(\phi_n(\mathbf{x}), \phi_n(\mathbf{z}_p), \phi_n(\mathbf{z}_{n_i})) \leq 0 \quad (6)$$

The probability of any such error happening satisfies

$$P(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{M_p}) \leq 4M_p \exp\left(-\frac{(\epsilon - \delta)^2 d}{50}\right)$$

When $(\epsilon - \delta) \geq 10\sqrt{\frac{\log M_p}{d}}$, the probability is bounded by $\frac{1}{M_p}$.

Discussion: An error event occurs when the sparse model makes a correct prediction but the neural model doesn't. If the neural model avoids all such errors, the performance should be at least as good as the sparse model, and Theorem 3 gives a bound of that probability.

The term δ measures the importance of selected keywords (smaller the more important), the error margin ϵ measures the difficulty the correctly predicted positive and negative pairs by the sparse model. The theorem states that the model achieves a lower bound performance as sparse classifier if the keywords are informative and error margin is non-trivial. Proofs are in section A.2 for interested readers and limitations are discussed in section 8.

5 Evaluation Design

Dataset	N_{train}	N_{test}	\bar{L}_d	L	$ \mathbb{L}_{tail} $
EURLex-4K	15,539	3,809	5.30	3,956	2,413
AmazonCat-13K	1,186,239	306,782	5.04	13,330	3,936
Wiki10-31K	14,146	6,616	18.64	30,938	26,545
Wiki-500K	1,779,881	769,421	4.75	501,070	338,719

Table 1: Corpus Statistics: N_{train} and N_{test} are the number of training and testing instances respectively; \bar{L}_d is the average number of labels per document, and L is the number of unique labels. $|\mathbb{L}_{tail}|$ is the number of tail labels with $1 \sim 9$ positive training instances.

5.1 Datasets

We conduct our experiments on 4 benchmark datasets: EURLex-4K, AmazonCat-13K, Wiki10-31K and Wiki-500K. The statistics of the datasets

are shown in Table 1. An unstemmed version of EURLex-4K is obtained from the APLC-XLNet github¹ and the rest are from the Extreme classification Repository².

For comparative evaluation of methods in tail label prediction, we consider the subset of labels with $1 \sim 9$ positive training instances. Those tail-label subsets correspond to 63.48%, 29.53%, 88.65% and 67.60% of the total labels in the 4 datasets respectively. With mostly more than half of the labels as tail labels, the distributions are indeed highly skewed.

5.2 Tail Label Evaluation Metrics

Micro-averaged PSP@k: The PSP (Jain et al., 2016a) metric re-weights the score of each instance according to the label frequency:

$$PSP@k = \frac{1}{k} \sum_{l=1}^k \frac{\mathbb{1}_y(\mathbf{p}_l)}{\text{prop}(\mathbf{p}_l)}$$

where the propensity score $\text{prop}(\mathbf{p}_l)$ in the denominator gives higher weights to tail labels.

Since the micro-averaged metric gives an equal weight to the per-instance scores, it can still be dominated by the system's performance on the head labels but not the tail labels. As an alternative, we adopt a macro-averaged metric to evaluate tail label performance.

Macro-averaged F1@k: The macro-averaged metric (Yang and Liu, 1999) gives an equal weight to all the labels (we apply it to tail labels specifically). It is defined as the average of the label-specific $F1@k$ values, calculated based on a contingency table for each label, as shown in table 2. The precision, recall and $F1$ for a predicted ranked list of length k are computed as $P = \frac{TP}{TP+FP}$, $R = \frac{TP}{TP+FN}$, and $F1 = 2 \frac{P \cdot R}{P+R}$.

Table 2: Contingency table for label l .

	l is true label	l is not true label
l predicted	True Positive (TP _{l})	False Positive (FP _{l})
l not predicted	False Negative (FN _{l})	True Negative (TN _{l})

For micro-averaged PSP@k, we choose $k = 1, 3, 5$ as in previous works. For macro-averaged F1@k, we choose $k = 19$ for Wiki10-31K because it has an average of 18.64 labels and $k = 5$ for the rest datasets.

¹https://github.com/huiyegit/APLC_XLNet.git

²<http://manikvarma.org/downloads/XC/XMLRepository.html>

5.3 Baselines

For the tail label evaluation, our method is compared with the SOTA deep learning models including X-Transformer (Chang et al., 2020), XLNet-APLC (Ye et al., 2020), LightXML (Jiang et al., 2021), and AttentionXML (You et al., 2018). X-Transformer, LightXML, and XLNet-APLC employ pre-trained Transformers for document representation. We reproduced the results of single model (given in their implementation) predictions with BERT as the base model for LightXML, BERT-large for X-Transformer, XLNet for XLNet-APLC, and LSTM for AttentionXML. The AttentionXML utilizes label-word attention to generate label-aware document embeddings, while the other models generate fixed document embedding.

We use the SVM model with tf-idf feature as our choice of sparse classifier and BERT-base as our dense model for neural retrieval and classification. Implementation details, more baselines and settings are discussed in appendix A.1.

6 Evaluation Results

Our experiments reveal the effectiveness of our model on the tail label prediction and we also include and discuss the performance on the overall prediction in appendix A.1.4.

6.1 Results in Tail Label Prediction

SVM on Tail Label Prediction The results evaluated with the F1 metric averaged on the tail labels are shown in figure 3. Surprisingly, a simple statistical SVM baseline achieves competitive results on the tail label predictions. We observe that SVM model can outperform most of the pretrained Transformer-based models on the tail label prediction, and outperform the AttentionXML on the Wiki10-31K dataset. This provides an empirical evidence for the robust performance of a sparse model on tail label prediction. As we analyzed in section 4, the SVM model utilizes the unsupervised statistical feature as document representation, which potentially suffers less from the data scarcity issue. The empirical result serves as an evidence for our theoretical analysis that the joint optimization of feature extractor and label embedding is difficult when data is limited.

Neural Classifier on Tail Label Prediction

The neural classifiers include LightXML, X-Transformer, XLNet-APLC and AttentionXML.

Specifically, the AttentionXML model leverages a label-word attention to calculate a label specific document representation. As we observe in figure 3, among the baseline models, the AttentionXML performs the best on the tail label predictions, beating the other baselines on 3 out of the 4 benchmark datasets. The superior performance could come from the local word and label matching which benefits the tail label prediction.

As mentioned in section 3, X-Transformer model ensembles a neural classifier and a SVM model by directly summing the prediction scores. Although X-Transformer outperforms SVM on the overall label prediction, it underperforms SVM on 3 out of 4 benchmark datasets. This shows that model performance on tail label is dragged down by the neural model prediction, and a simple ensemble does not fully exploit the advantage of the sparse model. Compared with the X-Transformer, our model achieves better performance on both macro-F1 and micro-PSP metrics, showing the advantage of leveraging the retrieval of augmented label descriptions rather than a pure ensemble.

DEPL Performance On the 3 smaller scale benchmark datasets, EURLex-4K, AmazonCat-13K and Wiki10-31K, our model directly ranks all the labels. On the large Wiki-500K dataset, our model leverages the prediction of cluster-based algorithm in X-Transformer and replaces the reranker with our retrieval model.

Our proposed models perform the best on the Macro-F1 metric with the DEPL model consistently and significantly showing the best performance on all the benchmark datasets. A macro t-test (Yang and Liu, 1999) is conducted to justify the significance of improvement over the SVM and previous best neural model. The significant performance gains over the SVM model shows that our retrieval framework can outperform the sparse model which serves as label keywords extractor. We attribute the success of model on tail label prediction to the retrieval module that focuses on the semantic matching between the document and label text. The DEPL performs better than the DEPL+cls as it is less affected by the large amount of training instances for head labels and thus more biased on the tail label prediction.

According to the evaluation with the PSP metric shown in table 3, it also confirms that our proposed models DEPL and DEPL+cls improves over the previous SOTA neural models on all the benchmark

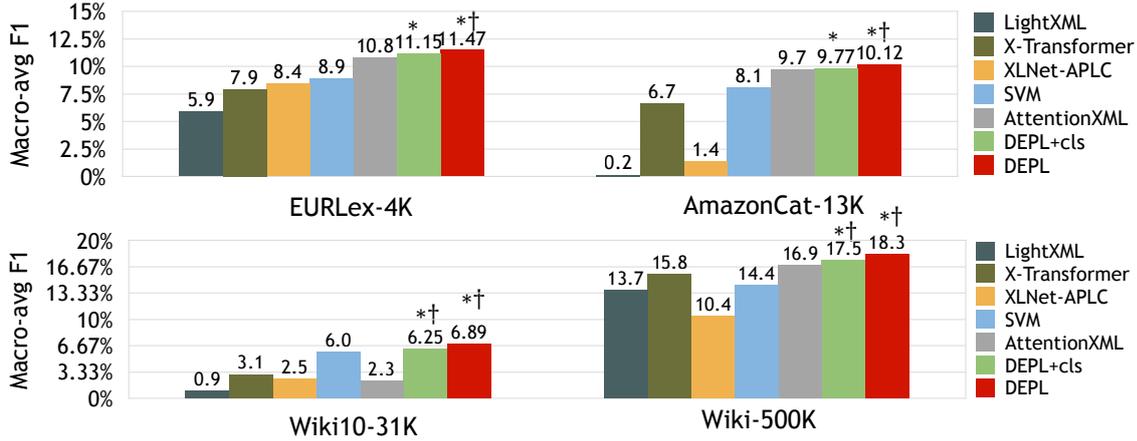


Figure 3: Tail-label prediction results in $F1@k$ on the labels with 1 ~ 9 positive training instances, with $k = 19$ for the Wiki10-31K dataset and $k = 5$ for the rest. * and † indicates the macro t-test is significant ($p < 0.05$) over SVM and previous best neural model respectively.

Table 3: Tail label prediction results of methods in $PSP@k$, with * indicating significant improvement ($p < 0.05$) over the previous best model on the micro sign test.

Methods	EURLex-4K			Wiki10-31K			AmazonCat-13K			Wiki-500K		
	PSP@1	PSP@3	PSP@5	PSP@1	PSP@3	PSP@5	PSP@1	PSP@3	PSP@5	PSP@1	PSP@3	PSP@5
X-Transformer	37.85	47.05	51.81	13.52	14.62	15.63	51.42	66.14	75.57	31.20	36.78	40.21
XLNet-APLC	42.21	49.83	52.88	14.43	15.38	16.47	52.55	65.11	71.36	29.73	30.26	30.59
LightXML	40.54	47.56	50.50	14.09	14.87	15.52	50.70	63.14	70.13	31.01	37.10	39.28
AttentionXML	44.20	50.85	53.87	14.49	15.65	16.54	53.94	68.48	76.43	30.05	37.31	41.74
SVM	39.18	48.31	53.37	11.84	14.00	15.81	51.83	65.41	72.82	32.12	32.75	35.20
DEPL	45.60*	52.28*	53.52	17.20*	16.90*	16.95	55.94*	70.01*	76.87*	32.07	40.60*	43.74*
DEPL+cls	44.60	52.74*	54.64	16.73*	16.84*	16.67	55.21*	69.73*	75.94	32.18	39.89*	41.46

Table 4: Examples of SVM generated keywords from Wiki10-31K. The classifier is trained with only 1 positive training instance per label. The top 20 keywords are shown. with meaningful words highlighted in red manually.

Label Text	#training instance	Top Keywords
phase4	1	trials clinical protection personal directive processed data trial drug phase eu processing patients sponsor controller legislation regulation art investigator study
ensemble	1	boosting kurtz ferrell weak algorithms learners misclassified learner kearns ensemble charges bioterrorism indictment doj indict cae correlated 2004 reweighted boost
kakuro	1	nikoli kakuro puzzles crossword clues entries entry values sums cells cross digits dell solvers racehorse guineas aa3aa digit clue kaji

datasets, with * indicates significant improvement ($p < 0.05$) over the previous best model on the micro sign test (Yang and Liu, 1999). The Wiki10-31K dataset has the most skewed distribution as the most frequent label covers more than 85% of the training instances, resulting in a low PSP score. Since DEPL relies on the semantic matching between the document and label text, it is less affected by the dominating training pairs, and thus the PSP@1, PSP@3 beats the SOTA models by a larger margin. The DEPL+cls achieves worse performance on this dataset, because the classification counterpart of the model would benefit more on

the head label predictions and tend to rank the head labels at the top.

Metric Comparison Although the PSP metric gives higher weight to the tail labels, it is a micro-averaged metric over the scores of each instance, which can still be affected by the performance on the more common categories that cover most of the instances. For example, SVM model doesn't stand out under the PSP metric, which has lower overall label performance. Since the F1 metric is calculated specifically on the set of tail labels, we argue that it provides a more accurate and fine-

grained evaluation on tail label prediction, which better reveals the success of XMTC models on predicting rare categories.

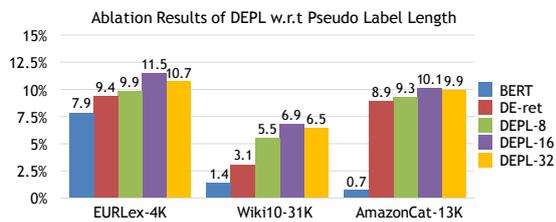


Figure 4: The ablation-test results of DEPL in Macro-averaged F1@k metric with varying length of pseudo label descriptions.

6.2 Ablation on Generated Pseudo Label

Table 4 shows examples of the SVM generated keywords trained on the Wiki10-31K dataset for labels with only 1 training example. We manually highlight the meaningful terms related to the label meaning. For example, the label name *phase4* is ambiguous, whose meaning needs to be inferred from the corresponding document. From the keywords *trial*, *clinical*, *drug*, *etc*, we deduce that the topic is about medical testing phase. In another example, *kakuro* is a Japanese logic puzzle known as a mathematical crossword and the game play involves in adding number in the cells. Generating a description for *kakuro* requires the background knowledge, but the keywords automatically learned from the sparse classifier provide the key concepts. Although not all the keywords can provide rich semantics to complement the original label name, they may serve as a context for the label to make it more distinguishable from others.

In figure 4, we conduct an ablation test on the length of the pseudo label and the performance is measured by Macro-avg F1@k. The BERT classifier is included as a baseline with no label text information. As we observe that the longer description of length 16 performs the better, but when length is 32, the performance doesn't increase as the text may become noisy with more unrelated keywords.

The DE-ret model is a pure retrieval baseline (avg length 3) with only the label name. While it achieves good performance on the EURLex-4K and AmazonCat-13K datasets, it still performs poorly on the Wiki10-31K dataset. This shows that generating the keywords from the sparse classifier can enhance the text quality. Furthermore, the gener-

ated text allows DEPL to use the semantic information of the label keywords, which is ignored in the SVM model. This could be another reason why our model performs better than the SVM baseline on the Wiki10-31K dataset.

7 Conclusion

In this paper, we propose a novel neural retrieval framework (DEPL) for the open challenge of tail-label prediction in XMTC. By formulating the problem as to capture the semantic mapping between input documents and system-enhanced label descriptions, DEPL combines the strengths of neural embedding based retrieval and the effectiveness of a large-margin BoW classifier in generating informative label descriptions under severe data sparse conditions. Our extensive experiments on very large benchmark datasets show significant performance improvements by DEPL over strong baseline methods, especially in tail label description.

8 Limitations

Our paper mainly focuses on the evaluation and improvement over the pretrained Transformer-based models such as X-Transformer, LightXML and APLC-XLNet by leveraging the recent advances in dense retrieval with BERT model. However, there are other works such as proposing reranking losses (Wei et al., 2021), regularization (Babbar and Schölkopf, 2019a) with other architectures are not included for comparison.

As pointed out by the reviewers, the performance bound analysis in section 4 adopts a strong assumption that the neural embeddings are random matrices. This could be very different in real application because the random matrices do not encode any semantic information. We acknowledge this limitation and provide more references on that. We rely on the mathematical tool based on random matrix theory, namely the *Johnson-Lindenstrauss* (JL) lemma. This tool was also adopted by Luan et al. (2020) under information retrieval setting, which provides the connection between dense and sparse retrievers. The bound is on its loose end because embeddings from BERT are more meaningful than random matrices (also verified from their empirical study). In our work, we study use the JL lemma to connect sparse and dense classifiers. The bound is reasonable considering that it is on its loose end, but, still, there is no guarantee when applied with real BERT embeddings.

References

- Rohit Babbar and Bernhard Schölkopf. 2017. Dismec: Distributed sparse machines for extreme multi-label classification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 721–729.
- Rohit Babbar and Bernhard Schölkopf. 2019a. Data scarcity, robustness and extreme multi-label classification. *Machine Learning*, 108(8):1329–1351.
- Rohit Babbar and Bernhard Schölkopf. 2019b. Data scarcity, robustness and extreme multi-label classification. *Machine Learning*, 108(8):1329–1351.
- Shai Ben-David, Nadav Eiron, and Hans Ulrich Simon. 2002. Limitations of learning via embeddings in euclidean half spaces. *Journal of Machine Learning Research*, 3(Nov):441–461.
- Duo Chai, Wei Wu, Qinghong Han, Fei Wu, and Jiwei Li. 2020. Description based text classification with reinforcement learning. In *International Conference on Machine Learning*, pages 1371–1382. PMLR.
- Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. 2020. Taming pre-trained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3163–3171.
- Kunal Dahiya, Ananye Agarwal, Deepak Saini, K Gururaj, Jian Jiao, Amit Singh, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. Siamesexml: Siamese networks meet extreme classifiers with 100m labels. In *International Conference on Machine Learning*, pages 2330–2340. PMLR.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.
- Siddharth Gopal, Yiming Yang, Bing Bai, and Alexandru Niculescu-Mizil. 2012. Bayesian models for large-scale hierarchical classification.
- Siddharth Gopal and Yiming Yang. 2010. Multilabel classification with meta-level features. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. 2019. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 528–536.
- Himanshu Jain, Yashoteja Prabhu, and Manik Varma. 2016a. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Himanshu Jain, Yashoteja Prabhu, and Manik Varma. 2016b. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 935–944.
- Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. *arXiv preprint arXiv:2101.03305*.
- William B Johnson and Joram Lindenstrauss. 1984. Extensions of lipschitz mappings into a hilbert space 26. *Contemporary mathematics*, 26.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.
- Sujay Khandagale, Han Xiao, and Rohit Babbar. 2019. [Bonsai – diverse and shallow trees for extreme multi-label classification](#).
- Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124.
- Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *arXiv preprint arXiv:2005.00181*.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Yashoteja Prabhu, Anil Kag, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. 2018. Parel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*, pages 993–1002.

Tong Wei, Wei-Wei Tu, Yu-Feng Li, and Guo-Ping Yang. 2021. Towards robust prediction on tail labels. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1812–1820.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49.

Hui Ye, Zhiyu Chen, Da-Han Wang, and Brian Davison. 2020. Pretrained generalized autoregressive model with adaptive probabilistic label clusters for extreme multi-label text classification. In *International Conference on Machine Learning*, pages 10809–10819. PMLR.

Ian EH Yen, Xiangru Huang, Wei Dai, Pradeep Ravikumar, Inderjit Dhillon, and Eric Xing. 2017. Ppdsparse: A parallel primal-dual sparse method for extreme classification. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 545–553.

Ronghui You, Zihan Zhang, Ziyue Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2018. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *arXiv preprint arXiv:1811.01727*.

A Appendix

A.1 Experiments

A.1.1 All-label Evaluation Metric

We introduce the micro-averaged $P@k$ as the metric for all-label prediction. Given a ranked list of the predicted labels for each test document, the **micro-averaged $P@k$** is:

$$P@k = \frac{1}{k} \sum_{i=1}^k \mathbb{1}_{y_i^+}(p_i) \quad (7)$$

where p_i is the i -th label in the list \mathbf{p} and $\mathbb{1}_{y_i^+}$ is the indicator function.

A.1.2 More Baseline

For the overall prediction of all labels, we also include the baselines of sparse classifiers: DisMEC (Babbar and Schölkopf, 2017), PfastreXML (Jain et al., 2016b), Parabel (Prabhu et al., 2018), Bonsai (Khandagale et al., 2019), and we

use the published results for comparison. We provide an implementation of linear SVM model with our extracted tf-idf features as another sparse baseline, and a BERT-base classifier as another dense classifier (used to initialize DEPL).

A.1.3 Implementation Details

For the sparse model, since the public available BoW feature doesn’t have a vocabulary dictionary, we generate the tf-idf feature by ourselves. We tokenize and lemmatize the raw text with the spaCy (Honnibal and Montani, 2017) library and extract the tf-idf feature with the Sklearn (Pedregosa et al., 2011) library, with unigram whose df count is ≥ 2 and df frequency $\leq 70\%$ of the total documents.

We use the BERT model as the contextualize function for our retrieval model, which is initialized with a pretrained dense classifier. Specifically, we fine-tune a 12 layer BERT-base model with different learning rates for the BERT encoder, BERT pooler and the classifier. The learning rates are $(1e - 5, 1e - 4, 1e - 3)$ for Wiki10-31K and $(5e - 5, 1e - 4, 2e - 3)$ for the rest datasets. For the negative sampling, we sample batch of 500 instances for Wiki10-31K, and 300 for EURLex-4K and AmazonCat-13K. For Wiki-500K dataset, we leverage the cluster-based algorithm in X-Transformer, and perform label re-ranking using our DEPL model to replace the linear model in X-Transformer. We use a negative batch size of 500 for to train the re-ranker.

We include 10 hard negatives predicted by the SVM model for each instances. We used learning rate $1e - 5$ for fine-tuning the BERT of our retrieval model and $1e - 4$ for the pooler and label embeddings. For the pseudo label descriptions, we concatenate the provided label description with the generated the top 20 keywords. The final length is truncated up to 32 tokens after BERT tokenization. We use length 16 of pseudo label description as the default setting for DEPL.

A.1.4 Results in All-label Prediction

The performance of our models evaluated on the all-label prediction by the micro-averaged $P@k$ metric is reported in table 5. Our model is compared against the SOTA sparse and dense classifiers. DEPL+c achieves the best or second best performance on all the 4 benchmark datasets, achieving comparable results to the previous best SOTA models.

Methods	EURLex-4K			Wiki10-31K			AmazonCat-13K			Wiki-500K		
	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5	P@1	P@3	P@5
DisMEC	83.21	70.39	58.73	84.13	74.72	65.94	93.81	79.08	64.06	70.21	50.57	39.68
PfastreXML	73.14	60.16	50.54	83.57	68.61	59.10	91.75	77.97	63.68	56.25	37.32	28.16
eXtremeText	79.17	66.80	56.09	83.66	73.28	64.51	92.50	78.12	63.51	65.17	46.32	36.15
Parabel	82.12	68.91	57.89	84.19	72.46	63.37	93.02	79.14	64.51	68.70	49.57	38.64
Bonsai	82.30	69.55	58.35	84.52	73.76	64.69	92.98	79.13	64.46	69.26	46.72	36.46
AttentionXML	85.12	72.80	61.01	86.46	77.22	67.98	95.53	82.03	67.00	75.20	56.42	44.10
X-Transformer	85.46	72.87	60.79	87.12	76.51	66.69	<u>95.75</u>	82.46	<u>67.22</u>	75.28	55.46	42.75
XLNet-APLC	86.83	74.34	61.94	88.99	78.79	69.79	94.56	79.78	64.59	72.95	51.23	38.64
LightXML	86.12	<u>73.87</u>	61.67	87.39	77.02	68.21	94.61	79.83	64.45	<u>75.96</u>	<u>56.55</u>	<u>44.22</u>
SVM	83.44	<u>70.62</u>	59.08	84.61	74.64	65.89	93.20	78.89	64.14	69.92	49.35	38.8
DEPL	85.38	71.86	59.91	84.63	74.80	65.96	94.86	80.85	64.55	74.69	55.72	42.71
DEPL+c	<u>86.43</u>	73.77	62.19	<u>88.57</u>	<u>78.04</u>	<u>68.75</u>	96.16	<u>82.23</u>	67.65	76.83	57.15	45.07

Table 5: The all-label prediction results of representative classification systems evaluated in the micro-avg P@k metric. The bold phase and underscore highlight the best and second best model performance.

We argue that though our models perform significantly better on the tail label prediction, the improvement is not announced in the overall label prediction. One of the problem is on the choice of evaluation metric: the micro-averaged precision metric is averaged over instances and can be dominated by the common categories with more test instances. Therefore, the metric is incapable of reflecting the tail label performance. We want to emphasize that over 26, 545 (88.65%) labels in the Wiki10-31K dataset belong to the tail labels with less than 10 training instances, constituting a majority of the label space. The overall classification precision (P@k) only reflects a part of the success of a classification system, and the tail label evaluation is yet another part. The results also shows while our model improves on the tail label prediction, the overall label prediction comparable to the other dense and sparse SOTA models.

When our model is compared on the Wiki-500K dataset, our backbone is the same as X-Transformer. DEPL achieve on par performance with Wiki-500k showing that the quality of overall ranking is similar. However, the DEPL+c achieves better performance, demonstrating the enhanced performance by combining retrieval with classification.

By comparing the DEPL+c and its retrieval-based counterpart DEPL, we uncover a trade-off between the head label and tail label prediction. We observe that the DEPL outperforms the DEPL+c on the tail label prediction, but not on the all-label prediction. This shows that incorporating a classifier with label embeddings trained from supervised

signal can boost performance on a high data regime. The dense classifier could learn more expressive label representation from the frequent co-occurrence of document and label pairs when the training instances are abundant, while the retrieval system is better at matching the semantic of document and label texts when data is scarce. Each of the modules captures a certain aspect of the data heuristic for text classification and a combination of them by sharing the BERT encoder yields better performance.

Lastly, the sparse classifiers generally underperform the neural models and are comparable to our implement of SVM. We observe that DEPL can outperform the sparse models, which agrees with our theoretical analysis. Although the pseudo labels are extracted from the SVM classifier, the neural retrieval model can additionally leverage the keyword semantic information and correlation of them, which is ignored in the SVM classifier. The pseudo label descriptions encode both the term importance and key semantics of labels.

A.1.5 More Ablation Tests

Model Pre-training We fine-tune our retrieval model on a pre-trained neural classifier (BERT) and table 6 shows that without using the pre-trained model, there is a significant drop in the precision and PSP metrics.

Negative Sampling We used the top negative predictions by the SVM model as the choice of hard negative labels. By default, we use 10 hard negatives for each instance in the batch. In table 6, we

Table 6: Ablation-test results of DEPL under different training conditions.

Methods	P@1	P@3	P@5	PSP@1	PSP@3	PSP@5
EUR-Lex						
DE-ret	-1.34	-1.18	-1.16	-3.2	-2.11	+0.18
w/o pre-train	-6.81	-6.72	-6.16	-6.87	-7.09	-5.87
w/o neg	-2.52	-2.63	-2.2	-3.59	-3.66	-1.9
5 hard negative	-1.55	-1.19	-1.12	-3.57	-2.98	-1.32
Wiki10-31K						
DE-ret	-3.40	-7.71	-10.74	-7.63	-5.09	-1.20
w/o pre-train	-4.81	-8.66	-12.1	-2.46	-2.00	-1.92
w/o hard negative	-2.01	-5.89	-4.93	-1.15	-1.17	-1.37
5 hard negative	-0.84	-3.03	-4.16	-1.22	-0.99	-0.92

Table 7: Ablation-test results of DEPL with CLS and mean-pooling.

Methods	PSP@1	PSP@3	PSP@5
EUR-Lex			
DE-ret	44.87	52.17	53.40
DEPL with cls	42.32	47.26	47.53
DEPL with mean-pooling	45.60	52.28	53.52
Wiki10-31K			
DE-ret	16.71	15.76	16.35
DEPL with cls	14.71	14.58	15.33
DEPL with mean-pooling	17.20	16.90	16.95

observe a performance drop when no hard negatives or only 5 hard negatives are used for training.

CLS vs. Mean-pooling Table 7 shows an ablation test for the design of label description encoder. We observe that using a mean-pooling over the last layer of label keyword embeddings outperforms that using the CLS embedding by a large margin. This could be because the label keywords are not natural language the optimization using CLS embedding is more difficult.

A.2 Proof

We include the assumptions and proofs of Theorem 3.

Assumptions Similar to Luan et al. (2020), we treat neural embedding as fixed dense vector $\mathbf{E} \in \mathbb{R}^{d \times v}$ with each entry sampled from a random Gaussian $N(0, d^{-1/2})$. $\phi_n(\mathbf{x}) = \mathbf{E}\phi_t(\mathbf{x})$ is weighted average of word embeddings by the sparse vector representation of text. According to the *Johnson-Lindenstrauss (JL) Lemma* (Johnson and Lindenstrauss, 1984; Ben-David et al., 2002), even if the entries of \mathbf{E} are sampled from a random normal distribution, with large probability, $\langle \phi_t(\mathbf{x}), \mathbf{v} \rangle$ and $\langle \mathbf{E}\phi_t(\mathbf{x}), \mathbf{E}\mathbf{v} \rangle$ are close.

Lemma 4. Let \mathbf{v} be the δ -bounded keyword-selected label embedding of \mathbf{w} . For two labels p, n , the error margins satisfy:

$$|\mu(\phi_t(\mathbf{x}), \mathbf{w}_p, \mathbf{w}_n) - \mu(\phi_t(\mathbf{x}), \mathbf{v}_p, \mathbf{v}_n)| \leq \delta$$

Proof. By the definition of δ -bounded keywords,

$$\langle \phi_t(\mathbf{x}), \mathbf{w}_p \rangle - \delta \leq \langle \phi_t(\mathbf{x}), \mathbf{v}_p \rangle \leq \langle \phi_t(\mathbf{x}), \mathbf{w}_p \rangle \quad (8)$$

$$-\langle \phi_t(\mathbf{x}), \mathbf{w}_n \rangle \leq -\langle \phi_t(\mathbf{x}), \mathbf{v}_n \rangle \leq -\langle \phi_t(\mathbf{x}), \mathbf{w}_n \rangle + \delta \quad (9)$$

Adding equation 8 and equation 9 finishes the proof:

$$\langle \phi_t(\mathbf{x}), \mathbf{w}_p - \mathbf{w}_n \rangle - \delta \leq \langle \phi_t(\mathbf{x}), \mathbf{v}_p - \mathbf{v}_n \rangle \leq \langle \phi_t(\mathbf{x}), \mathbf{w}_p - \mathbf{w}_n \rangle + \delta \quad (10)$$

□

Lemma 5. Let $\phi_t(\mathbf{x})$ and $\phi_n(\mathbf{x})$ be the sparse and dense (dimension d) document feature, \mathbf{w}_l be the label embedding and \mathbf{z}_l be the δ -bounded keywords. Let p be a positive label and n be a negative label ranked below p be the sparse classifier. The error margin is $\epsilon = \mu(\phi_t(\mathbf{x}), \mathbf{w}_p, \mathbf{w}_n)$. An error \mathcal{E} of neural classification occurs when $\mu(\phi_n(\mathbf{x}), \phi_n(\mathbf{z}_p), \phi_n(\mathbf{z}_n)) \leq 0$. The probability $P(\mathcal{E}) \leq 4 \exp(-\frac{(\epsilon-\delta)^2 d}{50})$.

Proof. By the JL Lemma (Ben-David et al., 2002): For any two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^v$, let $\mathbf{E} \in \mathbb{R}^{d \times v}$ be a random matrix such that the entries are sampled from a random Gaussian. Then for every constant $\gamma > 0$:

$$P\left(|\langle \mathbf{E}\mathbf{a}, \mathbf{E}\mathbf{b} \rangle - \langle \mathbf{a}, \mathbf{b} \rangle| \geq \frac{\gamma}{2} (\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2)\right) \leq 4 \exp\left(-\frac{\gamma^2 d}{8}\right) \quad (11)$$

Let $\gamma = \frac{2}{5}(\epsilon - \delta)$, $\mathbf{a} = \phi_t(\mathbf{x})$ and $\mathbf{b} = \mathbf{v}_p - \mathbf{v}_n$. Since $\|\mathbf{a}\|_2 = 1$ and $\|\mathbf{b}\|_2 \leq (\|\mathbf{v}_p\|_2 + \|\mathbf{v}_n\|_2)^2 \leq 4$, the JL Lemma gives

$$P(|\langle \mathbf{E}\phi_t(\mathbf{x}), \mathbf{E}(\mathbf{v}_p - \mathbf{v}_n) \rangle - \langle \phi_t(\mathbf{x}), \mathbf{v}_p - \mathbf{v}_n \rangle| \geq \epsilon - \delta) \quad (12)$$

$$\leq 4 \exp\left(-\frac{(\epsilon - \delta)^2 d}{50}\right) \quad (13)$$

To complete the proof, we need to show $P(\mathcal{E}) \leq Eq.12$:

$$\mathcal{E} \implies |\langle \mathbf{E}\phi_t(\mathbf{x}), \mathbf{E}(\mathbf{v}_p - \mathbf{v}_n) \rangle - \langle \phi_t(\mathbf{x}), \mathbf{w}_p - \mathbf{w}_n \rangle| \geq \epsilon \quad (14)$$

$$\implies |\langle \mathbf{E}\phi_t(\mathbf{x}), \mathbf{E}(\mathbf{v}_p - \mathbf{v}_n) \rangle - \langle \phi_t(\mathbf{x}), \mathbf{v}_p - \mathbf{v}_n \rangle| \geq \epsilon - \delta \quad (15)$$

where the equation 15 is derived by Lemma 4:

$$|\langle \mathbf{E}\phi_t(\mathbf{x}), \mathbf{E}(\mathbf{v}_p - \mathbf{v}_n) \rangle - \langle \phi_t(\mathbf{x}), \mathbf{v}_p - \mathbf{v}_n \rangle| \quad (16)$$

$$\geq |\langle \mathbf{E}\phi_t(\mathbf{x}), \mathbf{E}(\mathbf{v}_p - \mathbf{v}_n) \rangle - \langle \phi_t(\mathbf{x}), \mathbf{w}_p - \mathbf{w}_n \rangle| - \quad (17)$$

$$|\langle \phi_t(\mathbf{x}), \mathbf{w}_p - \mathbf{w}_n \rangle - \langle \phi_t(\mathbf{x}), \mathbf{v}_p - \mathbf{v}_n \rangle| \quad (18)$$

$$\geq \epsilon - \delta \quad (19)$$

Therefore $P(\mathcal{E}) \leq Eq.12$, which completes the proof. □

Proof of Theorem 3

Proof. The Lemma 2 shows that

$$P(\mathcal{E}_i) \leq 4 \exp\left(-\frac{(\epsilon_i - \delta)^2 d}{50}\right) \leq 4 \exp\left(-\frac{(\epsilon - \delta)^2 d}{50}\right) \quad (20)$$

By an union bound on the error events $\{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_{M_p}\}$,

$$P(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{M_p}) \leq \sum_{i=1}^{M_p} 4 \exp\left(-\frac{(\epsilon_i - \delta)^2 d}{50}\right) \quad (21)$$

$$= 4M_p \exp\left(-\frac{(\epsilon - \delta)^2 d}{50}\right) \quad (22)$$

□

When $(\epsilon - \delta)^2 \geq 10\sqrt{\frac{\log M_p}{d}}$, we have $\exp\left(-\frac{(\epsilon - \delta)^2 d}{50}\right) \leq \frac{1}{4M_p^2}$ and therefore $P(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{M_p}) \leq \frac{1}{M_p}$.

Unsupervised Keyphrase Extraction via Interpretable Neural Networks

Rishabh Joshi^{♣*} Vidhisha Balachandran^{♣*} Emily Saldanha[◇]

Maria Glenski[◇] Svitlana Volkova[◇] Yulia Tsvetkov[♣]

[♣]Language Technologies Institute, Carnegie Mellon University

[◇]Pacific Northwest National Laboratory

[♣]Paul G. Allen School of Computer Science & Engineering, University of Washington

{rjoshi2, vbalacha}@cs.cmu.edu,

{emily.saldanha, maria.glenski, svitlana.volkova}@pnnl.gov,

yuliats@cs.washington.edu

Abstract

Keyphrase extraction aims at automatically extracting a list of “important” phrases representing the key concepts in a document. Prior approaches for unsupervised keyphrase extraction resorted to heuristic notions of phrase importance via embedding clustering or graph centrality, requiring extensive domain expertise. Our work presents a simple alternative approach which defines keyphrases as document phrases that are salient for predicting the topic of the document. To this end, we propose INSPECT—an approach that uses self-explaining models for identifying influential keyphrases in a document by measuring the predictive impact of input phrases on the downstream task of the document topic classification. We show that this novel method not only alleviates the need for ad-hoc heuristics but also achieves state-of-the-art results in unsupervised keyphrase extraction in four datasets across two domains: scientific publications and news articles.¹

1 Introduction

Keyphrase extraction is crucial for processing and analysis of long documents in specialized (e.g., scientific, medical) domains (Mekala and Shang, 2020; Betti et al., 2020; Wang et al., 2019). The task is challenging, as the notion of phrase importance is context- and domain-dependent. Therefore, developing domain-agnostic keyphrase annotation guidelines and curating representative hand-labeled datasets is not feasible. This motivates the need for generalizable unsupervised approaches to keyphrase extraction.

Unsupervised keyphrase extraction methods have used heuristic notions of phrase importance (Mihalcea and Tarau, 2004; Shang et al., 2018; Campos et al., 2018). Popular proxies for phrase importance include phrase clustering based on statistical features like word density (Florescu and

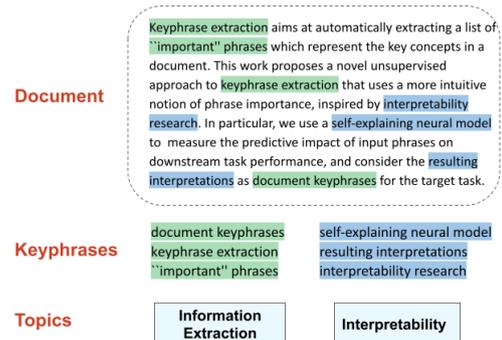


Figure 1: A comprehensive set of keyphrases should highlight important phrases for all major topics in a document. INSPECT identifies such keyphrases using interpretable neural models by measuring how use phrases are for predicting the topic of a text.

Caragea, 2017a; Campos et al., 2018) and structural features like graph centrality (Bougouin et al., 2013; Ding and Luo, 2022) or more recently neural embedding clustering techniques (Bennani-Smires et al., 2018; Zhang et al., 2022; Ding and Luo, 2021; Sun et al., 2020). However, such methods do not generalize to new domains as they require experts to carefully construct domain-specific heuristics (Mani et al., 2020).

Historically, topic models (Blei et al., 2001; Blei and McAuliffe, 2007; Wallach, 2006) have relied on salient words and phrases in a document, which are similar to the notion of keyphrases, although to the best of our knowledge there is no prior work that identified keyphrases using topic models. In this work, we hypothesize that end-to-end neural models for topic classification latently rely on salient phrases for document representation and topic classification. Consequently, if we can interpret model decisions via highlighting salient and influential features (phrases) used for neural topic prediction, we can identify such keyphrases.

Inspired by this intuition, we propose INSPECT—a novel and simple framework to identify keyphrases by leveraging interpretable text classi-

*Equal Contribution

¹Code: <https://github.com/rishabhjoshi/inspect>.

fiers to highlight phrases important for predicting the topics in a text. Specifically, we adapt an interpretable classifier SelfExplain (Rajagopal et al., 2021) to jointly predict the topic of an input document and to identify the salient phrases influencing the prediction. The model is distantly supervised using topic labels from off-the-shelf topic-models, eliminating the need for any human/expert annotations. We consider SelfExplain’s output interpretations as keyphrases for the input document (§2).

INSPECT can be trained on documents of any domain without keyphrase annotations and using distant topic supervision, making them easily adaptable to new domains. We contribute two versions of our method: i) INSPECT— individual models trained for topic-classification for each target dataset. ii) INSPECT-GEN—a more general model pre-trained on a large in-domain corpus, without finetuning on pre-specified target datasets.

We evaluate INSPECT and INSPECT-GEN on four benchmark datasets across two domains: scientific documents and news articles (§3). Our results in §4 show that INSPECT improves keyphrase extraction performance over strong baselines by 0.8% F1 on average, without any domain-specific processing. INSPECT-GEN further improves the performance, outperforming the state of the art in unsupervised keyphrase extraction on 3 out of 4 datasets by 2.7% F1 on average. Our experiments suggest that INSPECT-GEN has strong generalization capabilities, and can be used out-of-the-box without finetuning on individual datasets. Importantly, INSPECT alleviates the need for heuristics and expert-labelled annotations, and thus can be applied to a wide range of domains and problems where keyphrase extraction is important. Our results confirm that the latent keyphrases obtained from an interpretable model correlate with human annotated keyphrases, opening new avenues for research on interpretable models for information extraction.

2 The INSPECT Framework

The goal of the INSPECT framework is to extract important keyphrases in long documents. Following the hypothesis that neural text classifiers latently leverage important keyphrases for predicting topics in text, INSPECT extracts keyphrases through interpreting topic classification decisions. It builds upon an interpretable model, SelfExplain (Rajagopal et al., 2021), which learns to attribute

text classification decisions to relevant phrases in the input. However, SelfExplain was designed and tested in supervised settings and for single-sentence classification; in this work we explore its extension to unsupervised keyphrase extraction from long documents. In what follows, we describe the base SelfExplain model (§2.1) and the distant supervision setup for *topic classification* (§2.4). We outline the training mechanism to jointly predict topics and highlight salient phrases in the document as model interpretations (§2.2) and finally extract the resulting phrase interpretations as important keyphrases in the document (§2.3).

2.1 Base Interpretable Model

Feature attribution methods for model interpretability include two predominant approaches, (i) post-hoc interpretations of a trained model (Jin et al., 2020; Kennedy et al., 2020; Lundberg and Lee, 2017; Ribeiro et al., 2016), and (ii) intrinsically (by-design) interpretable models (Alvarez-Melis and Jaakkola, 2018; Rajagopal et al., 2021). We adopt the latter approach, specifically SelfExplain (Rajagopal et al., 2021) as our phrase attribution model, as the model directly produces interpretations, though in principle any phrase based interpretability techniques could be employed.

SelfExplain augments a pre-trained transformer-based model (RoBERTa (Liu et al., 2019) in our case) with a local interpretability layer (LIL) and a global interpretability layer (GIL) which are trained to produce local (relevant features from input sample) and global (relevant samples from training data) interpretations respectively. The model can be trained for any text classification tasks using gold task supervision, and produces local and global interpretations along with model predictions. Since our goal is to identify important phrases from the input sample, we use only the LIL layer. The LIL layer takes an input sentence and a set of candidate phrases and quantifies the contribution of a particular phrase for prediction through the activation difference (Shrikumar et al., 2017; Montavon et al., 2017) between the phrase and sentence representations.

2.2 Keyphrase Relevance Model

SelfExplain is designed to process single sentences and uses all the phrases spanning non-terminals in a constituency parser as units (candidate phrases) for interpretation. This is computationally expensive for our use-case. To facilitate long document topic

classification, we instead define the set of noun phrases (NPs) as the interpretable units, which aligns with prior work in keyphrase extraction of using noun phrases as initial candidate phrases (Shang et al., 2018; Mihalcea and Tarau, 2004; Bougouin et al., 2013). INSPECT splits a long document into constituent passages, extracts NPs as candidates, and attributes the contribution of each NP for predicting the topics covered in the passage.

For each text block \mathbf{X} in the input document, we preprocess and identify a set of candidate phrases $\mathbf{CP}_X = \mathbf{cp}_1, \mathbf{cp}_2, \dots, \mathbf{cp}_N$ where N is the number candidate phrases in \mathbf{X} . From the base RoBERTa model, we obtain contextual [CLS] representations of the entire text block $\mathbf{h}_{[CLS]}$ and individual tokens. We compute phrase representations $\mathbf{h}_1 \dots \mathbf{h}_N$ for each candidate by taking the sum of the RoBERTa representations of each token in the phrase.

To compute the relevance of each phrase, we construct a representation of the input without the contribution of the phrase, \mathbf{z}_i , using the activation differences between the two representations. We then pass it to a classifier layer in the local interpretability module to obtain the label distribution for prediction.

$$\mathbf{z}_i = g(\mathbf{h}_i) - g(\mathbf{h}_{[CLS]}); \quad \ell_i = f(\mathbf{W}^T \mathbf{z}_i + \mathbf{b}) \quad (1)$$

where g is the ReLU activation function and W and b are the weights and bias of the classifier. Here ℓ_i denotes the label distribution obtained on passing the phrase-level representations \mathbf{z}_i through a classification layer f which is either the sigmoid or the softmax function depending on the prediction task (multi-label versus multi-class). We denote the label distribution from the base RoBERTa model for predicting the output using the whole input block as $\ell_{[CLS]}$. We train the model using the cross entropy loss \mathcal{L}_y with respect to the multi-label gold topics \mathbf{Y}_i for instance i and an explanation specific loss \mathcal{L}_e using the mean of all phrase-level label distributions such that $\ell_e = \sum_{i=1}^P \ell_i$.

$$\mathcal{L}_y = - \sum_{j=1}^N \mathbf{y}_j \log(\ell_{[CLS]}), \quad \mathcal{L}_e = - \sum_{j=1}^N \mathbf{y}_j \log(\ell_e) \quad (2)$$

The classifier is regularized jointly with α regularization parameter² using explanation and classification loss: $\mathcal{L} = (1 - \alpha)\mathcal{L}_y + \alpha\mathcal{L}_e$.

² $\alpha = 0.5$

2.3 Inference

During inference, for each predicted label $y \in \mathbf{Y}$, where \mathbf{Y} denotes set of all predicted labels for input text \mathbf{X} , INSPECT calculates an importance score r_i^y with respect to the predicted label y using the difference between the label distribution ℓ_i^y for a candidate phrase \mathbf{cp}_i and the one obtained using the entire input $\ell_{[CLS]}^y$ as $r_i^y = \ell_{[CLS]}^y - \ell_i^y$.

This score denotes the influence of a candidate keyphrase on the predicted topic. This score denotes the influence of a phrase on the predicted topic—the closer ℓ_i^y is to $\ell_{[CLS]}^y$ the less important phrase i is for predicting the topic. Since the relevance scores are computed with respect to a particular predicted topic and its label distribution, the scores for the same input are not comparable across different predicted topics in multi-label classification (since label distributions can vary in magnitude). To aggregate important keyphrases across all predicted topics, we pick the ones that positively impact prediction for each topic (having a positive influence score) as a set of keyphrases.

$$\mathbf{KP}(x) = [\mathbf{cp}_i \mid \forall r_i^y > 0; y \in \mathbf{Y}; i \in \{1 : N\}]$$

2.4 Distant Supervision via Topic Prediction

Obtaining annotations for keyphrases in specialized domains is challenging for supervised keyphrase extraction (Mani et al., 2020). Instead, we train the interpretable model in a distant supervision setup for multi-class topic classification and use model interpretations to identify keyphrases, without any keyphrase annotations. Topical information about a document are known to be essential for identifying diverse keyphrases (Bougouin et al., 2013; Sterckx et al., 2015). Further, a comprehensive set of keyphrases should represent the various major topics in the document to be useful for different long document applications (Liu et al., 2010). We hypothesize that by using topic classification as our end-task, our model will learn to highlight—via interpretations it is designed to provide—important and diverse keyphrases in the input document.

While certain domains like news articles have extensive datasets with human annotated topic labels, others like scientific articles or legal documents require significant effort for human annotation. INSPECT can be trained using annotated topic labels when they exist. In other domains where such annotations are scarce, INSPECT can be trained using labels extracted unsupervisedly using topic models

Dataset	Type	Split	Total docs	Avg words per doc	Avg keyphrases per doc
SciERC	Scientific	Train	350	130	16
		Dev	50	130	16
		Test	100	134	17
SciREX	Scientific	Train	306	5601	353
		Dev	66	5484	354
		Test	66	6231	387
SemEval17	Scientific	Train	350	160	21
		Dev	50	193	27
		Test	100	186	23
500N-KPCrowd	News	Train	400	430	193
		Dev	50	465	86
		Test	50	420	116
BBC News	News	All	2225	385	-
ICLR	Scientific	All	8317	6505	-

Table 1: Description about the datasets. Average words and keyphrases per document are rounded to the nearest whole number. ICLR and BBC News are used in INSPECT-GEN setting for training and don’t have any labelled keyphrase data.

(Gallagher et al., 2017). Experiments in §4 show results using both settings.

3 Experimental Setup

3.1 Evaluation Datasets

We evaluate INSPECT in two domains using four popular keyphrase extraction datasets—scientific publications (SemEval-2017 (Augenstein et al., 2017a), SciERC (Luan et al., 2018), SciREX (Jain et al., 2020)) and news articles (500N-KPCrowd (Marujo et al., 2013)). Dataset details and statistics are shown in Table 1.

3.2 Topic Labels

We create distant supervision for INSPECT by labeling the above datasets using document topics as labels. We leverage existing topic annotations when such annotations exist. In the 500N-KPCrowd news based dataset, we use existing topic labels (tags or categories such as Sports, Politics, Entertainment) in a one-class classification setting. For the scientific publications domain, we use topic models (Gallagher et al., 2017) to extract $T = 75$ topics where each document can be labeled with multiple topics. The scientific domain datasets are trained in a multi-label classification setup.

3.3 Training Data and Settings

We train INSPECT in two settings:

1. **INSPECT** - Here we assume availability of training documents for each of our datasets. We train the model for topic prediction using only the documents and topic labels from the training set of each dataset obtained using the

approach outlined in §3.2). The training data in this setting, is most closely aligned to the test data, as the documents are of the same topic distribution.

2. **INSPECT-Gen** - We assume no access to training documents and train the model on a large external set of documents of a similar domain (ICLR papers for scientific, BBC News for news) but not necessarily of similar topic distribution as the test data (eg. SemEval-2017 has Physics papers). We use ICLR OpenReview dataset with topics obtained using off-the-shelf topic modeling³ for the scientific domain and BBC News corpus (Greene and Cunningham, 2006) with pre-labelled topics for the news domain.

The model from each setting is then evaluated on the held-out test data of each evaluation dataset.

For the external data, we collect over 8,317 full papers from ICLR and obtained 75 topic labels using topic modeling⁴. We removed 22 topic labels that were uninformative (list in Appendix Table 6) and used the rest to train our model in a multi-label classification setup. The BBC News corpus (Greene and Cunningham, 2006) consists of 2,225 news article documents, each annotated with one of five topics (business, entertainment, politics, sport, or tech).

We pre-process each document (for training and inference) by splitting it into text blocks of size 512 tokens, where consecutive blocks overlap with a stride size of 128. Following Shang et al. (2018),

³https://github.com/gregversteeg/corex_topic

⁴https://github.com/gregversteeg/corex_topic

for each block we consider all Noun Phrases (NPs) as candidate phrases and extract them using a Noun Phrase extractor from the Berkeley Neural Parser⁵. All hyperparameters were chosen based on development set performance on SciERC. Our final models were trained with a batch size of 8 a learning rate of $2e-5$ for 10 epochs. The classification layer dimension was 64 and α was 0.5. We provide more implementation details, including hyperparameter search in Appendix §A.2.

3.4 Baselines

We compare our method against seven unsupervised keyphrase extraction techniques — TF-IDF (Florescu and Caragea, 2017a), TopicRank (Bougouin et al., 2013), Yake (Campos et al., 2018), AutoPhrase (Shang et al., 2018; Liu et al., 2015), UKE-CCRank (Liang et al., 2021), MDERank (BERT)⁶ (Zhang et al., 2022) and SifRank (Sun et al., 2020). Out of the chosen baselines, Yake, TF-IDF and AutoPhrase are statistical, TopicRank is graph-based and SifRank, UKE-CCRank and MDERank are neural embedding based methods. For INSPECT setting, we compare with baselines that only use training data documents—TF-IDF, TopicRank, Yake, AutoPhrase, UKE-CCRank and MDERank. For the INSPECT-GEN setting, we compare with TF-IDF and AutoPhrase trained on our external corpora and SifRank which uses the external corpora to obtain prior likelihood scores for the phrases.

Following prior work and task guidelines (Augenstein et al., 2017a; Jain et al., 2020), INSPECT produces **span level** keyphrases and distinguishes each occurrence of a keyphrase. In contrast, methods like SifRank, AttentionRank, UKE-CCRank and MDERank are phrase level keyphrase extractors which don’t provide span level outputs. To maintain common evaluation, we adapt these methods to span level keyphrase extraction by matching each output keyphrase to all occurrences of the phrase in the document. As our method applies a cutoff on relevance scores and picks any phrase with a positive relevance score as a keyphrase, we cannot be directly compared with baselines which rank candidate phrases and pick top-K phrases as important. To establish a fair setting for evaluation, we choose the average of the number of keyphrase predictions from our model as the ‘K’ across all

⁵<https://pypi.org/project/benepar/>

⁶As of Oct 2022, the authors have not released their model.

Dataset	Method	F1 Score		
		Micro	Macro	Weighted
SciERC	RoBERTa	0.842	0.651	0.767
	INSPECT	0.836	0.658	0.771
SciREX	RoBERTa	0.609	0.404	0.641
	INSPECT	0.628	0.442	0.697
SemEval17	RoBERTa	0.819	0.613	0.731
	INSPECT	0.822	0.611	0.744
500N-KPCrowd	RoBERTa	0.916	0.880	0.910
	INSPECT	0.938	0.904	0.939
ICLR	RoBERTa	0.729	0.456	0.699
	INSPECT	0.743	0.492	0.733
BBC News	RoBERTa	0.880	0.851	0.876
	INSPECT	0.902	0.886	0.894

Table 2: Proxy Task (Topic prediction) performance. Our INSPECT method outperforms a strong RoBERTa baseline on Micro, Macro and Weighted F1 scores.

baselines.

3.5 Evaluation Metrics

Topic Prediction Evaluation: To ensure high-quality interpretations from our model, it is imperative that it performs well on topic prediction. We first evaluate INSPECT’s performance on topic prediction using micro, macro, and weighted F1 score of the classifier’s predictions compared to true labels across all labels.

Keyphrase Extraction Evaluation: For our primary evaluation of keyphrase extraction, we evaluate using span match of our predictions and the true labels (human annotated keyphrases). In addition to measuring quality of keyphrases, this evaluation also measures the quality of explanations from our interpretable topic model by measuring how well the keyphrases extracted by INSPECT align with human annotated keyphrases. Prior works (Shang et al., 2018; El-Beltagy and Rafea, 2009; Bougouin et al., 2013) have mainly focused on *exact match* performance. However, a recent survey highlights that the measure is highly restrictive (Papagiannopoulou and Tsoumakas, 2019) as simple variations in preprocessing can misalign phrases giving an inaccurate representation of the model’s capabilities (Boudin et al., 2016).

Alternatively, *partial span match* using the word level overlap between the predicted and gold span ranges, has also been explored (Rousseau and Vazirgiannis, 2015). But, it is sometimes lenient in scoring. Papagiannopoulou and Tsoumakas (2019) suggest *average of the exact and partial matching* as an appropriate metric based on empirical studies. Therefore, we evaluate performance using the average of the exact and partial match F1 scores

Dataset	Method	Exact Match F1	Partial Match F1	Avg Exact Partial F1
SciERC	TF-IDF	0.0627	0.2860	0.1743
	TopicRank	0.2533	0.5680	0.4110
	Yake	0.2230	0.5125	0.3678
	AutoPhrase	0.0961	0.3145	0.2053
	UKE CCRank	0.3584	0.4804	0.4194
	MDERank	0.3092	0.5102	0.4097
	INSPECT	0.3108	0.5524	0.4316
SciREX	TF-IDF	0.1521	0.3690	0.2605
	TopicRank	0.2298	0.4122	0.3210
	Yake	0.1840	0.3734	0.2787
	AutoPhrase	0.1814	0.4236	0.3025
	UKE CCRank	0.0419	0.0759	0.0589
	MDERank	0.1241	0.3776	0.2509
	INSPECT	0.2397	0.4127	0.3262
SemEval17	TF-IDF	0.0610	0.2698	0.1654
	TopicRank	0.2240	0.4312	0.3276
	Yake	0.1687	0.3644	0.2665
	AutoPhrase	0.0790	0.3404	0.2097
	UKE CCRank	0.2427	0.345	0.2938
	MDERank	0.2529	0.4818	0.3673
	INSPECT	0.2594	0.5185	0.3889
500N-KPCrowd	TF-IDF	0.1034	0.3520	0.2277
	TopicRank	0.1060	0.2346	0.1703
	Yake	0.1380	0.3551	0.2465
	AutoPhrase	0.1590	0.3608	0.2599
	UKE CCRank	0.1729	0.2873	0.2303
	MDERank	0.1522	0.4197	0.2859
	INSPECT	0.1608	0.3920	0.2764

Table 3: Span-match results for unsupervised keyphrase extraction across datasets in the INSPECT setting. Best performance is indicated in Bold. **Our model outperforms baselines on average of exact and partial F1 scores.**

between predicted and true phrases keyphrases.

4 Results

4.1 Topic Prediction with INSPECT

First, we compare INSPECT’s effectiveness in classifying the topics with the corresponding non-interpretable encoder baseline, using micro, macro, and weighted F1 score of the classifier’s predictions compared to gold standard annotations. The results in Table 2 show that our approach outperforms a strong RoBERTa (Liu et al., 2019) baseline for topic prediction across all of our evaluation datasets. The difference is more pronounced in larger datasets (SciREX, ICLR, and BBC News), and strong performance on the topic classification task provides confidence that highlighted interpretations are for relevant and major topics in the text.

4.2 Keyphrase Span Match Performance

Next, we study the utility of INSPECT in highlighting keyphrases via model interpretations. The results for INSPECT are detailed in Table 3 and, for INSPECT-GEN in Table 4.

Results in Table 3 show that even with access to only training set of documents from each dataset, on 3 out of 4 datasets INSPECT outperforms all

baselines with ~ 0.8 average F1 improvements. In the news domain (500-KPCrowd dataset) INSPECT performs comparably to prior best method. INSPECT has low exact match scores but higher partial match scores indicating misalignments between predicted and gold spans. Additionally, 500N-KPCrowd annotates all instances of a keyphrase as a reference span which favours phrase level methods like AttentionRank in the current evaluation setup. In SciREX, we observe very poor performance of UKE CCRank as it ranks common phrases like “image”, “label”, “method”, etc, very high.

In the INSPECT-GEN setting, with access to a larger dataset of external documents, our model outperforms prior methods in 3 out of 4 datasets with ~ 2.7 points average F1 improvements. In the 500N-KPCrowd dataset, INSPECT performs comparably to SifRank with improved Partial Match F1. As Table 4 illustrates, we notice that the model consistently performs better in the INSPECT-GEN setting when compared with the INSPECT setting, showing that the method benefits from more training data. We particularly see large improvements over the INSPECT setting in the scientific datasets, showing that training on a larger set of documents

Dataset	Method	Exact Match F1	Partial Match F1	Avg Exact Partial F1
SciERC	TF-IDF	0.2162	0.4434	0.3298
	AutoPhrase	0.2416	0.6130	0.4273
	SifRank	0.2248	0.7357	0.4803
	INSPECT-GEN	0.4371	0.7114	0.5743
SciREX	TF-IDF	0.1780	0.4008	0.2894
	AutoPhrase	0.2583	0.4993	0.3788
	SifRank	0.1234	0.3957	0.2595
	INSPECT-GEN	0.2601	0.4893	0.3747
SemEval17	TF-IDF	0.1810	0.3398	0.2604
	AutoPhrase	0.1104	0.4874	0.2989
	SifRank	0.2804	0.6336	0.4570
	INSPECT-GEN	0.3246	0.6218	0.4732
500N-KPCrowd	TF-IDF	0.1398	0.3578	0.2488
	AutoPhrase	0.1701	0.3918	0.2805
	SifRank	0.1847	0.4125	0.2986
	INSPECT-GEN	0.1776	0.4194	0.2985

Table 4: Span-match results for unsupervised keyphrase extraction in INSPECT-GEN (trained on ICLR and BBC News corpus). Best performance is indicated in Bold. **INSPECT outperforms most baselines.**

helps generalize the model in this setting. Our results further show that variations in topic distribution between training and test data don’t significantly impact results. INSPECT can thus benefit from large unlabeled documents from similar domains to improve results.

INSPECT improves performance in settings with human annotated topics (news) as well as when topics are extracted using unsupervised topic modeling (scientific). Additionally, most baselines rely on carefully constructed pre- and post-processing to eliminate common phrases and produce high-quality candidates (Liang et al., 2021; Ding and Luo, 2021; Sun et al., 2020). In contrast, INSPECT achieves competitive results without domain expertise and processing for extracting quality keyphrases. Therefore, INSPECT can be easily adapted to new domains without human annotations for topics and with minimal domain knowledge, as we show across two domains.

Our results demonstrate that phrase attribution techniques from interpretability literature can be leveraged to identify high-quality document keyphrases by measuring predictive impact of input phrases on topic prediction. These results also show that our interpretable model in INSPECT produces high quality keyphrases as phrase explanations which correlate with human annotated keyphrases, evaluating the interpretability aspect of our framework. Crucially, as these keyphrases correlate with human annotated keyphrases, our results validate our initial hypothesis that neural models latently use document keyphrases for tasks like topic classification.

Type	Recall	
	Exact	Partial
Metric	60.65	78.34
Task	58.27	90.45
Material	72.17	86.69
Scientific Term	78.87	95.13
Method	65.31	95.41
Generic	63.16	86.06

Table 5: Exact and partial span match recall scores for different types of keyphrases on the SciERC dataset.

5 Discussion

Here, we present an analysis on the common error types in INSPECT and discuss the strengths and weaknesses of INSPECT using qualitative examples.

Entity Type Analysis: We leverage the entity type information in SciERC to observe the performance of INSPECT on specific types of keyphrases. From Table 5, we see that INSPECT performs best on keyphrases labelled as *Scientific Terms* and *Materials*. *Generic* phrases and *Metrics* are usually not representative of topical content, and thus, our method performs poorly on them. On manual analysis, we noticed that many phrases marked as *Task* are very unique and infrequent, making them harder to identify. A high partial match recall but a low exact match recall for *Method* type suggest that many predicted keyphrases are misaligned with the gold labels. We believe that alternative downstream tasks can be explored in future to help tailor our approach to capture specific types of entities, based on application requirements.

Qualitative Analysis In Figure 2 we show two randomly selected abstracts from the SciERC

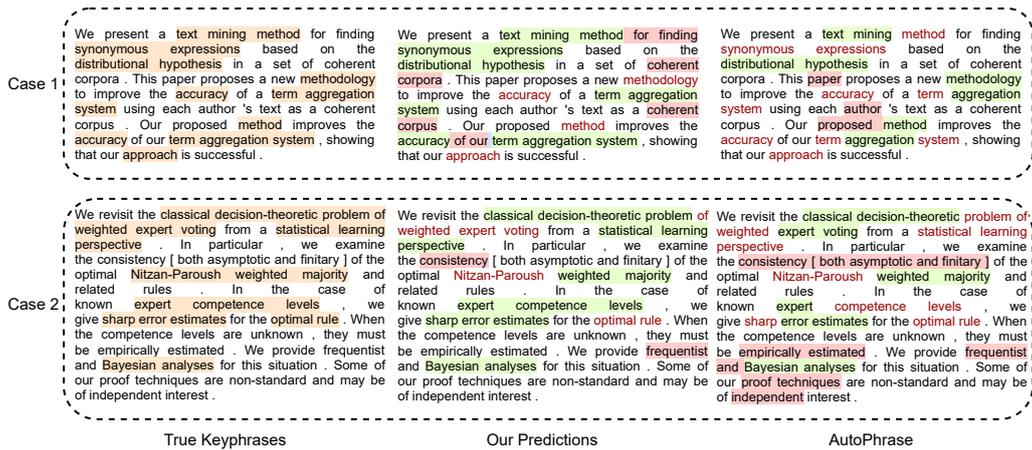


Figure 2: Two data points randomly chosen from the SciERC dataset. Orange spans represent gold standard annotations. Green spans in the predictions represent correctly predicted spans, whereas red spans are spans wrongly predicted as being keyphrases and red text are keyphrases that the model did not identify.

dataset. We see that INSPECT tends to extract longer phrases compared to AutoPhrase, which tends to extract mostly unigrams or bigrams. Overall, our approach is able to extract more relevant phrases than the baseline. Both INSPECT and AutoPhrase tend to miss generic phrases like ‘approach’ (e.g., as seen in case 1). Case 2 also demonstrates the INSPECT’s ability TO predict complete phrases, like ‘classical decision-theoretic problem’, instead of AutoPhrase’s prediction – ‘classical decision-theoretic’ which is incomplete. From both these examples, we see that INSPECT is usually able to correctly extract Scientific Terms, and struggles to extract Generic phrases and Metrics. This can be attributed to the usage of topic models to extract the content’s topical information.

6 Related Work

Unsupervised keyphrase extraction is typically treated as a ranking problem, given a set of candidate phrases (Shang et al., 2018; Campos et al., 2018; Florescu and Caragea, 2017a). Broadly, prior approaches can be categorized as statistical, graph-based, embedding-based, or language model based methods; Papagiannopoulou and Tsoumakas (2019) provide a detailed survey.

Statistical methods exploit notions of information theory directly. Common approaches include TF-IDF based scoring (Florescu and Caragea, 2017a) of phrases with other co-occurrence statistics to enhance performance (Liu et al., 2009; El-Beltagy and Rafea, 2009). Campos et al. (2018) shows the importance of incorporating statistical information of the context of each phrase to improve

performance. Statistical approaches typically treat different instances of a phrase equally, which is a limitation.

Graph-based techniques, on the other hand, broadly aim to form a graph of candidate phrases connected based on similarity to each other. Then core components of the graph are chosen as key phrases. Amongst these, PageRank (Brin and Page, 1998) and TextRank (Mihalcea and Tarau, 2004) assign scores to nodes based on their influence. A common extension is to use weights on the edges denoting the strength of connection (Wan and Xiao, 2008; Rose et al., 2010; Bougouin et al., 2013). Position Rank (Florescu and Caragea, 2017b) and SGRank (Danesh et al., 2015) combine the ideas from statistical, word co-occurrence and positional information. Some approaches, especially applied in the scientific document setting, make use of citation graphs (Gollapalli and Caragea, 2014; Wan and Xiao, 2008), and external knowledge bases (Yu and Ng, 2018) to improve keyphrase extraction. In this work, we focus our approach on a general unsupervised keyphrase extraction setting applicable to any domain where such external resources may not be present.

Finally, embedding based techniques (Bennani-Smires et al., 2018; Papagiannopoulou and Tsoumakas, 2018; Zhang et al., 2022) make use of word-document similarity using word embeddings (Sun et al., 2020; Liang et al., 2021), while language-model based techniques use word prediction uncertainty to decide informativeness (Tomokiyo and Hurst, 2003). Ding and Luo (2021) uses attention scores to calculate phrase importance

with the document in an unsupervised manner.

7 Conclusion and Future Work

In this work, we introduced INSPECT, a novel approach to unsupervised keyphrase extraction. Our framework uses a neural model that explains text classification decisions to extract keyphrases via phrase-level feature attribution. Using four standard datasets in two domains, we show that INSPECT outperforms prior methods and establishes state-of-art results in 3 out of 4 datasets. Through qualitative and quantitative analysis, we show that INSPECT can produce high-quality and relevant keyphrases. INSPECT presents applications of interpretable models beyond explanations for humans.

8 Limitations

Our method uses model explanations for each predicted topic to highlight keyphrases in text. A direct limitation of this method is that our importance scoring is topic-specific and cannot be used to provide an overall rank across topics. Our method therefore cannot provide a ranked list of top-5 or top-10 keyphrases as often done in prior work. While this is a limitation, our current technique of producing a set of all predicted keyphrases is useful in domains like scientific articles where keyphrases are used for downstream applications. Further, as our method produces topic-specific keyphrases, it could potentially miss some keyphrases which are not associated to any predicted topic. Therefore, our approach is beneficial in settings where topic prediction is accurate and feasible to ensure high quality and good coverage of keyphrases. Finally, this work was also limited by the specific choice of the downstream task - namely, topic prediction. Other downstream tasks, like summarization, can potentially help us gain additional insights from attribution.

Acknowledgements

We would like to thank Justin Lovelace, Sachin Kumar, Shangbin Feng, Dheeraj Rajagopal and other members of Tsvetshop Lab for feedback on the paper. This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200004. This material is also funded by the DARPA CMO under Contract No.

HR001120C0124 and by the United States Department of Energy (DOE) National Nuclear Security Administration (NNSA) Office of Defense Nuclear Nonproliferation Research and Development (DNN R&D) Next-Generation AI research portfolio. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

- David Alvarez-Melis and Tommi S Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. *Neurips*.
- Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017a. [SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.
- Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017b. [SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.
- Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. *arXiv preprint arXiv:1801.04470*.
- Arianna Betti, Martin Reynaert, Thijs Ossenkoppele, Yvette Oortwijn, Andrew Salway, and Jelke Bloem. 2020. [Expert concept-modeling ground truth construction for word embeddings evaluation in concept-focused domains](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6690–6702, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- David M. Blei and Jon D. McAuliffe. 2007. Supervised topic models. In *NIPS*.
- David M. Blei, A. Ng, and Michael I. Jordan. 2001. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.
- Florian Boudin, Hugo Mougard, and Damien Cram. 2016. How document pre-processing

- affects keyphrase extraction performance. In *NUT@COLING*.
- Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. **TopicRank: Graph-based topic ranking for keyphrase extraction**. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan. Asian Federation of Natural Language Processing.
- Sergey Brin and Lawrence Page. 1998. **The anatomy of a large-scale hypertextual web search engine**. *Computer Networks*, 30:107–117.
- Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. A text feature based automatic keyword extraction method for single documents. In *European conference on information retrieval*, pages 684–691. Springer.
- Soheil Danesh, Tamara Sumner, and James H. Martin. 2015. **SGRank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction**. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 117–126, Denver, Colorado. Association for Computational Linguistics.
- Haoran Ding and Xiao Luo. 2021. **AttentionRank: Unsupervised keyphrase extraction using self and cross attentions**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1928, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Haoran Ding and Xiao Luo. 2022. Agrank: Augmented graph-based unsupervised keyphrase extraction. In *AACL*.
- Samhaa R. El-Beltagy and Ahmed Rafea. 2009. **Kp-miner: A keyphrase extraction system for english and arabic documents**. *Information Systems*, 34(1):132–144.
- Corina Florescu and Cornelia Caragea. 2017a. A new scheme for scoring phrases in unsupervised keyphrase extraction. In *European Conference on Information Retrieval*, pages 477–483. Springer.
- Corina Florescu and Cornelia Caragea. 2017b. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115.
- Ryan J Gallagher, Kyle Reing, David Kale, and Greg Ver Steeg. 2017. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542.
- Sujatha Das Gollapalli and Cornelia Caragea. 2014. Extracting keyphrases from research papers using citation networks. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI’14*, page 1629–1635. AAAI Press.
- Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine Learning (ICML’06)*, pages 377–384. ACM Press.
- Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. **Scirex: A challenge dataset for document-level information extraction**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2020. **Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models**. In *International Conference on Learning Representations*.
- Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. 2020. **Contextualizing hate speech classifiers with post-hoc explanation**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5435–5442, Online. Association for Computational Linguistics.
- Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. **Unsupervised keyphrase extraction by jointly modeling local and global context**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 155–164, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining quality phrases from massive text corpora. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1729–1744.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *EMNLP*.
- Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 257–266.
- Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. **Multi-task identification of entities, relations, and coreference for scientific knowledge**

- [graph construction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.
- Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.
- Kaushik Mani, Xiang Yue, Bernal Jimenez Gutierrez, Yungui Huang, Simon Lin, and Huan Sun. 2020. Clinical phrase mining with language models. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1087–1090. IEEE.
- Luis Marujo, Márcio Viveiros, and João Paulo da Silva Neto. 2013. Keyphrase cloud generation of broadcast news. In *Proceeding of Interspeech 2011: 12th Annual Conference of the International Speech Communication Association*.
- Dheeraj Mekala and Jingbo Shang. 2020. [Contextualized weak supervision for text classification](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333, Online. Association for Computational Linguistics.
- Rada Mihalcea and Paul Tarau. 2004. Texttrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.
- Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222.
- Eirini Papagiannopoulou and Grigorios Tsoumakas. 2018. Local word vectors guiding keyphrase extraction. *Information Processing & Management*, 54(6):888–902.
- Eirini Papagiannopoulou and Grigorios Tsoumakas. 2019. [A review of keyphrase extraction](#). *CoRR*, abs/1905.05044.
- Dheeraj Rajagopal, Vidhisha Balachandran, E. Hovy, and Yulia Tsvetkov. 2021. Selfexplain: A self-explaining architecture for neural text classifiers. *ArXiv*, abs/2103.12279.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.
- François Rousseau and Michalis Vazirgiannis. 2015. Main core retention on graph-of-words for single-document keyword extraction. In *European Conference on Information Retrieval*, pages 382–393. Springer.
- Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.
- Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.
- Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2015. When topic models disagree: Keyphrase extraction with multiple topic models. *Proceedings of the 24th International Conference on World Wide Web*.
- Yi Sun, Hangping Qiu, Yu Zheng, Zhongwei Wang, and Chaoran Zhang. 2020. Sifrank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model. *IEEE Access*, 8:10896–10906.
- Takashi Tomokiyo and Matthew Hurst. 2003. [A language model approach to keyphrase extraction](#). In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18, MWE ’03*, page 33–40, USA. Association for Computational Linguistics.
- Hanna M. Wallach. 2006. Topic modeling: beyond bag-of-words. *Proceedings of the 23rd international conference on Machine learning*.
- Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, pages 855–860.
- Wenbo Wang, Yang Gao, He-Yan Huang, and Yuxiang Zhou. 2019. Concept pointer network for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3076–3085.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Yang Yu and Vincent Ng. 2018. Wikirank: Improving keyphrase extraction based on background knowledge. *arXiv preprint arXiv:1803.09000*.

Linhan Zhang, Qian Chen, Wen Wang, Chong Deng, ShiLiang Zhang, Bing Li, Wei Wang, and Xin Cao. 2022. *MDERank: A masked document embedding rank approach for unsupervised keyphrase extraction*. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 396–409, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

A.1 Evaluation Datasets

SemEval-2017 (Augenstein et al., 2017a) consists of 500 abstracts taken from 12 AI conferences covering Computer Science, Material Science, and Physics. The entities are annotated with Process, Task, and Material labels, which form the fundamental concepts in scientific literature. Identification of the keyphrases was subtask A of the ScienceIE SemEval task (Augenstein et al., 2017b).

SciERC (Luan et al., 2018) extends SemEval-2017 by annotating more entity types, relations, and co-reference clusters to include broader coverage of general AI. The dataset was annotated by a single domain expert who had high (76.9%) agreement with three other expert annotators on 12% subset of the dataset.

SciREX (Jain et al., 2020) is a document-level information extraction dataset, covering entity identification and n-ary relation formation using salient entities. Human and automatic annotations were used to annotate 438 full papers with salient entities, with a distant supervision from the Papers With Code⁷ corpus. This dataset can help verify the performance of models on full papers.

500N-KPCrowd (Marujo et al., 2013) is a keyphrase extraction dataset in the news domain. This data consists of 500 articles from 10 topics annotated by multiple Amazon Mechanical Turk workers for important keywords. Following the baselines on this datasets, we pick keywords that were among the top two most frequently chosen by the human annotators. Since no span-level information for these keywords is given, we annotate all occurrences of the chosen keywords in the document to obtain a list of span labels, which we use to evaluate all the models.

A.2 Implementation Details

Here, we present the hyper-parameters for all experiments along with their corresponding search space. We chose all hyperparameters based on the development set performance on the SciERC dataset.

We considered RoBERTa (Liu et al., 2019) and XL-NET (Yang et al., 2019) based encoders and finally chose RoBERTa for faster compute times. We experimented with learning-rates from the set of $1e-5, 2e-5, 5e-5, 1e-4$ and $2e-4$. We chose $2e-5$ as the final learning rate. Our batch size of 8 was chosen after experimenting with 4, 8, 12 and 16. The size of the weights matrix in the classification layer was chosen to be 64 from a set of 16, 32, 64 and 128. The α parameter used for regularization was fixed at 0.5. We tried values between 0.1 and 0.9 and did not find significant difference. We saved the model based on best weighted F1 on the topic prediction task. All training runs took less than 3 hours on 2 Nvidia 2080Ti GPUs, except on the ICLR dataset, which took 8 hours. All results are from a single run.

⁷<https://paperswithcode.com/>

S.No.	Top words from removed topic
1	proposed;propose novel;propose;proposed method;method
2	generalization;study;analysis;suggest;provide
3	outperforms;existing;existing methods;outperforms stateofheart;methods
4	state;art;state art;shortterm;current state
5	effectiveness;demonstrate effectiveness;source;effectiveness proposed;student
6	training;training data;training set;training process;model training
7	experimental;experimental results;results;results demonstrate;experimental results demonstrate
8	experiments;extensive;extensive experiments;experiments demonstrate;conduct
9	performance;improves;significantly;improve;improved
10	recent;shown;recent work;recent advances;success
11	achieves;introduce;competitive;achieves stateofheart;introduce new
12	trained;model trained;models trained;networks trained;trained using
13	present;paper present;present novel;work present;monte
14	widely;parameters;widely used;proposes;paper proposes
15	simple;benchmark datasets;benchmark;propose simple;simple effective
16	prior;approach;sampling;continuous;prior work
17	program;introduces;programs;future;paper introduces
18	solve;challenging;able;complex;challenging problem
19	challenge;current;challenges;open;current stateofheart
20	rate;good;good performance;!;regime
21	works;previous works;existing works;focus;scenarios
22	evaluate;evaluation;tackle;tackle problem;evaluate method

Table 6: 22 Generic topics removed from the 75 topic labels learned using topic modeling on ICLR data.

Large Language Models are few(1)-shot Table Reasoners

Wenhu Chen

University of Waterloo, Vector Institute

wenhuchen@uwaterloo.ca

Abstract

Recent literature has shown that large language models (LLMs) are generally excellent few-shot reasoners to solve text reasoning tasks. However, the capability of LLMs on table reasoning tasks is yet to be explored. In this paper, we aim at understanding how well LLMs can perform table-related tasks with few-shot in-context learning. Specifically, we evaluated LLMs on popular table QA and fact verification datasets like WikiTableQuestion, FetaQA, TabFact, and FEVEROUS and found that LLMs are competent at complex reasoning over table structures, though these models are not pre-trained on any table corpus. When combined with ‘chain of thoughts’ prompting, LLMs can achieve very strong performance with only a 1-shot demonstration, even on par with some SoTA models. We show that LLMs are even more competent at generating comprehensive long-form answers on FetaQA than tuned T5-large. We further manually studied the reasoning chains elicited from LLMs and found that these reasoning chains are highly consistent with the underlying semantic form. We believe that LLMs can serve as a simple yet generic baseline for future research. The code and data are released in <https://github.com/wenhuchen/TableCoT>.

1 Introduction

The problem of structured knowledge grounding has been extensively studied for many years. Tables, as one of the most popular (semi)-structured forms to store world knowledge receive significant attention from the natural language processing (NLP) community. Traditional approaches mostly rely on synthesizing executable languages like SQL or SPARQL to access the information inside the table. However, these symbolic languages normally make a rigid assumption about the table and cannot capture the semantics of text chunks inside the table. Such issues are even more pronounced with web tables due to their irregular forms. To fully

understand web tables, both structured reasoning and textual reasoning are required. Such challenges have attracted many researchers to work in the field. Recently, a wide range of table-based tasks have been proposed like table question answering (Pasupat and Liang, 2015; Chen et al., 2020c; Zhu et al., 2021; Chen et al., 2021b; Talmor et al., 2020; Chen et al., 2020a; Nan et al., 2022), table fact verification (Chen et al., 2019; Aly et al., 2021), table-based generation (Chen et al., 2020b; Parikh et al., 2020; Nan et al., 2021), and table-grounded conversation (Budzianowski et al., 2018; Nakamura et al., 2022). This wide range of table-based tasks all come with different input-output formats and domains. Due to the heterogeneity of these tasks, models achieving the best results on these tasks normally need to be fully fine-tuned on the specific downstream dataset with 10K-100K examples to achieve reasonable performance.

Recently, there have been efforts like Unified-SKG (Xie et al., 2022) aiming to unify these heterogeneous table-based tasks as a generic text-to-text format. UnifiedSKG has shown that using T5-3B (Raffel et al., 2020) with the text-to-text format can already achieve state-of-the-art performance on almost all the table-based tasks without task-specific designs. However, the proposed text-to-text models still need to be fully fine-tuned on the downstream tasks. UnifiedSKG also identified that T0-style (Sanh et al., 2022) cross-task transfer can only achieve almost random performance.

Wei et al. (2022); Wang et al. (2022); Zhou et al. (2022); Drozdov et al. (2022) have recently discovered that large language models (Brown et al., 2020; Chowdhery et al., 2022; Ouyang et al., 2022) can be used to solve complex mathematical and commonsense reasoning tasks with few-shot in-context learning. Inspired by this discovery, we aim at understanding whether these LLMs can also solve complex table-based reasoning tasks. Though the LLMs are not specifically designed to encode ta-

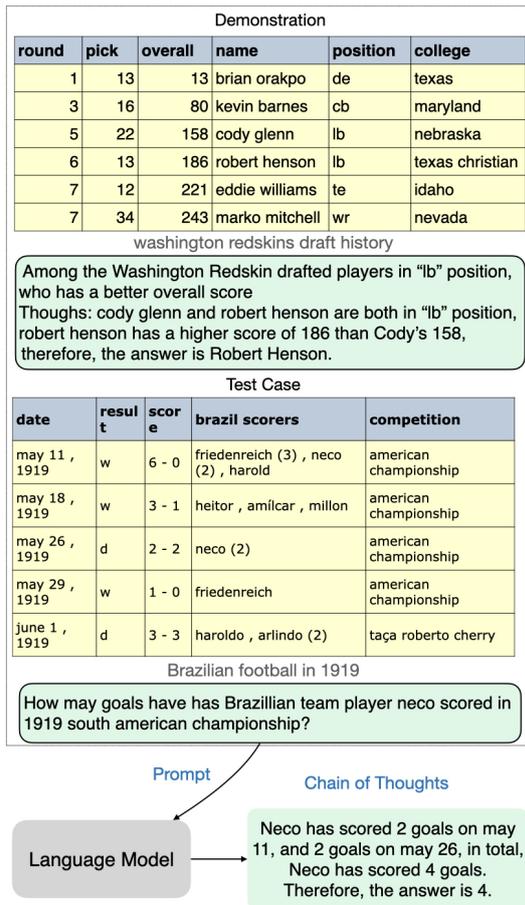


Figure 1: In-context learning for table-related tasks with chain-of-thoughts reasoning.

bles, given the enormous number of tables present in the pre-training corpus, we believe they are also competent at reasoning over table information.

In this paper, we experimented with few-shot in-context learning for LLMs as depicted in Figure 1. Instead of fine-tuning the model, we only provide a few examples to showcase the desired input-output format as the condition for the model to follow to solve unseen test examples. We experiment with several prompting variants including (1) direct prediction, (2) Chain of Thoughts (Wei et al., 2022) (CoT), (3) Chains of thoughts with self-consistency (Wang et al., 2022) (CoT+SC). We evaluate these methods on WikiTableQA (Pasupat and Liang, 2015), FetaQA (Nan et al., 2022), TabFact (Chen et al., 2019) and FEVEROUS (Aly et al., 2021). Our results reveal that LLMs (Ouyang et al., 2022; Chen et al., 2021a; Chowdhery et al., 2022) can achieve striking performance with only 1 or 2 demonstrations, e.g. 48.8% on WikiTableQuestions and 78.8% on TabFact, which are on par some near-SoTA models (Yu et al., 2021; Eisen-

schlos et al., 2020). On other datasets like FetaQA with long-form answers, our human evaluation reveals that GPT-3 can significantly outperform the fine-tuned T5-large by more than 30% in terms of correctness and adequacy.

Furthermore, we manually studied the chain of thoughts elicited from LLMs and found that the rationale is highly consistent with the ‘ground truth’ semantic forms when the model predictions are correct. We found that these models are surprisingly competent at performing symbolic operations over the table, like maximum, minimum, counting, comparison, addition, and difference. However, we also identify several issues of the LLMs on these table reasoning tasks: (1) due to the token limitation, the model is unable to generalize to ‘huge’ tables with 30+ rows, which is the major error source, (2) LLMs can sometimes make simple mistakes when performing symbolic operations.

Due to the simplicity and generality, we believe LLMs with CoT should be used as an important baseline for any future table-related research.

2 Related Work

2.1 Reasoning over Tables

Table-based reasoning is traditionally accomplished by semantic parsing to execute commands on tables like WikiTableQuestions (Pasupat and Liang, 2015), WikiSQL (Zhong et al., 2017), and Spider (Yu et al., 2018). These models aim to synthesize SQL/SPARQL to interact with tables. However, these machine languages have a rigorous requirement regarding the tables, e.g. the value in the same column should follow the same data type. Such rigorous assumptions are frequently violated by web tables containing unnormalized free-form text in cells. Therefore, language understanding inside the table is essential to achieve a better score. Recently, Yin et al. (2020); Herzig et al. (2020); Liu et al. (2021); Deng et al. (2022) have proposed to pre-train table and text to learn joint representation. These pre-trained models can use joint representation to perform reasoning implicitly without relying on symbolic execution. By pre-training the model on large-scale crawled or synthesized data, these models can normally achieve the best-known performance on table tasks. However, these models still require a significant amount of fine-tuning on the downstream datasets. Unlike these methods, we are interested in in-context learning, where the model can only learn with a

few examples (demonstration) without any fine-tuning. One contemporary work similar to ours is BINDER (Cheng et al., 2022), which utilizes Codex to synthesize SQL to execute logical forms against tables for question answering. One big difference is that BINDER (Cheng et al., 2022) involves logical form execution, if the execution fails, BINDER will fall back to using language models to answer the question, which is more similar to ours.

2.2 In-context Learning with LLMs

GPT-3 (Brown et al., 2020) and other large language models demonstrated strong abilities to perform few-shot predictions without fine-tuning, where the model is given a description of the task in natural language with few examples. Scaling model size, data, and computing are crucial to enable this learning ability. Recently, (Rae et al., 2021; Smith et al., 2022; Chowdhery et al., 2022; Du et al., 2022) have proposed to train different types of large language models with different training recipes. The LLMs have demonstrated a striking capability utilizing the few-shot prompts to accomplish unseen tasks without any fine-tuning, which is found to be an emergent capability not presented in smaller language models.

2.3 Chain of Thoughts Reasoning

Although LLMs (Brown et al., 2020; Chowdhery et al., 2022) have demonstrated remarkable success across a range of NLP tasks, their ability to demonstrate reasoning is often seen as a limitation. Such capability cannot be acquired simply by scaling up the model size. Recently, the ‘chain of thoughts’ prompting (Wei et al., 2022) has been discovered to empower LLMs to perform complex reasoning over text. By providing the model with several exemplars of reasoning chains, LLMs can learn to follow the template to solve difficult unseen tasks. Later, Wang et al. (2022) propose to use self-consistency with CoT to further improve performance. Later on, Kojima et al. (2022) discovered that LLMs can even perform reasoning without any demonstration by using appropriate prompts. These recent findings reveal the strong capability of LLMs to perform complex reasoning. However, the current studies are still heavily focused on text-based tasks like question answering, common sense reasoning, etc. The models’ capability to reason over tables is yet unknown. In this paper, we are specifically interested in understanding LLMs’ capability to

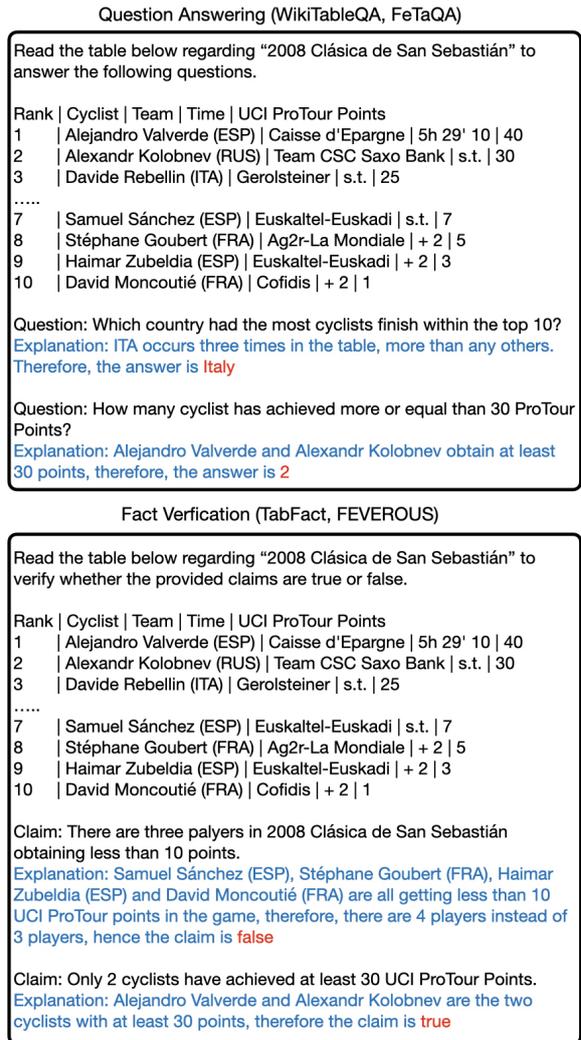


Figure 2: Prompts used for question answering and fact verification tasks.

reason over web tables with CoT prompting.

3 Method

We experiment with different in-context learning methods to solve the table-based reasoning tasks. To formulate the prompt, we linearize the table and concatenate it with a few examples as demonstrations of the language model to predict the output from an unseen test example. The format is described in Figure 2. We mainly investigate three different variants for language model prompting, including (1) Direct Prediction, (2) Chain of Thoughts (CoT), and (3) Chain of Thoughts + Self-Consistency decoding (CoT+SC). For self-consistency methods, we use LLMs to generate five diverse reasoning paths and then use majority voting to select the most voted answer.

To limit the budget and constrain the input token length, we truncate the input tables to contain only

the first 22 rows and the first 8 columns. For each cell, we truncate the word length to contain only the first 10 words. Through such truncation, we can restrict the input token length to within 2000 tokens. We will talk about the impact of input token length on the final performance.

4 Experimental Results

For the GPT-3 experiments, we used the four provided models, Ada, Babbage, Curie, and Davinci with 350M, 1.3B, 6.7B, and 175B parameters respectively. We mainly use Davinci-text-002 (Ouyang et al., 2022) in our experiments. We also report results for Codex (Chen et al., 2021a) (Davinci-code-002) on some datasets. We use a temperature of 0.7 without any frequency penalty and without top-k truncation. We found that the model performance is robust to the sampling strategies and the hyper-parameters. These models are mainly trained on web-crawled data and code data, without any specialized training on table corpus.

4.1 Datasets

Here we list all of our datasets as follows:

WikiTableQuestions Pasupat and Liang (2015) consists of complex questions annotated based on Wikipedia tables. Crowd Workers are asked to compose a series of complex questions that include comparisons, superlatives, aggregation, or arithmetic operations. The annotated dataset is cross-validated by other crowd workers. In our experiments, we use the unseen test set for evaluation. We evaluate the standard test set with roughly 4000 questions. In this dataset, we adopt the answer exact match as our evaluation metric.

FetaQA Nan et al. (2022) consists of free-form table questions. These questions are mostly complex questions that require integrating information from discontinuous chunks in the table. Instead of having short answers, the dataset annotates long free-form answers. Unlike other datasets using copies of short text spans from the source, the questions in FetaQA require a high-level understanding. We adopt sacre-BLEU and human evaluation as our evaluation metrics. The evaluation set contains a total of 2003 examples.

TabFact Chen et al. (2019) consists of both simple and complex claims annotated by crowd workers based on Wikipedia tables. In the simple

subset, the claims normally do not involve higher-order operations like max/min/count, etc. While the complex subset mainly contains claims involving higher-order operations. We evaluate the original test set containing 12,779 examples. We report binary classification accuracy on the set.

FEVEROUS Aly et al. (2021) consists of compositional claims annotated by crowd workers regarding Wikipedia tables. Since the dataset contains both table-supported and text-supported claims. We filter out text-supported claims and only keep the 2,295 table-supported claims as our test set. Different from TabFact, FEVEROUS consists of more complex tables with irregular structures like multi-row, multi-column, multi-table, etc. We report dev-set accuracy.

4.2 Baselines

In these experiments, we mainly consider the following baseline models.

Pre-trained Encoder-Decoder Model Pre-trained encoder-decoder model is one of our competitors, which aims to encode the table as a plain sequence into the encoder, and then apply the decoder to generate either an answer or a verdict. In this paper, we mainly compare against T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) as our baselines.

Pre-trained Table Understanding Model This family of models is specifically pre-trained on the table-related corpus, which utilizes specific architecture to encode table structure and handle symbolic computation. In this paper, we mainly consider TAPAS (Herzig et al., 2020), TABERT (Yin et al., 2020), and TAPEX (Liu et al., 2021).

Neural Symbolic Model This family of models includes a non-pre-trained neural symbolic model, which can synthesize machine language to interact with the table. This line of work includes LogicFactChecker (Zhong et al., 2020), Neural-Symbolic Machine (Liang et al., 2018), etc.

4.3 Main Results

Here we show our main results for different datasets as follows.

WikiTableQuestions As can be seen from Table 1, directly asking GPT-3 to generate answers can only lead to 26% EM score. However, if we prompt the model with the CoT demonstrations,

Type	Model	Test EM
Train	Pasupat and Liang (2015)	37.1
Train	Zhang et al. (2017)	43.7
Train	Liang et al. (2018)	43.7
Train	Agarwal et al. (2019)	44.1
Train	Wang et al. (2019)	44.5
PT + FT	Herzig et al. (2020)	48.8
PT + FT	Yu et al. (2021)	52.7
1-shot	GPT-3 Direct	24.0
2-shot	GPT-3 Direct	27.3
1-shot	GPT-3 CoT	44.2
2-shot	GPT-3 CoT	45.7
2-shot	Codex CoT	48.8

Table 1: Experimental Results on WikiTableQuestions. PT means pre-training and FT means fine-tuning.

GPT-3 is more likely to follow the logical operation to derive the answers. With two demonstrations, GPT-3 can achieve roughly 46% EM score. By switching from GPT-3 to Codex, we are able to further improve the EM score to over 48.8%. These results are particularly surprising given that TAPAS has a built-in module to complete symbolic operations, while GPT-3 was not trained on any table-specific dataset. These results demonstrate GPT-3’s built-in capabilities to perform diverse types of reasoning over tables.

FetaQA As demonstrated in Table 2, we compare GPT-3 with different fine-tuned models from Nan et al. (2022). Unlike the other datasets with short phrase answers, the goal of this dataset is to generate a complete long-form answer. Unlike WikiTableQuestion, the questions normally do not involve complex operations like max, min, compare, average, etc. The long-form answer is similar to the role of CoT. Therefore, we only applied ‘direct generation’ in this experiment. In terms of BLEU score (Papineni et al., 2002), GPT-3 is still a bit behind the fine-tuned T5-large. However, the BLEU score cannot reflect the faithfulness and correctness of the model generation. Thus, we follow Nan et al. (2021) to do human evaluation over the four aspects: (1) fluency (whether the generated sentence contains the linguistic error), (2) correctness (whether the generated sentence answers the question correctly), (3) faithfulness (whether the generated sentence is grounded on the input table), and (4) adequacy (whether the generated sentence is comprehensive enough to cover all the answers). We list our results in Table 3. Similarly, we also sample 100 model predictions and manually evaluate their quality and adopt binary scores for each

Type	Model	sacreBLEU
zero-shot	Pipeline (Nan et al., 2022)	9.16
FT	Pipeline (Nan et al., 2022)	11.00
FT	T5-small (Nan et al., 2022)	21.60
FT	T5-base (Nan et al., 2022)	28.14
FT	T5-large (Nan et al., 2022)	30.54
1-shot	GPT-3 Direct	26.88
2-shot	GPT-3 Direct	27.02

Table 2: Experimental Results on FetaQA. PT means pre-training and FT means fine-tuning.

Source	Fluency	Correct	Adequate	Faithful
Pipeline	85.2	25.4	23.6	23.6
T5-large	94.6	54.8	50.4	50.4
Human	95.0	92.4	95.6	95.6
GPT-3	98.0	84.0	78.0	90.0

Table 3: Human Evaluation Results on FetaQA.

example. As can be seen, GPT-3 can significantly outperform T5-large over all the aspects, i.e. more than 30% improvement over correctness, adequacy, and faithfulness. The evaluation indicates that the model output is almost on par with the average human performance on this dataset.

TabFact As demonstrated in Table 4, we compare GPT-3 against the other pre-trained and fine-tuned models including TAPAS (Eisenschlos et al., 2020), TAPEX (Liu et al., 2021), etc. We show that GPT-3 direct prediction is already getting a decent accuracy of 72%, which is slightly higher than Logic FactChecker (Zhong et al., 2020). When combined with CoT reasoning, the model accuracy increases to over 77%. Similar to before, we found that Codex can generate more accurate reasoning chains, thus achieving better accuracy of 78.8%, which is only 2% lower than pre-trained table understanding model TAPAS (Eisenschlos et al., 2020). The more intriguing property about LLM + CoT is that the intermediate rationale can be produced without any training. All the existing trained models do not have the capability to produce the intermediate reasoning steps due to the lack of annotation in the dataset.

FEVEROUS We demonstrate our results on FEVEROUS dev-set in Table 5 and compare different-sized UnifiedSKG models (built with T5). We found that GPT-3’s performance with direct prediction is similar to UnifiedSKG-base. Similar to TabFact, we found that the model performance can be boosted with ‘chain of thoughts’ prompt-

Type	Model	Overall
FT	Chen et al. (2019)	65.1
FT	Zhong et al. (2020)	71.1
FT	Zhang et al. (2020)	73.2
FT	Yang et al. (2020)	74.4
FT	Lewis et al. (2020)	82.5
PT + FT	Eisenschlos et al. (2020)	81.0
PT + FT	Liu et al. (2021)	84.2
1-shot	GPT-3 Direct	72.0
2-shot	GPT-3 Direct	73.9
1-shot	GPT-3 CoT	75.5
2-shot	GPT-3 CoT	76.0
1-shot	GPT-3 CoT+SC	77.3
2-shot	Codex CoT	78.8

Table 4: Experimental Results on TabFact. PT means pre-training and FT means fine-tuning.

Type	Model	Dev Set
FT	Aly et al. (2021)	82.23
FT	UnifiedSKG-base (Xie et al., 2022)	75.05
FT	UnifiedSKG-large (Xie et al., 2022)	79.81
FT	UnifiedSKG-3B (Xie et al., 2022)	82.40
1-shot	GPT-3 Direct	74.20
2-shot	GPT-3 Direct	75.22
1-shot	GPT-3 CoT	75.70
2-shot	GPT-3 CoT	76.44
1-shot	GPT-3 CoT+SC	77.22

Table 5: Experimental Results on FEVEROUS. PT means pre-training and FT means fine-tuning.

ing. The best-performing model is roughly between UnifiedSKG-base and UnifiedSKG-large. Compared to TabFact, the model’s overall performance is weaker mainly because the table structure in FEVEROUS is more irregular, containing lots of segments and subtables. Such structural difficulties pose great challenges to GPT-3.

Model Scaling We investigate the model scaling’s impact on the final performance and plot our findings in Figure 3. On the WebTableQuestions dataset, we found that model size is essential for achieving the best performance. As can be seen, the 6.7B GPT-3 model is only achieving half of the performance of the 175B GPT-3 model. Similarly, on TabFact, we found that the smaller models with 6.7B or fewer parameters are almost getting random accuracy, which is even worse than QA tasks. This again suggests that LLMs’ reasoning ability over web tables is emergent as the model scales up.

4.4 Case Study

We demonstrate a few examples in Figure 4 where GPT-3 makes correct predictions. In the first example, GPT-3 is able to first identify all the Belgian

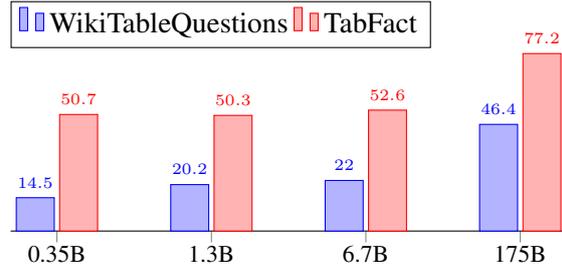


Figure 3: The model performance with respect to model size on WikiTableQuestions and TabFact.

riders from the table and then perform the addition of $3+3+1=7$ precisely. In the second example, GPT-3 can identify the players with the position of ‘d’ and count the number correctly to refute a false claim. In the third example, we can see that GPT-3 is able to associate multiple blocks of information to generate a comprehensive long-form answer. The elicited ‘chain of thoughts’ in these examples are highly aligned with the underlying semantic forms. These findings suggest that LLMs like GPT-3 can provide high-quality explanations to justify their decision-making.

We also provide a few mistakes made by GPT-3 in Figure 5. In the first example, GPT-3 miscounts the ‘number of countries above 1 billion box office’ because it misidentifies ‘world’ also as a country. In the second example, GPT-3 misunderstood ‘2nd highest’ as ‘highest’, which leads to prediction error. In the last example, GPT-3 misunderstands the semantics of the question and answers ‘left office time’ instead of ‘took office time’. These examples show the typical errors of grounding the inputs to the wrong rows or columns of the table.

4.5 Analysis

Impact of Number of Shots First of all, we conduct an ablation study to understand the impact of a number of shots in the final performance. In order to control the budget, we only sample 200 samples from WikiTableQuestions, TabFact and FEVEROUS for this ablation study. As can be seen from Figure 7, GPT-3 is not quite sensitive to the number of provided demonstrations. Increasing from 1-shot to 2-shot can often benefit the model, however, increasing the shot number further does not yield more performance gain. We conjecture that instruct fine-tuning used in GPT-3 (Ouyang et al., 2022) can easily extrapolate the task meaning, thus, having a single demonstration is already enough for the model to understand the task.

Place	Rider	Country	Team	Points	Wins
1	Sylvain Geboers	Belgium	Suzuki	3066	3
2	Adolf Weil	Germany	Maico	2331	2
3	Torlief Hansen	Sweden	Husqvarna	2052	0
4	Roger De Coster	Belgium	Suzuki	1865	3
5	Joel Robert	Belgium	Suzuki	1730	1
6	Heikki Mikkola	Finland	Husqvarna	1680	2
7	Willy Bauer	Germany	Maico	1276	0
8	Gaston Rahier	Belgium	ČZ	1112	0
9	Pierre Karsmakers	Netherlands	Husqvarna	1110	0
10	Dave Bickers	United Kingdom	ČZ	1076	0

Question: total wins by belgian riders?

According to the table, Sylvain Geboers, Roger De and Joel Robert are Belgian riders and they have totally 3+3+1=7 wins. Therefore, the answer is 7.

name	pos	height	2012 club
Merrill Moses	gk	m	New York athletic club
Peter Varellas	d	m	the Olympic club
Peter Hudnut	cb	m	Los Angeles wp club
Jeff Powers	cf	m	Newport wp foundation
Adam Wright	d	m	New York athletic club
Shea Buckner	d	m	New York athletic club
Layne Beaubien	d	m	New York athletic club
Tony Azevedo	d	m	New York athletic club
Ryan Bailey	cf	m	Newport wp foundation
Chay Lapin	gk	m	long beach shore aquatics

Claim: more players have the position of gk than any other position.

There are 2 players with the position of gk and 5 players with the position of d. 2 is less than 5, therefore, the claim is false.

Year	Title	Role	Notes
1970	House of Dark Shadows.	Willie Loomis	-
1971	Daughters of Darkness	Stefan	-
1971	Night of Dark Shadows	Alex Jenkins	-
1976	A Small Town in Texas	Deputy Lenny Lutz	-
1978	Killer's Delight	Danny	-
1981	Pennies from Heaven	The Detective	-
1984	Racing with the Moon	Mr. Nash	-
1984	Impulse	Bob Russell	-
1984	Gimme an 'F'	Bucky Berkshire	-
1986	Native Son	Max	-
1991	The Dark Wind	Jake West	-
1993	Surf Ninjas	Mac	-
2016	The Job Interview	The Applicant	Short film

Question: What were John Karlen's roles in 1970 and 1978?

John Karlen's roles in 1970 were Willie Loomis in House of Dark Shadows and Stefan in Daughters of Darkness. His role in 1978 was Danny in Killer's Delight.

Figure 4: 'Correct' predictions from WikiTableQuestions, TabFact, and FetaQA datasets, where the 'blue' text are the outputs from the GPT-3, 'red' means the correct rows to reference.

Quality Evaluation of Reasoning Chains We conduct a human evaluation to assess whether GPT-3 is making the correct prediction with the correct reasons. Specifically, we sample 100 reasoning paths from the correctly predicted examples and manually study whether these reasoning chains are grounded on the table or simply 'hallucination'. As can be seen from Figure 7, we found that around 90% of reasoning chains are faithful to the infor-

Rank	Country	Box Office	Year	office from national
1	Canada/United States	\$10.8 billion	2012	-
2	China	\$3.6 billion	2013	59% (2013)
3	Japan	\$1.88 billion	2013	61% (2013)
4	United Kingdom	\$1.7 billion	2012	36.1% (2011)
5	France	\$1.7 billion	2012	33.3% (2013)
6	South Korea	\$1.47 billion	2013	59.7% (2013)
7	India	\$1.4 billion	2012	-
8	Germany	\$1.3 billion	2012	-
9	Russia	\$1.2 billion	2012	-
10	Australia	\$1.2 billion	2012	4.1% (2011)
11	Italy	\$0.84 billion	2013	30% (2013)
12	Brazil	\$0.72 billion	2013	17% (2013)
-	World	\$34.7 billion	2012	-

Question: how many countries had at least \$1 billion in box office?

Canada/United States, China, Japan, United Kingdom, France, South Korea, India, Germany and Russia, Australia and world all had at least \$1 billion in box office. Therefore, the answer is 11.

no in series	title	directed by	original air date
47	knight of the drones (part 1)	sidney hayers	september 30 , 1984
48	knight of the drones (part 2)	sidney hayers	september 30 , 1984
49	the ice bandits	georg fenady	october 7 , 1984
...
61	buy out	jeffrey hayden	february 10 , 1985
62	knightlines	charles watson sanford	march 3 , 1985
63	the nineteenth hole	georg fenady	march 10 , 1985
64	knight & knerd	georg fenady	march 17 , 1985
65	ten wheel trouble	bob bralver	march 24 , 1985
66	knight in retreat	roy campanella ii	march 29 , 1985
67	knight strike	georg fenady	april 5 , 1985

Claim: knight in retreat is the 2nd highest no in series.

Knight in strike has no of series of 67, which is the highest no in series, therefore, the claim is false.

Took office	Left office	President	Party
7 Oct 1985	2 May 1988	Dick Ukeiwé	The Rally-UMP
2 May 1988	11 June 1989	Albert Etuve	The Rally-UMP
28 June 1989	7 July 1995	Simon Loueckhote	The Rally-UMP
31 July 1995	16 July 1997	Pierre Frogier	The Rally-UMP
16 July 1997	3 June 1998	Harold Martin	The Rally-UMP
3 June 1998	9 May 2004	Simon Loueckhote	The Rally-UMP
21 May 2004	3 July 2007	Harold Martin	L'Avenir Ensemble
31 July 2007	10 May 2009	Pierre Frogier	The Rally-UMP
22 May 2009	3 March 2011	Harold Martin	L'Avenir Ensemble
1 April 2011	29 August 2012	Roch Wamytan	Kanak and Socialist
29 Aug 2012	7 August 2013	Gérard Poadja	Caledonia Together
23 May 2014	17 July 2015	Gaël Yanno	Popular Caledonian
17 July 2015	30 July 2018	Thierry Santa	The Rally-UMP
30 July 2018	Incumbent	Gaël Yanno	Popular Caledonian

Question: On which dates did Harold Martin take office as President of the Congress of New Caledonia?

Harold Martin took office as President of the Congress of New Caledonia on 3 June 1998, 3 July 2007 and 3 March 2011.

Figure 5: 'Wrong' predictions from WikiTableQuestions, TabFact, and FetaQA datasets, where 'blue' text are the outputs from the GPT-3, 'red' means the region of the correct cell to reference, and 'green' means the reference trusted by GPT-3.

mation in the table, and only less than 10% of the reasoning chains are hallucinated. Based on this

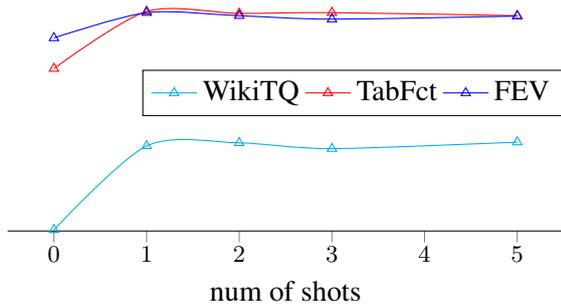


Figure 6: k-shot ablation study over WikiTableQuestions and TabFact and FEVEROUS.

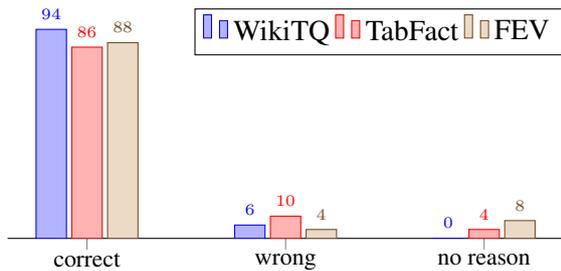


Figure 7: human evaluation of ‘reasoning chains’ in WikiTableQuestions, TabFact, and FEVEROUS.

evaluation, we believe that LLMs are not guessing the answers correctly by chance.

We believe these ‘reasoning chains’ are useful in many aspects: (1) the chains can provide a rationale to humans to justify the decision-making process. (2) one of the notorious annotation tasks is to annotate the ‘underlying’ semantic form for many NLP tasks, which require expertise for human annotators, on the other hand, the annotation cost is huge. Using GPT-3 to demonstrate useful natural language ‘semantic forms’ could potentially greatly lower the annotation burden of these tasks.

Impact of Table Size An important factor for model performance is the size of the table. Here we want to understand how relevant the model performance is w.r.t the input table length. We group the table token length into different groups like ‘0-100’, ‘100-200’, etc, and plot the group-wise accuracy for WikiTables and TabFact in Figure 8. As can be seen from the table, we found that GPT-3’s performance is highly sensitive to the table size. As the table size grows, the accuracy almost decreases monotonically. After the table size exceeds 1000 tokens (e.g. 1500 word pieces), GPT-3’s performance almost degrades to random guesses. This ablation study reveals one of the drawbacks of using LLMs for table reasoning. To further enhance

LLMs’ performance, we need to develop better methods to maintain more consistent performance across different-sized tables.

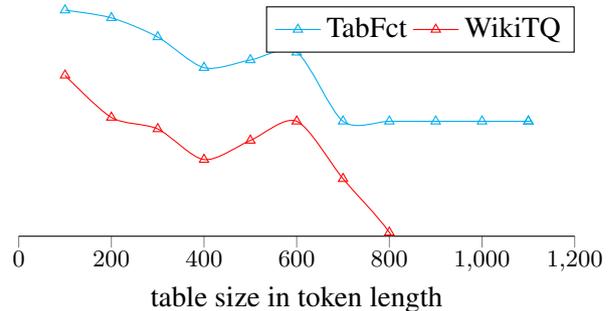


Figure 8: Model performance on WikiTableQuestions and TabFact w.r.t the input table size.

Discussions In this study, we investigate the possibilities of prompting LLMs to perform complex reasoning tasks over tables. However, we do not believe LLM prompting can replace the existing symbolic methods. LLMs have several favorable properties: (1) no annotation is needed, and (2) the functional coverage is broader than symbolic methods. However, LLM prompting exhibits unpredictable randomness and cannot generalize to large tables. In contrast, symbolic models are (1) agnostic to the table size, and (2) can reliably perform designed functions without much randomness. But they in general require a significant amount of annotated data to learn.

In conclusion, these two types of models are complementary to each other. To push the limit forward, we need to investigate how to combine the merits of these two types of methods. For example, the symbolic methods can perform certain operations to narrow down to a targeted region in the table, and then LLMs can be used to reason over the limited information.

5 Conclusion

In this paper, we investigate whether the current LLMs (GPT-3) can be directly utilized to perform table reasoning tasks. Surprisingly, though LLMs are not optimized for table-based tasks, we found these models highly competent in performing complex table reasoning tasks, especially when combined with ‘chain of thoughts’ prompting. We believe this study can open new possibilities for LLM application in table-related tasks to either directly predict the output or to serve as an auxiliary tool for annotating complex intermediate forms.

Limitations

Our approach has several limitations: (1) the proposed approach is still far from state-of-the-art performance, and there is still room for improve before it can be used as an alternative. (2) the method is still costly, we show that the model can only achieve superior performance when scaling up. Smaller-sized models are still weak at table reasoning. Therefore, we need to consider how to empower smaller models with such reasoning capabilities.

References

- Rishabh Agarwal, Chen Liang, Dale Schuurmans, and Mohammad Norouzi. 2019. Learning to generalize from sparse and underspecified rewards. In *International conference on machine learning*, pages 130–140. PMLR.
- Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and verification over unstructured and structured information (feverous) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021a. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W Cohen. 2020a. Open question answering over tables and text. In *International Conference on Learning Representations*.
- Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020b. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942.
- Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.
- Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020c. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036.
- Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. 2021b. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.
- Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, et al. 2022. Binding language models in symbolic languages. *arXiv preprint arXiv:2210.02875*.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record*, 51(1):33–40.
- Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. 2022. Compositional semantic parsing with large language models. *arXiv preprint arXiv:2209.15003*.
- Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.
- Julian Eisenschlos, Syrine Krichene, and Thomas Mueller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296.
- Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Mueller, Francesco Piccinno, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333.

- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.
- Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. *Advances in Neural Information Processing Systems*, 31.
- Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2021. Tapex: Table pre-training via learning a neural sql executor. In *International Conference on Learning Representations*.
- Kai Nakamura, Sharon Levy, Yi-Lin Tuan, Wenhua Chen, and William Yang Wang. 2022. Hybridialogue: An information-seeking dialogue dataset grounded on tabular and textual data. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 481–492.
- Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Nick Schoelkopf, Riley Kong, Xiangru Tang, et al. 2022. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.
- Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, et al. 2021. Dart: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186.
- Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480.
- Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations*.
- Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deep-speed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.
- Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hananeh Hajishirzi, and Jonathan Berant. 2020. Multimodalqa: complex question answering over text, tables and images. In *International Conference on Learning Representations*.
- Bailin Wang, Ivan Titov, and Mirella Lapata. 2019. Learning semantic parsers from denotations with latent structured alignments and abstract programs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3774–3785.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

- Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*.
- Xiaoyu Yang, Feng Nie, Yufei Feng, Quan Liu, Zhigang Chen, and Xiaodan Zhu. 2020. Program enhanced fact verification with verbalization and graph attention network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7810–7825.
- Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426.
- Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir R Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: Grammar-augmented pre-training for table semantic parsing. In *ICLR*.
- Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921.
- Hongzhi Zhang, Yingyao Wang, Sirui Wang, Xuezhi Cao, Fuzheng Zhang, and Zhongyuan Wang. 2020. Table fact verification with structure-aware transformer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1629.
- Yuchen Zhang, Panupong Pasupat, and Percy Liang. 2017. Macro grammars and holistic triggering for efficient semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223.
- Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.
- Wanjun Zhong, Duyu Tang, Zhangyin Feng, Nan Duan, Ming Zhou, Ming Gong, Linjun Shou, Daxin Jiang, Jiahai Wang, and Jian Yin. 2020. Logicalfactchecker: Leveraging logical operations for fact checking with graph module network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6065.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.
- Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287.

Realistic Citation Count Prediction Task for Newly Published Papers

Jun Hirako Ryohei Sasano Koichi Takeda
Graduate School of Informatics, Nagoya University
hirako.jun.e5@s.mail.nagoya-u.ac.jp
{sasano, takedasu}@i.nagoya-u.ac.jp

Abstract

Citation count prediction is the task of predicting the future citation counts of academic papers, which is particularly useful for estimating the future impacts of an ever-growing number of academic papers. Although there have been many studies on citation count prediction, they are not applicable to predicting the citation counts of newly published papers, because they assume the availability of future citation counts for papers that have not had enough time pass since publication. In this paper, we first identify problems in the settings of existing studies and introduce a realistic citation count prediction task that strictly uses information available at the time of a target paper’s publication. For realistic citation count prediction, we then propose two methods to leverage the citation counts of papers shortly after publication. Through experiments using papers collected from arXiv and bioRxiv, we demonstrate that our methods considerably improve the performance of citation count prediction for newly published papers in a realistic setting.

1 Introduction

In recent years, the number of academic papers in various fields has increased drastically. Accordingly, the demand for techniques for predicting papers that will become influential in the future is growing to help readers identify those papers and support efficient knowledge acquisition. In this study, we adopt the citation count as a measure of future impact, following several previous studies (e.g., Chubin and Garfield, 1979; Aksnes, 2006), and we address the citation count prediction task, which entails predicting how many times a target paper will be cited in the future.

There have been many studies on citation count prediction (e.g., Fu and Aliferis, 2008; van Dongen et al., 2020). However, none of those settings is strictly applicable to predicting the citation count of newly published papers, because they assume

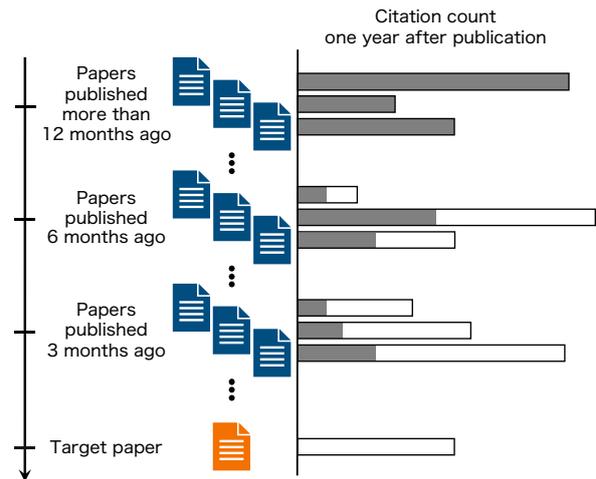


Figure 1: Comparison of a realistic citation count prediction setting with existing research settings. Each bar (■+□) represents the citation count one year after publication, which existing studies assume to be available, while the gray part (■) represents the citation count that is actually available at the time of a target paper’s publication.

the availability of future citation counts for papers shortly after publication. For example, consider the case of predicting the citation counts one year after publication. For training and testing, the correct citation count of the target paper one year after publication must be known; hence, only papers published more than one year ago are used in the experiments. Consequently, even for papers published less than one year before the target paper, the number of citations one year after publication is available, and these citation counts are commonly used to train the prediction model. The bars (■+□) in Figure 1 represent the citation count information used in such settings. However, in actually predicting the future citation count of newly published papers, the correct citation counts one year after the publication of papers published less than one year ago are not available; what is actually available is the gray part of each bar (■) in the figure.

The unrealistic assumption in previous studies might appear to have a limited impact on the performance of a prediction model. However, information on the future citation counts of recently published papers could cause leakage of research trends in the near future, which turns out to have a non-negligible impact on performance. Hence, in this study, we first show that the settings of existing studies leak future information that contributes significantly to the prediction performance. We then introduce a realistic citation count prediction task that strictly uses information available at the time of a target paper’s publication.

Furthermore, we propose two methods to capture research trends in the near future that are applicable even in our realistic setting. The first method is **citation count complementation**, which uses papers published less than one year ago as training data by estimating the citation count one year after publication from the current citation count. The second method leverages the **degree of early adoption** by using the property that papers that cite more recent papers and papers that cite more frequently cited papers tend to receive more attention in the future.

2 Datasets

For the experiments here, we used two datasets: a CL dataset, consisting of papers in the field of computational linguistics, and a Bio dataset, consisting of papers in the field of biology.

To construct the **CL dataset**, we collected 16,940 papers submitted to arXiv in the Computation and Language (cs.CL) category¹ from June 2014 to June 2020. We considered preprints suitable for this study because they include papers that have not been peer-reviewed and are expected to have a large variance in their future impact. We then obtained the publication dates of papers that cited the collected papers from Semantic Scholar² to calculate the citation count for each elapsed month after the publication of each paper in the dataset.

We created 13 subsets, each of which consists of papers published in one of the months from June 2019 to June 2020 and papers published in the five years prior to that month. Within each subset, the papers published in the latest month were used for evaluation, and the remainder was used for training. For example, one subset consists

of papers published from May 2015 to May 2020, of which papers published in May 2020 were used for evaluation and the remainder for training. The subsets created in this way have the same properties as cross-validation, where there is overlap in the papers for training, but the papers for evaluation are completely different. The average numbers of papers per subset for training and evaluation are 13,227 and 500.2, respectively. In the experiments, we used the subset that used papers published in June 2019 for evaluation as the development set and the remaining 12 subsets to train and evaluate the model.

To construct the **Bio dataset**, we collected 7,535 papers submitted to the Biochemistry and Plant Biology, Pharmacology and Toxicology areas of bioRxiv³ from May 2015 to April 2021. As with the CL dataset, we created 12 subsets with papers published in each month from May 2020 to April 2021 as the papers for evaluation. The average numbers of papers per subset for training and evaluation were 5,913 and 257, respectively.⁴

3 Task Formulation

3.1 Leakage in Existing Settings

Most previous studies on citation count prediction adopted the citation count n years after publication as the target citation count for prediction (e.g., Fu and Aliferis, 2008; van Dongen et al., 2020). Those studies used datasets consisting of papers published in a specific time period. Specifically, they used a set of newly published papers by year or a set of randomly selected papers as the evaluation set, and the rest as the training set. The citation count prediction model was then trained using the citation counts n years after the publication of each paper in the training set, and the prediction performance was evaluated by predicting the citation counts of the papers in the evaluation set.

In reality, the citation counts n years after publication are available only for papers published more than n years after publication, but existing settings use those citation counts even for papers published less than n years after publication (Fu and Aliferis, 2008; Davletov et al., 2014; Singh et al., 2015; Abrishami and Aliakbary, 2019; van Dongen et al., 2020). The use of future citation counts that are not actually available in the existing settings may lead

¹<https://arxiv.org/list/cs.CL/recent>

²<https://www.semanticscholar.org/>

³<https://www.biorxiv.org/>

⁴Statistics for each subset of the two datasets are provided in Appendix A.

to leakage of future research trends. Accordingly, we conducted a preliminary experiment to examine the effect of this leakage. We found that, with the same number of papers used for training, the use of future citation counts of newly published papers, which are not actually available, achieves higher performance than the use of papers published more than n years ago.⁵ Hence, we introduce a realistic citation count prediction task that prevents such leakage and is applicable to the prediction of citation counts for newly published papers.

3.2 Realistic Citation Count Prediction

Our realistic citation count prediction task restricts the citation count information used for training to information that is strictly available as of the publication of the target papers for evaluation. Specifically, in the case of predicting the citation count n years after publication, the citation count n years after publication is used for training with papers that were published more than n years after publication. On the other hand, for papers published less than n years after publication, the citation counts as of the publication of the target papers are used for training.

3.3 Target Citation Counts for Prediction

In this study, to determine an appropriate value of n for predicting citation counts, we first investigated the datasets described in Section 2. Specifically, we assumed that the citation counts five years after publication are stable, and we extracted papers published more than five years after publication from each dataset. We then calculated Spearman’s rank correlation between the citation counts m months after publication and five years after publication.⁶ As a result, we found that Spearman’s rank correlation between the citation count one year after publication against the count five years after publication was 0.86 for the CL dataset and 0.71 for the Bio dataset. This indicates that the citation count one year after publication is a good indicator of a paper’s final citation count. Hence, we adopt the citation count one year after publication as the target citation count for prediction.

4 Citation Count Complementation

We propose a method to estimate the citation count one year after the publication of papers that were

published less than one year ago. Our method uses the citation counts of those papers at the time the target paper was published to estimate the counts one year after they were published. Specifically, we estimate the citation counts of a paper m months after publication with a citation count c_m by the following two methods:

Case-based Extract all papers in the training set that have a citation counts c_m at m months after publication, and use the median of those papers’ counts one year after publication as the estimate.

Ratio-based For the training set, calculate the ratio of the average citation count m months after publication to the average count one year after publication, and multiply it by the citation count c_m to obtain the estimate.

While case-based estimation is expected to be accurate for less-cited papers, where there are many other papers with the same citation count, it is not suitable for highly-cited papers that have no or few other papers with the same citation count. Thus, if the citation count c_m is associated with a paper in the list of top 10% papers, it is estimated using the ratio-based method. Otherwise, it is estimated using the case-based method. The rank order of c_m is calculated from the distribution of citation counts m months after publication for the papers in the training set.

To confirm the appropriateness of this citation count complementation, we calculated Spearman’s rank correlation between the correct citation counts one year after publication against the predicted citation count before and after complementation (c_m and complemented citation count). For this investigation, we used the training portion of the 12 subsets to train and evaluate the model on the CL dataset, and we compared the average Spearman’s rank correlations for each subset. As a result, we found that the correlation improved from 0.88 to 0.92, which demonstrates that the citation count complementation is appropriate.

5 Degree of Early Adoption

In realistic citation count prediction, the full citation counts of papers published less than one year after publication cannot be used for training, yet papers that are frequently cited in such a short term are likely to be impactful. In addition, papers that

⁵Details of the experiment are provided in Appendix B.

⁶Detailed results are provided in Appendix C.

	Top 0–1%	1–2.5%	2.5–5%	5–10%	10–25%	25–100%	No citation
Within 3 months	15.5 (4.6%)	14.3 (3.8%)	10.6 (3.6%)	8.8 (5.0%)	7.5 (6.6%)	6.5 (4.9%)	5.0 (71.5%)
Within 6 months	14.3 (9.6%)	12.6 (7.3%)	9.8 (6.2%)	7.6 (8.8%)	6.3 (10.7%)	4.6 (9.4%)	3.7 (48.1%)
Within 9 months	13.8 (15.4%)	11.2 (10.3%)	7.7 (7.8%)	6.6 (10.1%)	5.3 (12.9%)	3.5 (12.4%)	2.5 (31.0%)
Within 12 months	12.7 (21.6%)	10.1 (12.0%)	6.4 (9.1%)	5.7 (10.5%)	4.4 (13.5%)	2.5 (12.9%)	2.0 (20.6%)

Table 1: Average citation counts one year after publication for papers citing at least one paper with the top $k_1\%$ to $k_2\%$ citation counts published within m months in the CL dataset. “No citation” indicates papers that did not cite any paper published within m months. The numbers in parentheses give the ratio of papers belonging to each group in each column.

cite such frequently cited papers earlier—i.e., papers with a high degree of early adoption—can be considered as adequately recognizing the latest trends and are likely to receive more attention in the future because of their novelty and technical contributions. To validate this hypothesis, we investigated whether papers that cite frequently cited papers at an early date tend to be cited more in the future.

Specifically, we examined the average citation count one year after publication for those papers that cite at least one paper with the top $k_1\%$ to $k_2\%$ citation counts published within m months. For this investigation, we used 15,962 papers published in arXiv’s cs.CL category between June 2015 and May 2020, which form the training portion of the subset described in Section 2. In the case of multiple citations of papers published within m months, we used the highest rank order of the citation counts among them. For (k_1, k_2) , we used 6 pairs: (0, 1), (1, 2.5), (2.5, 5), (5, 10), (10, 25), and (25, 100). For m , we used four values: 3, 6, 9, and 12. We then calculated the average citation count for each combination of (k_1, k_2) and m .

Table 1 lists the results. In the table, “no citation” indicates papers that did not cite any paper published within m months. We confirmed an overall trend that papers citing more recent papers and papers citing more frequently-cited papers have higher average citation counts. The average citation count of papers that cited papers in the top 1% of citations within 3 months of publication was 15.5, which was about 2.4 times higher than the average citation count of 6.5 for all papers. On the basis of these results, we attempted to leverage the degree of early adoption in citation count prediction, and we describe the specific methods for this in Section 6.1.

6 Experiments

We conducted experiments on the datasets described in Section 2 to validate the effectiveness of using citation count complementation and the degree of early adoption in realistic citation count prediction.

6.1 Setup

Task Following Maillette de Buy Wenniger et al. (2020), we defined the citation score as $\log(c_n + 1)$, where c_n is the citation count n years after a paper’s publication. In this study, we sought to predict the citation score one year after the publication by using the target paper’s title and abstract.

Prediction Model We adopted a model based on BERT (Devlin et al., 2019) to predict the citation scores. We treated the paper’s title as the first sentence of the input and the abstract as the second sentence. For the output of BERT, we used the vector representation of a special token [CLS]. The [CLS] vector was then passed through a fully connected layer and linearly transformed to obtain a prediction of the citation score. During training, we applied dropout (Srivastava et al., 2014) to the [CLS] vector and minimized the mean squared error (MSE) between the predicted and actual citation scores.

We also represented the degree of early adoption via a special token sequence, which was inserted at the beginning of the input sentence to BERT. Specifically, we created seven special tokens: “top 0–1%,” “top 1–2.5%,” “top 2.5–5%,” “top 5–10%,” “top 10–25%,” “top 25–100%,” and “no citation.” This enabled us to represent the degree of early adoption by arranging the four special tokens corresponding to the highest-ranking citation counts of the papers cited by the target paper within 3, 6, 9, and 12 months, respectively. For example, if a paper cited no paper published within 3 months, a paper published within 6 to 9 months with a top 5–

10% citation count, and a paper published within 12 months with a top 0–1% citation count, the special token sequence would be “[no citation][top 5–10%][top 5–10%][top 0–1%].”

Experimental Setting We used two BERT-based pre-trained language models (PLMs): BERT⁷ pre-trained on a general-domain corpus such as Wikipedia, and SciBERT⁸ pre-trained on a scientific-domain corpus built from a large number of papers. All models were trained with 3 epochs, a batch size of 32, the AdamW optimizer (Loshchilov and Hutter, 2019), and a learning-rate schedule with warm-up at 10% of the total training steps and linear decays in the remaining steps. Following Devlin et al. (2019), the learning rate was set to 2e-5, which achieved the highest Spearman’s rank correlation for all models on the development set, after searches conducted at rates of 2e-5, 3e-5, and 5e-5. We experimented with three different random seeds for each model and calculated the mean and standard deviation of the evaluation scores.⁹

Compared Methods We compared the following five methods to validate the effectiveness of using citation count complementation and leveraging the degree of early adoption.

- **Baseline:** A method that used only papers more than one year after publication for training.
- **+CCC:** A method that used all papers in the training set, including those published less than one year after publication, with Citation Count Complementation.
- **+CCC*:** A method that used the same number of papers as the Baseline model, in order from the newest in the training set, with Citation Count Complementation.
- **+DEA:** A method that was based on the Baseline model but used the Degree of Early Adoption.
- **+CCC+DEA:** A method that used all papers in the training set with Citation Count Complementation and the Degree of Early Adoption.

We also considered applying the proposed method to the existing citation count prediction models based on deep learning such as NNCP (Abrihami and Aliakbary, 2019), BIL_A (Ma et al.,

2021), and SChuBERT (van Dongen et al., 2020), but discarded the idea for the following reasons. First, NNCP and BIL_A were designed under the assumption that citation counts several years after a target paper’s publication are available, and thus these models were not applicable to our setting. SChuBERT was excluded from the experiments because preliminary experiments showed that its performance was equal to or lower than the Baseline, even though it is a model that predicts citation counts using the entire body of a paper. The low performance of SChuBERT is probably due to the fact that it does not perform fine-tuning since it would be computationally expensive to perform fine-tuning for SChuBERT.

Evaluation We evaluated the models with three metrics: Spearman’s rank correlation (ρ) to assess the overall ranking quality, the mean squared error (MSE) to assess the amount of error, and a metric defined as the percentage of the actual top n% of papers in the top k% of the output (n%@k%) to intuitively understand the results.

As mentioned in Section 2, because the average number of papers for evaluation in each subset of the datasets was not large, the evaluation scores would not have been stable if each subset were evaluated individually. Therefore, to yield stable results, we computed each metric across all subsets of the papers. That is, while each subset was used to train the prediction model and the citation counts of the papers for evaluation were predicted by using the model for each subset, the evaluation scores were calculated by combining the predictions for all 12 subsets.

6.2 Experimental Results

Table 2 summarizes the experimental results. For both the CL and Bio datasets, the models based on BERT and SciBERT improved the citation count prediction performance by leveraging either the citation count complementation or the degree of early adoption. The performance was further improved by using both. The SciBERT-based model outperformed the BERT-based model, which demonstrated the effectiveness of pre-training on a scientific-domain corpus for citation count prediction.¹⁰

By comparing the Baseline and +CCC* models,

⁷<https://huggingface.co/bert-base-uncased>

⁸https://huggingface.co/allenai/scibert_scivocab_uncased

⁹Training took about 10 minutes per epoch and inference took a few seconds per evaluation set on a single GV100 GPU.

¹⁰We also experimented with domain-specific models such as PubMedBERT (Gu et al., 2021) on the Bio dataset, but we could not confirm further performance improvement.

Dataset	PLM	Method	ρ	MSE	5%@5%	5%@25%	10%@10%	10%@50%
CL	BERT	Baseline	36.6 \pm 0.4	1.504 \pm 0.022	21.1 \pm 1.7	63.7 \pm 2.3	28.1 \pm 0.9	83.2 \pm 0.4
		+CCC	39.1 \pm 0.2	1.275 \pm 0.018	28.8 \pm 1.9	72.6 \pm 1.0	34.5 \pm 0.4	84.6 \pm 0.6
		+CCC*	39.6 \pm 0.1	1.176 \pm 0.041	28.2 \pm 1.5	73.4 \pm 1.0	34.4 \pm 0.7	84.9 \pm 1.1
		+DEA	40.4 \pm 0.5	1.394 \pm 0.019	22.4 \pm 0.6	69.4 \pm 0.9	31.1 \pm 0.8	86.7 \pm 0.7
		+CCC+DEA	41.8 \pm 0.3	1.173 \pm 0.008	28.6 \pm 2.1	75.3 \pm 2.2	35.7 \pm 1.1	87.0 \pm 1.0
	SciBERT	Baseline	38.3 \pm 0.3	1.390 \pm 0.042	27.4 \pm 1.2	67.5 \pm 1.5	32.0 \pm 1.0	84.7 \pm 0.7
		+CCC	40.1 \pm 0.5	1.147 \pm 0.010	33.2 \pm 1.7	72.8 \pm 0.6	37.7 \pm 0.4	86.2 \pm 0.2
		+CCC*	40.9 \pm 0.1	1.063 \pm 0.013	33.1 \pm 0.9	75.5 \pm 0.9	37.8 \pm 0.9	86.0 \pm 0.4
		+DEA	41.1 \pm 0.4	1.307 \pm 0.015	28.0 \pm 1.4	70.3 \pm 0.4	33.8 \pm 1.2	86.2 \pm 0.2
		+CCC+DEA	42.8 \pm 0.1	1.104 \pm 0.012	34.2 \pm 0.5	76.0 \pm 1.1	36.7 \pm 1.1	87.9 \pm 0.2
Bio	BERT	Baseline	24.1 \pm 2.0	0.593 \pm 0.012	20.1 \pm 1.1	41.3 \pm 4.6	26.8 \pm 2.4	67.5 \pm 3.4
		+CCC	36.4 \pm 1.1	0.487 \pm 0.010	50.4 \pm 1.0	83.1 \pm 0.0	48.4 \pm 0.9	86.7 \pm 0.9
		+CCC*	32.9 \pm 0.9	0.499 \pm 0.011	49.8 \pm 1.0	80.5 \pm 1.3	47.1 \pm 0.6	84.2 \pm 2.1
		+DEA	32.7 \pm 3.0	0.559 \pm 0.018	21.9 \pm 0.4	47.4 \pm 4.7	29.9 \pm 1.6	77.1 \pm 6.7
		+CCC+DEA	40.6 \pm 0.6	0.461 \pm 0.005	50.0 \pm 0.0	87.7 \pm 0.6	49.6 \pm 0.7	89.9 \pm 1.5
	SciBERT	Baseline	30.3 \pm 1.0	0.588 \pm 0.011	21.0 \pm 2.6	51.7 \pm 2.6	29.9 \pm 0.9	73.2 \pm 2.6
		+CCC	40.5 \pm 0.3	0.446 \pm 0.007	54.3 \pm 0.4	86.8 \pm 0.7	52.5 \pm 0.7	89.4 \pm 0.7
		+CCC*	37.2 \pm 0.7	0.472 \pm 0.006	53.7 \pm 0.4	84.4 \pm 1.3	48.4 \pm 0.3	88.3 \pm 1.7
		+DEA	37.0 \pm 2.3	0.555 \pm 0.018	25.1 \pm 1.5	57.8 \pm 6.2	33.7 \pm 1.8	79.4 \pm 5.3
		+CCC+DEA	42.5 \pm 1.2	0.436 \pm 0.010	52.4 \pm 0.4	90.3 \pm 2.6	52.6 \pm 0.6	91.8 \pm 1.8

Table 2: Experimental results from comparing methods that use papers published less than one year after publication in realistic citation count prediction. Each score besides the MSE is multiplied by 100.

BERT for Coreference Resolution: Baselines and Analysis	
Abstract: We apply BERT to coreference resolution, achieving strong improvements on the OntoNotes (+3.9 F1) and GAP (+11.5 F1) benchmarks. A qualitative analysis of model predictions indicates that, compared to ELMo and BERT-base, BERT-large is particularly better at distinguishing between related but distinct entities (e.g., President and CEO). However, there is still room for improvement in modeling document-level context, conversations, and mention paraphrasing. Our code and models are publicly available.	Ground truth: top 0.9%
	Baseline: top 14.5%
	+CCC: top 2.8%
	+DEA: top 7.1%
	+CCC+DEA: top 0.8%

Figure 2: Example of a paper for which the citation count complementation and degree of early adoption improved the prediction. The left part shows the papers title and abstract (Joshi et al., 2019), and the right part shows the relative position of the citation count one year after publication of the target paper (ground truth) and the relative positions predicted by SciBERT-based models.

which used the same number of papers for training, we can see that the +CCC* model performed better on both datasets; thus, we confirmed the effectiveness of using papers published less than one year after publication with citation count complementation for training. We had predicted that the +CCC model, which used a larger number of papers for training, would perform better than the +CCC* model. This was true for the Bio dataset, but surprisingly for the CL dataset, the +CCC* model performed better. We speculate that older papers could serve as noise if the number of papers is sufficiently large, but we leave further investigation of this point to a future work. From the result for the Baseline, +CCC, and +CCC* models on the Bio dataset, we confirmed performance improvement due to the increased number of papers for training

and the leverage of newer papers. In particular, the performance gains from using new papers for training were considerable.

As for the actual predictive performance, the SciBERT-based model using both citation count complementation and the degree of early adoption achieved a score of 90.3 for the 5%@25% metric on the Bio dataset. This means that if we read only the top 25% of the papers predicted by the model for a given set of papers, we could cover 90.3% of the papers expected to have future citation counts within the top 5%. Hence, we believe that this method is highly useful from a practical viewpoint.

Figure 2 shows an example of a paper for which the citation count complementation and degree of early adoption improved the prediction. Although the citation count one year after the paper’s publi-

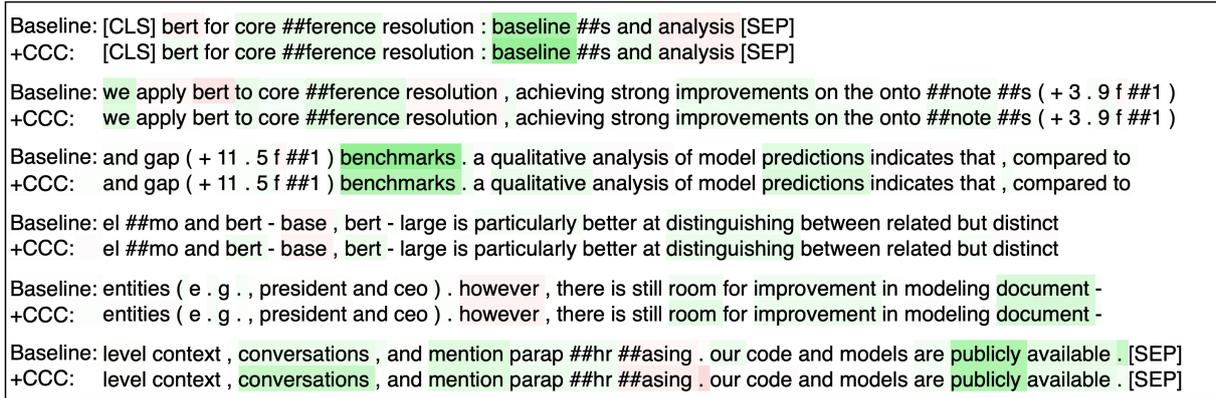


Figure 3: Visualization of the contribution of each token in predicting the citation count of the paper shown in Figure 2. Darker green represents a higher contribution to the prediction, while darker red represents a lower contribution.

citation was in the top 0.9% in the evaluation set, the Baseline model underestimate the citation count. This is likely because the paper was published 10 months after the original paper on BERT, and the Baseline model thus could not leverage the “latest” information that BERT was going to get enormous attention. The prediction was improved by applying either of the two proposed methods, and it was quite accurate when both methods were applied. The use of papers published less than one year after publication for training by citation count complementation would enable the model to use information about BERT for prediction. In addition, this paper cited the top 5% to 10% of papers within 3 months of publication and the top 0% to 1% of papers within 6 months of publication, which indicates that it captured the latest trends. We believe that the proposed method successfully incorporated these properties of the paper into citation count prediction by leveraging the degree of early adoption.

6.3 Analysis and Discussion

To investigate what words the model came to emphasize by leveraging papers shortly after publication for training, we performed an analysis using Integrated Gradients (Sundararajan et al., 2017). The Integrated Gradients method computes each input feature’s contribution to a deep network’s prediction by integrating gradients; thus, it enables analysis of each input token’s contribution to a prediction by BERT. Similar to Schwarzenberg et al. (2021) and Bharadwaj and Shevade (2022), we used a sequence of [PAD] tokens as the baseline input for Integrated Gradients to estimate the contribution of each token.

Figure 3 shows a visualization of the contribu-

tion of each token in predicting the citation count of the example paper shown in Figure 2, for the Baseline model, which does not use papers published after the BERT paper for training, and the +CCC model, which uses papers published after the BERT paper for training. The darker green represents a higher contribution to the prediction, while the darker red represents a lower contribution. The figure shows that the Baseline model did not know about BERT, and the token *bert* had a negative impact, whereas the +CCC model knew that BERT was a state-of-the-art model, and the token had a positive impact. We also observed that both models emphasized tokens that are intuitively important, such as the higher contribution of *publicly available*, which is thought to facilitate subsequent research and growth in citation counts when codes and models are made publicly available.

Furthermore, we quantitatively analyzed the tokens whose contribution to the prediction was increased by using papers shortly after publication for training. To extract these tokens, we calculated each token’s contribution in the +CCC model and its contribution in the Baseline model for the same paper. Then, we took the difference to obtain the score increase due to the use of papers published less than one year after publication. We computed this increase by using all the papers for evaluation in each of the two datasets, took the average for each token, and extracted the top 10 tokens for that average. If a word was divided into subwords, its contribution was determined by summing the subwords’ contributions. In addition, stop words, tokens containing symbols, and tokens with a document frequency of less than 10 were deleted.

Table 3 lists the extracted words. In the CL

Rank	CL	Bio
1	trec	coronavirus
2	coronavirus	coronaviruses
3	revisiting	sars
4	semeval	cov
5	rethinking	computationally
6	finnish	tumors
7	wmt	nucleocapsid
8	bert	hydroxychloroquine
9	propaganda	cannabis
10	specaugment	pandemic

Table 3: Tokens that the model came to emphasize by using papers shortly after publication for training by citation count complementation.

dataset, the conference names *trec*, *semeval*, and *wmt* were at the top of the list. This could mean that more and more papers have evaluated models on datasets that were published at those conferences in recent years. Other words such as *revisiting* and *rethinking* may be associated with an increase in the number of papers that have revised existing models and methods in recent years. In fact, the number of papers published at ACL that included these words in their titles increased from three (0.15%) in 2013-2018 to 15 (0.53%) in 2019-2022. The model also increasingly focused on technologies that have gained attention in recent years, such as *bert* and *specaugment*. In particular, SpecAugment (Park et al., 2019) is a high-profile technology in the speech-processing field that has been cited more than 2,000 times since it was published in April 2019, and the model was able to capture it here as an important technology.

As for the Bio dataset, a number of COVID-19-related words appeared at the top of the list. This indicates that the model captured the increasing number of relevant papers and increasing overall citation counts due to the COVID-19 pandemic. Also, we attribute the large performance improvement with citation count completion on the Bio dataset to the capability to focus more on COVID-19-related words.

7 Related Work

Early works on citation count prediction formulated the task and explored effective features. Castillo et al. (2007) formulated citation count pre-

diction as a regression problem and used author reputation to predict the citation count. Fu and Aliferis (2008) formulated citation count prediction as a classification problem and investigated several features that are effective for such prediction, including a paper’s title, abstract, and author information.

Other studies have sought to improve the prediction performance by using various features. One such feature is a citation graph constructed from citation relationships among papers. Davletov et al. (2014) proposed a method to use the graph’s temporal and topological features. Pobiedina and Ichise (2015) achieved high prediction performance by mining frequent graph patterns. Singh et al. (2015) proposed a method to use the citation context, which is the text in a paper that mentions other cited papers. Bhat et al. (2015) found that the interdisciplinarity of authors is effective in predicting citation counts. Li et al. (2019) proposed a method to use peer-reviewed text from multiple aspects.

Several studies have focused on aspects other than features. Chakraborty et al. (2014) and te Li et al. (2015) found several patterns in the growth of citation counts by analyzing a large number of papers, and they proposed a two-step prediction method, first classifying papers into each pattern and then predicting counts for each pattern. Xiao et al. (2016) proposed a method to predict the citation count at an arbitrary point in time from the publication of a paper, with the aim of predicting its future potential impact.

In recent years, there has been research on the use of deep learning techniques to predict citation counts. Abrishami and Aliakbary (2019) proposed an RNN-based method to predict a paper’s future citation count by using the citation counts for each elapsed year since its publication. van Dongen et al. (2020) proposed a method to predict the citation count by dividing a paper’s text into chunks and encoding the paper’s entire body with BERT. Ma et al. (2021) proposed a method to predict the citation count by extracting semantic features from a paper’s title and abstract via Doc2Vec and Bi-LSTM with an attention mechanism.

8 Conclusion

In this paper, we introduced a realistic citation count prediction task that is applicable to newly published papers, by using only citation count information that is strictly available at the time of pub-

lication of a target paper for training. We further proposed two methods to use papers published less than one year after publication for citation count prediction, as these papers cannot be directly used for training because their citation counts one year after publication are unknown. The first method is citation count complementation, which uses recent papers for training by estimating their citation counts one year after publication. The second method is to leverage the degree of early adoption, which incorporates the tendency for papers that cite highly cited papers earlier to have higher average citation counts. Through experiments using papers collected from arXiv and bioRxiv, we demonstrated that the use of papers published less than one year after publication improves the performance of realistic citation count prediction. For future work, we intend to build models that incorporate information from papers that was not used in this study, such as the body, figures, tables, and author information.

Limitations

Both methods proposed in this paper focus on fields in which technology is rapidly evolving and the latest research results are increasingly important. Because of this, these methods' effectiveness could be limited in fields for which the latest research results are not particularly important. Also, the model in this study only uses the titles and abstracts of papers as inputs, and it does not leverage the body, figures, or tables.

Acknowledgements

This work was partly supported by JST Moonshot R&D (Grant Number JPMJMS2033).

References

- Ali Abrishami and Sadegh Aliakbary. 2019. [Predicting citation counts based on deep neural network learning techniques](#). *Journal of Informetrics*, 13(2):485–499.
- Dagfinn W. Aksnes. 2006. [Citation rates and perceptions of scientific contribution](#). *J. Assoc. Inf. Sci. Technol.*, 57:169–185.
- Shikhar Bharadwaj and Shirish Shevade. 2022. [Efficient constituency tree based encoding for natural language to bash translation](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2022)*, pages 3159–3168.

- Harish S. Bhat, Li-Hsuan Huang, Sebastian Rodriguez, Rick Dale, and Evan Heit. 2015. [Citation prediction using diverse features](#). *2015 IEEE International Conference on Data Mining Workshop (ICDMW 2015)*, pages 589–596.
- Carlos Castillo, Debora Donato, and Aristides Gionis. 2007. [Estimating number of citations using author reputation](#). In *String Processing and Information Retrieval (SPIRE 2007)*, pages 107–117.
- Tanmoy Chakraborty, Suhansanu Kumar, Pawan Goyal, Niloy Ganguly, and Animesh Mukherjee. 2014. [Towards a stratified learning approach to predict future citation counts](#). In *IEEE/ACM Joint Conference on Digital Libraries*, pages 351–360.
- Daryl E. Chubin and Eugene Garfield. 1979. [Is citation analysis a legitimate evaluation tool?](#) *Scientometrics*, 2:91–94.
- Feruz Davletov, Ali Selman Aydin, and Ali Cakmak. 2014. [High impact academic paper prediction using temporal and topological features](#). *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM 2014)*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (NAACL 2019)*, pages 4171–4186.
- Lawrence D. Fu and Constantin F. Aliferis. 2008. [Models for predicting and explaining citation count of biomedical articles](#). *AMIA ... Annual Symposium proceedings. AMIA Symposium vol. 2008 (AMIA 2008)*, pages 222–226.
- Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. [Domain-specific language model pretraining for biomedical natural language processing](#). *ACM Trans. Comput. Healthcare*, 3(1).
- Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. [BERT for coreference resolution: Baselines and analysis](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 5803–5808.
- Siqing Li, Wayne Xin Zhao, Eddy Jing Yin, and Ji-Rong Wen. 2019. [A neural citation count prediction model based on peer review text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 4914–4924.

Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations (ICLR 2019)*.

Anqi Ma, Yu Liu, Xiujuan Xu, and Tao Dong. 2021. [A deep-learning based citation count prediction model with paper metadata semantic features](#). *Scientometrics*, 126:6803–6823.

Gideon Maillette de Buy Wenniger, Thomas van Dongen, Eleri Aedmaa, Herbert Teun Kruitbosch, Edwin A. Valentijn, and Lambert Schomaker. 2020. [Structure-tags improve text classification for scholarly document quality prediction](#). In *Proceedings of the First Workshop on Scholarly Document Processing (SDP 2020)*, pages 158–167.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. [SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition](#). In *Proceedings of Interspeech 2019*, pages 2613–2617.

Nataliia Pobiedina and Ryutaro Ichise. 2015. [Citation count prediction as a link prediction problem](#). *Applied Intelligence*, 44:252–268.

Robert Schwarzenberg, Nils Feldhus, and Sebastian Möller. 2021. [Efficient explanations from empirical explainers](#). In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP 2021)*, pages 240–249.

Mayank Singh, Vikas Patidar, Suhansanu Kumar, Tanmoy Chakraborty, Animesh Mukherjee, and Pawan Goyal. 2015. [The role of citation context in predicting long-term citation profiles: An experimental study based on a massive bibliographic text dataset](#). *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM 2015)*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: A simple way to prevent neural networks from overfitting](#). *Journal of Machine Learning Research (JMLR 2014)*, 15(56):1929–1958.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. [Axiomatic attribution for deep networks](#). In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML 2017)*, page 3319–3328.

Cheng te Li, Yu-Jen Lin, Rui Yan, and Mi-Yen Yeh. 2015. [Trend-based citation count prediction for research articles](#). In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2015)*.

Thomas van Dongen, Gideon Maillette de Buy Wenniger, and Lambert Schomaker. 2020. [SCHuBERT: Scholarly document chunks with BERT-encoding boost citation count prediction](#). In *Proceedings of the First Workshop on Scholarly Document Processing (SDP 2020)*, pages 148–157.

Shuai Xiao, Junchi Yan, Changsheng Li, Bo Jin, Xiangfeng Wang, Xiaokang Yang, Stephen M. Chu, and Hongyuan Zha. 2016. [On modeling and predicting individual paper citation count over time](#). In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*.

A Detailed Dataset Statistics

Table 4 lists the numbers of papers for training and evaluation for each subset in the CL and Bio datasets described in Section 2.

Dataset	Subset	Training	Evaluation
CL	6/2019	10,459	620
	7/2019	11,026	404
	8/2019	11,404	479
	9/2019	11,854	720
	10/2019	12,529	550
	11/2019	13,031	564
	12/2019	13,552	345
	1/2020	13,820	260
	2/2020	14,049	326
	3/2020	14,339	334
	4/2020	14,617	747
	5/2020	15,305	713
Bio	6/2020	15,962	440
	5/2020	4,451	292
	6/2020	4,743	303
	7/2020	5,046	286
	8/2020	5,331	268
	9/2020	5,597	233
	10/2020	5,827	261
	11/2020	6,088	221
	12/2020	6,307	219
	1/2021	6,524	245
	2/2021	6,769	246
	3/2021	7,012	258
4/2021	7,264	252	

Table 4: Numbers of papers for training and evaluation for each subset in the CL and Bio datasets. The subset names correspond to the year and month of publication of the papers that a subset used for evaluation.

B Details of Leakage Investigation in Existing Settings

To investigate the impact of leakage in the existing setting on the performance of citation count prediction, we conducted an experiment using the CL dataset described in Section 2. The experiment basically used the Baseline model described

PLM	Setting	Avg. train size	ρ	MSE	5%@5%	5%@25%	10%@10%	10%@50%
BERT	w/ future citation count	13,277	40.5 \pm 0.3	1.373 \pm 0.010	28.7 \pm 2.0	72.8 \pm 1.0	34.6 \pm 0.1	87.1 \pm 0.3
	w/ future citation count	8,571	39.0 \pm 0.2	1.358 \pm 0.030	26.0 \pm 0.2	73.5 \pm 1.0	33.7 \pm 1.2	85.1 \pm 0.4
	w/o future citation count	8,571	36.6 \pm 0.4	1.504 \pm 0.022	21.1 \pm 1.7	63.7 \pm 2.3	28.1 \pm 0.9	83.2 \pm 0.4
SciBERT	w/ future citation count	13,277	41.8 \pm 0.3	1.220 \pm 0.024	31.1 \pm 1.1	73.5 \pm 1.5	37.1 \pm 0.8	87.9 \pm 0.4
	w/ future citation count	8,571	40.4 \pm 0.9	1.232 \pm 0.019	31.3 \pm 2.1	72.6 \pm 1.0	35.8 \pm 0.7	86.2 \pm 1.1
	w/o future citation count	8,571	38.3 \pm 0.3	1.390 \pm 0.042	27.4 \pm 1.2	67.5 \pm 1.5	32.0 \pm 1.0	84.7 \pm 0.7

Table 5: Experimental results of the investigation of the leaks in the existing setting (w/ future citation count). Each score besides the MSE is multiplied by 100.

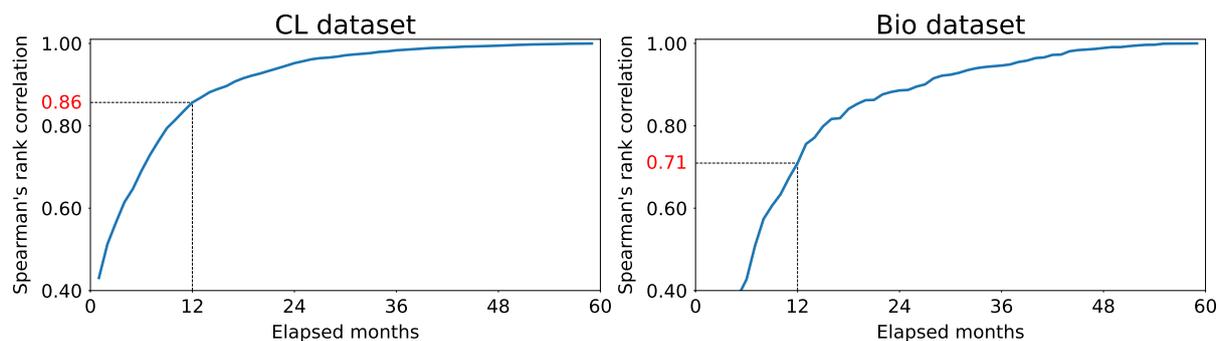


Figure 4: Spearman’s rank correlation between the citation counts m months after publication against the citation count five years after publication. The left part shows the results on the CL dataset and the right part shows the results on the Bio dataset.

in Section 6.1, and only the papers for training were changed. We compared settings that used future citation counts with those that do not. In the setting that did not use future citation counts (*w/o future citation count*), only papers published more than one year after publication as of the target paper’s publication were used for training. For example, if the subset that used papers published in June 2020 for evaluation, papers published between July 2019 and May 2020 were excluded from the training set, and only papers published between June 2015 and June 2019 were used for training. This reduced the average number of papers for training from 13,227 to 8,571. In the setting that used future citation counts (*w/ future citation count*), we used the citation counts one year after publication for all papers in the training set, including papers published less than one year after publication as of the target paper’s publication.

In the *w/ future citation count* setting, the number of papers that can be used for training was larger than in the *w/o future citation count* setting, and thus the impact of the leakage could not be fairly investigated. For a fair comparison, we also experimented with settings that align the number of papers for training used in the *w/ future citation count* setting with the *w/o future citation count* set-

ting. The number of papers for training was aligned by grouping the papers for training by year and month of publication and randomly reducing the papers in each group by the same ratio. By aligning the number of papers, we could fairly compare *w/* and *w/o future citation count* settings.

Table 5 shows the experimental results. For all metrics, the *w/ future citation count* setting, which was trained using all citation count that was actually unavailable, outperforms the *w/o future citation count* setting, which was trained using only available information. The results show that the existing setting improperly improves the performance of the prediction model. In particular, even when the number of papers for training was aligned, the *w/ future citation count* setting outperformed the *w/o future citation count* setting. This demonstrates that the future citation count of papers published close to the target causes leakage of research trends that grow in citation count in the future.

C Transition of Spearman’s Rank Correlation

Figure 4 shows Spearman’s rank correlation between the citation counts m months after publication and five years after publication in the CL and Bio datasets.

“Why do I feel offended?” Korean Dataset for Offensive Language Identification

San-Hee Park^{1*} Kang-Min Kim^{2*} O-Joun Lee² Youjin Kang¹
Jaewon Lee³ Su-Min Lee² SangKeun Lee¹

¹ Korea University, Seoul, Republic of Korea

² The Catholic University of Korea, Bucheon, Republic of Korea

³ Seoul National University, Seoul, Republic of Korea

carpediem20@korea.ac.kr, {kangmin89, ojlee}@catholic.ac.kr yjkang10@korea.ac.kr
enotchi@snu.ac.kr, sumini0516@catholic.ac.kr, yalphy@korea.ac.kr

Abstract

Warning: This paper contains some offensive expressions.

Offensive content is an unavoidable issue on social media. Most existing offensive language identification methods rely on the compilation of labeled datasets. However, existing methods rarely consider low-resource languages that have relatively less data available for training (e.g., Korean). To address these issues, we construct a novel KOrean Dataset for Offensive Language Identification (KODOLI). KODOLI comprises more fine-grained offensiveness categories (i.e., not offensive, likely offensive, and offensive) than existing ones. A likely offensive language refers to texts with implicit offensiveness or abusive language without offensive intentions. In addition, we propose two auxiliary tasks to help identify offensive languages: abusive language detection and sentiment analysis. We provide experimental results for baselines on KODOLI and observe that pre-trained language models suffer from identifying "LIKELY" offensive statements. Quantitative results and qualitative analysis demonstrate that jointly learning offensive language, abusive language and sentiment information improves the performance of offensive language identification.

1 Introduction

Data-driven approaches for detecting and measuring offensive content have steadily grown from statistical methodologies to deep learning models for natural language processing (Balayn et al., 2021). Although various methods for detecting offensive language have been proposed, most of them rely on composing training datasets to determine whether a statement is offensive (Fortuna and Nunes, 2018; Mishra et al., 2019; Vidgen and Derczynski, 2020). In South Korea, most of the population actively

* These authors contributed equally to this work.

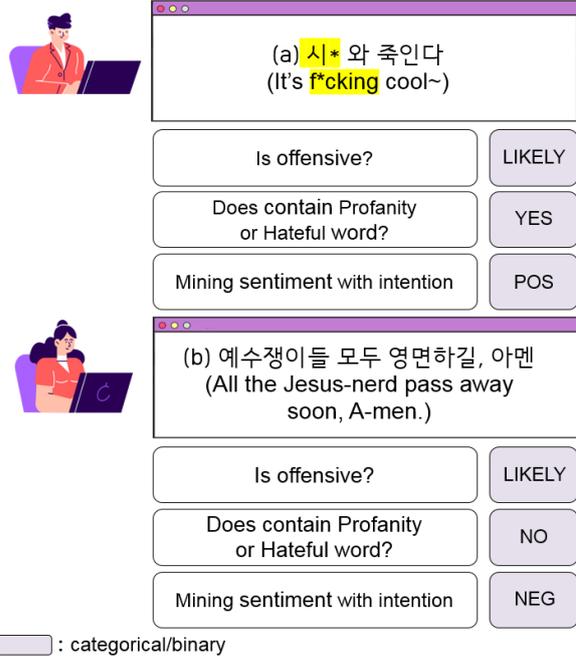


Figure 1: Understanding offensive text (a) and (b) in real-world scenarios considering three questions: identification of offense, existence of abusive language, and underlying sentiment with intention. We supplement the description with examples.

uses the Internet, and the size of online communities is large compared with the population (Park et al., 2021b). The social problems caused by offensive comments have also increased (BBC, 2022). Therefore, we need to analyze and discuss Korean texts and their offensiveness.

Recent approaches have been studied to understand offensive language based on the typology of (Waseem et al., 2017), which differentiates whether the abusive language is directed to a specific individual or group, and whether it is explicit or implicit (Zampieri et al., 2019a; Caselli et al., 2020). This typology helps to identify the offensive language from the statement. However, most existing studies (Sigurbergsson and Derczynski, 2019; Zampieri et al., 2019b) have considered the offens-

ive language detection problem as a binary classification task for distinguishing offensive languages. Although a few studies distinguish profanity and insults under offense (Wiegand et al., 2018), they are limited in classifying various types of offensive language. For instance, offensive intention can be hidden under rhetorical expressions or abusive language can be used without offensive intentions. In particular, in online communities, users freely express their opinions without self-censorship. For instance, users often emphasize emotions with profanity without any offensive intention, as shown in Figure 1(a). In addition, comments on news media (i.e., strictly regulated platforms) are sophisticated in their expressions (i.e., sarcasm or twists) to avoid blocking, as shown in Figure 1(b)¹.

To address these issues, we propose a novel offensive language identification (OLI) task that has three classes: not offensive, likely offensive, and offensive (we extend the existing OLI task by adding a likely offensive class). Moreover, we analyze the attributes of the offensive language. Offensiveness is closely associated with abuse (Caselli et al., 2020). Several studies (Alorainy et al., 2018; Rodriguez et al., 2019) have revealed that negative sentiment messages occur frequently in offensive languages. Therefore, we propose two auxiliary tasks to effectively identify offensive languages: abusive language detection (ALD) and sentiment analysis (SA). The ALD task aims to detect literally abusive language, whereas the SA task extracts the speaker’s subjectivity beyond the sentence. A combination of tasks can be useful for detecting various offensive cases and interpreting the attributes of offensiveness.

We use KODOLI to build classifiers using pre-trained language models (PLMs) (Park, 2020; Park et al., 2021c) and feature-based models (Schuster and Paliwal, 1997; Kim, 2014). We observe that these models struggle to identify likely offensive comments. We utilize a multi-task learning (MTL) technique to utilize related tasks (i.e., ALD and SA). In a qualitative analysis, models that integrate information from offensive language, abusive language, and sentiment exhibit consistent and better-contextualized predictions than those that use only offensive language information.

The contributions of this study are as follows:

- We introduce KODOLI (Korean Dataset for Offensive Language Identification), a new

dataset annotating offensive language, abusive language, and sentiment. We provide a fine-grained annotation scheme for each class to analyze offensive texts in Korean.²

- We find that the PLMs struggle to identify "LIKELY" offensive comments, including implicitly offensive comments and abusive with no intention.
- Quantitative and qualitative analyses demonstrate that learning offensive language, abusive language, and sentiment information improves the performance of OLI.

2 Related Work

Offensive language datasets Offensive language is correlated with several other linguistic and social phenomena including abusive and aggressive language, cyberbullying, racism, extremism, radicalization, toxicity, profanity, and hate speech (Caselli et al., 2020). As hate speeches increased, the number of corpora annotating offensive languages increased (Fortuna and Nunes, 2018; Poletto et al., 2021; Sigurbergsson and Derczynski, 2019; Moon et al., 2020). A previous study (Zampieri et al., 2019a) proposed a novel dataset that provides a scheme for classifying the type and target as well as offensive language. Other studies (Waseem et al., 2017; Sap et al., 2020a; Caselli et al., 2020; Wiegand et al., 2021) have been categorized into explicit and implicit offensive instances. However, none of the aforementioned studies handles the Korean offensive language. To the best of our knowledge, the present study is one of only a few studies that address the Korean offensive language by introducing related auxiliary tasks. Most recently, the concurrent study (Jeong et al., 2022) has proposed Korean offensive language dataset that includes target group, offensive span, and target span annotations as well as offensiveness annotation. They focus on justifying the decision for offensiveness through auxiliary tasks (i.e., target of insult, offensive span). In this study, we focus on subdividing the degree of offensiveness by adding the likely offensive category and auxiliary tasks (i.e., ALD and SA).

Abusive language detection Abuse encompasses many types of fine-grained negative expressions. For instance, Nobata et al. used the term ‘abuse’ to refer collectively to hate speech, derogatory

¹ Blasphemy using phonetic similarity

² <https://github.com/cardy20/KODOLI>

language, and profanity, whereas Mishra et al. considered racism and sexism as abuse. We follow the definition of abusive language suggested by Park et al.: (i) *Profanity* is a word or phrase that insults or curses others; (ii) *Hate speech* is an act of hostile expression based on negative prejudice against a group that has been historically discriminated against because of race, ethnicity, religion, gender, sexual orientation, and gender identity (Cho and Moon, 2020; Madukwe et al., 2020).

Sentiment analysis SA identifies and measures opinions, specifically in determining whether a writer’s attitude toward a particular topic is positive, negative, or neutral (Pang and Lee, 2008; Rodriguez et al., 2019; Liu, 2020). Recent studies have investigated the benefits of using sentiment features in OLI. For instance, Rodriguez et al. applied SA to detect posts suspected of instigating hatred containing highly negative tones. In addition, Plaza-del Arco et al. demonstrated that polarity knowledge can be useful for detecting hate speech and offensive languages more accurately across datasets in Spanish tweets. Inspired by the prior studies, we propose KODOLI, which contains ALD and SA tasks as auxiliary tasks.

3 Task Description

We provide a comprehensive overview of the three tasks for framing the offensive language phenomenon as follows: (i) whether a comment is offensive, likely offensive or not, (ii) whether it contains abusive language (profanity and hate speech), and (iii) whether it has sentiment with intention.

3.1 Main Task: Offensive Language Identification

This task recognizes whether a comment includes offensive language. We consider two factors from previous studies for offensive comments (Wiegand et al., 2018) as follows: (i) Is offensive language directed toward a specific individual or group? (ii) Is an offensive comment explicit or implicit? Unlike previous studies (Zampieri et al., 2019a,b), we establish three categories as follows:

- **Offensive (OFFEN):** Comments that contain surface evidence of non-acceptable language (e.g., profanity) and a targeted offense (i.e., group or individual). This category can be direct or generalized and includes insults, threats, and sexual harassment.

- **Likely offensive (LIKELY):** Comments that could be likely offensive, as they can hide the offensive intention behind sarcasm, irony, and backhanded rude jokes based on stereotypes. The LIKELY class also includes abusive language without malicious intent (additional guidelines that draw a borderline for the likely offensive class can be found in Appendix A.1.).
- **Not offensive (NOT):** Comments that do not contain direct or indirect offense. They do not have profanity or hate speech.

We construct a dataset following the aforementioned guidelines (Appendix A.1 provides details). Owing to the nature of the real-world data collected, many cases in which abusive words expressed intimacy or vitality are observed.

3.2 Auxiliary Task 1: Abusive Language Detection

Auxiliary Task 1 seeks to detect explicit expressions such as profanity and hate speech (see the definition in Section 2). These remarks can be offensive and cause discomfort and conflict within the group. Excessively explicit sexual and obscene expressions are also annotated as abusive language.

- **Abuse (ABS):** Comments that contain profanity and hate speech.
Profanity: e.g., “개같은 *들... , *들 자*하는 이유도 모름?” (you guys are *b*tches*... I do not know why you are masturbating *assh*l*s*?)
Hate speech: e.g., “시* 페미들 너무 싫다.” (I don’t like *f*cking feminist*.), “와 지금 맥날에 백인여자랑 한남 왔는데 존* 이쁘다.” (Wow, a white woman and a *f*cking Korean man* came to McDonald’s right now, and she’s freaking pretty.)
- **Non-abuse (NON):** Comments that do not contain any profanity or hate speech.

3.3 Auxiliary Task 2: Sentiment Analysis

Auxiliary Task 2 analyzes the polarity and intention of the documents and sentences, following the criteria used in the previous sentiment analysis studies (Patwa et al., 2020; Plaza-del Arco et al., 2021).

- **Positive (POS):** Comments that express happiness and support for a person, group, country,

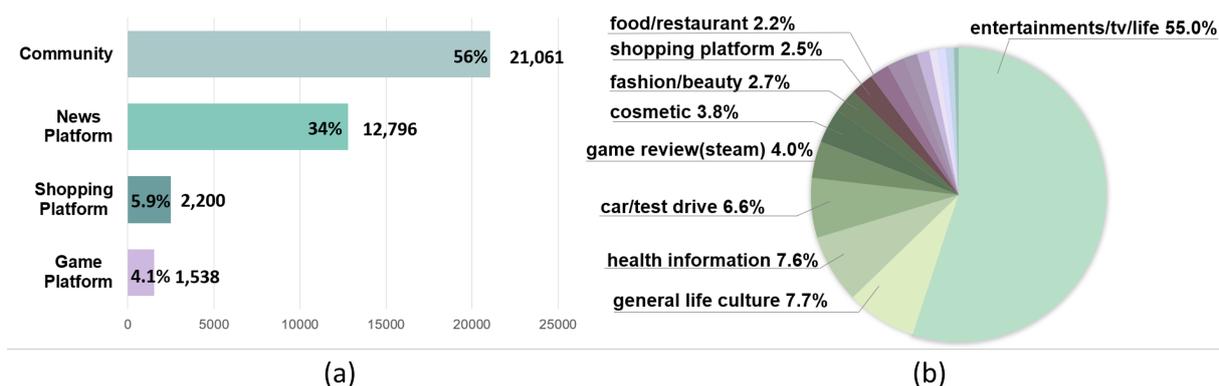


Figure 2: (a) and (b) show the KODOLI’s source and domain, respectively.

or product. e.g., “얼굴 오지게 잘 생겼네” (Your face looks f*cking good.)

- Negative (NEG): Comments that attack a specific target such as a person, group, product, or country. These make people uncomfortable and unhappy. e.g., “현기차 티비 광고보면 성능 품질관련 내용은 별로없고 오로지 감성팔이 ㅋㅋㅋ 개 극혐” (There is not much content related to performance and quality in Hyundai Motor’s TV commercials, only sentimentality haha. It is very hateful.)
- Neutral (NEU): Comments that state a fact or convey news. In general, those that do not fall into these two categories. They also exclude emotional words. e.g., “야채는 건강에 도움이 되니 우리 모두 먹도록 권장합니다.” (Vegetables are good for our health; thus, we encourage you to eat them.)

4 KODOLI

4.1 Data Collection

KODOLI aims to enhance the ability of a system to recognize offensive comments. We collect comments that convey opinions and feelings in explicit and implicit forms. Our dataset is primarily collected and sampled from online communities and news articles, as shown in Figure 2(a). Comments from popular online Korean communities, such as DC-inside³ (from October 2020 to December 2020). The comments on DC-inside contain profanity, hateful speech, and sexual harassment through sub-communities. Therefore, KODOLI is practically similar to a raw representation. We also collect comments from articles from July 2021

³ <https://www.dcinside.com/>

to September 2021. The data are collected from various fields on the Naver news platform⁴. We collect comments from top-ranked articles on pages to ensure contentiousness. To diversify the collected comments, articles are randomly selected from the topic categories of the platform, and from each article, a maximum of 500 comments are collected. Approximately 15 domains are shown in Figure 2(b). Entertainment, TV shows, and life domains constitute the majority of the sample. Although the collected comments are distributed unevenly among domains, they reflect the interests of real-world users.

Duplicates and unnecessary special characters are removed. In addition, during comment collection, special attention is paid to preventing bias on specific topics. For instance, we first count the words that frequently appear by topic. We then replace a certain percentage of comments containing a specific word to comments with the same label collected from a new domain to match the proportions⁵. Comments with sentiment polarity are supplemented by sampling reviews from open-source databases⁶ collected from the game community⁷ and Naver shopping platforms⁸. Finally, 39,589 comments are retained.

4.2 Annotation

We collect at least three annotators per post and attempt to balance gender and diversify educational backgrounds. During the annotation process, we

⁴ <https://news.naver.com/>

⁵ We found after applying this technique, the difference in occurrence between the most frequent (except for stopwords) and least frequent words was about 2%.

⁶ <https://github.com/bab2min/corpus/tree/master/sentiment>

⁷ <https://store.steampowered.com/>

⁸ <https://shopping.naver.com/>

OLI	Abusive Language Detection		Sentiment Analysis			Total
	NON	ABS	POS	NEU	NEG	
NOT	22,453	2,513	10,548	10,865	3,553	24,966 (65.4%)
LIKELY	2,461	3,122	207	1,436	3,940	5,583 (14.6%)
OFFEN	751	6,875	99	1,164	6,363	7,626 (20.0%)
TOTAL	25,665 (67.2%)	12,510 (32.8%)	10,854 (28.4%)	13,465 (35.3%)	13,856 (36.3%)	38,175

Table 1: Distribution of label combinations in the KODOLI. Herein OLI denotes Offensive Language Identification. ABS and NON denote the abuse and non-abuse for the abuse class. POS, NEU, and NEG denote positive, neutral, and negative, respectively, for the sentiment class.

contact undergraduate and graduate students. Even Korean speakers are selected using crowdsourcing. For each comment, the annotators indicate whether a comment is offensive, likely offensive, or not. Thereafter, they categorize whether the comment contains abusive language, such as profanity and hate speech in Korean, and simultaneously annotated intention in terms of sentiment polarity (Cho and Moon, 2020; Park et al., 2021a; Sohn et al., 2012). If the comments are free of profanity and hate speech, the participants are asked to judge the intended support or attack nature within the comments, following abusive language and sentiment guidelines.

Inter-annotator agreement The inter-annotator agreement is calculated based on Krippendorff’s alpha (α) (Krippendorff, 2011), a reliability coefficient developed to measure agreement among annotators. Annotators agree on an offensive comment at a rate of 82.8% (Krippendorff’s $\alpha=0.42$). In particular, we compute Krippendorff’s α using only the LIKELY label, which is 0.41. Sentiment indicates an average Krippendorff’s α of 0.45, indicating moderate agreement (Hughes, 2021; Sap et al., 2020b). For the ALD task, we obtain Krippendorff’s α of 0.72. The final dataset consists of 38,525 Korean comments.

4.3 Data Statistics

Table 1 presents the statistics of comments per task. Comment counts are provided for six and nine combinations. In our corpus, we observe the tendency of each class in terms of offensive language. For example, many comments with abusive language in ALD (6,875) and negative labels in SA (6,363) are offensive. We observe 2,513 comments with abusive language but non-offensive. These use swear words to lay emphasis and to express enthusiasm with positive sentiment, for example, ‘짹짹지’ (f*ck cool), ‘존나 잘한다’(damn good). Most of the comments with LIKELY have a negative sen-

timent. They relatively have the abusive language with no target; for instance, they express their emotion with the abusive language ‘술치먹으면 감수성더예민해져서 *같음’(If I drink alcohol, I will become more sensitive and I hate this shit).

Table 1 also shows the distribution of each label. Comments are categorized to binary depending on the abusive content and two ternary classes for identifying offensive language: NOT, LIKELY, and OFFEN, and sentiment polarity: POS, NEU, and NEG. Our corpus’s offensive and abusive category distributions are skewed, whereas the sentiment distribution is balanced. Each task’s label distribution also follows the real-world comments’ nature (i.e., about two-thirds of the comments contain no profanity and are not offensive).

We analyze the frequency of comments tagged as abusive. We observe the obscene and identity terms for demographic groups (e.g., gender, race, and political orientation). We guide more in detail in A.2.

5 Modeling

Preprocessing We randomly shuffle and split the dataset into training (26,967), validation (5,778), and testing (5,780) sets. We apply the morpheme-level pre-tokenization, which is effective for character-rich languages (Park et al., 2021c). Specifically, we select Mecab-ko⁹ (Kudo, 2006), a pre-tokenizer adapted for Koreans. In the case of BERT-family models, we apply the WordPiece tokenizer following the work (Devlin et al., 2019).

Multi-task learning MTL has been widely used to train with data from multiple tasks, and we use the hard parameter sharing technique (Crawshaw, 2020). This is the practice of sharing model weights between related tasks; therefore, each weight is trained to minimize multiple loss func-

⁹ <https://bitbucket.org/eunjeon/mecab-ko/src/master/>

Model	NOT			LIKELY			OFFEN			Macro Average		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CNN	88.49	90.58	89.52	33.87	40.47	36.88	76.54	62.44	68.77	66.30	64.50	65.06
BiLSTM	88.39	90.11	89.24	33.46	36.74	35.02	75.74	66.99	71.10	65.86	64.61	65.12
KLUE-BERT	91.07	88.48	89.75	39.65	41.44	39.06	73.55	74.72	74.13	67.19	68.21	67.65
KLUE-RoBERTa	92.34	87.19	89.69	36.73	43.37	39.77	71.76	76.40	74.01	66.94	68.99	67.82
KoELECTRA	91.81	89.90	90.84	37.90	43.92	40.69	77.91	75.68	76.78	69.21	69.83	69.44

Table 2: Results for offensive language identification task on the KODOLI test set. We report the precision (P), recall (R) and F1-score for the classifiers (best in bold).

tions jointly. We construct two kinds of parts: a shared part and task-specific parts. We share the encoder layer and construct a task-specific layer for each task based on the shared encoder.

Let $x_1, x_2, \dots, x_k \in U$ be the given text with k words from input sentence U . In PLMs, we add a special symbol [CLS] at the beginning of the text and add the [SEP] symbol at the end. The embedding layer transforms a fixed-length sequence into an embedding matrix. The embedding matrix is fed to each shared encoder. The hidden states, h_1, h_2, \dots, h_k , are obtained from the encoder. We obtain the output vector \mathbf{h} from the max-pooling layer in feature-based models while using the special token [CLS] to construct the pooled output \mathbf{h} in the PLMs. After feeding the output vector into each task-specific layer, we obtain the output logit, \mathbf{z} . It passes through the softmax layer to calculate the cross-entropy loss. L_{OLI}, L_{ALD} , and L_{SA} denote cross-entropy losses for OLI, ALD, and SA tasks, respectively. $L_{CE}(U)$ is the weighted sum of the joint objective functions L_{OLI}, L_{ALD} and L_{SA} ,

$$L_{CE}(U) = \lambda_o L_{OLI}(U) + \lambda_a L_{ALD}(U) + \lambda_s L_{SA}(U), \quad (1)$$

where λ_o, λ_a and λ_s denote the weights for the OLI, ALD, and SA tasks, respectively.

6 Experimental Results

We first experiment with the single-task learning (STL) method for the OLI task using our dataset, KODOLI, in the popular and powerful NLP models (the implementation details are described in Appendix A.4). Further, we experiment with the MTL method by combining the OLI task with auxiliary task 1 (ALD) or auxiliary task 2 (SA), which are our proposed approaches. We evaluate the experimental performance using the following metrics: precision (P), recall (R), F1-score (F1) for each class and macro-averaging scores.

6.1 Experimental Settings

- BiLSTM (Schuster and Paliwal, 1997): This model consists of two layers of bidirectional long short-term memory initialized randomly. The outputs of the second layer are max-pooled to predict the result using a multi-layer perceptron.
- CNN (Kim, 2014): This model takes individual token representations as the input and then transforms sequence representations for the output using 1D convolution and max-over-time pooling.
- KLUE-BERT (Park et al., 2021c): This model follows the BERT (Devlin et al., 2019) structure. It is designed to pre-train language representation from unlabeled Korean texts¹⁰.
- KLUE-RoBERTa (Park et al., 2021c): This model follows the RoBERTa (Liu et al., 2019) architecture, which uses dynamic masking strategy and whole-word masking. It is pre-trained using the same corpora as KLUE-BERT.
- KoELECTRA (Park, 2020): This model follows the ELECTRA (Clark et al., 2020) architecture¹¹ trained with masked language modeling and replaced token detection objectives.

6.2 Results of Offensive Language Identification Task

Table 2 presents the results of the experiments with the five baseline models for the OLI task. KoELECTRA performs best in most evaluation metrics, in-

¹⁰ It was pre-trained on five Korean corpora of approximately 62GB consisting of formal documents, such as news and books, colloquial texts, multilingual web pages, encyclopedia, and petitions.

¹¹ It is trained with 34GB of crawled news data and the MODU corpus (<https://corpus.korean.go.kr/>).

Model	NOT			LIKELY			OFFEN			Macro Average		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
KoELECTRA _{OLI}	91.81	89.90	90.84	37.90	43.92	40.69	77.91	75.68	76.78	69.21	69.83	69.44
KoELECTRA _{OLI+ALD}	92.48	89.64	91.04	38.50	47.38	42.48	78.16	75.04	76.57	69.71	70.68	70.03
KoELECTRA _{OLI+SA}	92.15	90.45	91.29	38.14	45.30	41.41	78.62	74.48	76.49	69.64	70.08	69.73
KoELECTRA _{OLI+ALD+SA}	92.73	89.27	90.97	38.03	48.48	42.62	79.03	75.44	77.19	69.93	71.06	70.26

Table 3: The MTL results on the KODOLI test set using KoELECTRA. *OLI* means a model that trained only OLI task in the STL method. *OLI+ALD* and *OLI+SA* mean models trained in MTL for OLI task with ALD task or SA task, respectively. *OLI+ALD+SA* means a model jointly trained on OLI, ALD, and SA tasks in the MTL method.

cluding precision, recall, F1-score for all classes and macro-averaging scores. CNN and BiLSTM show similar results for the macro average F1-score, both of which have lower performance than the PLMs (i.e., KLUE-BERT, KLUE-RoBERTa, and KoELECTRA). These results indicate that the PLM series outperforms the non-PLM series in the OLI task. We observe that performance for the LIKELY class has a significantly lower F1-score compared to not offensive and offensive classes in all models. These results indicate that existing models suffer from the LIKELY class. In particular, the non-PLMs (i.e., CNN and BiLSTM) perform poorly in LIKELY class. We observe that models tend to predict ‘non-offensive’ about comments that hide the offensive intention and have no lexical cues regards to be patterned easily (i.e. f*ck). For example, “센텀시티는 바벨탑이라. 전부 무너져내릴 것이다.” (Centum City is the Tower of Babel. Someday it will completely collapse.) In addition, models easily predict ‘offensive’ if there is abusive language in a sentence. The results of the offensive class show higher precision, recall, and F1-score, which is interpreted as high consistency and sensitivity compared to the likely offensive instances.

6.3 Results on Multi-task Learning

Does training with auxiliary tasks improve the performance of OLI? We evaluate the performance of the MTL based on KoELECTRA (which performed best on the STL) in the OLI task. Table 3 summarizes the experimental results of KoELECTRA trained on the combination of all tasks, including the OLI. First, when learning the OLI, ALD, and SA tasks simultaneously, we observe the best precision, recall, and F1-score in most classes and the macro average. In addition, all the MTL models outperform the STL framework in all metrics except recall in OFFEN. In particular, MTL models with auxiliary tasks are effective in the LIKELY class. We observe a 1.79-point F1-

score improvement in the LIKELY class when we jointly learn ALD and OLI. The LIKELY class contains instances of abusive language but no targeted offense. In the case of jointly learning the OLI and the SA tasks, it shows 0.72 points up F1-score performance in the LIKELY class. This indicates that sentiment features are also effective for KODOLI, including the LIKELY class. We also observe that MTL outperforms STL in the other baseline models (Appendix A.5). We can see that the ALD and SA tasks complement each other to help the model identify offensive languages.

6.4 Qualitative Analysis

We qualitatively examine the model’s ability to understand various offensive cases more effectively. Models that integrate information from offensive languages, abusive terms, and sentiment show consistent and better-contextualized predictions than those that only use offensive language information. In particular, the model trained jointly on OLI and ALD is more effective in the LIKELY examples. In Table 4, although profanity or derogatory language in comments (a) and (b) are not used for offensive purposes, they can cause discomfort and shame. A model trained using offensive language with sentiment performs better in qualitative analysis. For instance, example (c) illustrates a sarcastic case without abusive terms that is implicitly offensive. The model trained with offensive and abusive language and sentiment information correctly predicts all examples (a) ~ (f), which are misclassified in the model trained with the OLI task. These results indicate that training the model with two auxiliary tasks provides a more delicate and accurate identification of offensive language.

6.5 Error Analysis

For further investigation into closing the gap, we inspect approximately 750 instances misclassified as false positives and false negatives from the MTL

Class	Comment	OLI	OLI+ALD	OLI+SA	OLI+ALD+SA
LIKELY	(a) 졸려시발 (Sleepy sh*t)	✗	✓	✗	✓
	(b) 근데 았창 살빼면 돈도모이고 건강해지고 존나좋은데 (Losing weight saves money and makes you healthier, so that's great. Or, my mother is a wh*re.)	✗	✓	✗	✓
	(c) 기자 참 아무나한다 (It seems that anyone can easily become a journalist.)	✗	✗	✓	✓
OFFEN	(d) 힙플이새끼들은 힙합을 제발 음악이라고 포장하지마라 (Hip-hop b*st*rds, please don't treat hip-hop as music.)	✗	✓	✓	✓
	(e) 콜빈 특등 머저리의 헛 소리 누가 믿나 그리고도 밥은 목구멍에 잘 넘어 갈꺼야 더러운 (Who believes this b*llsh*t of the special grade idiot with an empty skull? Do you get up each morning, too?)	✗	✗	✓	✓
	(f) 범죄자 10 8 새들...저것들부터 불태워버리자!! (Puc b*s-crim-tard Let's burn them down!!)	✗	✗	✗	✓

Table 4: Qualitative examples comparing offensive language only, and offensive language with the auxiliary tasks combination models.

model (KoELECTRA). In false positive cases, the model struggles to predict comments as offensive or likely offensive for not offensive comments. The opposite is true for false negatives. We additionally analyze likely offensive class in Appendix (A.6).

False positive types :

- The mixture of swearing but the opposite intention: e.g., *물싸개는 ㄹㅇ 별로 심한욕처럼 안느껴지는데. (I do not sound s*men excreter like a very harsh insult.)
- Using abusive language as an expression of emphasizing emotion: e.g., 와 씨* 테이블에 있는데 창문에 자꾸 하얀게 지나가는거야 (Wow, f*cking I'm at the table and something white passes repeatedly.)

False negative types :

- Implicitly offensive: e.g., 여고생이 맛있나요 여대생이 맛있나요? (How do you feel that high school girls are more tasty? or female college students?¹²)
- Modified profanity: e.g., 야 이 뽕신아 ㅋㅋ (Hey, you bbastard haha), 닥치고 일본가서 살어.. (Shudd¹³ up and live in Japan.)

¹² Sexual harassment expressions

¹³ Similar pronunciation

7 Conclusion

In this paper, we introduced KODOLI, a new Korean dataset for OLI. To this end, we collected various offensive comments from online communities and news articles in diverse domains. In particular, we expanded a fine-grained label called 'likely offensive' to distinguish the implicitly offensive and abusive comments with no targeted offense. We proposed two auxiliary tasks to help models identify offensive languages: ALD and SA. Finally, we released 38k comments annotated with offensive language, abusive language, and sentiment information. Using KODOLI, we demonstrated that modeling offensive language using abusive language and sentiment was effective in quantitative and qualitative analyses. We expect our research will benefit further studies that analyze offensiveness in Korean.

Limitations

Risk in annotation Perceptions of "offensiveness" can vary from person to person. Therefore, we outsourced our data. In addition to typical offensive norms, which refer to expert opinions, the majority decided on annotations. Eleven annotators participated in this study. The definitions in our guidelines are not representative of all possible perspectives. It is important to include the opinions of the targeted minorities when dealing with the an-

notation of offensive language. We tried to balance gender among annotators (57% men, 43% women); however, another specific target demographic remains challenging. For the consistency and quality of the data, when the concordance rate was lower than the threshold 0.5, examples were put on hold in favor of consistency. For instance, if 2 NOT, 4 LIKELY, and 5 OFFEN for a sample, the OFFEN label got the most voted, but $5/11 = 0.45 \leq 0.5$, so it is excluded from the dataset. In the future, these examples should be further studied and dealt with.

Coverage Although we collected data from various sources, we acknowledge that the data do not represent all of them. In addition, there could be bias depending on the collection period, and it could be difficult to cover neologisms.

Ethics Statement

To protect the privacy, we only collected comments rejecting all personally identifiable information, including the user IDs. Subsequently, we removed comments containing personal information, such as phone numbers and emails. Our dataset contains real-life examples of abusive language obtained from actual web data. Therefore, we notified the dangers of the postings in advance. To mitigate the risks, we limited the number of maximum comments workers worked per day, and they were given sufficient time to work. We paid workers above minimum wage. We are aware that our topics could have side effects, such as KODOLI's potential malicious use such as generating bad words. Nevertheless, we urge the practical use of KODOLI, such as filtering offensive comments explicitly and identifying potentially offensive content from multiple points of view. This can prevent the negative influence of users intentionally leaving malicious comments.

Acknowledgements

This work was supported by the Basic Research Program through the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C3010430), the NRF grant funded by the Korea government (MSIT) (No. 2022R1C1C1010317), and Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program (Korea University)).

References

2022. [Korea: High-profile suicides spark cyber-bullying petition.](#)
- Wafa Alorainy, Pete Burnap, Han Liu, Amir Javed, and Matthew L Williams. 2018. Suspended accounts: A source of tweets with disgust and anger emotions for augmenting hate speech data sample. In *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 581–586.
- Agathe Balayn, Jie Yang, Zoltan Szlavik, and Alessandro Bozzon. 2021. Automatic identification of harmful, aggressive, abusive, and offensive language on the web: A survey of technical biases informed by psychology literature. *ACM Transactions on Social Computing (TSC)*, 4(3):1–56.
- Tommaso Caselli, Valerio Basile, Mitrović Jelena, Kar-toziya Inga, Granitzer Michael, et al. 2020. I feel offended, don't be abusive! implicit/explicit messages in offensive and abusive language. In *Language Resources and Evaluation Conference (LREC)*, pages 6193–6202.
- Won Ik Cho and Jihyung Moon. 2020. A study on the construction of korean hate speech corpus: Based on the attributes of online toxic comments. In *Annual Conference on Human and Language Technology*, pages 298–303. Human and Language Technology.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations (ICLR)*.
- Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4171–4186.
- Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30.
- John Hughes. 2021. krippendorffsalph: An r package for measuring agreement using krippendorff's alpha coefficient. *R Journal*, 13:413.
- Younghoon Jeong, Juhyun Oh, Jaimeen Ahn, Jongwon Lee, Jihyung Mon, Sungjoon Park, and Alice Oh. 2022. Kold: Korean offensive language dataset. In *arXiv:2205.11315*.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

- Klaus Krippendorff. 2011. Computing krippendorff’s alpha-reliability.
- Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer. <http://mecab.sourceforge.jp>.
- Bing Liu. 2020. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. In *arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Kosisochukwu Madukwe, Xiaoying Gao, and Bing Xue. 2020. In data we trust: A critical analysis of hate speech detection datasets. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 150–161.
- Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1088–1098.
- Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2019. Tackling online abuse: A survey of automated abuse detection methods. *arXiv preprint arXiv:1908.06024*.
- Jihyung Moon, Won Ik Cho, and Junbum Lee. 2020. BEEP! Korean corpus of online news comments for toxic speech detection. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*, pages 25–31.
- Chikashi Nobata, Joel R. Tetreault, Achint Oommen Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pages 145–153.
- B Pang and L Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135.
- Jangwon Park. 2020. KoELECTRA: Pretrained electra model for korean. <https://github.com/monologg/KoELECTRA>.
- Jin Won Park, Young-Yun Na, and Kyubyong Park. 2021a. A new dataset for korean toxic comment detection. In *Proceedings of the Korea Information Processing Society Conference*, pages 606–609.
- San-Hee Park, Kang-Min Kim, Seonhee Cho, Jun-Hyung Park, Hyuntae Park, Hyuna Kim, Seongwon Chung, and SangKeun Lee. 2021b. KOAS: Korean text offensiveness analysis system. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 72–78.
- Sungjoon Park, Sungdong Kim, Jihyung Moon, Won Ik Cho, Kyunghyun Cho, Jiyeon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, et al. 2021c. KLUE: Korean language understanding evaluation. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.
- Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Tamar Solorio, and Amitava Das. 2020. SemEval-2020 Task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790.
- Flor Miriam Plaza-del Arco, Sercan Halat, Sebastian Padó, and Roman Klínger. 2021. A multi-task learning approach to hate speech detection leveraging sentiment analysis. *IEEE Access*, 9:112478–112489.
- Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, 55(2):477–523.
- Axel Rodriguez, Carlos Argueta, and Yi-Ling Chen. 2019. Automatic detection of hate speech on facebook using sentiment and emotion analysis. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, pages 169–174.
- Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020a. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5477–5490.
- Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020b. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5477–5490.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- Gudbjartur Ingi Sigurbergsson and Leon Derczynski. 2019. Offensive language and hate speech detection for danish. *arXiv preprint arXiv:1908.04531*.

Sun-Ju Sohn, Mi-Sook Park, Ji-Eun Park, and Jin-Hun Sohn. 2012. Korean emotion vocabulary: Extraction and categorization of feeling words. *Science of Emotion and Sensibility*, 15(1):105–120.

Bertie Vidgen and Leon Derczynski. 2020. Directions in abusive language training data, a systematic review: Garbage in, garbage out. *Plos one*, 15(12):e0243300.

Zeerak Waseem, Thomas Davidson, Dana Warmlesley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84.

Michael Wiegand, Maja Geulig, and Josef Ruppenhofer. 2021. Implicitly abusive comparisons—a new dataset and linguistic analysis. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 358–368.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 shared task on the identification of offensive language.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1415–1420.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

A Supplemental Material

A.1 Guidelines Detail

We have established additional guidelines for offensive language that workers can refer to when distinguishing offensive areas from likely offensive areas during the crowdsourcing process. Especially the clearer the expression of hate, profanity, and obscenity, and the more specific the target is, the closer it is to the OFFEN label. Therefore, harassment and threats are also classified as the OFFEN.

- A comment where profanity and hate speech appears explicitly and the target is clear : e.g., 노라주다 저*깁 ㄹㅇ *패고싶게생겼노. (Norajuda that bas*ard’s face really makes me want to beat him.)

- A comment which reveals a hostile attitude toward an individual or group : e.g., 이런 *게이새*들 보면 쥐패서 죽*버리고싶은데. (When I face these f*cking g*y bas*ards, I want to beat up and kill them.)
- A comment in which an expression of direct threat is present : same as second example.
- A comment which contains expressions that sexually harass or humiliate : e.g., 이 짤보고 고로상 빙의됐다 애* 존맛겠다. (After seeing this meme, I was possessed with Goro and wanted to f*ck your m*ther.)

We define the terms in the guidelines to the participants well before the annotation work, and the annotators were trained follow them precisely. Through the pilot study, we analyze the types and set the following guidelines with examples to draw a borderline for the LIKELY class.

- A comment judged to have offensive intent due to intrinsic bias: e.g., 장애인들 몸비틀고 잘 노내. (Disabled people twist their bodies and play well.)
- A comment with disrespectful expressions (reproaching, admonishing, etc.): e.g., 신이 있어? 그만건 없음. 정신차리고 사세요. (Did you say that there is a god? Nothing like that. Calm down.)
- A comment may cause shame: e.g., 섹*할때 필수용품.. (A must-have item for s*x.)
- A comment with a cynical tone: e.g., 조각일 보 다운 기사네.. (It is an article from a daily forgery..)
- A comment considered to be implicitly disparaging: e.g., 근데 화*는 왜 빠는거야? (Why did you suck Hw*s*¹⁴?)
- A comment with abusive language but judged to be acceptable: e.g., 와 미친 개잘한다. (Wow, it’s crazy, you are doing f*cking great!)

A.2 Abusive Language in KODOLI

We analyze comments with abusive labels, extract the profane term and hate term based on the frequency, and organize them into a bag of words.¹⁵

¹⁴ Celebrity

¹⁵ <https://github.com/cardy20/KODOLI/tree/main/bow>

A.3 Experimental Results on Each Auxiliary Task in the STL Settings

We evaluate both the auxiliary tasks, ALD and SA. Table 5 and Table 6 summarize the baseline results of the STL setup.

Model	Abusive Language Detection		
	P	R	F1
BiLSTM	89.03	87.27	88.05
CNN	90.53	88.22	89.22
KLUE-BERT	88.60	88.22	88.41
KLUE-RoBERTa	88.96	88.61	88.78
KoELECTRA	90.96	90.02	90.47

Table 5: Precision, recall, F1-score of abusive language detection

Model	Sentiment Analysis		
	P	R	F1
BiLSTM	73.31	72.16	72.61
CNN	74.32	73.46	73.81
KLUE-BERT	77.02	76.78	76.85
KLUE-RoBERTa	76.88	76.51	76.68
KoELECTRA	77.70	77.69	77.64

Table 6: Precision, recall, F1-score of sentiment analysis

A.4 Implementation Details

- a. Hyperparameters: We used a batch size of 32 examples for each model and a fixed sentence length of 128. We used the AdamW optimizer (Loshchilov and Hutter, 2017). We set 48 seed and explored the learning rate to obtain the best results for each model. For CNN and BiLSTM, the learning rate was searched for between 1e-04, 2e-04, 3e-04, 4e-04, 5e-04, 6e-04, 7e-04. We searched for the following learning rates: 7e-06, 9e-06, 1e-05, 2e-05, 3e-05, 4e-05, for KLUE-BERT, KLUE-RoBERTa, and KoELECTRA. In the case of MTL, we initially set all lambda weights to 1.0. We searched for an appropriate lambda weight by using a grid search.
- b. Training conditions: We implemented the model using PyTorch (Paszke et al., 2019) and used an NVIDIA GeForce RTX 3090 with 24 GB of VRAM to train all baseline models. We used the HuggingFace library for our BERT-family models¹⁶.

¹⁶ <https://huggingface.co/klue/bert-base>

Model	Task	Macro Average		
		P	R	F1
CNN	OLI	66.30	64.50	65.06
	OLI + ALD + SA	67.42	67.03	66.33
BiLSTM	OLI	65.86	64.61	65.12
	OLI + ALD + SA	66.98	65.06	65.91
KLUE-BERT	OLI	67.19	68.21	67.65
	OLI + ALD + SA	68.12	69.17	68.53
KLUE-RoBERTa	OLI	66.94	68.99	67.82
	OLI + ALD + SA	68.10	70.22	68.67
KoELECTRA	OLI	69.21	69.83	69.44
	OLI + ALD + SA	69.93	71.06	70.26

Table 7: STL(OLI) vs MTL(OLI+ALD+SA)

A.5 Experimental Results on the Baseline Models in the MTL Settings

Table 7 presents the experimental results obtained using KODOLI on the STL method for the OLI task and the MTL method combining the OLI task with auxiliary task 1 (ALD) and auxiliary task 2 (SA) in the five baseline models. This result indicates that the performance is improved when two auxiliary tasks are jointly learned in all baseline models.

A.6 Error Analysis Details

We conduct an in-depth analysis of the LIKELY class, which shows relatively low performance on classifiers, with auxiliary labels. Of the 718 examples of the LIKELY class in the validation set, 208 examples misclassified LIKELY as NOT and 197 LIKELY examples as OFFEN. Among the cases misclassified as NOT, 136 cases are labeled as non-abusive language, which means that they have no explicit expression (i.e., hate words, profane). We find that a large portion of the cases is sarcastically or twisted as considering the context of the sentence. Especially, if a comment is likely offensive under the social and cultural background (e.g., first and fourth examples in A.1), the distribution of prediction scores tends to appear evenly. In addition, most misclassified cases as OFFEN (72%) contain an explicit and emphasized expression. We conjecture that classifiers predict OFFEN by looking at the specific word itself. However, humans take it differently in feeling offended.

Empirical Investigation of Neural Symbolic Reasoning Strategies

Yoichi Aoki,¹ Keito Kudo,¹ Tatsuki Kuribayashi,^{1,2} Ana Brassard,^{3,1}

Masashi Yoshikawa,¹ Keisuke Sakaguchi,^{1,3} Kentaro Inui^{1,3}

¹Tohoku University, ²Langsmith, Inc., ³RIKEN

{youichi.aoki.p2, keito.kudo.q4}@dc.tohoku.ac.jp,

kuribayashi@tohoku.ac.jp, ana.brassard@riken.jp,

{yoshikawa, keisuke.sakaguchi, kentaro.inui}@tohoku.ac.jp

Abstract

Neural reasoning accuracy improves when generating intermediate reasoning steps. However, the source of this improvement is yet unclear. Here, we investigate and factorize the benefit of generating intermediate steps for symbolic reasoning. Specifically, we decompose the reasoning strategy w.r.t. step granularity and chaining strategy. With a purely symbolic numerical reasoning dataset (e.g., $A=1$, $B=3$, $C=A+3$, $C?$), we found that the choice of reasoning strategies significantly affects the performance, with the gap becoming even larger as the extrapolation length becomes longer. Surprisingly, we also found that certain configurations lead to nearly perfect performance, even in the case of length extrapolation. Our results indicate the importance of exploring effective strategies for neural reasoning models.¹

1 Introduction

Artificial intelligence researchers have been attempting neural-symbolic integration for a long time (d’Avila Garcez and Lamb, 2020; Hamilton et al., 2022). Neural models tend to perform better when generating intermediate reasoning steps in addition to the answer. This phenomenon was seen across various reasoning tasks, such as math word problems (Wei et al., 2022; Cobbe et al., 2021; Kojima et al., 2022; Recchia, 2021; Lewkowycz et al., 2022), commonsense reasoning (Wei et al., 2022; Wang et al., 2022), and symbolic reasoning (Wei et al., 2022; Kojima et al., 2022). However, it is yet unclear which factors in the intermediate step generation bring the benefit. Previous studies often used different strategies for step generation in an ad-hoc manner. To investigate this, we break down the neural reasoning process into two strategies: *output strategy* and *chaining strategy* (Figure 1). The output strategy (§2.1) determines the granularity of

¹Code available at: <https://github.com/ao1neko/reasoning-strategy>

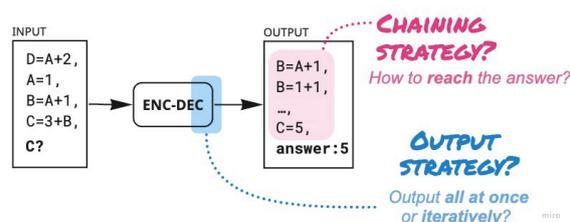


Figure 1: In a controlled setting, we found that output and chaining strategy choice significantly impact performance when conducting multi-step reasoning.

intermediate reasoning step generation (all at once vs. step-by-step vs. token-by-token). Some studies trained the models to generate reasoning steps and a conclusion derived from them at once (Nye et al., 2021; Lewkowycz et al., 2022; Wei et al., 2022; Kojima et al., 2022; Wang et al., 2022; Recchia, 2021), some generated a single reasoning step given the input and iterated this process until achieving a conclusion (Sanyal et al., 2022; Picco et al., 2021; Tafjord et al., 2021), and others iteratively generated sub-goals as well as reasoning steps (Liang et al., 2021; Shwartz et al., 2020).

In turn, the chaining strategy (§2.2) defines the reasoning path direction (shortest path vs. exhaustive path vs. backward path). For example, some studies used a backward chaining process (Picco et al., 2021; Rocktäschel and Riedel, 2017; Cingillioglu and Russo, 2019), while others adopted exhaustive searches (Tafjord et al., 2021; Liang et al., 2021; Yang et al., 2022).

To compare the strategies, we prepared a test bed of numerical reasoning problems in a simplified language (Figure 1). This format allows for more controlled testing while serving as a necessary condition—should a model fail to solve it, it cannot be expected to adequately generalize to more complex math word problems.

We found that both strategies substantially affect the symbolic reasoning performance of neural seq2seq learners. Overall, iterative generation

"ALL AT ONCE"

INPUT $\xrightarrow{\text{ENC-DEC}}$ $B=A+1, B=1+1, \dots, C=5, \text{answer}:5$

"STEP BY STEP"

INPUT $\xrightarrow{\text{ENC-DEC}}$ $B=A+1$
INPUT, $B=A+1$ $\xrightarrow{\text{ENC-DEC}}$ $B=1+1$
 \vdots
INPUT, $B=A+1, \dots, C=5$ $\xrightarrow{\text{ENC-DEC}}$ $\text{answer}:5$

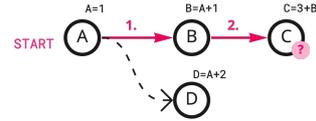
"TOKEN BY TOKEN"

INPUT $\xrightarrow{\text{ENC-DEC}}$ B
INPUT, B $\xrightarrow{\text{ENC-DEC}}$ $=$
 \vdots
INPUT, $B=A+1, \dots, \text{answer}:$ $\xrightarrow{\text{ENC-DEC}}$ 5

(a) *All-at-once*: output the entire reasoning chain and answer in a single call. *Step-by-step*: iteratively build the output with a single calculation step per call. *Token-by-token*: iteratively output only one *token* per call.

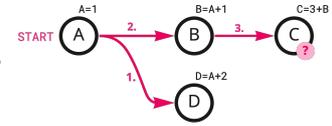
"SHORTEST PATH"

$B=A+1, B=1+1, B=2, C=3+B,$
 $C=3+2, C=5, \text{answer}: 5$



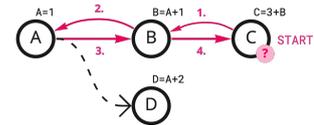
"EXHAUSTIVE"

$D=A+2, D=1+2, D=3, B=A+1,$
 $B=1+1, B=2, C=3+2, C=5,$
 $\text{answer}: 5$



"BACKWARD"

$C=3+B, B=A+1, A=1, B=1+1,$
 $B=2, C=3+2, C=5, \text{answer}: 5$



(b) The graph nodes represent variables and edges their dependencies. *Shortest path*: a minimal chain starting from the first necessary equation. *Exhaustive*: greedily solve all equations until the target is reached. *Backward*: start from the target's equation, backtrack over dependencies until a known value is reached, then solve each equation in order.

Figure 2: Overview of (a) output and (b) chaining strategies given the INPUT: $D=A+2, A=1, B=A+1, C=3+B, C?$

outperformed all-at-once outputting, and roughly granular reasoning steps (i.e., shortest-path chaining) lagged behind finely granular steps (i.e., exhaustive and backward chaining). Surprisingly, some settings had near-perfect performance even in generalization tests which extrapolate over greater reasoning depths and unseen numbers during training.

2 Experimental settings

Problem definition. We evaluated the models' ability to iteratively perform arithmetic operations over given symbols. Given a series of equations, the task is to answer the value of a target variable (Figure 1). Each question also has a certain reasoning depth—the number of *necessary* equations to reach the answer. For example, the depth of the question $A=1, B=2+A, C=3+B, D=2, C?$ is 3 ($A=1, B=2+A, C=3+B$).

Each equation defines either an assignment (e.g., $A=1$) or a modular addition and an assignment (e.g., $B=3+1$). The addition is mod 100. The question contexts also contain distractors that are not necessary to calculate the answer (e.g., $D=A+2$ in Figure 1). A value assigned to a particular variable is typically referred to in different equations (e.g., $A=1, B=A+1$). Numbers, variables, and the ordering of equations are randomly assigned.

Motivation for using artificial data There are mainly three advantages to this dataset. First, the

symbolic format allows easier control of reasoning depth for generalization tests. Specifically, we trained a model using instances with shallow (1-5) depths and evaluated them with instances with shallow/deep (1-12) depths. On the other hand, math word problems are harder to control for reasoning depth (e.g., it is not easy to come up with various instances which have a reasoning depth of 10). Second, we wanted to avoid the "spurious bias" that natural (math word) texts implicitly bring into the model (Gururangan et al., 2018; Gupta et al., 2021; Al-Negheimish et al., 2021; Sugawara et al., 2018; Jia and Liang, 2017; McCoy et al., 2019). Third, we assume that our setting is the necessary condition for solving math word problems. It is unreasonable to expect that a model that can't solve this pure numerical reasoning task can solve more complex tasks.

In total, we prepared 5K instances for training and 2.4K for testing.

2.1 Output strategies

We compared three configurations: all-at-once, step-by-step, and token-by-token (Figure 2a).

All-at-once: The model outputs the entire reasoning chain and the final answer in a single call (i.e., *chain-of-thought* style) (Wei et al., 2022; Cobbe et al., 2021; Yavuz et al., 2022; Schwartz et al., 2020). In this setting, the more reasoning steps, the longer the sequence the decoder must generate at once.

Step-by-step: The model outputs a single reasoning step per call. Each generated step is concatenated to the past input, and the model again generates the next step (i.e., *proofwriter* style) (Liang et al., 2021; Sanyal et al., 2022; Picco et al., 2021; Tafjord et al., 2021; Shwartz et al., 2020). This process is iterated until the model outputs the answer or until a set maximum number of iterations is reached (100). **Token-by-token:** This is the same as step-by-step chaining, but the decoder outputs only a single *token* per call. We set the maximum number of steps to 500.

Comparing *all-at-once* and the others reveals the effect of changing the sequence length that the decoder outputs in a single call. In addition, comparing *step-by-step* and *token-by-token* quantifies the advantage of breaking a problem into meaningful units.

2.2 Chaining strategies

Particular variables sometimes depend on another variable; the key to reaching the correct answer is determining the order in which the equations are referred to. Regarding existing studies, we compared three chaining strategies: *shortest-path*, *exhaustive*, and *backward* chaining (Figure 2b).

Shortest-path chaining: The model straightforwardly solves the equations starting from the first solvable one (i.e., involving a known value) and ending with the target (Wei et al., 2022; Cobbe et al., 2021; Yavuz et al., 2022; Shwartz et al., 2020). Here, the reasoning behind determining the shortest path is not outputted by the model.

Exhaustive chaining: The model greedily solves all given equations until the target value is reached (Tafjord et al., 2021; Liang et al., 2021; Yang et al., 2022). Specifically, the model calculates the left-most solvable equation in each step. Note that this strategy typically derives a long reasoning chain; from an engineering perspective, this strategy is inefficient.

Backward chaining: The model starts from the equation for the target variable and backtracks over the dependent equations until it reaches a known value (Picco et al., 2021; Rocktäschel and Riedel, 2017; Cingillioglu and Russo, 2019). Then, it solves each equation in order by inserting known or calculated values until the target one is reached.

No chaining: As a baseline, we also examined the setting where the model was trained to directly output the answer.

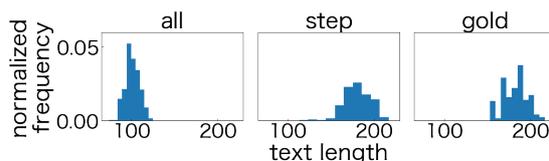


Figure 3: Distributions of the total reasoning chain length (num. characters). The all-at-once and step-by-step generate those at depth 12.

3 Results

Models: We used the pre-trained T5-base, T5-large² (Raffel et al., 2020), and BART-base³ (Lewis et al., 2020). Results of BART-base are in Appendix C.

Note that their pre-defined tokenizers have all the numbers from 0 to 9, and the numerical values in our dataset are divided into digits (e.g., “12” should be “@@1 @@2”) in advance, following Kim et al. (2021).

Training: The models were first pre-trained using a 10K *simple* dataset for 30 epochs, then trained with the 5K training set (1K training instances for each reasoning depth.) for 2000 epochs. The experiment setting details are in Appendix A. In addition, we prepared 0.2K test instances for each reasoning depth. This pre-training is intended to teach the models primitive operations (i.e., assignment, reference, and addition). The pre-training dataset contains two types of single-depth instances: *assign-refer* type (e.g., A=1, A?) and *operate-assign-refer* type (e.g., A=1+3, A?). All the results in the paper are averages of the results on three different seeds.

3.1 Output strategies

We compared the output strategies while fixing the chaining strategy to the shortest path. Figure 4a shows the accuracy per reasoning depth. Note that the accuracy score here denotes whether the answer (e.g., C=6) is correct. We observed the following: (i) **generating intermediate reasoning steps enhance the performance**, and (ii) among the output strategies, **step-by-step works the best**, and **all-at-once works the worst**. The format of the dataset in this study is simple. Therefore, this result indicates the low symbolic reasoning ability of neural models and the necessity of the choice of

²https://huggingface.co/docs/transformers/model_doc/T5

³https://huggingface.co/docs/transformers/model_doc/bart

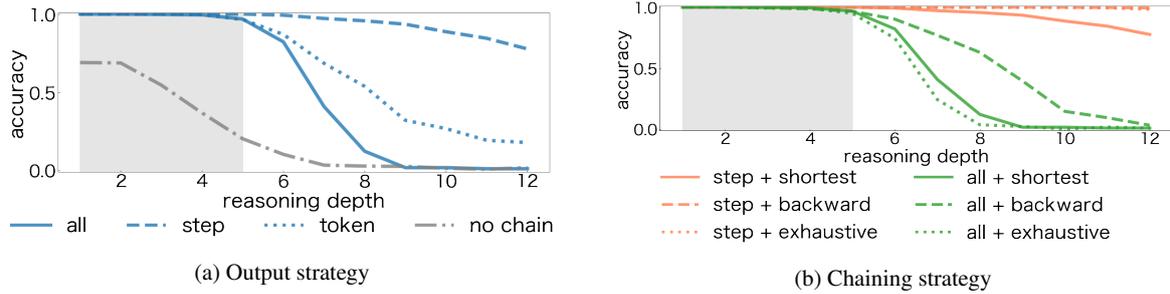


Figure 4: Accuracy changes of the models against reasoning depth. The gray range represents the training data domain (1-5 depth). Figure (4a) shows the performance degradation with the increase of reasoning steps when using the all-at-once strategy. Figure (4b) shows that the combination of step-by-step output and backward/exhaustive chaining leads to successful generalization.

Depth	Shortest	Backward	Exhaustive
6	99.3/99.3	100/ 100	99.7/99.7
8	95.5/95.7	100/ 100	99.8/99.8
12	76.7/77.7	99.5/99.5	98.2/98.3

Table 1: Accuracy of the T5-base model with the step-by-step output strategy at each depth (chain/answer).

Question: $A=1$, $B=2+A$, $B?$		
Error types	Gold	Prediction
Copying error	$B=2+A$, $B=2+1$, $B=3$	$B=6+A$, $B=6+1$, $B=7$
Hasty assignment	$B=2+A$, $B=2+1$, $B=3$	(skip step) $B=2+2$, $B=4$

Table 2: Illustrative examples of the errors under the step-by-step, shortest-path chaining settings. (skip step) denotes that the reasoning steps is accidentally skipped.

an appropriate reasoning strategy.

We hypothesized that the source of all-at-once’s inferiority was that the decoder overfitted to output a similar length of reasoning steps as those in the (shallower) training data. In fact, the models generated relatively shorter reasoning steps in the out-of-domain (e.g., depth of 12) setting when using the all-at-once strategy (Figure 3); this supports our hypothesis.

The advantage of step-by-step over token-by-token suggests the advantage of breaking the problem into meaningful units (reasoning step) and modeling each step in a single call of the encoder-decoder.

3.2 Chaining strategies

Figure 4b and Table 1 show the results on each depth with a fixed step-by-step output strategy.

Note that the accuracy of the chain (left side of the scores) was measured based on not an exact match but mathematically. For example, even if the order of generated equations is different, it is correct. The results of a fixed token-by-token output strategy are in Appendix B.

While the performance dropped in the shortest-path setting as the reasoning depth increased, with either the exhaustive or backward chaining, models successfully solved the task even when extrapolating to depths 6-12. The models correctly generated the intermediate steps (nearly perfect) as well as the final answer in the exhaustive and backward chaining settings (Table 1). Note that these strategies were ineffective with all-at-once outputting.

Gontier et al. (2020) compared chaining strategies and concluded that models that *didn’t* generate reasoning steps had better generalization performance than models that did when the reasoning chains were long. However, our results suggest that the choice of the appropriate output strategy improves the reasoning ability of the model.

We considered that the source of shortest-path inferiority was the rough granularity of the given reasoning steps. The models don’t know the shortest path before outputting the reasoning steps. Therefore, both the exhaustive and shortest path chaining approaches must search for variables other than those on the shortest path. As shown in Figure 2b, the exhaustive chaining approach is taught this process explicitly. On the other hand, the shortest-path chaining approach must be learned that by training data that don’t include this process. We thought this difference affected the accuracy and concluded that **the accuracy is higher when the granularity of given intermediate steps is**

finer, even though they are long.

Therefore, we concluded that **the accuracy is higher when the granularity of intermediate steps is finer**, even though they are long.

3.3 Error analysis

We also analyzed the errors of the depth-12 instances under the shortest-path strategy.⁴ We observed two types of errors: (i) copying errors and (ii) hasty assignment. Table 2 shows an illustrative example of each error type and the percentage of these errors. The most frequent one (53%) was a simple copying error, where the model failed to accurately copy an original equation into the reasoning chain. This erroneous copying ability is consistent with Xu et al. (2020) and supports the advantage of introducing a copy mechanism to the model (Ontanon et al., 2022). Second, a hasty assignment is the model skipping the step to copy the equation from context and instead assigned it a random value. Note that these errors were almost addressed in the other strategies; this could stem from the difficulty of the implicit calculation of the shortest path.

3.4 Models' scalability

To investigate the scalability, we compared T5-large with T5-base. Figure 5 shows the result. T5-large had a similar trend but slightly lower accuracy on all-at-once and step-by-step compared to T5-base. The reason may be that T5-large needs more data for updating the weights of the entire model. On the other hand, the accuracy of T5-large is higher than T5-base on token-by-token. It's because the data size of token-by-token is as token lengths of output sequence times as the data size of all-at-once, as shown in Figure 2a. This result indicates that the parameter size of the model needs to be larger to output token-by-token.

4 Conclusions

We investigated and factorized the reasoning strategy in symbolic numerical reasoning with neural seq2seq models. We found that the combination of step-by-step output and finely granular reasoning leads to successfully performing symbolic reasoning. Our results support the potential of neural models for symbolic reasoning.

⁴In total, 32 instances were analyzed. That is the total number of incorrect answers on one seed.

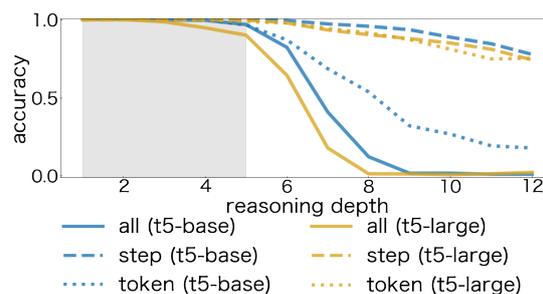


Figure 5: Accuracy changes of the T5-base and T5-large against reasoning depth. The gray range presents the training data domain (1-5 depth). This figure shows that the accuracy of T5-large with token-by-token is higher.

Limitations

We found that even simple symbolic reasoning requires the appropriate selection of reasoning strategy. It is unclear whether our findings generalize to more complex symbolic reasoning and/or problems written in natural language. If our findings do not generalize in these different settings, we must address the gap in future work. For example, we start with one of the simplest tasks and find out when models fail as we add complexity to tasks one by one.

From the engineering perspective, the iterative strategies are limited to the input length of the model. For example, in our experiments, when adopting the setting where reasoning depths are greater than 13, the input length of step-by-step and token-by-token became longer than the input length limit of T5 (i.e., 512 tokens).

In addition, gigantic language models (e.g., GPT-3) have recently been used. Including these models in our study is one of our future works.

Acknowledgements

We thank four anonymous reviewers who provided valuable feedback. This work was supported by JSPS KAKENHI Grant Numbers JP22H00524, 21K21343 and JST CREST Grant Number JP-MJCR20D2, Japan.

References

Hadeel Al-Negheimish, Pranava Madhyastha, and Alessandra Russo. 2021. [Numerical reasoning in machine reading comprehension tasks: are we there yet?](#) In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*,

- pages 9643–9649. Association for Computational Linguistics.
- Nuri Cingilloglu and Alessandra Russo. 2019. **Deep-logic: Towards end-to-end differentiable logical reasoning**. In *Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019) Stanford University, Palo Alto, California, USA, March 25-27, 2019., Stanford University, Palo Alto, California, USA, March 25-27, 2019*, volume 2350 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. **Training verifiers to solve math word problems**. *CoRR*, abs/2110.14168.
- Artur d’Avila Garcez and Luís C. Lamb. 2020. **Neurosymbolic AI: the 3rd wave**. *CoRR*, abs/2012.05876.
- Nicolas Gontier, Koustuv Sinha, Siva Reddy, and Christopher Pal. 2020. **Measuring systematic generalization in neural proof generation with transformers**. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. **BERT & family eat word salad: Experiments with text understanding**. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12946–12954. AAAI Press.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. **Annotation artifacts in natural language inference data**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 107–112. Association for Computational Linguistics.
- Kyle Hamilton, Aparna Nayak, Bojan Bozic, and Luca Longo. 2022. **Is neuro-symbolic AI meeting its promise in natural language processing? A structured review**. *CoRR*, abs/2202.12205.
- Robin Jia and Percy Liang. 2017. **Adversarial examples for evaluating reading comprehension systems**. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2021–2031. Association for Computational Linguistics.
- Jeonghwan Kim, Giwon Hong, Kyung-min Kim, Junmo Kang, and Sung-Hyon Myaeng. 2021. **Have you seen that number? investigating extrapolation in question answering models**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7031–7037. Association for Computational Linguistics.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. **Large language models are zero-shot reasoners**. *CoRR*, abs/2205.11916.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. **BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. **Solving quantitative reasoning problems with language models**. *CoRR*, abs/2206.14858.
- Zhengzhong Liang, Steven Bethard, and Mihai Surdeanu. 2021. **Explainable multi-hop verbal reasoning through internal monologue**. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1225–1250. Association for Computational Linguistics.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. **Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference**. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3428–3448. Association for Computational Linguistics.
- Maxwell I. Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. **Show your work: Scratchpads for intermediate computation with language models**. *CoRR*, abs/2112.00114.
- Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. 2022. **Making transformers solve compositional tasks**. In *Proceedings of the 60th Annual Meeting of the Association for Computational*

- Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland. Association for Computational Linguistics.
- Gabriele Picco, Thanh Lam Hoang, Marco Luca Sbodio, and Vanessa López. 2021. [Neural unification for logic reasoning over natural language](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3939–3950. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Gabriel Recchia. 2021. [Teaching autoregressive language models complex tasks by demonstration](#). *CoRR*, abs/2109.02102.
- Tim Rocktäschel and Sebastian Riedel. 2017. [End-to-end differentiable proving](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3788–3800.
- Soumya Sanyal, Harman Singh, and Xiang Ren. 2022. [Fairr: Faithful and robust deductive reasoning over natural language](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1075–1093. Association for Computational Linguistics.
- Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. [Unsupervised commonsense question answering with self-talk](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4615–4629. Association for Computational Linguistics.
- Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. [What makes reading comprehension questions easier?](#) In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4208–4219. Association for Computational Linguistics.
- Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. [Proofwriter: Generating implications, proofs, and abductive statements over natural language](#). In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3621–3634. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, and Denny Zhou. 2022. [Self-consistency improves chain of thought reasoning in language models](#). *CoRR*, abs/2203.11171.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). *CoRR*, abs/2201.11903.
- Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. [Self-attention guided copy mechanism for abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1355–1362. Association for Computational Linguistics.
- Kaiyu Yang, Jia Deng, and Danqi Chen. 2022. [Generating natural language proofs with verifier-guided search](#). *CoRR*, abs/2205.12443.
- Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, Nitish Shirish Keskar, and Caiming Xiong. 2022. [Modeling multi-hop question answering as single sequence prediction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 974–990. Association for Computational Linguistics.

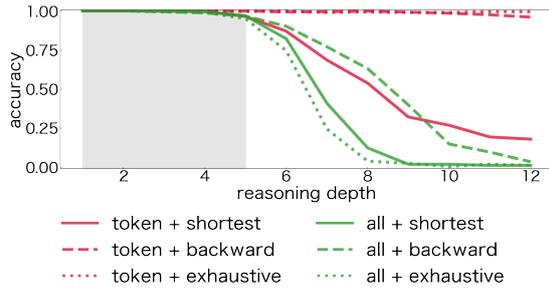


Figure 6: Accuracy changes of token-by-token per reasoning depth. The gray range presents the training data domain (depths 1-5).

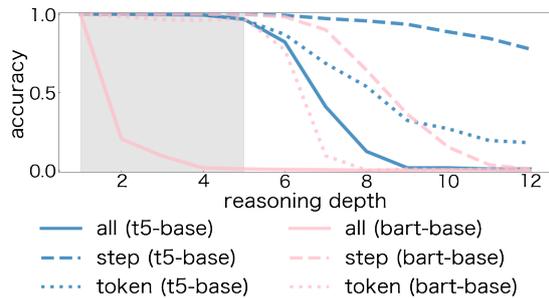


Figure 7: Accuracy changes of the T5-base and BART-base models per reasoning depth. The gray range presents the training data domain (depths 1-5). T5 seems to outperform BART.

A Details on Experimental Settings

We first examined the learning rate from 10^{-3} , 10^{-4} , and 10^{-5} ; among them, we used the largest rate at which the loss converged. After training models, we used the model with the lowest validation loss among the per-epoch checkpoints during the training reported. We used four NVIDIA V100 GPUs for NVLink 16GiB HBM2.

B Results of Token-by-token

Figure 6 shows the results on each depth with a fixed token-by-token output strategy. Like step-by-step, the performance drops in the shortest-path setting as the reasoning depth increases. In addition, the exhaustive or backward successfully solves the task even when extrapolating to depths 6-12.

C Different Architectures

We also tested BART-base (Lewis et al., 2020) as a baseline to investigate the effectiveness of the NLP-task-oriented objectives used in the T5-style pre-training. Figure 7 shows this result. In this

particular setting, T5 was superior to BART. This suggests that the NLP-task-oriented objectives benefit symbolic reasoning.

D Other errors

We analyzed the cases where the answer is correct and the chain is wrong. Table 3 shows examples of chain errors. Ignoring the incorrect step is an example of the model outputting the correct reasoning step after outputting an incorrect one. Correct assignment is an example in which the assignment accidentally makes the model output the correct step. Finally, Non-affecting error is an example in which a variable not on the shortest path is wrongly assigned a value.

Question: $A=1$, $C=5+B$, $B=2+A$, $D=3+A$, $C?$

Chain error types	Gold	Prediction
Ignoring the incorrect step	$A=1$, $B=2+A$, $B=2+1$, $B=3$, $C=5+B$, $C=5+3$, $C=8$	$A=1$, $B=2+D$, $B=2+A$, $B=2+1$, $B=3$, $C=5+B$, $C=5+3$, $C=8$
Correct assignment	$A=1$, $B=2+A$, $B=2+1$, $B=3$, $C=5+B$, $C=5+3$, $C=8$	$A=1$, $B=2+D$, $B=2+1$, $B=3$, $C=5+B$, $C=5+3$, $C=8$
Non affecting error	$A=1$, $B=2+A$, $B=2+1$, $B=3$, $C=5+B$, $C=5+3$, $C=8$	$A=1$, $B=2+A$, $B=2+1$, $B=3$, $D=3+A$, $D=3+2$, $D=5$, $C=5+B$, $C=5+3$, $C=8$

Table 3: These instances are examples of chain errors. Note that the final answers are correct.

Analyzing the Effectiveness of the Underlying Reasoning Tasks in Multi-hop Question Answering

Xanh Ho,^{*} ^{◇,♡} Anh-Khoa Duong Nguyen,^{*} [♣] Saku Sugawara,[♡] and Akiko Aizawa^{◇,♡,♣}

[◇]The Graduate University for Advanced Studies, Kanagawa, Japan

[♡]National Institute of Informatics, Tokyo, Japan

[♣]Independent Researcher

[♠]The University of Tokyo, Tokyo, Japan

{xanh, saku, aizawa}@nii.ac.jp

dnanhkhoa@live.com

Abstract

To explain the predicted answers and evaluate the reasoning abilities of models, several studies have utilized underlying reasoning (UR) tasks in multi-hop question answering (QA) datasets. However, it remains an open question as to how effective UR tasks are for the QA task when training models on both tasks in an end-to-end manner. In this study, we address this question by analyzing the effectiveness of UR tasks (including both sentence-level and entity-level tasks) in three aspects: (1) QA performance, (2) reasoning shortcuts, and (3) robustness. While the previous models have not been explicitly trained on an entity-level reasoning prediction task, we build a multi-task model that performs three tasks together: sentence-level supporting facts prediction, entity-level reasoning prediction, and answer prediction. Experimental results on 2WikiMultiHopQA and HotpotQA-small datasets reveal that (1) UR tasks can improve QA performance. Using four debiased datasets that are newly created, we demonstrate that (2) UR tasks are helpful in preventing reasoning shortcuts in the multi-hop QA task. However, we find that (3) UR tasks do not contribute to improving the robustness of the model on adversarial questions, such as sub-questions and inverted questions. We encourage future studies to investigate the effectiveness of entity-level reasoning in the form of natural language questions (e.g., sub-question forms).¹

1 Introduction

The task of multi-hop question answering (QA) requires a model to read and aggregate information from multiple paragraphs to answer a given question (Figure 1a). Several multi-hop QA datasets have been proposed, such as QAngaroo (Welbl et al., 2018), HotpotQA (Yang et al., 2018), and

MuSiQue (Trivedi et al., 2022). In HotpotQA, the authors provide sentence-level supporting facts (SFs) to test the reasoning ability and explainability of the models. However, owing to the design of the sentence-level SFs task (binary classification) and the redundant information in the sentences, Inoue et al. (2020) and Ho et al. (2020) show that the sentence-level SFs are insufficient to explain and evaluate multi-hop models in detail. To address this issue, R⁴C (Inoue et al., 2020) and 2WikiMultiHopQA (2Wiki; Ho et al., 2020) datasets provide an entity-level reasoning prediction task to explain and evaluate the process of answering questions. Entity-level reasoning information is defined as a set of triples that describes the reasoning path from question to answer (Figure 1b).

Several previous studies (Chen et al., 2019; Fu et al., 2021a) utilize sentence-level SFs and/or entity-level reasoning information to build explainable models by using question decomposition (Min et al., 2019b; Perez et al., 2020) or predicting sentence-level SFs. The advantages of these pipeline models are that they can exploit the underlying reasoning (UR) process in QA and their predicted answers are more interpretable. However, the question remains as to how effective training on UR tasks is for the QA task in an end-to-end manner. Although a few end-to-end models have also been introduced (Qiu et al., 2019; Fang et al., 2020), these models are not explicitly trained on entity-level and answer prediction tasks.

In addition to the triple form, the sub-question form is another way to utilize entity-level reasoning information. Specifically, Tang et al. (2021) utilize question decomposition as an additional sub-question evaluation for bridge questions (there are two types of questions: bridge and comparison) in HotpotQA. They only use sub-questions for evaluation and do not fine-tune the models on them. In addition, Ho et al. (2022) use sub-questions for both evaluation and training. However, they only

^{*}Equal contribution.

¹Our data and code are available at <https://github.com/Alab-NII/multi-hop-analysis>

<p>Question: Who is the paternal grandfather of Joan of Valois, Countess of Beaumont?</p> <p>Paragraph A: Joan of Valois, Countess of Beaumont</p> <p>[1] Joan of Valois (1304 – 1363) was the daughter of Charles of Valois and his second wife ...</p> <p>Paragraph B: Charles, Count of Valois</p> <p>[2] Charles of Valois (12 March 1270 – 16 December 1325), the third son of Philip III of France and, ... [3] ...</p> <p>Answer: Philip III of France</p>	<p>Sentence-level supporting facts: 1, 2</p> <p>Entity-level reasoning prediction (Evidence):</p> <p>Step 1: ("Joan of Valois, Countess of Beaumont", "father", "Charles of Valois") &</p> <p>Step 2: ("Charles of Valois", "father", "Philip III of France")</p> <p>⊕ QA Performance ⊖ Robustness</p> <p>⊕ Reasoning Shortcuts</p>	<p>Paragraph A: Joan of Valois, Countess of Beaumont</p> <p>[1] We can also establish the global weak solution ... [2] Joan of Valois (1304 – 1363) was the daughter of Charles of Valois and his second wife ...</p> <p>Paragraph B: Charles, Count of Valois</p> <p>[3] This gives a clear impulse to develop ... [4] Charles of Valois (12 March 1270 – 16 December 1325), the third son of Philip III of France and, ... [5] ...</p> <p>Adversarial Question: Who is the father of Joan of Valois, Countess of Beaumont?</p>
a) Standard QA task format	b) UR tasks and three aspects	c) Debiased and Adversarial examples

Figure 1: Example of (a) a standard multi-hop question, (b) two underlying reasoning tasks in the QA process and three aspects in our analysis, ‘+’ and ‘-’ indicate that the UR tasks have a positive and negative impacts, respectively, and (c) debiased and adversarial examples that are used in our study.

focus on comparison questions for date information. In contrast, we focus on the triple form of the entity-level information and conduct experiments using two datasets, 2Wiki and HotpotQA-small (obtained by combining HotpotQA and R⁴C), which include both types of questions.

In this study, we analyze the effectiveness of UR tasks (including both sentence-level and entity-level) in three aspects: (1) *QA performance*, (2) *reasoning shortcuts*, and (3) *robustness*. First, QA performance is the final objective of the QA task. We aim to answer the following question: **(RQ1)** *Can the UR tasks improve QA performance?* For the second aspect, previous studies (Chen and Durrett, 2019; Jiang and Bansal, 2019a; Min et al., 2019a; Trivedi et al., 2020) demonstrate that many questions in the multi-hop QA task contain biases and reasoning shortcuts (Geirhos et al., 2020), where the models can answer the questions by using heuristics. Therefore, we aim to ask the following: **(RQ2)** *Can the UR tasks prevent reasoning shortcuts?* For the final aspect, to ensure safe development of NLP models, robustness is one of the important issues and has gained tremendous amount of research (Wang et al., 2022). In this study, we aim to test the robustness of the model by asking modified versions of questions, such as sub-questions and inverted questions. Our question is **(RQ3)** *Do the UR tasks make the models more robust?*

There are no existing end-to-end models that can perform three tasks simultaneously (sentence-level SFs prediction, entity-level prediction, and answer prediction); therefore, we first build a multi-hop BigBird-base model (Zaheer et al., 2020) to perform these three tasks simultaneously. We then evaluate our model using two multi-hop datasets: 2Wiki and HotpotQA-small. To investigate the ef-

fectiveness of the UR tasks, for each dataset, we conduct three additional experiments in which the model is trained on: (1) answer prediction task, (2) answer prediction and sentence-level prediction tasks, and (3) answer prediction and entity-level prediction tasks. We also create four debiased sets (Figure 1c) for 2Wiki and HotpotQA-small for **RQ2**. We create and reuse adversarial questions for 2Wiki and HotpotQA-small for **RQ3**.

The experimental results indicate that the UR tasks can improve QA performance from 77.9 to 79.4 F1 for 2Wiki and from 66.4 to 69.4 F1 for HotpotQA-small (**RQ1**). The results of the models on the four debiased sets reveal that the UR tasks can be used to reduce reasoning shortcuts (**RQ2**). Specifically, when the model is trained on both answer prediction and UR tasks, the performance drop of the model on the debiased sets is lower than that when the model is trained only on answer prediction (e.g., 8.9% vs. 13.4% EM). The results also suggest that the UR tasks do not make the model more robust on adversarial questions, such as sub-questions and inverted questions (**RQ3**). Our analysis shows that correct reconstruction of the entity-level reasoning task contributes to finding the correct answer in only 37.5% of cases. This implies that using entity-level reasoning information in the form of triples does not answer adversarial questions, in this case, the sub-questions. We encourage future work to discover the effectiveness of the entity-level reasoning task in the form of sub-questions that have the same form as multi-hop QA questions.

2 Background

Reasoning Tasks in Multi-hop QA In this study, we consider UR tasks in multi-hop QA including two levels: *sentence-level* and *entity-level*. The

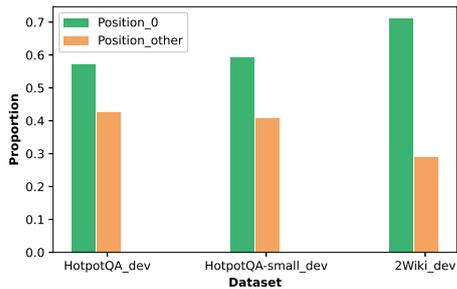


Figure 2: Information on the position of sentence-level SFs in the dev. sets of the three datasets.

sentence-level SFs prediction task was first introduced by Yang et al. (2018). This task requires a model to predict a set of sentences that is necessary to answer a question (Figure 1).

To evaluate the UR process of the models, derivation and evidence information were introduced in R⁴C and 2Wiki, respectively. Both derivation and evidence are sets of triples that represent the reasoning path from question to answer. The difference is the form; derivation in R⁴C uses a semi-structured natural language form, whereas evidence in 2Wiki uses a structured form. We conduct experiments with both R⁴C (HotpotQA-small) and 2Wiki. For consistency, we use the term *entity-level reasoning prediction task* to denote the derivation task in R⁴C and the evidence task in 2Wiki.

Reasoning Shortcuts and Biases In this study, we consider both reasoning shortcuts and biases to be similar. These are spurious correlations in the dataset that allow a model to answer the question correctly without performing the expected reasoning skills, such as comparison and multi-hop reasoning. Following previous studies (Jiang and Bansal, 2019a; Ko et al., 2020), we use the terms *word overlap shortcut* and *position bias*.

To check whether the UR tasks can prevent reasoning shortcuts, we first identify the types of shortcuts that exist in HotpotQA-small and 2Wiki. We use heuristics to identify the word overlap shortcut (Appendix A). We find that the word overlap shortcut is common in HotpotQA-small, but not in 2Wiki. The small sample size of HotpotQA-small (Section 4) increases the uncertainty of the obtained results. Therefore, within the scope of this study, we mainly experiment with position bias.

We observe that many examples in 2Wiki contain answers in the first sentence. Therefore, we divide every sentence-level SF in each gold para-

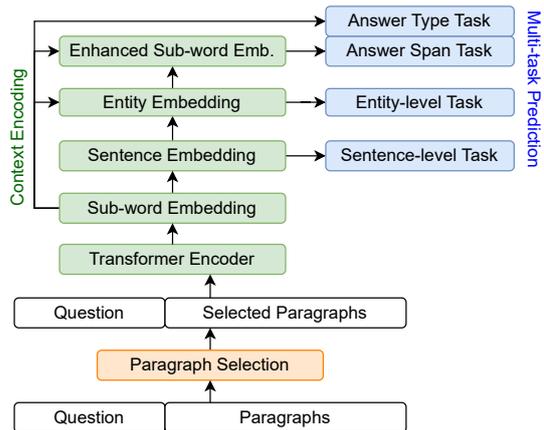


Figure 3: Our model has three main steps: paragraph selection, context encoding, and multi-task prediction.

graph into two levels: the first sentence (position_0) and the remaining sentences (position_other). Subsequently, we obtain the percentage of each level by dividing the total number of each level (e.g., position_0) by the total number of SFs. Figure 2 illustrates the information on the position of sentence-level SFs in dev. sets of three datasets. We find that all three datasets have a bias toward the first sentence. We also find that 2Wiki has more position biases than HotpotQA and HotpotQA-small.

3 Our Multi-task Model

To investigate the usefulness of UR tasks for the QA task, we jointly train the corresponding tasks: sentence-level SFs prediction, entity-level prediction, and answer prediction. Figure 3 illustrates our model. To handle long texts, we use the Big-Bird model (Zaheer et al., 2020), which is available in Hugging Face’s transformers repository.² Our model comprises three main steps: (1) paragraph selection, (2) context encoding, and (3) multi-task prediction. We use the named entity recognition (NER) models of Spacy³ and Flair (Akbik et al., 2019) to extract all entities in the context and use them for the entity-level prediction task.

Paragraph Selection Following previous models (Qiu et al., 2019; Fang et al., 2020; Tu et al., 2020), instead of using all the provided paragraphs, we first filter out answer-unrelated paragraphs. We follow the paragraph selection process described in Fang et al. (2020). First, we retrieve first-hop

²https://huggingface.co/transformers/model_doc/bigbird.html

³<https://spacy.io/>

paragraphs by using title matching or entity matching. We then retrieve second-hop paragraphs using the hyperlink information available in Wikipedia. When we retrieve paragraphs, we reuse a paragraph ranker model⁴ from the hierarchical graph network (HGN) model (Fang et al., 2020) to rank input paragraphs using the probability of whether they contain sentence-level SFs.

Context Encoding To obtain vector representations for sentences and entities, we first combine all the selected paragraphs into one long paragraph and then concatenate it with the question to form a context C . Specifically, $C = [[\text{CLS}], q_1, \dots, q_m, [\text{SEP}], p_1, \dots, p_n, [\text{SEP}]]$, where m and n are the lengths of the question q and the combined paragraph p (all selected paragraphs), respectively. The context C is then tokenized into l sub-words before feeding into BigBird to obtain the contextual representation C' of the sub-words:

$$C' = \text{BigBird}(C) \in \mathbb{R}^{l \times h}, \quad (1)$$

where h is the hidden size of the BigBird model. Next, we obtain the representation $s_i \in \mathbb{R}^{2h}$ of the i -th sentence and the representation $e_j \in \mathbb{R}^{4h+d_t}$ of the j -th entity, as follows:

$$\begin{aligned} s_i &= C'_{S_{\text{start}}^i}; C'_{S_{\text{end}}^i} \\ e_j &= C'_{E_{\text{start}}^j}; C'_{E_{\text{end}}^j}; t_j; s_k, \end{aligned} \quad (2)$$

where $[\cdot]$ denotes the concatenation of the two vectors, $C'_{S_{\text{start}}^i}$ and $C'_{E_{\text{start}}^j}$ denote the first sub-word representations of the i -th sentence and j -th entity, respectively. $C'_{S_{\text{end}}^i}$ and $C'_{E_{\text{end}}^j}$ denote the last sub-word representations of the i -th sentence and j -th entity, respectively. We enrich the entity embedding e_j by concatenating it with a d_t -dimensional type embedding t_j and a sentence embedding s_k , where k is the index of the sentence containing the j -th entity.

We also leverage the entity information to improve the contextual representation of sub-words C' as it is mainly used for the answer prediction task, which will be described in the next section. Thus, the enhanced sub-word representation C''_i of the i -th sub-word is calculated as follows:

$$C''_i = C'_i; e_k \in \mathbb{R}^{5h+d_t}, \quad (3)$$

where e_k is the embedding of the k -th entity containing the i -th sub-word. Otherwise, e_k is a null vector with the same dimension.

⁴<https://github.com/yuwfan/HGN>

Multi-task Prediction After context encoding, we train our model on three main tasks together: (1) sentence-level prediction, (2) entity-level prediction, (3) and answer prediction. We split the answer prediction task into two sub-tasks, similar to previous studies (Yang et al., 2018; Fang et al., 2020), including answer type prediction and answer span prediction. We train our model by minimizing the joint loss for all tasks, as follows:

$$\begin{aligned} L_{\text{joint}} &= \lambda_{\text{sent}} L_{\text{sent}} + \lambda_{\text{ent}} L_{\text{ent}} + \\ &\lambda_{\text{ans}} (L_{\text{start}} + L_{\text{end}} + L_{\text{type}}), \end{aligned} \quad (4)$$

where λ_{sent} , λ_{ent} , and λ_{ans} are the hyper-parameters for three tasks: sentence-level prediction, entity-level prediction, and answer prediction (details are given in Appendix B.1).

For the sentence-level prediction task, we use a binary classifier to predict whether a sentence is a supporting fact. For the answer type prediction task, we use a 4-way classifier to predict the probabilities of *yes*, *no*, *span*, and *no answer*. Two linear classifiers are used for the answer span prediction task to independently predict the start and end tokens of the answer span.

Different from existing end-to-end models (Qiu et al., 2019; Fang et al., 2020), our model is explicitly trained on the entity-level prediction task. We formalize the entity-level reasoning prediction task as a relation extraction task (Zhang and Wang, 2015). The input is a pair of entities, and the output is the relationship between two entities. From all named entities obtained by using the NER models, we generate a set of entity pairs; for example, given N entities, we obtain $N \times (N - 1)$ pairs. For each pair, we predict a relationship in a set of predefined relationships obtained from the training set. We then use cross-entropy as the learning objective.

4 Datasets and Evaluation Metrics

We mainly experiment with 2Wiki and HotpotQA-small. We also train and evaluate our model on the full version of HotpotQA. We reuse and create debiased and adversarial sets for the evaluation. Table 1 presents the statistics for 2Wiki, HotpotQA-small, and additional evaluation sets. The details of HotpotQA and 2Wiki are presented in Appendix B.2. It should be noted that all datasets are in English.

4.1 HotpotQA-small

R⁴C (Inoue et al., 2020) is created by adding entity-level reasoning information to the samples

Split	2Wiki	HotpotQA-small
Train	167,454	3,671
Dev.	12,576	917
Test	12,576	-
Debiased	12,576 (x4)	917 (x4)
Adversarial	12,576	659 & 134

Table 1: Statistics for 2Wiki and HotpotQA-small. There are four debiased sets in 2Wiki and HotpotQA-small. There are one adversarial set in 2Wiki and two adversarial sets in HotpotQA-small.

in HotpotQA. We obtain HotpotQA-small by combining HotpotQA (Yang et al., 2018) with R⁴C. HotpotQA-small comprises three tasks as in 2Wiki: (1) sentence-level SFs prediction, (2) entity-level prediction, and (3) answer prediction. First, we re-split the ratio between the training and dev. sets; the new sizes are 3,671 and 917 for the training and dev. sets, respectively (the original sizes are 2,379 and 2,209, respectively). In R⁴C, there are three gold annotations for the entity-level prediction task; in 2Wiki, there is only one gold annotation. For consistency in the evaluation and analysis, we randomly choose one annotation from the three annotations for every sample in R⁴C.

The entity-level reasoning in R⁴C is created by crowdsourcing. We observe that there are many similar relations in the triples in R⁴C, and these relations can be grouped into one. For example, *is in*, *is located in*, *is in the*, and *is located in the* indicate location relation. We also group the relations by removing the context information in the relations; for example, *is a 2015 book by* and *is the second book by* are considered similar to the relation *is a book by*. After grouping, the number of relations in R⁴C is 2,526 (it is 4,791 before).

4.2 Debiased Dataset

The objective of our debiased dataset is to introduce a small perturbation in each paragraph to mitigate a specific type of bias, in our case, the position bias shown in Figure 2. For both 2Wiki and HotpotQA-small, we use the same method to generate four debiased sets: ADDUNRELATED, ADDRELATED, ADD2, and ADD2SWAP. The differences between these four sets are whether the sentence is related or unrelated to the paragraph and whether we add one or two sentences into the paragraph. The details of each set are as follows.

ADDUNRELATED: One sentence unrelated to the paragraph is added. In our experiment, we use a list of sentences in the sentence-level revision dataset (Tan and Lee, 2014). We randomly choose one sentence that has a number of tokens greater than eleven but less than twenty-one.

ADDERELATED: One sentence that does not have an impact on the meaning or flow of the paragraph is added. In our experiment, we write multiple templates for each entity type (e.g., for a film entity, “#Name is a nice film”, where #Name is the title of the paragraph), then randomly choose one template, and add it to the paragraph. To detect the type of the paragraph, we use the question type information in 2Wiki and HotpotQA-small, the results of the NER model, and the important keywords in the question (e.g., who, magazine, album, and film).

ADD2: ADDRELATED and ADDUNRELATED are combined in order.

ADD2SWAP: The order of ADDRELATED and ADDUNRELATED in ADD2 is swapped.

4.3 Adversarial Dataset

The objective of our adversarial dataset is to check the robustness of the model by asking modified versions of questions. For HotpotQA-small, we reuse two versions of adversarial examples in Geva et al. (2022). The first one is automatically generated by using the ‘Break, Perturb, Build’ (BPB) framework in Geva et al. (2022). The BPB framework performs three main steps: (1) breaking a question into multiple reasoning steps, (2) perturbing the reasoning steps by using a list of defined rules, and (3) building new QA samples from the perturbations in step #2. The second version is a subset of the first version and is validated by crowd workers. We only use the examples in these two versions that the original examples appear in HotpotQA-small.

For 2Wiki, no adversarial dataset is available. Based on the idea of the BPB framework in Geva et al. (2022), we apply two main rules from BPB for 2Wiki: (1) replace the comparison operation for comparison questions, and (2) use the prune step for bridge questions. For the first rule, we replace the operation in the comparison questions (e.g., “Who was born first, A or B?” is converted to “Who was born later, A or B?”). For the second rule, we use a sub-question in the QA process as the main question (e.g., for Figure 1, we ask, “Who is the father of Joan of Valois?”).

Dataset	Task Setting	Answer		Sentence-level		Entity-level	
		EM	F1	EM	F1	EM	F1
2Wiki	(1) Ans	72.03	77.87	-	-	-	-
	(2) Ans + Sent	72.82	78.65	78.06	92.38	-	-
	(3) Ans + Ent	72.33	78.21	-	-	46.11	76.65
	(4) Ans + Sent + Ent	73.60	79.37	78.46	92.68	45.97	76.69
HotpotQA-small	(1) Ans	52.89	66.43	-	-	-	-
	(2) Ans + Sent	54.42	69.03	75.35	91.00	-	-
	(3) Ans + Ent	54.74	69.08	-	-	6.54	31.31
	(4) Ans + Sent + Ent	54.74	69.44	75.14	90.88	6.43	31.05

Table 2: Ablation study results (%) of our model in the dev. sets of 2Wiki and HotpotQA-small. *Ans*, *Sent*, and *Ent* represent the answer prediction task, sentence-level SFs prediction task, and entity-level prediction task, respectively. ‘Task Setting’ represents the tasks that the model is trained on. ‘-’ indicates the tasks the model is not trained on.

4.4 Evaluation Metrics

Each task in HotpotQA and 2Wiki is evaluated by using two metrics: exact match (EM) and F1 score. Following the evaluation script in HotpotQA and 2Wiki, we use joint EM and joint F1 to evaluate the entire capacity of the model. For HotpotQA, they are the products of the scores of two tasks: sentence-level prediction and answer prediction. For 2Wiki and HotpotQA-small, they are the products of the scores of three tasks: sentence-level prediction, entity-level prediction, and answer prediction.

5 Results

Currently, there are no existing end-to-end models that explicitly train all three tasks together; therefore, in this study, we use our proposed model for analysis. We also compare our model with other previous models on the HotpotQA and 2Wiki datasets. In general, the experimental results indicate that our model is comparable to previous models and can be used for further analyses. We focus more on the analysis; therefore, the detailed results of the comparison are presented in Appendix B.3.

5.1 Effectiveness of the UR Tasks

To investigate the effectiveness of the UR tasks, we train the model in four settings: (1) answer prediction only, (2) answer prediction and sentence-level SFs prediction, (3) answer prediction and entity-level prediction, and (4) all three tasks together.

QA Performance (RQ1) Our first research question is whether the UR tasks can improve QA performance. To answer this question, we compare the

results of different task settings described above. The results are presented in Table 2. For 2Wiki, using sentence-level and entity-level separately (settings #2 and #3), the QA performance does not change significantly. The improvement is significant when we combine both the sentence-level and entity-level (setting #4). Specifically, the scores when the model is trained on the answer prediction task only (setting #1) and on both the answer prediction task and UR tasks (setting #4) are 77.9 and 79.4 F1, respectively. In contrast to 2Wiki, using sentence-level and entity-level separately, there is a larger QA performance improvement in HotpotQA-small. Specifically, the F1 scores of settings #2 and #3 are 69.0 and 69.1, respectively, whereas, the F1 score of the first setting is 66.4. Similar to 2Wiki, there is a large gap between the two settings, #1 and #4 (66.4 F1 and 69.4 F1, respectively).

In summary, these results indicate that both sentence-level and entity-level prediction tasks contribute to improving QA performance. These results align with the findings in Yang et al. (2018), which shows that incorporating the sentence-level SFs prediction task can improve QA performance. We also find that when combining both sentence-level and entity-level prediction tasks, the scores of the answer prediction task are the highest.

Reasoning Shortcuts (RQ2) To investigate whether explicitly optimizing the model on the UR tasks can prevent reasoning shortcuts, we evaluate the four settings of the model on the four debiased sets of 2Wiki and HotpotQA-small. The generation of the debiased sets includes stochastic steps. To minimize the impact of randomness on our re-

Dataset	Task Setting	Reduction (%) on Four Debiased Sets							
		ADDUNRELATED		ADDRELATED		ADD2		ADD2SWAP	
		EM	F1	EM	F1	EM	F1	EM	F1
2Wiki	(1) Ans	<u>13.40</u>	<u>12.13</u>	3.55	3.46	<u>12.32</u>	<u>11.72</u>	<u>18.99</u>	<u>17.51</u>
	(2) Ans + Sent	11.00	9.71	<u>4.16</u>	<u>4.22</u>	11.22	10.69	17.62	16.24
	(3) Ans + Ent	7.73	6.94	2.80	2.77	8.38	7.76	13.12	12.21
	(4) Ans + Sent + Ent	8.86	8.11	3.16	3.13	9.09	8.58	14.53	13.77
HotpotQA-small	(1) Ans	3.01	1.53	<u>4.04</u>	<u>1.50</u>	1.65	1.01	3.96	2.47
	(2) Ans + Sent	1.13	1.35	-0.51	0.19	0.08	0.85	1.77	1.96
	(3) Ans + Ent	<u>6.73</u>	<u>5.60</u>	-0.92	0.03	<u>4.02</u>	<u>3.54</u>	<u>6.89</u>	<u>5.46</u>
	(4) Ans + Sent + Ent	5.05	4.65	1.26	1.25	1.83	2.46	3.58	3.64

Table 3: Average performance drop from five times running (smaller is better) of the four settings on the four debiased sets of 2Wiki and HotpotQA-small. The best and worst scores are boldfaced and underlined, respectively.

ported results, we generate the debiased sets five times and report the average evaluation scores. The average performance drops are presented in Table 3 (detailed scores are given in Appendix B.4).

Overall, for 2Wiki, when the model is trained on only one task (#1), the drop is the largest (except for ADDRELATED, which is the second largest). When the model is trained only on the answer prediction task, the drops are always higher than those when the model is trained on three tasks. Specifically, the gaps between the two settings, #1 (only answer task) and #4 (all three tasks), are 4.5%, 0.4%, 3.2%, 4.5% (EM score) for ADDUNRELATED, ADDRELATED, ADD2, and ADD2SWAP, respectively. These scores indicate that the two tasks, sentence-level and entity-level, positively affect the answer prediction task when the model is trained on three tasks simultaneously.

For HotpotQA-small, we observe that the effectiveness of the UR tasks is inconsistent. For example, for ADDUNRELATED, when training the model on the three tasks (setting #4), the reduction is larger than that when training on answer task only (setting #1) (5.1 vs. 3.0 EM). However, for ADDRELATED, the reduction on setting #4 is smaller than that on setting #1 (1.3 vs. 4.0 EM). One possible reason is that the performance of the entity-level task is not good (6.4 EM), which affects the answer prediction task when the model is trained on the three tasks together. Another possible reason is that the position bias in HotpotQA-small is not sufficiently large. We present a detailed analysis in Section 5.2 to explain this case.

Robustness (RQ3) To test whether the UR tasks can help to improve the robustness of the model,

Task Setting	Dev-adver		Reduction %	
	EM	F1	EM	F1
Ans	37.09	46.07	48.51	40.84
Ans + Sent	34.26	43.64	52.95	44.51
Ans + Ent	32.67	39.43	54.83	49.58
Ans + Sent + Ent	34.19	42.74	53.55	46.15

Table 4: Results of our model in the dev-adversarial set of 2Wiki and the performance drop.

we evaluate the four settings of the model on the adversarial sets. For 2Wiki, the results are presented in Table 4. The scores for all four settings decrease significantly on the adversarial set. The reduction is the smallest when the model is trained on the answer task only. The UR tasks do not make the model more robust on this adversarial set. For HotpotQA-small, we observe the same behavior, that is, when the model is trained on the answer task only, the reduction is the smallest. All results are presented in Table 5. These results indicate that both sentence-level and entity-level prediction tasks do not contribute to improving the robustness of the models on adversarial questions, such as sub-questions and inverted questions. We analyze the results in Section 5.2.

5.2 Analyses

Details of RQ2 To investigate the results concerning RQ2 in more depth, we first analyze the position biases of different types of questions in 2Wiki and HotpotQA-small. We find that the comparison questions have more position biases than the bridge questions in both 2Wiki and HotpotQA-

Task Setting	Dev		Dev-Adver		Adver↓ (%)		Dev-Adver-val		Adver-val↓ (%)	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
(1) Ans	52.89	66.43	40.36	51.23	23.69	22.88	37.31	46.69	29.46	29.72
(2) Ans + Sent	54.42	69.03	41.73	52.50	23.32	23.95	34.33	43.86	36.92	36.46
(3) Ans + Ent	54.74	69.08	42.79	52.16	21.83	24.49	27.61	36.86	49.56	46.64
(4) Ans + Sent + Ent	54.74	69.44	40.52	51.14	25.98	26.35	31.34	38.22	42.75	44.96

Table 5: Results of our model in the dev. and two dev-adversarial sets of HotpotQA-small. ‘Adver’ denotes adversarial and ‘Adver-val’ denotes the adversarial set that was validated by crowd workers.

Task Setting	Correct Ans	Correct Ent	Correct Both Ans & Ent
(3) Ans + Ent	4,109	6,851	2,249 (32.8%)
(4) Ans + Sent + Ent	4,300	6,450	2,420 (37.5%)

Table 6: Number of correct predicted answers, number of correct predicted entity-level reasoning, and number of examples that have both correct predicted answers and correct predicted entity-level reasoning.

small (Appendix B.5). To evaluate the effectiveness of the position bias for each type of question, we evaluate the four settings of the model on the four debiased sets for each type of question in both datasets. All the results are presented in Appendix B.5.

For 2Wiki, we find that most of the answers are in the first sentences in the comparison questions. This large bias is the main reason for the significant reduction in the scores in the comparison questions. 2Wiki has 46.0% of comparison questions. The reduction in comparison questions contributes to the reduction in the entire dataset. In other words, the results of 2Wiki are affected by those of the comparison questions. HotpotQA-small has only 22.0% of comparison questions, and the position bias in the comparison questions was not sufficiently large. Therefore, the position bias does not have a significant impact on the main QA task. In other words, the UR tasks do not have a significant effect.

Details of RQ3 The adversarial questions used in RQ3 are the sub-questions in the QA process for bridge questions and the inverted questions for comparison questions. We observe that the triple in the entity-level task is helpful in answering the sub-questions. For example, the triple is: (*Charles of Valois, father, Philip III of France*) and the sub-question is “Who is the father of Charles of Valois?”. To understand more on the behaviors of the model, we analyze the results from 2Wiki in two

settings: (3) Ans + Ent and (4) Ans + Sent + Ent. Table 6 presents the detailed results for these two settings. We find that correct reconstruction of the entity-level reasoning task contributes to finding the correct answer only in 32.8% of cases in setting #3 and only in 37.5% of cases in setting #4. Entity-level reasoning in the form of triples has no significant effect on the main QA process. Several examples are presented in Appendix B.5.

We conjecture that there are three possible reasons why the UR tasks cannot contribute to the adversarial dataset. The first one is the difference in the form and design of the tasks. Specifically, the entity-level reasoning task is formulated as a relation extraction task; the input is a pair of entities, and the output is a relation label. Meanwhile, the adversarial dataset is formulated as a QA task; the input is a natural language question, and the output is an answer. The second reason is the incompetence of the entity-level reasoning information. As discussed in Ho et al. (2022), the entity-level reasoning in the comparison questions does not describe the full path from question to answer, and other reasoning operations are required to obtain the answer. The final reason is the manner in which we utilize the entity-level reasoning information. Our model does not consider the order of the triples in the reasoning chain. For example, we do not consider the order of the two steps in Figure 1b. We hope that our research will inspire future studies to investigate the effectiveness of the UR tasks in the form of a natural language question, which has the same form as a multi-hop QA question.

6 Related Work

Multi-hop Datasets and Analyses To test the reasoning abilities of the models, many multi-hop QA datasets (Welbl et al., 2018; Talmor and Berant, 2018; Yang et al., 2018) have been proposed. Recently, Trivedi et al. (2022) introduced MuSiQue, a

multi-hop dataset constructed from a composition of single-hop questions. The reason why do we not conduct experiments on MuSiQue is explained in the limitations section.

In addition to Tang et al. (2021) and Ho et al. (2022), the most similar to our research mentioned in the Introduction, there are some other existing studies (Chen and Durrett, 2019; Jiang and Bansal, 2019a; Min et al., 2019a; Trivedi et al., 2020) on the analysis and investigation of the multi-hop datasets and models. However, most of them do not utilize the internal reasoning information when answering questions.

Multi-hop Models Various directions have been proposed for solving multi-hop datasets, including question decomposition (Talmor and Berant, 2018; Jiang and Bansal, 2019b; Min et al., 2019b; Perez et al., 2020; Wolfson et al., 2020; Fu et al., 2021a), iterative retrieval (Feldman and El-Yaniv, 2019; Asai et al., 2020; Qi et al., 2021), graph neural networks (Song et al., 2018; De Cao et al., 2019; Ding et al., 2019; Qiu et al., 2019; Tu et al., 2019; Fang et al., 2020), and other approaches such as single-hop based models (Yang et al., 2018; Nishida et al., 2019) or transformer-based models (Devlin et al., 2019; Zaheer et al., 2020). Our model is based on the BigBird transformer model.

Other QA Reasoning Datasets In addition to multi-hop reasoning datasets, several other existing datasets also aim to evaluate the reasoning abilities of the models. Some of them are: DROP (Dua et al., 2019) for numerical reasoning; CLUTRR (Sinha et al., 2019), ReClor (Yu et al., 2020), and LogiQA (Liu et al., 2020) for logical reasoning; Quoref (Dasigi et al., 2019) for coreference reasoning; CommonsenseQA (Talmor et al., 2019), MCScript2.0 (Ostermann et al., 2019), and CosmosQA (Huang et al., 2019) for commonsense reasoning. Many of these datasets consist of only a single paragraph in the input or lack explanation information that describes the reasoning process from question to answer. However, our focus is on multi-hop reasoning datasets that contain multiple paragraphs in the input and provide explanatory information for the QA process.

7 Conclusion

We analyze the effectiveness of the underlying reasoning tasks using two multi-hop datasets: 2Wiki and HotpotQA-small. The results reveal that the

underlying reasoning tasks can improve QA performance. Using four debiased sets, we demonstrate that the underlying reasoning tasks can reduce the reasoning shortcuts of the QA task. The results also reveal that the underlying reasoning tasks do not make the models more robust on adversarial examples, such as sub-questions and inverted questions. We encourage future studies to investigate the effectiveness of the entity-level reasoning task in the form of sub-questions.

Limitations

Our study has two main limitations. The first one is the small size of HotpotQA-small. Currently, there are no other multi-hop datasets that contain a large number of examples with the entity-level reasoning prediction task. MuSiQue is the most potential option. The entity-level reasoning information in MuSiQue includes two types of formats: triple format and natural language question format. We do not experiment with MuSiQue because the number of examples that have entity-level reasoning information in the form of a triple is small: 2,253 out of 19,938 in the training set and 212 out of 2,417 in the dev. set.

The second limitation is that our model does not consider the order of the triples in the entity-level reasoning prediction task. As shown in Figure 1b, the two triples are ordered. However, our model formulates the entity-level prediction task as a relation extraction task. We predict a relation given the two entities detected by the NER models. Therefore, the order of the triples is not considered. We conjecture that this may be one of the reasons why the entity-level reasoning prediction task (e.g., a triple (*Film A*, *director*, *D*)) does not support the model when answering sub-questions (e.g., *Who is the director of Film A?*) using the same information.

Acknowledgments

We would like to thank Viktor Schlegel and the anonymous reviewers for their valuable comments and suggestions. This work was supported by JSPS KAKENHI Grant Numbers 21H03502 and 22K17954 and JST AIP Trilateral AI Research and PRESTO Grant Number JPMJCR20G9.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019.

- FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.
- Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*.
- Jifan Chen and Greg Durrett. 2019. [Understanding dataset design choices for multi-hop reasoning](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4026–4032, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jifan Chen, Shih ting Lin, and Greg Durrett. 2019. [Multi-hop question answering via reasoning chains](#). *arXiv:1910.02610*.
- Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. [Quoref: A reading comprehension dataset with questions requiring coreferential reasoning](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. [Question answering by reasoning across documents with graph convolutional networks](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2306–2317, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. [Cognitive graph for multi-hop reading comprehension at scale](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2694–2703, Florence, Italy. Association for Computational Linguistics.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. [DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuo-hang Wang, and Jingjing Liu. 2020. [Hierarchical graph network for multi-hop question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online. Association for Computational Linguistics.
- Yair Feldman and Ran El-Yaniv. 2019. [Multi-hop paragraph retrieval for open-domain question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2309, Florence, Italy. Association for Computational Linguistics.
- Ruiliu Fu, Han Wang, Xuejun Zhang, Jun Zhou, and Yonghong Yan. 2021a. [Decomposing complex questions makes multi-hop QA easier and more interpretable](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 169–180, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ruiliu Fu, Han Wang, Jun Zhou, and Xuejun Zhang. 2021b. [Na-reviewer: Reviewing the context to improve the error accumulation issue for multi-hop qa](#). *Electronics Letters*.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. [Shortcut learning in deep neural networks](#). *Nature Machine Intelligence*, 2(11):665–673.
- Mor Geva, Tomer Wolfson, and Jonathan Berant. 2022. [Break, perturb, build: Automatic perturbation of reasoning paths through question decomposition](#). In *Transactions of the Association for Computational Linguistics (TACL)*.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Xanh Ho, Saku Sugawara, and Akiko Aizawa. 2022. [How well do multi-hop reading comprehension models understand date information?](#) In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 470–479, Online only. Association for Computational Linguistics.

- Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. [Cosmos QA: Machine reading comprehension with contextual commonsense reasoning](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.
- Naoya Inoue, Pontus Stenetorp, and Kentaro Inui. 2020. [R4C: A benchmark for evaluating RC systems to get the right answer for the right reason](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6740–6750, Online. Association for Computational Linguistics.
- Yichen Jiang and Mohit Bansal. 2019a. [Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736, Florence, Italy. Association for Computational Linguistics.
- Yichen Jiang and Mohit Bansal. 2019b. [Self-assembling modular networks for interpretable multi-hop reasoning](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4474–4484, Hong Kong, China. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. [Look at the first sentence: Position bias in question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1109–1121, Online. Association for Computational Linguistics.
- Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. [LogiQA: A challenge dataset for machine reading comprehension with logical reasoning](#). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.
- Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. [Compositional questions do not necessitate multi-hop reasoning](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4249–4257, Florence, Italy. Association for Computational Linguistics.
- Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. [Multi-hop reading comprehension through question decomposition and rescoring](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy. Association for Computational Linguistics.
- Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. [Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345, Florence, Italy. Association for Computational Linguistics.
- Simon Ostermann, Michael Roth, and Manfred Pinkal. 2019. [MCScript2.0: A machine comprehension corpus focused on script events and participants](#). In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 103–117, Minneapolis, Minnesota. Association for Computational Linguistics.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS Autodiff Workshop*.
- Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. [Unsupervised question decomposition for question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. Association for Computational Linguistics.
- Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. 2021. [Answering open-domain questions of varying reasoning steps from text](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3599–3614, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. [Dynamically fused graph network for multi-hop reasoning](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150, Florence, Italy. Association for Computational Linguistics.
- Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. [CLUTRR: A diagnostic benchmark for inductive reasoning from text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

- Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. [Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks](#). *arXiv:1809.02040*.
- Alon Talmor and Jonathan Berant. 2018. [The web as a knowledge-base for answering complex questions](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [CommonsenseQA: A question answering challenge targeting commonsense knowledge](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.
- Chenhao Tan and Lillian Lee. 2014. [A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 403–408, Baltimore, Maryland. Association for Computational Linguistics.
- Yixuan Tang, Hwee Tou Ng, and Anthony Tung. 2021. [Do multi-hop question answering systems know how to answer the single-hop sub-questions?](#) In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3244–3249, Online. Association for Computational Linguistics.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2020. [Is multihop QA in DiRe condition? measuring and reducing disconnected reasoning](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8846–8863, Online. Association for Computational Linguistics.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [Musique: Multi-hop questions via single-hop question composition](#). *Transactions of the Association for Computational Linguistics*, 10(0):539–554.
- Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. [Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9073–9080. AAAI Press.
- Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. [Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2704–2713, Florence, Italy. Association for Computational Linguistics.
- Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022. [Measure and improve robustness in NLP models: A survey](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.
- Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. [Constructing datasets for multi-hop reading comprehension across documents](#). *Transactions of the Association for Computational Linguistics*, 6:287–302.
- Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. [Break it down: A question understanding benchmark](#). *Transactions of the Association for Computational Linguistics*, 8:183–198.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. [Reclor: A reading comprehension dataset requiring logical reasoning](#). In *International Conference on Learning Representations*.
- Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. [Big bird: Transformers for longer sequences](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.
- Dongxu Zhang and Dong Wang. 2015. [Relation classification via recurrent neural network](#). *arXiv:1508.01006*.

A Word Overlap Shortcut

Using adversarial methods, Jiang and Bansal (2019a) show that examples in HotpotQA often contain word overlap shortcut, where the models can answer the questions by performing word-matching between the question and a sentence in the context.

Based on this finding, we automatically calculate the word overlap shortcut for 2Wiki and HotpotQA-small. We observe that the word overlap shortcut is common in bridge questions; therefore, we only calculate the word overlap shortcut for bridge questions in 2Wiki and HotpotQA-small. To check whether a sample contains the word overlap shortcut, we do the following steps:

- Obtain a set of surrounding words S by getting the five words immediately to the left and right of the answer span, then remove stopwords in S .
- Obtain a set of overlapping words (O) between S and a question.
- We consider a sample containing the word overlap shortcut if there are at least two words in O and $\frac{|O|}{|S|} \geq 0.65$. These numbers (threshold) are chosen based on the evaluation of 40 examples that are manually annotated by the authors.

We find that there are 56 out of 5,791 and 151 out of 715 examples (5,791 and 715 are the numbers of bridge questions in 2Wiki and HotpotQA-small) in the dev. sets of 2Wiki and HotpotQA-small containing the word overlap shortcut.

It is noted that there is another type of shortcut, namely, entity-type matching shortcut. Based on the experimental results and human performance, Min et al. (2019a) reveal that examples in HotpotQA contain the entity type matching shortcut, where the models can answer the questions by using the first five tokens in the questions; meanwhile, humans can answer the questions by using the entity type of the paragraphs. Currently, there is no dataset that can prevent the entity-type shortcut; therefore, we do not use this type of shortcut in our experiments.

B Experimental Details

B.1 Implementation Details

We use Pytorch (Paszke et al., 2017) and Hugging Face when building our model. For the context encoding step, we use a pre-trained BigBird model as the encoder; the hidden dimension is 768. For the entity-level reasoning prediction task, we obtain 33 relations for 2Wiki and 2,526 relations for R⁴C, from all triples in the training set, including a non-relation type. We use entity type embedding d_t of

50. We fine-tuned our model with a total batch size of 32 on a single GPU (NVIDIA A100 80GB) using mixed precision and a gradient accumulation step of 8. Following the hyperparameters in the BERT model (Devlin et al., 2019), for optimization, we use the Adam Optimizer (Kingma and Ba, 2015) with a learning rate of 3e-5, weight decay of 0.01, learning rate warmup over the first 10% of the total number of training steps, and linear decay of the learning rate. We also use a dropout probability of 0.1 on all layers.

For multi-task prediction, we use λ_{sent} as 4, λ_{ent} as 15, and λ_{ans} as 1 for 2Wiki and HotpotQA-small; we use λ_{sent} as 7 and λ_{ans} as 1 for HotpotQA. We do not run all experiments with different values of λ_{sent} , λ_{ent} , and λ_{ans} ; instead we run several experiments, based on the results, we then adjust the parameters. We find that when running with λ_{sent} as 4 for 2Wiki and 7 for HotpotQA, λ_{ent} as 15, and λ_{ans} as 1, it produces the best results. We fix the random seed for the reproducibility of the results. We observe that the final epoch often produces the best scores, and its scores are stable on adversarial datasets; therefore, we choose the final epoch for all settings in our experiment.

B.2 Datasets

HotpotQA HotpotQA was created by crowdsourcing. Due to the design of the dataset, there are only two tasks in HotpotQA: sentence-level SFs prediction and answer prediction. R⁴C was created based on HotpotQA and contained 4,588 questions. The dataset requires systems to provide an answer and derivation in a semi-structured natural language form. There are two types of questions in HotpotQA: bridge and comparison.

2Wiki 2Wiki was constructed by utilizing a Knowledge Base and Wikipedia, and the questions were created by using templates. There are three different tasks in the dataset: (1) sentence-level SFs prediction, (2) evidence generation (for consistency, we use the term *entity-level prediction*), and (3) answer prediction. The context consists of ten paragraphs, including two or four gold paragraphs and eight or six distractor paragraphs. The gold paragraph contains the information required to find the answer. Meanwhile, the purpose of the distractor paragraph is to distract the models. There are four different types of questions in the dataset: comparison, inference, compositional, and bridge-comparison. Inference and compositional

<p>Question: Who was born first, Albert Einstein or Abraham Lincoln?</p> <p>Paragraph A: Albert Einstein</p> <p>[1] Albert Einstein (14 March 1879 – 18 April 1955) was a ...</p> <p>Paragraph B: Abraham Lincoln</p> <p>[2] Abraham Lincoln (February 12, 1809 – April 15, 1865) was a ...</p> <p>Answer: Abraham Lincoln</p> <p>Sentence-level supporting facts: 1, 2</p> <p>Entity-level reasoning prediction (Evidence):</p> <p>Step 1: ("Albert Einstein", "date of birth", "14 March 1879") &</p> <p>Step 2: ("Abraham Lincoln", "date of birth", "February 12, 1809")</p> <p>Adversarial question: Who was born later, Albert Einstein or Abraham Lincoln?</p>
--

Figure 4: Example of a comparison question from the 2Wiki dataset.

questions are two sub-types of the bridge question. For the convenience of analysis, we consider comparison and bridge-comparison questions as comparison questions. Figure 4 presents an example of a comparison question from the 2Wiki dataset.

2Wiki was designed to focus on the entire reasoning process from question to answer. The entire capacity of the model is evaluated by using two metrics: joint EM and joint F1. To obtain the joint F1 score, they first calculate the joint precision and joint recall as follows: $P^{joint} = P^{ans} P^{ent} P^{sent}$ and $R^{joint} = R^{ans} R^{ent} R^{sent}$. (P^{ans}, R^{ans}) , (P^{ent}, R^{ent}) , (P^{sent}, R^{sent}) represent the precision and recall for three tasks: answer prediction, entity-level reasoning prediction, and sentence-level SFs prediction. The joint EM is 1 when all three tasks achieve an exact match and otherwise 0.

B.3 Results Comparison

We compare our results with three previous models: BiDAF, CRERC, and NA-Reviewer. BiDAF is a baseline model in Ho et al. (2020). CRERC (Fu et al., 2021a) is a pipeline model that includes three modules: relation extractor, reader, and comparator. NA-Reviewer (Fu et al., 2021b) is an improved version of CRERC, as it addresses the error accumulation issue. It is noted that both CRERC and NA-Reviewer models are evaluated on only 2Wiki.

Table 7 presents the results of our model and previous models in the dev. set of HotpotQA and in the test set of 2Wiki. It also shows the performance of our model in the dev. set of HotpotQA-small and human performance in Ho et al. (2020).

Results on HotpotQA Our score is comparable to the BERT-base version of two strong models, SAE (Tu et al., 2020) and HGN (Fang et al., 2020)

in the dev. set of the distractor setting in HotpotQA. Specifically, our joint F1 is 67.8, while for SAE-BERT, it is 66.5, and for HGN-BERT, it is 66.9. However, our score is smaller than the RoBERTa-base of SAE and HGN. They are 72.8 and 74.4 F1 for SAE-RoBERTa and HGN-RoBERTa, respectively. It is noted that we use the BigBird-ITC version in our model. Although the BigBird-ETC version performs better than the BigBird-ITC version, it is not available in Hugging Face. We do not use SAE and HGN for our analyses because these models are not designed to train on the entity-level reasoning prediction task.

Results on HotpotQA-small The scores on HotpotQA-small are lower than those on HotpotQA in the answer prediction task. This result may be explained by the fact that the training size of HotpotQA-small is smaller than HotpotQA (3,671 vs. 90,564). Due to the small size, we only use the gold paragraphs for experiments. That is why the scores on HotpotQA-small are higher than those on HotpotQA in the sentence-level task. For the entity-level task, the EM score is quite low (6.4 EM). A possible reason for this is that there are many relations in HotpotQA-small (2,526 relations); meanwhile, there are only 33 relations in 2Wiki. We observe that the F1 score (31.1 F1) is much better than the EM score. Therefore, we keep using HotpotQA-small for analyses.

Results on 2Wiki Our model significantly outperforms BiDAF in all tasks. Our results are comparable to CRERC. The EM score of our model in the entity-level task is lower than that of CRERC. A possible explanation for this might be that the relation extractor module in CRERC is fine-tuned on 2Wiki; therefore, it can extract entities better than the NER models from Spacy and Flair that are used in our model. However, the F1 score of our model in the entity-level task is higher than that of CRERC. This indicates that our model can correctly obtain a few triples in a set of gold triples for many samples. All our scores (except the F1 score of the entity-level task) are lower than those on NA-Reviewer. Our target is to analyze the UR tasks in an end-to-end model. Although the pipeline models (CRERC and NA-Reviewer) are easy to interpret, we cannot determine how the UR tasks affect answer prediction in an end-to-end model. Therefore, we use the design of our model to perform the analyses in this study.

Dataset	Model	Answer		Sentence-level		Entity-level		Joint	
		EM	F1	EM	F1	EM	F1	EM	F1
HotpotQA	HGN-BERT [‡] (Fang et al., 2020)	<i>N/A</i>	74.76	<i>N/A</i>	86.61	×	×	<i>N/A</i>	66.90
	HGN-RoBERTa (Fang et al., 2020)	68.93	82.18	63.09	88.59	×	×	46.46	74.34
	SAE-BERT (Tu et al., 2020)	61.32	74.81	58.06	85.27	×	×	39.89	66.45
	SAE-RoBERTa (Tu et al., 2020)	67.70	80.75	63.30	87.38	×	×	46.81	72.75
	Our BigBird-base	61.90	76.09	58.54	86.93	×	×	39.39	67.81
HotpotQA-small	Our BigBird-base	54.74	69.44	75.14	90.88	6.43	31.05	4.25	21.69
2Wiki	BiDAF (Ho et al., 2020)	36.53	43.93	24.99	65.26	1.07	14.94	0.35	5.41
	CRERC (Fu et al., 2021a)	69.58	72.33	82.86	90.68	54.86	68.83	49.80	58.99
	NA-Reviewer (Fu et al., 2021b)	76.73	81.91	89.61	94.31	53.66	70.83	52.75	65.23
	Our BigBird-base	74.05	79.68	77.14	92.13	45.75	76.64	39.30	63.24
	Human UB (Ho et al., 2020)	91.00	91.79	88.00	93.75	64.00	78.81	62.00	75.25

Table 7: Results (%) of our model and previous models in the dev. set of HotpotQA and in the test set of 2Wiki. We also show the performance of our model in the dev. set of HotpotQA-small. *Answer*, *Sentence-level*, and *Entity-level* represent the answer prediction task, sentence-level prediction task, and entity-level prediction task, respectively. For HGN-BERT, the scores that we obtained (from left to right: 58.93 73.18 54.64 85.34 35.11 64.24) are lower than the reported scores in HGN (Fang et al., 2020); therefore, we show the reported F1 scores in HGN.

B.4 Effectiveness of the UR Tasks

Reasoning Shortcuts (RQ2) Table 8 presents the performance drop (smaller is better) for five times running of the four settings of the model on the four debiased sets of 2Wiki and HotpotQA-small. As depicted in the table, for 2Wiki, the gap between two settings #1 (answer prediction task only) and #4 (all three tasks) is consistent in all five times running. Meanwhile, for HotpotQA-small, the gap between two settings #1 (answer prediction task only) and #4 (all three tasks) is inconsistent in all five times running. This observation supports our explanation in Section 5.2 that the position bias in HotpotQA-small does not have a large impact on the main QA task.

B.5 Analyses

Details of RQ2 Figure 5 illustrates the information on the position of sentence-level SFs of comparison and bridge questions in the dev. sets of the two datasets: 2Wiki and HotpotQA-small. As shown in the Figure, the comparison questions have more position biases than the bridge questions in both 2Wiki and HotpotQA-small. Furthermore, we observe that the position bias in the comparison questions in HotpotQA-small is smaller than that in 2Wiki.

Table 9 presents the performance drop for two types of questions, comparison and bridge questions, in 2Wiki and HotpotQA-small.

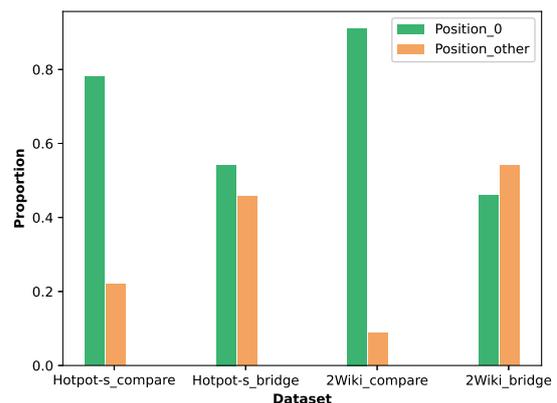


Figure 5: Information on the position of sentence-level SFs of comparison and bridge questions in the dev. sets of the two datasets: 2Wiki and HotpotQA-small.

Details of RQ3 Table 10 presents examples of the outputs predicted by our model, which is trained on three tasks simultaneously.

Dataset	Task Setting	Reduction (%)									
		Time #1		Time #2		Time #3		Time #4		Time #5	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
2Wiki											
ADDUNRELATED	Ans	<u>13.26</u>	<u>12.11</u>	<u>13.06</u>	<u>11.87</u>	<u>13.59</u>	<u>12.15</u>	<u>13.31</u>	<u>12.08</u>	<u>13.79</u>	<u>12.44</u>
	Ans + Sent	10.82	9.56	10.94	9.68	10.93	9.83	11.15	9.83	11.16	9.66
	Ans + Ent	8.09	7.15	7.77	6.98	7.71	6.99	7.78	7.08	7.31	6.52
	Ans + Sent + Ent	8.41	7.80	8.85	7.99	9.10	8.33	8.97	8.23	8.97	8.18
ADDRELATED	Ans	3.72	3.61	3.61	3.57	3.54	3.39	3.29	3.26	3.57	3.49
	Ans + Sent	<u>4.22</u>	<u>4.23</u>	<u>4.44</u>	<u>4.44</u>	<u>4.11</u>	<u>4.18</u>	<u>4.08</u>	<u>4.20</u>	<u>3.94</u>	<u>4.04</u>
	Ans + Ent	2.61	2.62	2.89	2.88	2.85	2.83	2.89	2.81	2.75	2.71
	Ans + Sent + Ent	3.18	3.12	3.18	3.14	3.06	3.06	3.23	3.25	3.14	3.07
ADD2	Ans	<u>12.26</u>	<u>11.63</u>	<u>12.59</u>	<u>12.10</u>	<u>12.29</u>	<u>11.69</u>	<u>12.29</u>	<u>11.72</u>	<u>12.16</u>	<u>11.48</u>
	Ans + Sent	11.10	10.48	11.03	10.57	11.26	10.77	11.18	10.72	11.51	10.92
	Ans + Ent	8.38	7.91	8.74	8.12	8.31	7.63	8.07	7.48	8.41	7.67
	Ans + Sent + Ent	9.13	8.54	8.90	8.45	8.94	8.54	8.95	8.43	9.51	8.92
ADD2SWAP	Ans	<u>19.06</u>	<u>17.61</u>	<u>18.87</u>	<u>17.40</u>	<u>19.20</u>	<u>17.59</u>	<u>18.80</u>	<u>17.31</u>	<u>19.03</u>	<u>17.63</u>
	Ans + Sent	17.71	16.16	17.73	16.40	17.74	16.41	17.34	15.99	17.59	16.25
	Ans + Ent	13.02	12.19	13.09	12.19	13.38	12.30	13.13	12.30	12.97	12.07
	Ans + Sent + Ent	14.28	13.56	14.31	13.70	14.18	13.41	14.89	13.99	15.00	14.17
HotpotQA-small											
ADDUNRELATED	Ans	4.33	2.89	1.44	0.68	0.21	-0.45	4.33	2.66	4.75	1.85
	Ans + Sent	4.01	3.65	0.81	1.07	-0.20	0.88	1.01	0.85	0.00	0.28
	Ans + Ent	6.17	4.97	5.97	3.98	<u>7.76</u>	<u>7.06</u>	<u>6.38</u>	<u>6.20</u>	<u>7.36</u>	<u>5.80</u>
	Ans + Sent + Ent	<u>6.76</u>	<u>5.83</u>	<u>6.76</u>	<u>5.18</u>	1.79	2.97	4.77	4.94	5.17	4.32
ADDRELATED	Ans	<u>3.71</u>	<u>1.14</u>	<u>4.12</u>	1.10	<u>4.54</u>	<u>2.12</u>	3.91	<u>1.46</u>	<u>3.91</u>	1.70
	Ans + Sent	-0.79	0.10	0.20	0.71	0.61	1.23	-1.40	-0.59	-1.19	-0.52
	Ans + Ent	-0.80	0.23	-0.60	-0.19	-1.21	-0.13	-0.40	0.68	-1.61	-0.42
	Ans + Sent + Ent	0.38	0.59	2.37	<u>1.96</u>	1.19	0.95	0.00	0.59	2.37	<u>2.17</u>
ADD2	Ans	1.04	1.01	1.04	0.17	1.04	0.53	1.64	0.51	3.50	2.83
	Ans + Sent	1.01	1.51	-0.79	-0.19	0.00	1.43	0.00	0.32	0.20	1.20
	Ans + Ent	<u>4.57</u>	<u>3.59</u>	2.19	2.32	<u>5.17</u>	<u>4.85</u>	<u>3.78</u>	<u>3.20</u>	<u>4.38</u>	<u>3.73</u>
	Ans + Sent + Ent	0.00	0.62	<u>3.38</u>	<u>3.63</u>	1.19	2.68	1.59	2.56	2.98	2.79
ADD2SWAP	Ans	5.16	3.55	3.10	1.40	3.71	1.82	3.29	2.12	4.54	3.45
	Ans + Sent	3.82	3.77	0.81	1.17	2.00	2.03	1.62	1.04	0.61	1.81
	Ans + Ent	<u>5.57</u>	<u>4.20</u>	<u>6.38</u>	<u>5.07</u>	<u>7.96</u>	<u>6.83</u>	<u>6.56</u>	<u>5.39</u>	<u>7.96</u>	<u>5.83</u>
	Ans + Sent + Ent	3.58	3.54	5.77	4.74	1.59	2.52	2.37	3.04	4.57	4.38

Table 8: Performance drop (smaller is better) for five times running of the four settings of the model on the four debiased sets of 2Wiki and HotpotQA-small. The best and worst scores are boldfaced and underlined, respectively. The debiased datasets are newly created for each time running.

Dataset	Task Setting	Comparison				Bridge			
		Answer		Answer↓ (%)		Answer		Answer↓ (%)	
		EM	F1	EM	F1	EM	F1	EM	F1
2Wiki									
Dev	Ans	78.98	83.74			66.10	72.85		
	Ans + Sent	79.45	84.21			67.16	73.90		
	Ans + Ent	78.86	83.60			66.75	73.61		
	Ans + Sent + Ent	80.35	85.08			67.84	74.49		
ADDUNRELATED	Ans	59.51	64.49	<u>24.65</u>	<u>22.99</u>	65.01	71.81	1.65	1.43
	Ans + Sent	65.55	71.11	17.50	15.56	64.42	71.14	<u>4.08</u>	<u>3.73</u>
	Ans + Ent	67.67	72.84	14.19	12.87	65.47	72.44	1.92	1.59
	Ans + Sent + Ent	69.38	74.01	13.65	13.01	65.72	72.48	3.13	2.70
ADDRELATED	Ans	73.60	78.22	<u>6.81</u>	<u>6.59</u>	65.73	72.36	0.56	0.67
	Ans + Sent	74.87	79.43	5.76	5.68	65.38	71.82	<u>2.65</u>	<u>2.81</u>
	Ans + Ent	75.57	80.17	4.17	4.10	66.06	72.75	1.03	1.17
	Ans + Sent + Ent	76.69	81.28	4.56	4.47	66.63	73.14	1.78	1.81
ADD2	Ans	61.61	65.54	<u>21.99</u>	<u>21.73</u>	64.55	71.60	2.34	1.72
	Ans + Sent	64.93	69.13	18.28	17.91	64.58	71.50	<u>3.84</u>	<u>3.25</u>
	Ans + Ent	67.16	71.43	14.84	14.56	65.51	72.52	1.86	1.48
	Ans + Sent + Ent	67.85	72.19	15.56	15.15	66.06	72.94	2.62	2.08
ADD2SWAP	Ans	51.13	55.50	<u>35.26</u>	<u>33.72</u>	64.42	71.55	2.54	1.78
	Ans + Sent	55.19	60.21	30.53	28.50	63.96	70.83	<u>4.76</u>	<u>4.15</u>
	Ans + Ent	60.42	64.80	23.38	22.49	65.04	71.99	2.56	2.20
	Ans + Sent + Ent	60.25	64.37	25.02	24.34	65.51	72.23	3.43	3.03
HotpotQA-small									
Dev	Ans	56.44	61.86			51.89	67.72		
	Ans + Sent	57.92	63.44			53.43	70.61		
	Ans + Ent	57.92	63.14			53.85	70.75		
	Ans + Sent + Ent	57.43	64.44			53.99	70.86		
ADDUNRELATED	Ans	50.00	56.24	11.41	9.09	50.77	66.85	2.16	1.28
	Ans + Sent	52.97	60.64	8.55	4.41	52.03	68.17	2.62	3.46
	Ans + Ent	51.49	57.43	11.10	9.04	51.33	67.97	<u>4.68</u>	<u>3.93</u>
	Ans + Sent + Ent	47.03	55.59	<u>18.11</u>	<u>13.73</u>	52.17	68.16	3.37	3.81
ADDRELATED	Ans	53.96	60.48	4.39	2.23	50.07	67.14	<u>3.51</u>	<u>0.86</u>
	Ans + Sent	57.43	63.37	0.85	0.11	54.13	70.54	-1.31	0.10
	Ans + Ent	58.91	64.11	-1.71	-1.54	54.13	70.27	-0.52	0.68
	Ans + Sent + Ent	53.96	61.23	<u>6.04</u>	<u>4.98</u>	54.69	71.24	-1.30	-0.54
ADD2	Ans	54.46	59.52	<u>3.51</u>	<u>3.78</u>	51.75	67.53	0.27	0.28
	Ans + Sent	58.91	64.31	-1.71	-1.37	52.45	69.03	1.83	2.24
	Ans + Ent	56.93	62.33	1.71	1.28	50.91	67.81	<u>5.46</u>	<u>4.16</u>
	Ans + Sent + Ent	55.94	62.58	2.59	2.89	54.41	70.82	-0.78	0.06
ADD2SWAP	Ans	48.51	53.94	<u>14.05</u>	<u>12.80</u>	50.63	66.94	2.43	1.15
	Ans + Sent	53.47	60.30	7.68	4.95	52.03	68.16	2.62	3.47
	Ans + Ent	53.96	60.51	6.84	4.17	51.05	67.78	<u>5.20</u>	<u>4.20</u>
	Ans + Sent + Ent	50.99	58.64	11.21	9.00	53.29	69.33	1.30	2.16

Table 9: Performance drop (smaller is better) for two types of questions (comparison and bridge questions) of the four settings of the model on the four debiased sets of 2Wiki and HotpotQA-small. The best and worst scores are boldfaced and underlined, respectively. For both 2Wiki and HotpotQA-small, we choose the results from the first time running to perform the analysis.

Type	Example
Bridge - Prune	<p>Paragraph A: Polish-Russian War (Wojna polsko-ruska) is a 2009 Polish film directed by Xawery Żuławski based on ...</p> <p>Paragraph B: Xawery Żuławski (born 22 December 1971 in Warsaw) is a Polish film director. ... He is the son of actress Małgorzata Braunek and director Andrzej Żuławski. ...</p> <p>Q: Who is the director of Polish-Russian War?</p> <p>Predicted answer: Andrzej Żuławski ✗</p> <p>Predicted entity-level: (“Polish-Russian War”, “director”, “Xawery Żuławski”) ✓</p>
Bridge - Prune	<p>Paragraph A: Francesca von Habsburg (born 7 June 1958) is an art collector and the estranged wife of Karl von Habsburg, current head of the House of Habsburg- Lorraine.</p> <p>Paragraph B: Michaela von Habsburg was born ... She is the twin sister of Monika von Habsburg, and daughter of Otto von Habsburg and Princess Regina of Saxe - Meiningen.</p> <p>Q: Who is the spouse of Francesca von Habsburg?</p> <p>Predicted answer: Princess Regina of Saxe - Meiningen ✗</p> <p>Predicted entity-level: (“Francesca von Habsburg”, “spouse”, “Karl von Habsburg”) ✓</p>
Comparison - Inverted	<p>Paragraph A: Montréal/Les Cèdres Airport is a general aviation aerodrome located approximately west of Montreal, Quebec, Canada near Autoroute 20 west of ...</p> <p>Paragraph B: Flying J Ranch Airport is a privately owned, public use ... The airport is located southwest of the central business district of Pima, a city in Graham County, Arizona, United States and northeast of Tucson International Airport. ...</p> <p>Q: Are Montréal/Les Cèdres Airport and Flying J Ranch Airport located in different countries?</p> <p>Predicted answer: no ✗</p> <p>Predicted entity-level: (“Flying J Ranch Airport”, “country”, “United States”) & (“Montréal/Les Cèdres Airport”, “country”, “Canada”) ✓</p>
Comparison - Inverted	<p>Paragraph A: A Romance of the Air is a 1918 American silent drama film based ... Directed by Harry Revier, the film was ...</p> <p>Paragraph B: Harry Revier (16 March 1890 – 13 August 1957) was ... American director ...</p> <p>Paragraph C: How Moscha Came Back is a 1914 silent film comedy short directed by Phillips Smalley. ...</p> <p>Paragraph D: Phillips Smalley (August 7, 1865 – May 2, 1939) was an American silent film director and actor.</p> <p>Q: Which film has the director who was born later, A Romance of the Air or How Moscha Came Back?</p> <p>Predicted answer: How Moscha Came Back ✗</p> <p>Predicted entity-level: (“A Romance of the Air”, “director”, “Harry Revier”), (“How Moscha Came Back”, “director”, “Phillips Smalley”), (“Harry Revier”, “date of birth”, “16 March 1890”), & (“Phillips Smalley”, “date of birth”, “August 7, 1865”) ✓</p>

Table 10: Examples of the outputs predicted by our model, which is trained on three tasks simultaneously.

PubMedCLIP: How Much Does CLIP Benefit Visual Question Answering in the Medical Domain?

Sedigheh Eslami, Christoph Meinel, Gerard de Melo

Hasso Plattner Institute / University of Potsdam

{sedigheh.eslami, christoph.meinel, gerard.demelo}@hpi.de

Abstract

Contrastive Language–Image Pre-training (CLIP) has shown remarkable success in learning with cross-modal supervision from extensive amounts of image–text pairs collected online. Thus far, the effectiveness of CLIP has been investigated primarily in general-domain multimodal problems. In this work, we evaluate the effectiveness of CLIP for the task of Medical Visual Question Answering (MedVQA). We present PubMedCLIP, a fine-tuned version of CLIP for the medical domain based on PubMed articles. Our experiments conducted on two MedVQA benchmark datasets illustrate that PubMedCLIP achieves superior results improving the overall accuracy up to 3% in comparison to the state-of-the-art Model-Agnostic Meta-Learning (MAML) networks pre-trained only on visual data. The PubMedCLIP model with different back-ends, the source code for pre-training them and reproducing our MedVQA pipeline is publicly available at <https://github.com/sarahESL/PubMedCLIP>.

1 Introduction

Medical visual question answering (MedVQA) seeks answers to natural language questions about a given medical image. The development of MedVQA has considerable potential to benefit healthcare systems, as it may aid clinicians in interpreting medical images and obtaining more accurate diagnoses by consulting a second opinion. Thus, it has become a very active area of research, with competitive benchmarks and yearly competitions (Abacha et al., 2021). Yet, visual question answering in the medical domain in particular remains non-trivial, as we suffer from a general lack of large balanced training data, in part due to privacy concerns. To solve the multimodal task of MedVQA, a system must understand both medical images and textual questions and infer the associations between them sufficiently well to produce a correct answer (An-

tol et al., 2015). Thus, the success of these solutions is tied to the effectiveness of their visual and question encoders. Current approaches for MedVQA adopt deep artificial neural network encoders to interpret the image and the question. Previous studies in MedVQA (Nguyen et al., 2019; Zhan et al., 2020; Pan et al., 2021; Gong et al., 2022) commonly exploit the Mixture of Enhanced Visual Features (MEVF) model (Nguyen et al., 2019) as their visual encoder to overcome data limitations. However, MEVF is custom-tailored for the particular challenges encountered in the VQA-RAD (Lau et al., 2018) dataset, i.e., specifically designed for the organs present in this dataset, limiting its generalizability to other settings.

In non-medical settings, recent work (Su et al., 2019; Zhang et al., 2020; Cho et al., 2021; Wang et al., 2021; Radford et al., 2021; Yu et al., 2022) has shown improvements of visual encoders when learning from multimodal image–text pairs in comparison to learning from just visual images. Among these approaches, the contrastive pre-training of language–image data in OpenAI’s CLIP (Radford et al., 2021) has been particularly prominent. CLIP is trained using a vast number of image–text pairs acquired from the Internet with close to zero additional human annotation. We argue that this is particularly promising for the medical domain, since data annotation requires expert medical knowledge, making it expensive and time-consuming. Following CLIP, we investigate to what extent learning from publicly available medical image–text pairs without any further annotation can aid in the MedVQA task. To this end, we use image–text pairs obtained from PubMed articles to train a new version of CLIP called PubMedCLIP. We then examine the outcomes when incorporating PubMedCLIP into state-of-the-art MedVQA methods, investigating whether CLIP benefits MedVQA.

To the best of our knowledge, this is the first study introducing a PubMed-optimized CLIP and

assessing the effectiveness of its visual and textual encoders for VQA. Unlike prior work on MedVQA, PubMedCLIP is trained using medical images from a diverse range of body regions and is not restricted to only a few organs. We conduct extensive experiments on two MedVQA benchmark datasets and employ diverse back-end visual encoders in PubMedCLIP. Our experiments show that using PubMedCLIP as a pre-trained visual encoder improves previous models by up to 3%. Our experiments further reveal question type distributional differences in the two MedVQA benchmark datasets that have not been imparted in previous work and cause different back-end visual encoders in PubMedCLIP to exhibit different behavior on these datasets.

2 Related Work

Shen et al. (2021) showed the benefits of CLIP for general-domain visual question answering. However, MedVQA approaches generally need to be able to learn from small amounts of training data and be able to pick up fine-granular details such as subtle medical abnormalities. Recent MedVQA approaches typically employ deep pre-trained neural encoders and consist of four main components: a visual encoder, question encoder, attention-based fusion of vision and text features, and an answer classifier (Nguyen et al., 2019; Vu et al., 2020; Zhan et al., 2020; Pan et al., 2021; Liu et al., 2021a; Gong et al., 2022). Skip-thought vectors, LSTM, and GRU recurrent neural networks have been popular question encoders in prior work. Due to the lack of diversity in the semantics of the questions in the ImageCLEF VQA-Med 2021 Challenge (Abacha et al., 2021), the winning teams (Gong et al., 2021; Eslami et al., 2021) were able to treat MedVQA as a multi-class image classification task, without any need to encode and interpret the questions. Bilinear attention networks (Kim et al., 2018), multimodal compact bilinear pooling (Fukui et al., 2016), stacked attention networks (Yang et al., 2016), and element-wise production are popular as multimodal pooling approaches in MedVQA. With regard to the visual encoder, the winning teams in the ImageCLEF VQA-Med Challenges (Abacha et al., 2020, 2021) often fine-tune an ensemble of pre-trained VGG (Simonyan and Zisserman, 2014) and various ResNet (Lei et al., 2018) encoders. A notable number of papers (Nguyen et al., 2019; Zhan et al., 2020; Pan et al., 2021; Gong et al., 2022) employ the Mixture

of Enhanced Visual Features (MEVF; Nguyen et al. 2019) in order to overcome image data limitations. MEVF consists of two modules: 1. the pre-trained meta-learning module, which uses Model-Agnostic Meta-Learning (MAML; Finn et al. 2017) with the objective of solving a k -shot n -way classification problem with the abnormality status of chest, abdomen, and brain organs as classes, 2. the Convolutional Denoising Autoencoder (CDAE; Masci et al. 2011) module in order to have a robust visual encoder for noisy medical images. The pre-training of MEVF is custom-tailored for the particular organs that are present in the VQA-RAD (Lau et al., 2018) dataset, i.e., chest, brain, abdomen. Another study (Do et al., 2021) similarly trained multiple meta-models confined to these three body regions, combined with a scoring mechanism to select the n most robust and accurate encoders and concatenate their outputs to represent the visual features. Liu et al. (2021a) also restricted the objective of their visual encoding to chest, brain, and abdomen, and pre-trained three separate visual encoder teacher models for these respective body regions. They distilled the three teacher models into a smaller student model by contrastive representation distillation. As opposed to previous work, which learns from just visual data, we design an alternative encoder, PubMedCLIP, which not only uses natural language as supervision for visual representation learning, but also learns features in medical images of various modalities and diverse body organs, and hence, is not limited to only a few body regions.

3 PubMedCLIP

Our first step is to fine-tune the original general-domain CLIP using medical image–text pairs. We refer to the fine-tuned version as PubMedCLIP. Figure 1 (A) shows an overview of the training procedure for PubMedCLIP. Texts and images are encoded separately using CLIP, which we denote by $\mathbf{e}_t \in \mathbb{R}^{b \times d}$, $\mathbf{e}_v \in \mathbb{R}^{b \times d}$, respectively, for a batch of size b . For each image–text pair, a label $y \in \mathbb{R}$ represents the correspondence of the pairing of image and text. The cosine similarities between text and image features are computed to represent the respective visual and textual logits \hat{y}_v , \hat{y}_t , i.e.,

$$\hat{y}_v = \frac{\mathbf{e}_v^\top \mathbf{e}_t}{\|\mathbf{e}_v\| \|\mathbf{e}_t\|}, \quad \hat{y}_t = \frac{\mathbf{e}_t^\top \mathbf{e}_v}{\|\mathbf{e}_t\| \|\mathbf{e}_v\|}. \quad (1)$$

As formulated in Eq. 2, a weighted sum of the vision and language loss values is computed to

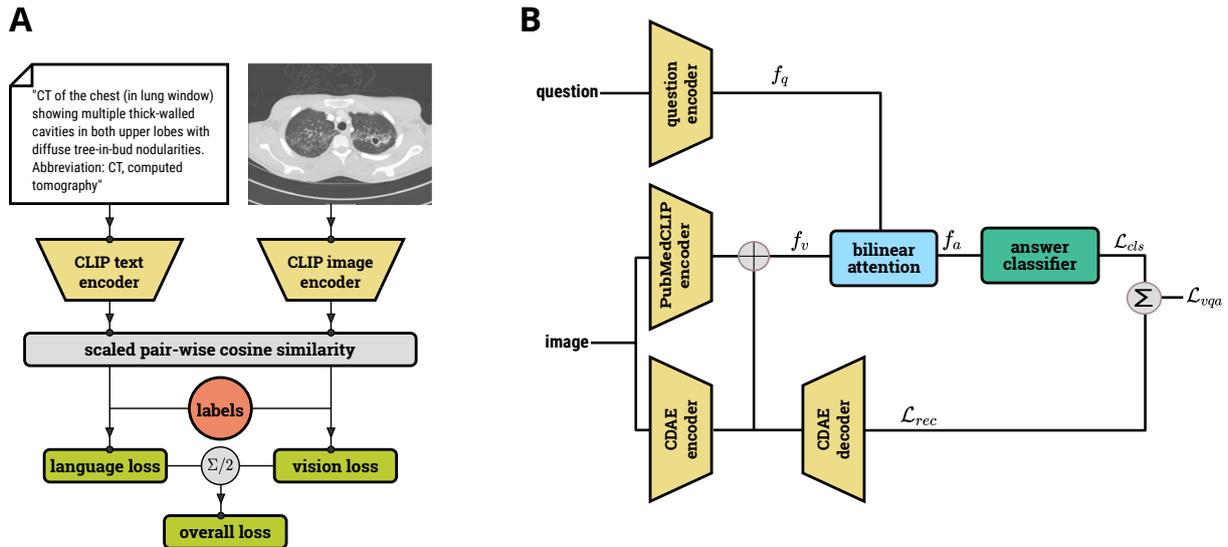


Figure 1: (A) Overview of how PubMedCLIP is pre-trained. (B) Schematic of MedVQA backbone with PubMedCLIP pre-trained visual encoder.

represent the overall loss. $Y \in \mathbb{R}^b$ denotes the set of labels y for a total of b image–text pairs in the batch. In this work, we use the cross-entropy loss

$$\mathcal{L} = \lambda H(\hat{y}_v, Y) + (1 - \lambda)H(\hat{y}_t, Y). \quad (2)$$

Following CLIP, we set $\lambda = 0.5$ to obtain the average of vision and language losses.

For training PubMedCLIP, we drew on the Radiology Objects in COntext (ROCO) dataset (Pelka et al., 2018). Previous work (Rajpurkar et al., 2017; Wang et al., 2017; Irvin et al., 2019; Johnson et al., 2019a) also proposes large-scale multi-modal datasets in the medical domain. However, they include images of only one imaging modality, i.e., X-ray, for a very limited number of body regions. In contrast, ROCO includes over 80K samples of diverse imaging modalities such as ultrasound, X-rays, PET scans, CT scans, MRI, angiography, from various human body regions, e.g., head, neck, spine, chest, abdomen, hand, foot, knee, and pelvis. Learning visual representations of diverse organs with various imaging modalities is valuable for a MedVQA system, as it is expected to interpret images given such diversities. The image–text pairs in ROCO stem from PubMed articles. The texts are taken from the relatively short captions (average length of 20 words) associated with images in the articles, which provide rich explanatory information about the content of images. In this work, the training and validation data splits from the original paper (Pelka et al., 2018) were used to train PubMedCLIP, with ViT-B/32 Vision

Transformer (Dosovitskiy et al., 2021), ResNet RN-50 (He et al., 2016), and RN-50x4 visual encoder back-ends. With respect to the maximum text length accepted by CLIP, which is 76, we trimmed any longer captions, while zero-padding shorter ones. PubMedCLIP was trained for 50 epochs with a batch size of 64, and Adam optimization (Kingma and Ba, 2014) with a learning rate of 10^{-5} . The trained models, source code as well as further implementation details are available online at <https://github.com/sarahESL/PubMedCLIP>.

Figures 2 and 3 show PCA visualizations of the caption and image embeddings, respectively, for the ROCO validation set. Comparing CLIP and PubMedCLIP embeddings, PubMedCLIP appears to obtain more semantic-aware visual and textual features with regard to body locations. For instance, looking at chest, abdomen, and head body locations, the corresponding embeddings form clusters for PubMedCLIP. However, the original CLIP embeddings are scattered without much separation.¹

4 PubMedCLIP for MedVQA

Given a MedVQA training dataset represented as $T = \{(v_i, q_i, a_i)\}_{i=1}^D$ of size D , where v_i is a medical image, q_i is the corresponding natural language question, and a_i is natural language answer, our goal is to learn to emit correct answer a_i given

¹In Appendix A, we provide more information on our approach for proxy-labeling the unannotated captions from the ROCO dataset. The proxy-labels have been merely used for the purpose of visualisations in this paper.

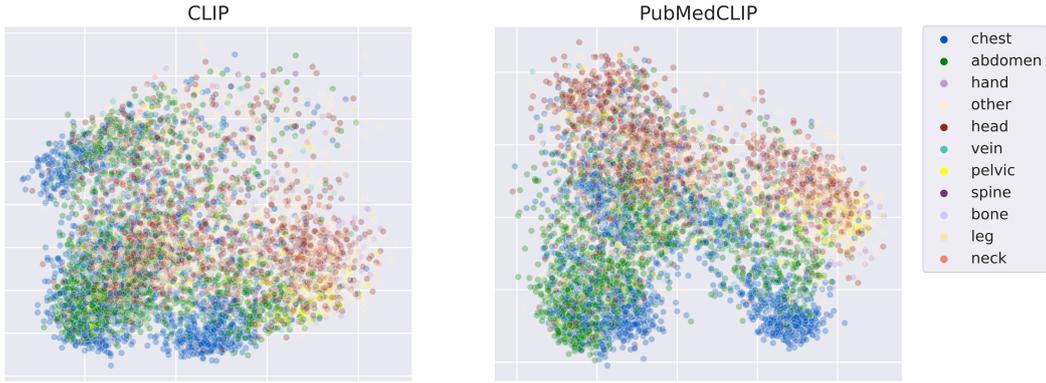


Figure 2: PCA visualizations of image embeddings.

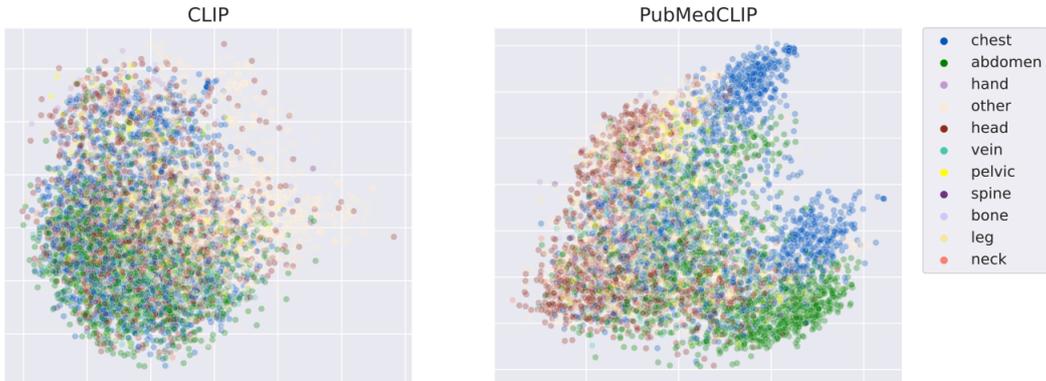


Figure 3: PCA visualizations of text embeddings.

image-question pair (v_i, q_i) . For this, we assume appropriate encoding functions to obtain $\mathbf{f}_v \in \mathbb{R}^n$ as an n -dimensional vector encoding for image v_i and the sequence embedding of $\mathbf{f}_q \in \mathbb{R}^{m \times l}$ for the question q_i with length l . We then cast MedVQA as a multi-label classification function $F : \mathbb{R}^n \times \mathbb{R}^{m \times l} \rightarrow \{0, 1\}^{|A|}$ where A is the overall set of possible answers and $F(\mathbf{f}_v, \mathbf{f}_q) = \mathbf{a}_i$ for the one-hot encoded answer \mathbf{a}_i .

Our goal is to investigate the effect of employing PubMedCLIP as the pre-trained visual encoder in MedVQA. To this end, we considered two prominent MedVQA methods, MEVF (Zhan et al., 2020) and QCR (Nguyen et al., 2019), that adopt MAML as their pre-trained visual encoder and GloVe word embeddings followed by a Recurrent Neural Network (RNN) as their question encoder. We substitute the pre-trained MAML module in MEVF and QCR with the pre-trained visual encoder from PubMedCLIP. A schematic architecture of our pipeline is shown in Figure 1 (B). The representative visual feature \mathbf{f}_v in this solution is the concatenation of the outputs of the PubMedCLIP network and the CDAE encoder. The objective of CDAE’s encoder

is to robustly encode the noisy version v'_i of an image v_i while the decoder learns to reconstruct the original non-noisy images. Denoting the reconstructed image as v_i^{rec} , Equation 3 defines the image reconstruction loss of CDAE as the mean squared error.

$$\mathcal{L}_{\text{rec}} = \|v_i - v_i^{\text{rec}}\|^2 \quad (3)$$

The multimodal pooling mechanism for combining \mathbf{f}_v and \mathbf{f}_q is BAN (Kim et al., 2018) to obtain the answer feature vector \mathbf{f}_a , as illustrated in Figure 1 (B). For answer prediction, which is a classification task in our case, a sigmoid layer preceding a binary cross-entropy loss is utilized in order to allow multiple correct answers per question. Eq. 4 formulates the answer classification loss function.

$$\mathcal{L}_{\text{cls}} = -\frac{1}{D} \sum_{i=1}^D \sum_{c=1}^A a_{i,c} \log(\hat{a}_{i,c}) + (1 - a_{i,c}) \log(1 - \hat{a}_{i,c}) \quad (4)$$

Here, $\hat{a}_{i,c} = \sigma(\mathcal{M}(\mathbf{f}_a))$, where σ represents the sigmoid function. Following BAN (Kim et al., 2018), the answer classifier \mathcal{M} is a two-layer feed-forward network with ReLU activation.

The objective of MedVQA is to simultaneously minimize the error of answer classification and image reconstruction, denoted as:

$$\mathcal{L}_{\text{vqa}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{rec}}. \quad (5)$$

5 Experiments

5.1 Datasets and Setup

We conducted our experiments using two well-known MedVQA datasets:

1. **VQA-RAD** (Lau et al., 2018) consists of 315 images and 3,515 English language question-answer pairs. Following previous work, we adopt the data split proposed in MEVF (Nguyen et al., 2019). We notice that all the images in the test dataset are also present in the training set. However, the set of question-answer pairs for these images in the test set are unseen in the training set.
2. The **SLAKE** (Liu et al., 2021b) dataset consists of English and Chinese questions. In this work, we utilize the English subset of the dataset, comprising 642 images and more than 7,000 question-answer pairs. Using the original data split, we observe that in contrast to VQA-RAD, all the images in the test set of SLAKE are unseen in the training set.

To ensure a fair comparison, our experiments followed the same setups used in the original MEVF and QCR studies. MEVF was trained for 20 epochs, QCR for 200, both with Adam optimization. When using PubMedCLIP as either the pre-trained visual encoder or the text encoder, we set the learning rate to 1×10^{-3} and 2×10^{-3} and the batch size to 16 and 32 in QCR and MEVF, respectively. All implementations are based on the PyTorch framework (Paszke et al., 2019). We ran the original MEVF and QCR on our machine and report the results here to have a fair comparison. Due to the non-deterministic behaviour of the cuDNN library in CUDA convolution operations (Pham et al., 2020), we observed non-deterministic results in different runs of the original MEVF and QCR. For a more robust comparison, we repeated all experiments 10 times and report the average accuracy scores.

5.2 Results and Analysis

The results of our experiments using PubMedCLIP’s visual encoder are given in Table 1. In

order to see the effectiveness of PubMedCLIP in comparison to the general domain CLIP, we also report the results when using CLIP. We provide the overall accuracy along with the accuracy of answering only open-ended or closed-ended questions.

When using CLIP and PubMedCLIP as the pre-trained visual encoder only, it is observed that the results of both the MEVF and QCR approaches improve. Furthermore, PubMedCLIP yields an absolute improvement of up to 1% in comparison with the original CLIP. On the VQA-RAD dataset, PubMedCLIP with the ResNet-50 backend achieves the best results, improving the overall accuracy of MEVF up to 6% and for QCR up to 3%. Results on the SLAKE dataset show that PubMedCLIP with ViT-B/32 Vision Transformer encoder back-end attains the best accuracy. It enhances MEVF by up to 3% and QCR up to 2%. We witness the same trend of improvement among overall, open-ended, and closed-ended accuracy scores.

In Figure 4, a comparison of image embeddings when using MAML as apposed to PubMedCLIP’s visual encoder is shown using PCA analysis for the VQA-RAD dataset. We find that in contrast to the MAML encoder, PubMedCLIP’s visual encoding results in organ-aware visual embeddings i.e., images of head, chest, and abdomen form more coherent and distinct clusters.

In Table 2, we compare the performance of PubMedCLIP with the recent state-of-the-art models in MedVQA. All the models use BAN as the fusion mechanism. In Table 2, PubMedCLIP refers to using PubMedCLIP as the pre-trained visual encoder in QCR. The comparison shows that PubMedCLIP achieves the best results on open-ended, closed-ended, and overall accuracies.

Behavior of visual encoder back-ends. The fact that PubMedCLIP with ResNet-50 back-end achieves the best results for VQA-RAD, while PubMedCLIP with ViT performs best on the SLAKE dataset points us to underlying differences in the question type distribution in these datasets. As Figure 5 shows, the majority of the questions in the VQA-RAD ask about the presence of an abnormality in the images. This requires the visual encoder to detect local features and local abnormalities. Thus, the CNN-based ResNet model with better visual localization outperforms the Vision Transformer. However, on SLAKE, the majority of questions are of the type “organ”, asking which organ is present in the image. For such cases, the

MedVQA Model	Question Encoder	Visual Encoder	VQA-RAD Accuracy			SLAKE Accuracy		
			Open	Closed	Overall	Open	Closed	Overall
MEVF	GloVe+RNN	MAML + AE (*)	42.1%	73.2%	60.8%	74.1%	77.5%	75.5%
		CLIP-ViT-B + AE	50.8%	75%	65.4%	75.8%	80.5%	77.7%
		CLIP-RN50 + AE	47%	77.4%	65.4%	75.7%	79.6%	77.2%
		CLIP-RN50x4 + AE	46.8%	76.6%	64.8%	75.9%	79.1%	77.2%
		PubMedCLIP-ViT-B + AE	48.9%	76.7%	65.5%	76.5%	80.4%	78%
		PubMedCLIP-RN50 + AE	48.6%	78.1%	66.5%	76.2%	79.9%	77.6%
		PubMedCLIP-RN50x4 + AE	47.1%	77.8%	65.6%	76.6%	79.1%	77.6%
QCR	GloVe+RNN	MAML + AE (+)	56%	77.9%	69.2%	76.8%	80.6%	78.3%
		CLIP-ViT-B + AE	57.6%	79.5%	70.7%	78.6%	81%	79.5%
		CLIP-RN50 + AE	58.3%	80%	71.3%	78.2%	81.5%	79.7%
		CLIP-RN50x4 + AE	59.9%	79.4%	71.3%	77.6%	80.5%	78.7%
		PubMedCLIP-ViT-B + AE	58.4%	79.5%	71.1%	78.4%	82.5%	80.1%
		PubMedCLIP-RN50 + AE	60.1%	80%	72.1%	77.8%	81.4%	79.3%
		PubMedCLIP-RN50x4 + AE	60%	79.7%	71.8%	77.7%	81.3%	79.1%

Table 1: Accuracy scores on VQA-RAD and SLAKE datasets. (*) denotes the original MEVF (Nguyen et al., 2019) and (+) denotes the original QCR (Zhan et al., 2020). Bold numbers represent the rows that achieved best overall accuracy. Light cyan, yellow, and green highlight correspond to the results when using MAML, CLIP and PubMedCLIP as the visual encoder only, respectively.

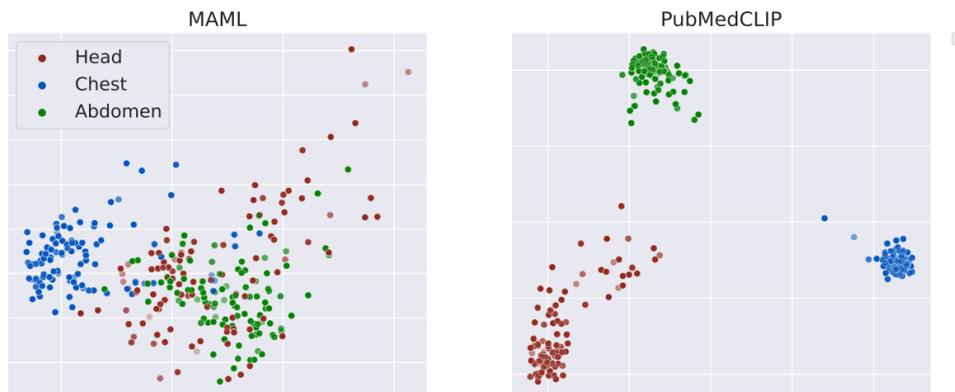


Figure 4: PCA visualizations of MAML and PubMedCLIP image embeddings for VQA-RAD dataset.

MedVQA Model	VQA-RAD Accuracy			SLAKE Accuracy		
	Open	Closed	Overall	Open	Closed	Overall
MEVF (Nguyen et al., 2019)	42.1%	73.2%	60.8%	74.1%	77.5%	75.5%
QCR (Zhan et al., 2020)	56%	77.9%	69.2%	76.8%	80.6%	78.3%
MMQ (Do et al., 2021)	53.7%	75.8%	67%	—	—	—
VQAMix (Gong et al., 2022)	56.6%	79.6%	70.4%	—	—	—
PubMedCLIP + BAN (ours)	60.1%	80%	72.1%	78.4%	82.5%	80.1%

Table 2: Comparison of PubMedCLIP with state-of-the-art MedVQA models. Results for the SLAKE dataset are not reported in the MMQ and VQAMix papers.

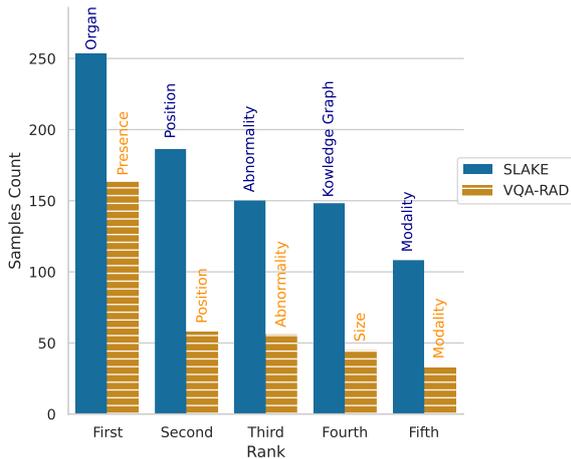


Figure 5: Distribution of top 5 most frequent question types in VQA-RAD and SLAKE.

visual encoder needs to be able to acquire a holistic overall understanding of the image and thus capture long-range dependencies of image patches. Vision Transformers indeed are capable of accounting for such features (Yu et al., 2021), and hence perform better on the SLAKE dataset.

PubMedCLIP as the text encoder. We expanded our experiments to investigate the effects of PubMedCLIP’s text encoder in MedVQA. To this end, we replaced the question encoder in the MEVF model with PubMedCLIP’s text encoder, i.e., instead of using GloVe word embeddings and an RNN network to model the question, we use PubMedCLIP’s text tokenizer and encoder, which receives the question q_i with l words and outputs a sequence-level embedding $\mathbf{f}_q \in \mathbb{R}^m$. Note that the size of image and text embeddings when using PubMedCLIP is equal. The results of our experiments in Table 3 suggest that invoking PubMedCLIP to encode questions in MedVQA is not as successful as using it for images. Furthermore, Table 3 shows that using both the visual and textual encoders of PubMedCLIP achieves absolute improvements of up to 5% in comparison to the original MEVF model. However, the best results are achieved with PubMedCLIP as the visual encoder together with GloVe+RNN for encoding questions.

In order to have a better understanding of the PubMedCLIP’s text encoder, a PCA visualization of the question embeddings is provided in Figure 6. The top row shows the embeddings when annotated according to their respective body location and the bottom row depicts them when labeled with question types, i.e., whether the question asks

Visual Encoder	Question Encoder	VQA-RAD Accuracy		
		Open	Closed	Overall
MAML	GloVe+RNN(*)	42.1%	73.2%	60.8%
	PubMedCLIP	26.5%	72.9%	54.3%
PubMedCLIP	GloVe+RNN	48.6%	78.1%	66.5%
	PubMedCLIP	48%	77.4%	65.6%

Table 3: Accuracy of PubMedCLIP as text encoder in the MEVF model. (*) denotes the original MEVF.

about the *Presence* of abnormality, *Position* of abnormality, type of *Abnormality*, etc. For having a comprehensible analysis, we visualize the top five frequent question types shown in Figure 5. Observations from Figure 6 suggest that PubMedCLIP’s text encoder emits organ-aware textual embeddings in contrast to GloVe+RNN. However, PubMedCLIP does not separate embeddings based on the question type, while GloVe+RNN results in better question type clusters. These findings suggest that question type awareness when encoding questions might be more beneficial than organ awareness for the MedVQA task. Based on our experiments, exploiting PubMedCLIP as the visual encoder in the QCR model is the most effective solution.

Furthermore, we sampled a few questions from the VQA-RAD test set and compared their pairwise cosine similarities when using GloVe+RNN versus PubMedCLIP encoding. We seek to examine the power of PubMedCLIP text encoder in identifying semantic differences. Figure 7 reports the cosine similarities when using PubMedCLIP in contrast to GloVe+RNN embeddings. As can be seen, when using PubMedCLIP text encoder, different questions about “lung abnormality” and “image plane” are equally similar to the “rib fracture” question, i.e., 0.77, and the encoder does not distinguish them. However, the cosine similarities are more intuitive when using GloVe+RNN. For instance, questions “Is there a rib fracture?” and “Describe the lung abnormalities?” have a small similarity of 0.27, while questions “Which plane is this image taken?” and “What is the plane of this image?” have a high similarity of 0.86.

In addition, it is observed that PubMedCLIP generally results in embeddings that are highly close to each other, with cosine similarities of more than 0.7 for different questions on disparate topics. In contrast, similarities of GloVe+RNN encoding are spread in the range of $[-0.09, 1]$, meaning that these embeddings are scattered over the m -

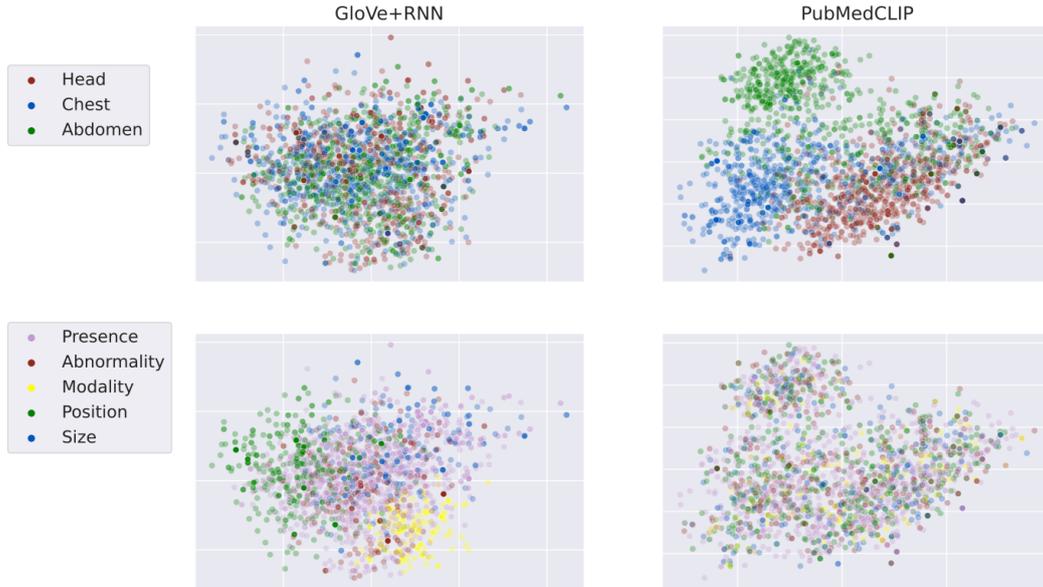


Figure 6: PCA of question embeddings. (Top) Labeled with body locations. (Bottom) Labeled with question types.

dimensional embedding space. We conclude that GloVe+RNN distinguishes the semantics of questions more effectively in comparison to PubMedCLIP’s text encoder for the MedVQA task.

CLIP versus PubMedCLIP. In order to better see the impact of fine-tuning PubMedCLIP, we additionally looked into the intermediate task of image–text matching using nearest neighbors vector retrieval. Considering that the pre-training objective in CLIP and PubMedCLIP is to minimize the cosine distance between paired image and text embeddings while maximizing this distance for non-paired image–text combinations, we argue that with a rich representation learning model, a nearest neighbor approach using the cosine distance metric should be fairly successful in retrieving matching image–text pairs. We randomly selected a subset of $D' = 10,000$ samples from the ROCO training data and used them to compare the outcomes of image–text matching in the medical domain. We exploit the text encoder as well as the visual encoder in CLIP and PubMedCLIP. Using Faiss (Johnson et al., 2019b) for vector retrieval, we investigated KNN with $K = 1$ on batches of size b . For each batch, the objective was to find the closest encoded text for a given encoded image, using the cosine distance metric. The evaluation metric for this setting is the overall accuracy of image–text matching over all batches:

$$\text{acc} = \frac{\sum_{i=1}^S \# \text{ correct matches in batch } i}{D'}, \quad (6)$$

V-L encoder	Batch size	ViT-B/32	RN50	RN50x4
CLIP	8	58.1%	49.1%	57.7%
	16	44%	36.1%	45.1%
	32	21.6%	25.5%	33.1%
PubMedCLIP	8	93.1%	89.2%	92.2%
	16	87.6%	81.1%	85.7%
	32	80.1%	70.6%	76.2%

Table 4: Accuracy scores of image-text matching using CLIP and PubMedCLIP vision–language encoders.

where $S = \lceil \frac{D'}{b} \rceil$. Table 4 summarizes the results for batch sizes of 8, 16, and 32. PubMedCLIP achieves over 40% improvement in comparison to CLIP across all batch sizes, with the ViT-B/32 back-end achieving the best results. This shows the effectiveness of our fine-tuning in PubMedCLIP.

Comparison of qualitative examples. In Figure 8, examples from the VQA-RAD and SLAKE datasets are provided that illustrate the performance of the original MEVF and QCR in comparison with PubMedCLIP, used here as either the visual or question encoder for QCR. PubMedCLIP_TE_VE, PubMedCLIP_TE and PubMedCLIP_VE refer to the scenarios of PubMedCLIP as both visual and textual encoders, as textual encoder only, and as the visual encoder only, respectively.

We find that the MEVF model often has difficulties discerning which organ is depicted in the image. For instance, regardless of the asked ques-

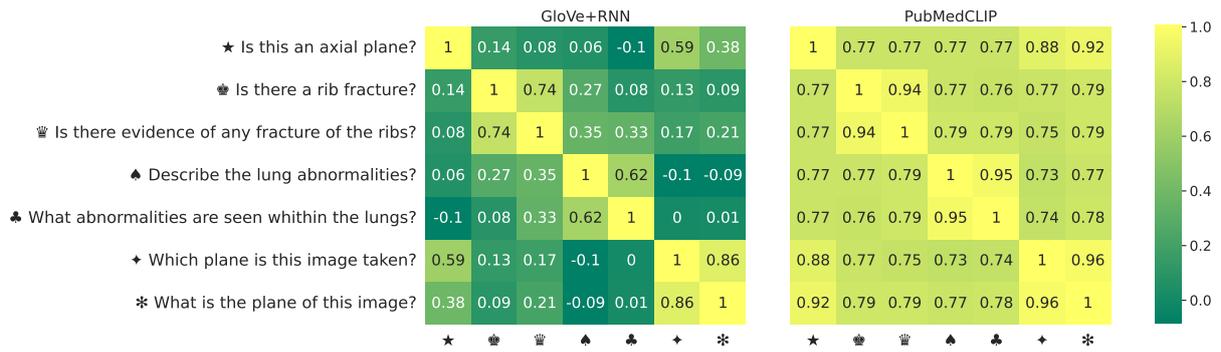


Figure 7: Pair-wise cosine similarities of questions from VQA-RAD encoded with GloVe+RNN compared with PubMedCLIP. Each question is associated with a symbol and represented only by the symbol on the horizontal axis.

	A	B	C
Question:	What are the bright white, structures, almost forming an X?	Where does the image represent in the body?	Are there multiple or just 1 metastatic focus?
Answer:	lateral ventricles	chest	one
MEVF:	chest tightness ... ✗	atelectasis, effusion ✗	right chest ✗
QCR:	extremities ✗	lower left lung ✗	no ✗
PubMedCLIP_TE_VE:	diffuse ✗	no ✗	yes ✗
PubMedCLIP_TE:	extremities ✗	no ✗	both sides ✗
PubMedCLIP_VE:	lateral ventricles ✓	chest ✓	yes ✗

Figure 8: (A) Example from VQA-RAD dataset. (B) Example from SLAKE dataset. (C) Example from VQA-RAD dataset that all models fail to answer correctly.

tion in Figure 8 (A), MEVF provides an answer related to the chest region, while the image is of the brain. This behaviour is also seen in Figure 8 (B) and 8 (C). From this perspective, QCR appears to be providing answers that are at least relevant to the given image. As Figure 8 (B) shows, the answer provided by QCR is related to the chest X-ray, although it is not a correct answer. Furthermore, it is observed that when PubMedCLIP is used as the question encoder, the model has difficulties providing the correct answers and often misinterprets open-ended questions as close-ended. In contrast, PubMedCLIP as the visual encoder successfully yields the correct answers.

Figure 8 (C) shows an example from the VQA-RAD that all models fail to answer correctly. MEVF again provides irrelevant answers about body organs not present in the image. QCR and PubMedCLIP misinterpret the question as a yes/no one. In spite of this, the fact that PubMedCLIP_VE answers with “yes” may illustrate that it has at least

detected the “one” metastatic focus in the image. In comparison, QCR answers with “no”, showing its troubles in interpreting the image and recognizing the metastatic focus. Figure 8 (C) reveals that these models still have shortcomings in understanding questions and correctly relating them to the images.

6 Conclusion

This work introduces PubMedCLIP, a pre-trained vision–language encoder for the medical domain trained via contrastive learning of medical image–caption pairs from PubMed articles. We demonstrated that PubMedCLIP results in organ-aware vision and language embeddings and evaluated its effectiveness for the task of MedVQA in comprehensive experiments across two heterogeneous MedVQA benchmarks. While PubMedCLIP’s text encoder is found to be less powerful for MedVQA, we showed that PubMedCLIP’s visual encoder outperforms previously used pre-trained visual encoders by up to 3%, leading to state-of-the-art results.

Limitations

Although we envision that in the long term, MedVQA systems can be sufficiently successful and trustworthy to aid medical practitioners towards better interpreting medical images and providing better healthcare, we emphasize that the development of these systems is still in its infancy stage and they are not yet ready for fully automated and unsupervised use in real-world clinical settings. Despite the notable improvement of accuracy in MedVQA brought by PubMedCLIP, further evaluations of these models from the vantage points of scalability, trustworthiness, explainability, and generalizability are required before they can be deployed for sensitive clinical tasks. In future work, we plan to perform further analysis of these models using explainable AI techniques such as Grad-CAM visualizations to assess the regions of focus within the image from the class activation maps. Furthermore, due to a lack of suitable data to train large-scale models for other languages, our current experiments are limited to English language MedVQA, so different findings may be observed for typologically different languages. By releasing PubMedCLIP, we hope to enable further research investigating these aspects as well as its effectiveness in other use cases, e.g., image classification for medical diagnosis and radiology report generation.

Discussions on Ethics

As remarked above, MedVQA models are still in their early stages of development and have limitations that should be considered before being used in any real-world scenarios.

Acknowledgements

The authors acknowledge the financial support by the German Federal Ministry for Education and Research (BMBF) within the project »KI-Servicezentrum Berlin Brandenburg« 01IS22092.

References

- Asma Ben Abacha, Vivek V Datla, Sadid A Hasan, Dina Demner-Fushman, and Henning Müller. 2020. Overview of the VQA-Med task at ImageCLEF 2020: Visual question answering and generation in the medical domain. In *CLEF (Working Notes)*.
- Asma Ben Abacha, Mourad Sarroui, Dina Demner-Fushman, Sadid A Hasan, and Henning Müller. 2021. Overview of the VQA-Med task at ImageCLEF 2021:

Visual question answering and generation in the medical domain. In *CLEF (Working Notes)*.

- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR.
- Tuong Do, Binh X Nguyen, Eрман Tjiputra, Minh Tran, Quang D Tran, and Anh Nguyen. 2021. Multiple meta-model quantifying for medical visual question answering. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 64–74. Springer.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Sedigheh Eslami, Gerard de Melo, and Christoph Meinel. 2021. [TeamS at VQA-Med 2021: BBN-Orchestra for long-tailed medical visual question answering](#). In *Working Notes of CLEF 2021*, number 2936 in CEUR Workshop Proceedings, pages 1211–1217. CEUR-WS.
- Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.
- Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Conference on Empirical Methods in Natural Language Processing*, pages 457–468. ACL.
- Haifan Gong, Guanqi Chen, Mingzhi Mao, Zhen Li, and Guanbin Li. 2022. VQAMix: Conditional triplet mixup for medical visual question answering. *IEEE Trans. on Medical Imaging*.
- Haifan Gong, Ricong Huang, Guanqi Chen, and Guanbin Li. 2021. SYSU-HCP at VQA-Med 2021: A data-centric model with efficient training methodology for medical visual question answering. In *CLEF 2021 Working Notes*, volume 201. CEUR-WS.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. [Deep residual learning for image recognition](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

- Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silvana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. 2019. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597.
- Alistair EW Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. 2019a. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 6(1):1–8.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019b. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. *Advances in neural information processing systems*, 31.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):1–10.
- Li Lei, Haogang Zhu, Yuxin Gong, and Qian Cheng. 2018. A deep residual networks classification algorithm of fetal heart CT images. In *2018 IEEE international conference on imaging systems and techniques (IST)*, pages 1–4. IEEE.
- Bo Liu, Li-Ming Zhan, and Xiao-Ming Wu. 2021a. Contrastive pre-training and representation distillation for medical visual question answering based on radiology images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 210–220. Springer.
- Bo Liu, Li-Ming Zhan, Li Xu, Lin Ma, Yan Yang, and Xiao-Ming Wu. 2021b. SLAKE: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1650–1654. IEEE.
- Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks*, pages 52–59. Springer.
- Binh D Nguyen, Thanh-Toan Do, Binh X Nguyen, Tuong Do, Erman Tjiputra, and Quang D Tran. 2019. Overcoming data limitation in medical visual question answering. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 522–530. Springer.
- Haiwei Pan, Shuning He, Kejia Zhang, Bo Qu, Chunling Chen, and Kun Shi. 2021. MuVAM: A multi-view attention-based model for medical visual question answering. *arXiv preprint arXiv:2107.03216*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Obioma Pelka, Sven Koitka, Johannes Rückert, Felix Nensa, and Christoph M Friedrich. 2018. Radiology Objects in COntext (ROCO): a multimodal image dataset. In *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, pages 180–189. Springer.
- Hung Viet Pham, Shangshu Qian, Jiannan Wang, Thibaud Lutellier, Jonathan Rosenthal, Lin Tan, Yao-liang Yu, and Nachiappan Nagappan. 2020. Problems and opportunities in training deep learning software systems: an analysis of variance. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, pages 771–783.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.
- Pranav Rajpurkar, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang, Kaylie Zhu, Dillon Laird, Robyn L Ball, et al. 2017. Mura: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv preprint arXiv:1712.06957*.
- Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can CLIP benefit vision-and-language tasks? In *International Conference on Learning Representations*.
- Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.
- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. VL-BERT: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*.
- Minh H Vu, Tommy Löfstedt, Tufve Nyholm, and Raphael Sznitman. 2020. A question-centric model for visual question answering in medical imaging. *IEEE transactions on Medical Imaging*, 39(9):2856–2868.

- Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106.
- Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021. SimVLM: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*.
- Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29.
- Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. CoCa: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.
- Qihang Yu, Yingda Xia, Yutong Bai, Yongyi Lu, Alan L Yuille, and Wei Shen. 2021. Glance-and-gaze vision transformer. *Advances in Neural Information Processing Systems*, 34:12992–13003.
- Li-Ming Zhan, Bo Liu, Lu Fan, Jiabin Chen, and Xiaoming Wu. 2020. Medical visual question answering via conditional reasoning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2345–2354.
- Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. 2020. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*.

A Proxy-labeling ROCO dataset for visualization purposes

In order to have a better analysis of the PCA visualizations when comparing CLIP and PubMedCLIP encodings, we created proxy body location labels by identifying organ-specific keywords in ROCO captions. The complete list of keywords used for each body location is provided in Listing 1. Furthermore, the distribution of these proxy labels in the ROCO validation dataset is shown in Figure 9.

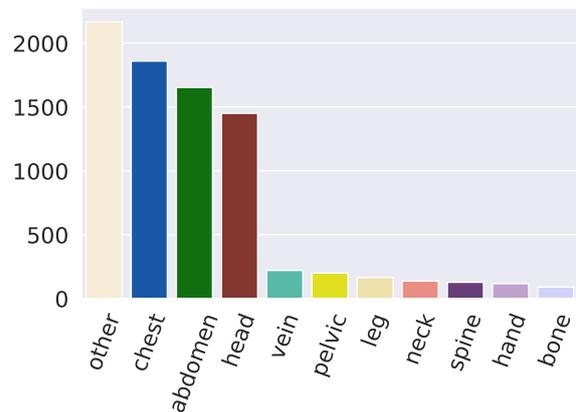


Figure 9: Distribution of proxy labels in ROCO.

```

1 chest = ['breast', 'lung', 'rib', 'thoracotomy', 'pulmonary', 'mediastinal',
2         'bronchus', 'bronchoscopic', 'bronchiectasis', 'bronchial',
3         'tuberculosis', 'heart', 'ventricle', 'myocardial', 'valve',
4         'thorax', 'thoracic', 'echocardiogram', 'echocardiography',
5         'angioplasty', 'diaphragm', 'coronary', 'cardiac', 'coronaries',
6         'thoracique', 'chest', 'mitral annulus', 'empyema']
7 #####
8 abdomen = ['gastro-oesophageal', 'gastrointestinal', 'gastric',
9           'abdomen', 'abdomenal', 'abdominal', 'bowel', 'colon', 'liver',
10          'kidney', 'renal', 'stomach', 'ventral', 'esophagus', 'pancreas',
11          'pancreatic', 'pancreatitis', 'hernia', 'bladder', 'gallstones',
12          'gallbladder', 'spleen', 'splenic', 'appendi', 'intestine',
13          'duodenum', 'ileum', 'jejunum', 'rectum', 'ovary', 'uterus',
14          'vagina', 'cervix', 'pregnancy', 'cervical', 'prostate', 'penis',
15          'testicle', 'testis', 'testicular', 'urethrogram', 'urethra',
16          'ureteral', 'ureter', 'peritoneum']
17 #####
18 head = ['head', 'skullbase', 'skull', 'zygoma', 'parieto-occipital',
19         'parietooccipital', 'parieto occipital', 'cerebellar', 'cerebellum',
20         'brain', 'caudate nucleus', 'caudate', 'ear', 'auditory canal',
21         'facial', 'eye', 'sinus', 'gland', 'temporal lobe', 'frontal lobe',
22         'frontal bone', 'parietal bone', 'parietal lobe', 'occipital lobe',
23         'lymph', 'nose', 'nasal', 'mouth', 'tongue', 'cheek', 'jaw',
24         'root canal', 'tooth', 'teeth', 'obturation', 'periapical', 'premolars',
25         'dental', 'parotid', 'orthopantomograph', 'orthopantomogram',
26         'myelinolysis']
27 #####
28 neck = ['neck', 'throat', 'theroid', 'thyroid', 'carotid']
29 #####
30 spine = ['foraminal', 'spine', 'disk', 'disc', 'spinal', 'lumbosacral',
31         'thoracic spine', 'lubmar']
32 #####
33 pelvic = ['pelvic', 'pelvis', 'hip', 'perineum', 'iliac', 'gluteal']
34 #####
35 hand = ['arm', 'shoulder', 'elbow', 'wrist', 'hand', 'nail', 'finger',
36         'humerus', 'thumb']
37 #####
38 leg = ['tibias', 'leg', 'thigh', 'foot', 'feet', 'talus', 'toe', 'knee',
39        'calcaneus', 'fibula', 'femur', 'femoral', 'femural', 'prosthesis',
40        'prostheses', 'limb']
41 #####
42 vein = ['vein', 'vessel', 'vascular', 'artery', 'angioplasty', 'angiography',
43         'artial', 'aorta', 'aortogram']
44 #####
45 bone = ['bone']

```

Listing 1: Proxy-label keywords

Multilingual BERT has an accent: Evaluating English influences on fluency in multilingual models

Isabel Papadimitriou* and Kezia Lopez* and Dan Jurafsky

Computer Science Department

Stanford University

{isabelvp,keziak1,jurafsky}@stanford.edu

Abstract

While multilingual language models can improve NLP performance on low-resource languages by leveraging higher-resource languages, they also reduce average performance on all languages (the ‘curse of multilinguality’). Here we show another problem with multilingual models: grammatical structures in higher-resource languages bleed into lower-resource languages, a phenomenon we call *grammatical structure bias*. We show this bias via a novel method for comparing the fluency of multilingual models to the fluency of monolingual Spanish and Greek models: testing their preference for two carefully-chosen variable grammatical structures (optional pronoun-drop in Spanish and optional Subject-Verb ordering in Greek). We find that multilingual BERT is biased toward the English-like setting (explicit pronouns and Subject-Verb-Object ordering) as compared to our monolingual control language model. With our case studies, we hope to bring to light the fine-grained ways in which multilingual models can be biased, and encourage more linguistically-aware fluency evaluation.

1 Introduction

Multilingual language models share a single set of parameters between many languages, opening new pathways for multilingual and low-resource NLP. However, not all training languages have an equal amount, or a comparable quality of training data in these models. In this paper, we investigate if the hegemonic status of English influences other languages in multilingual language models. We propose a novel method for evaluation, whereby we ask if model predictions for lower-resource languages exhibit structural features of English. This is similar to asking if the model has learned some languages with an “English accent”, or an English *grammatical structure bias*.

We demonstrate this bias effect in Spanish and Greek, comparing the monolingual models BETO

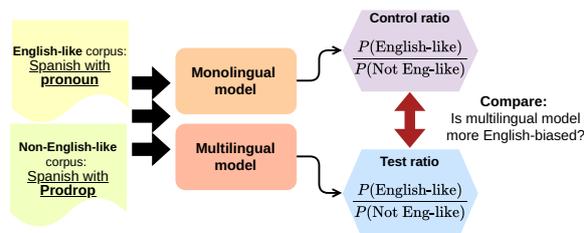


Figure 1: Our method for evaluating English structural bias in multilingual models. We compare monolingual and multilingual model predictions on two sets of natural sentences in the target language: one which is structurally parallel to English, and one which is not.

(Cañete et al., 2020) and GreekBERT (Koutsikakis et al., 2020) to multilingual BERT (mBERT), where English is the most frequent language in the training data. We show that *mBERT prefers English-like sentence structure in Spanish and Greek* compared to the monolingual models. Our case studies focus on Spanish pronoun drop (pro-drop) and Greek subject-verb order, two structural grammatical features. We show that multilingual BERT is structurally biased towards explicit pronouns rather than pro-drop in Spanish, and subject-before-verb order in Greek: the structural forms parallel to English.

Though the effect we showcase here is likely not captured by the downstream classification tasks often used to evaluate multilingual models (Hu et al., 2020), it demonstrates the type of fluency that can be lost with multilingual training — something that current evaluation methods miss. In fact, though we choose two clear-cut syntactic features to investigate, there are many less-measurable features that make language production fluent: subtleties in lexical choice, grammatical choice, and discourse expression, among many others. With this paper, beyond showing a trend for two specific grammatical features, we wish to highlight fluency discrepancies in multilingual models, and also call for more evaluations focused on fluency.

S_{parallel} : English-like structure	$S_{\text{different}}$: Different structure
Spanish explicit pronoun (pron in orange, verb in blue)	Spanish prodrop (verb in blue)
Yo <u>volveré</u> para averiguarlo I will return to figure it out El 2004, <u>ella</u> <u>hizo</u> doblaje a el Inglés [...] In 2004, <u>she</u> <u>did</u> dubbing to English [...] <u>Ella</u> <u>decide</u> pasar sus vacaciones en la granja <u>She</u> <u>decides</u> to spend her vacation in the country	Jamás <u>dan</u> soluciones y siempre [...] [They] Never <u>give</u> solutions and always [...] <u>Jugó</u> de centrocampista en el Real Zaragoza [He/She/You] <u>Played</u> as a midfielder in Real Zaragoza <u>Habita</u> en Perú. [He/She/You] <u>Lives</u> in Peru
Greek Subject-Verb (subject in orange, verb in blue)	Greek Verb-Subject (subject in orange, verb in blue)
<u>Πηγές</u> της Αντιπολίτευσης <u>αναφέρουν</u> ότι [...] Sources of the Opposition <u>mention</u> that [...] Σε άλλες πλευρές ο <u>ποταμός</u> <u>κυλά</u> από ψηλούς βράχους On other sides, the <u>river</u> <u>flows</u> from tall boulders Η <u>εκπαίδευση</u> και η μόρφωση <u>απέκτησαν</u> επιτέλους προτεραιότητα Training and education have finally <u>acquired</u> priority	Το σκορ του αγώνα <u>άνοιξε</u> ο <u>Γουέν</u> Ρούνι The score of the game <u>opened</u> <u>Wayne</u> Rooney Εδώ πρέπει να <u>γίνουν</u> μεγαλύτερες <u>προσπάθειες</u> . Here must <u>happen</u> bigger <u>efforts</u> Απασχόληση στο εξωτερικό <u>ψάχνουν</u> οι μισοί <u>Έλληνες</u> σε παραγωγική ηλικία Employment in foreign countries <u>search</u> half of <u>Greeks</u>

Table 1: Examples from our dataset for S_{parallel} and $S_{\text{different}}$ in Spanish and Greek, along with roughly word-by-word gloss translations in English. In all cases, we’ve underlined $w(x)$, the word we use to represent the construction in our calculations. These examples are not randomly selected and have been chosen to be significantly shorter than the average sentence in our datasets in order to be presentable in a table.

Our proposed method can be expanded, without the need for manual data collection, to any language with a syntactic treebank and a monolingual model. Since our method focuses on fine-grained linguistic features, some expert knowledge of the target language is necessary for evaluation. Multilingual evaluation so far has been largely translated or automatically curated, and the methods for creating such datasets have allowed for the creation of resources in many languages for which there were none. Fluency evaluation requires some linguistic expertise to set up, and as such is more restricted in the languages the research community can reach. Nevertheless, such evaluation has been missing from the multilingual NLP literature, and our work bridges this gap by proposing fluency testing for multilingual models.

Our work builds off of a long literature on multilingual evaluation which has until now mostly focused on downstream classification tasks (Conneau et al., 2018; Ebrahimi et al., 2022; Clark et al., 2020; Liang et al., 2020; Hu et al., 2020; Raganato et al., 2020; Li et al., 2021). With the help of these evaluation methods, research has pointed out the problems for both high- and low-resource languages that come with adding many languages to a single model (Wang et al., 2020; Turc et al., 2021;

Lauscher et al., 2020, inter alia). Methods for creating more equitable models have been proposed, through identifying or reserving language-specific parameters for each language (Ansell et al., 2022; Pfeiffer et al., 2022), through training models without typologically distant languages that dominate the training data (Ogueji et al., 2021; Virtanen et al., 2019; Ògúnrè mí and Manning, 2023), as well as through adding model capacity (Conneau et al., 2020; Xue et al., 2021; Lepikhin et al., 2021; Liang et al., 2023). We hope that our work can add to these analyses and methodologies by pointing out issues beyond downstream classification performance that can arise with multilingual training, and aid towards building and evaluating more equitable multilingual models.

2 Method

Our method relies on finding a variable construction in the target language which can take two structural surface forms: one which is parallel to English (S_{parallel}) and one which is not ($S_{\text{different}}$). Surface forms parallel to English are those which mirror English structure. For example, English has strict Subject-Verb-Object word order, and so a *parallel* structure in another language is one where the verb

and its arguments appear in Subject-Verb-Object order, while a *different* structure is one where the verb appears before the subject (see Table 1 for examples).

Once we have identified such a construction in our target language, we can ask: are multilingual models biased towards S_{parallel} ? For a native speaker of the target language, structural, semantic, and discourse features determine whether they will use S_{parallel} or $S_{\text{different}}$ in a given context — with the alternative option usually being less fluent. We assume that a BERT-sized monolingual model in the target language will have a sufficiently accurate representation of this fluent variation between S_{parallel} and $S_{\text{different}}$ without being influenced by other languages. Therefore, to understand if multilingual models have an English structural bias, we now just have to answer: do multilingual models prefer S_{parallel} over $S_{\text{different}}$ *more* than the fluent distribution defined by a monolingual model?

2.1 Collecting model judgements

By design, both S_{parallel} and $S_{\text{different}}$ are constructions that occur naturally in the target language. Therefore, we should be able to use the syntactic treebank annotations to pick out sentences that exhibit the structures S_{parallel} or $S_{\text{different}}$. We can put these extracted sentences into two corpora, C_{parallel} and $C_{\text{different}}$. Note that the sentences in C_{parallel} and $C_{\text{different}}$ are unrelated and not paired, and that the two corpora can have different sizes. Crucially, we have to use natural sentences for both of our corpora: we cannot artificially alter sentences from S_{parallel} to $S_{\text{different}}$, or use templates to create sentences. This is because our evaluation is about the subtleties of fluency, while altered or templated stimuli are not naturally produced and are therefore often awkward, confounding any effect we might want to measure.

Each model gives us a ratio r_{model} : the average probability of a sentence in C_{parallel} divided by the average probability of a sentence in $C_{\text{different}}$ according to the model. That is:

$$r_{\text{model}} = \frac{\sum_{x \in C_p} P_{\text{model}}(x) / |C_p|}{\sum_{x \in C_d} P_{\text{model}}(x) / |C_d|} \quad (1)$$

We want to compare judgements on these corpora from two models: a monolingual model `mono` and a multilingual model `multi`. Our experimental question then boils down to asking if r_{multi} is significantly larger than r_{mono} .

2.2 From model outputs to construction probability

How can we calculate $P_{\text{model}}(x)$ for a given sentence x , focusing on the probability of a specific construction in x ? Looking at model judgements over long natural sentences introduces a lot of noise that is unrelated to the structural construction in question, reducing the statistical power of our experiment. Furthermore, since we are looking at encoder-only bidirectional models, there is no canonical or controlled way of extracting the probability of a whole sentence. To get a better model judgement for each sentence, we can extract the probability of *one word* in each sentence that best represents the construction. For example, if we are looking at pronoun drop, it makes sense to use main verb of the sentence as the target word, as this is the syntactic head of the pronoun that is present or dropped. Using a carefully chosen word as a proxy for the probability of a construction is a methodological choice also made in reading time psycholinguistics experiments (Levy, 2011; Levy and Keller, 2013).

Going back to our problem of calculating $P_{\text{model}}(x)$, we define w to be a function that returns the structurally-relevant word from each sentence. Using this, we approximate $P_{\text{model}}(x)$ in Eq. (1) with $P_{\text{model}}(w(x)|x)$. The probability $P(w(x)|x)$ is simple to calculate for BERT-style masked language models: it is simply the logit of the word $w(x)$ when we encode the sentence x using `model`.

2.3 Extending to more languages

Extending our fluency evaluation to a new language requires three language-specific steps: (1) decide on an appropriate construction with two structural forms S_{parallel} and $S_{\text{different}}$, (2) decide on an appropriate $w(x)$: which word in each structural form can represent the form, and (3) use treebank annotations to pull out sentences which exhibit S_{parallel} or $S_{\text{different}}$, and identify the relevant word. Below, we detail these steps for our two case studies.

2.4 Case Study: Spanish Pro-drop

In Spanish, the subject pronoun is often dropped: person and number are mostly reflected in verb conjugation, so the pronoun is realized or dropped depending on semantic and discourse factors. English, on the other hand, does not allow null subjects except in rare cases, and expletive syntactic subjects like “there” are even added when there

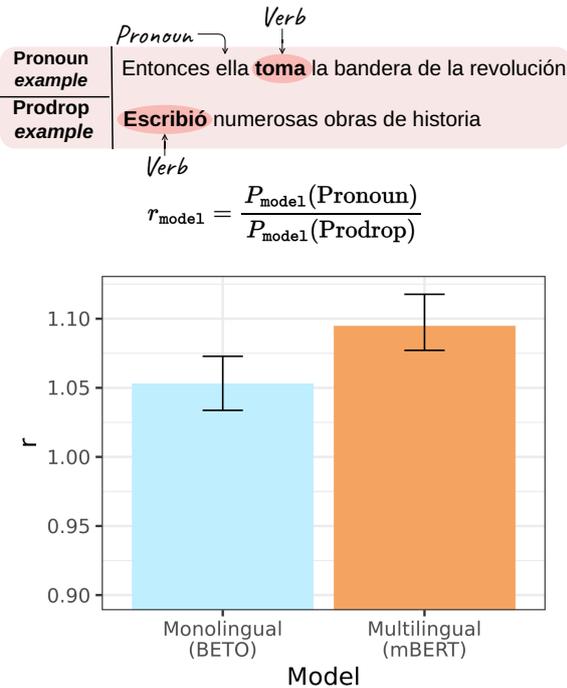


Figure 2: Results from our experiment on the Spanish GSD treebank, along with two examples from the treebank to illustrate S_{parallel} (with pronoun) and $S_{\text{different}}$ (pro-drop). We compare model logits for the main verb of the sentence, which is bold and highlighted in the examples. Error bars represent 95% bootstrap confidence intervals. We find that r_{mono} is significantly smaller than r_{multi} (bootstrap sampling, $p < 0.05$).

is no clear subject. For our Spanish experiment, we define S_{parallel} to be sentences which have the subject pronoun of the main verb, as is necessary in English, and $S_{\text{different}}$ to be pro-drop sentences which have a main verb with no realized subject. We define w to be the main verb of the sentence, which is always present in our extracted examples.

To extract our corpora C_{parallel} and $C_{\text{different}}$, we use the Spanish GSD treebank from the Universal Dependencies dataset (De Marneffe et al., 2021). We ignore all sentences not verb-rooted (i.e. noun phrases), those rooted with “haber” (which in its copula-like existential form cannot take an explicit subject, “There is” in English), and those using the impersonal-“se” passive construction (e.g. “se nos fue permitido”, “it was permitted of us”). We then take all sentences with a pronoun subject (i.e. a pronoun dependent of the root verb) and add them to C_{parallel} and all sentences where there is no nsubj relation to root verb and add them to $C_{\text{different}}$. We always pick the main root verb of the sentence as our w . We collect 283 sentences in C_{parallel} and 2,656 sentences in $C_{\text{different}}$.

2.5 Case Study: Greek Subject-Verb order

English is a fixed word order language: with few exceptions, the order of a verb and its arguments is Subject-Verb-Object. Greek, on the other hand, has mostly free word order (Mackridge, 1985), meaning that the verb and arguments can appear in any order that is most appropriate given discourse context. For our experiment, we define S_{parallel} to be cases in Greek when the subject precedes the verb, as is the rule in English. $S_{\text{different}}$ is then the cases when the verb precedes the subject, which almost never happens in English.

We define w to be the first element of the subject and verb: the subject when the subject comes first or the verb when the verb comes first. This first element is closer to the surrounding context, and so gives us a word-order-sensitive measurement of how the subject-verb construction is processed as a whole within the context. Though this choice means that our w is a noun in S_{parallel} and a verb in $S_{\text{different}}$, this does not constitute a confounder between models: we are comparing the same noun-verb probability ratio between different models.

To extract our corpora C_{parallel} and $C_{\text{different}}$, we use the Greek Dependency Treebank, the Universal Dependencies treebank for Greek (Prokopidis and Papageorgiou, 2017). We take all sentences where the main verb has a lexical subject, and we add to C_{parallel} if the subject appears before the verb and to $C_{\text{different}}$ if it appears after. We collect 1,446 sentences in C_{parallel} and 425 sentences in $C_{\text{different}}$.

3 Results

Results are shown in Figures 2 and 3, showing for both of our case studies that multilingual BERT has a greater propensity for preferring English-like sentences which exhibit S_{parallel} . Multilingual BERT significantly prefers pronoun sentences over pro-drop compared with monolingual BETO (bootstrap sampling, $p < 0.05$), and significantly prefers subject-verb sentences over verb-subject sentences over GreekBERT (bootstrap sampling, $p < 0.05$).

4 Discussion

In this paper, we proposed fluency evaluation as a further way of understanding the curse of multilinguality: what can be lost when we train many languages together. The discrepancies that we point out in these experiments are not going to seriously affect multilingual LM performance, especially in the more coarse-grained classification tasks that

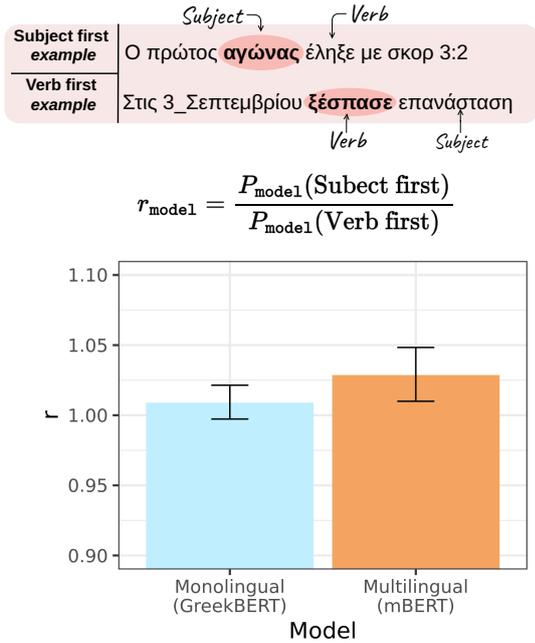


Figure 3: Results from our experiment on the Greek Dependency Treebank, along with two examples from the treebank to illustrate S_{parallel} (Subject-Verb) and $S_{\text{different}}$ (Verb-Subject). We measure and compare model logits for the bold words: the subject in subject-verb sentences and the verb in verb-subject sentences. Error bars represent 95% bootstrap confidence intervals. r_{mono} is significantly smaller than r_{multi} (bootstrap sampling, $p < 0.05$).

are most commonly used for evaluation. But, as we demonstrate here, not all levels of language learning can be evaluated from such datasets.

Our experiments do not pinpoint the reasons behind the effects that we measure: there are different possible explanations for the English-like trends that we showcase. On the one hand, the effects we measure might stem from training with a language that’s more dominant in the training data, like English is for many multilingual models. Such training could lead to an English-biased representation space which the representations of other languages conform to. On the other hand, the effects we show might be down to the data: the non-English datasets used to train a multilingual model may be more limited in domain, may contain a high proportion of data that’s actually been translated from English (Multilingual Wikipedia is often translated, Adar et al., 2009), or might be more polluted with irrelevant or non-linguistic elements. Domain limitations and translationese stemming from the data are separate, but related issues to fluency: fluency can be grammatical, but

also involves proficiency in a range of registers or possibilities. It is also possible that the effects we show are due to a combination of both multilingual representation learning artifacts, and training data quality. Further controlled fluency experimentation on the limits and abilities of multilingual models is needed to disentangle these effects. We hope the case studies in this paper can inspire more fine-grained evaluation of multilingual models, so that we understand the “accent”-like effects of hegemonic languages more fully.

5 Limitations

This study is meant to highlight the kinds of modeling flaws that have so far gone undetected and that can arise for lower-resource languages in multilingual models. However, our study does not focus on languages that are truly low-resource. In fact, as designed it could not do so: our methodology relies on having an available monolingual model, which of course requires a large amount of training data. This is because our method requires a control: we can only judge multilingual models against what we can believe to be a non-biased language model in the language. There are ways to test for fluency in low-resource languages that would not require a monolingual model as a control, but would require dataset collection in the target language for features that reflect fluency and linguistic acceptability (similar to what Warstadt et al. (2019) achieve with the CoLA dataset for English). We hope our study can create motivation for such work in linguistically-aware, fine-grained multilingual evaluation for languages of all resource levels.

Our experiments focus on BERT-style models, since this is mostly the size of model available for monolingual, non-English models (in our case BETO and GreekBERT). However, it is not necessary from these experiments that our findings extrapolate to larger models that are commonplace at the time of writing.

Lastly, both pro-drop and subject-verb order are largely discourse-dependent constructions. For example, pro-drop is more likely when the subject of the sentence is very clear from the discourse, while subject-verb order in Greek is changed to achieve different discourse focus, similar to how intonation changes the focus of a sentence in English (e.g., stressing the verb in “Mary helped John” puts the focus on the verb, which in Greek can be done by putting the verb first). Despite this, all of our ex-

periments are done on isolated sentences from the UD treebanks and do not contain discourse content. Though this means that the models do not have the full relevant context for each input, we do not expect that having more context should favor one model more than another for our evaluation. Since this work compares models on the same inputs, we did not consider this a significant confounder.

6 Acknowledgements

We thank Anjalie Field and Mirac Suzgun for comments on drafts. This research was funded in part by NSF award number IIS-2128145.

References

- Eytan Adar, Michael Skinner, and Daniel S. Weld. 2009. [Information arbitrage across multi-lingual wikipedia](#). In *Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM '09*, page 94–103, New York, NY, USA. Association for Computing Machinery.
- Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796.
- José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Joun-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish Pre-trained BERT Model and Evaluation Data. In *PMLADC at ICLR 2020*.
- Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.
- Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational linguistics*, 47(2):255–308.
- Abteem Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios Gonzales, Ivan Meza-Ruiz, et al. 2022. AmericasNLI: Evaluating Zero-shot Natural Language Understanding of Pre-trained Multilingual Models in Truly Low-resource Languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.
- John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. Greek-BERT: The Greeks visiting sesame street. In *11th Hellenic Conference on Artificial Intelligence*, pages 110–117.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From Zero to Hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499.
- Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. [{GS}hard: Scaling giant models with conditional computation and automatic sharding](#). In *International Conference on Learning Representations*.
- Roger Levy. 2011. Integrating surprisal and uncertainty in online sentence comprehension: Formal techniques and empirical results. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1055–1065.
- Roger P. Levy and Frank Keller. 2013. [Expectation and locality effects in German verb-final structures](#). *Journal of Memory and Language*, 68(2):199–222.
- Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962.
- Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. Xlm-v: Overcoming the vocabulary bottleneck in multilingual masked language models. *arXiv preprint arXiv:2301.10472*.
- Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroan

- Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. [XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.
- P. Mackridge. 1985. *The Modern Greek Language: A Descriptive Analysis of Standard Modern Greek*. Oxford University Press.
- Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. [Small Data? No Problem! Exploring the viability of pretrained multilingual language models for low-resourced languages](#). In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Tolúloṣé Ògúnṛẹ̀mí and Christopher D. Manning. 2023. [Mini but Mighty: Efficient multilingual pretraining with linguistically-informed data selection](#). In *Findings of EACL 2023*.
- Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. [Lifting the curse of multilinguality by pre-training modular transformers](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.
- Prokopis Prokopidis and Haris Papageorgiou. 2017. [Universal Dependencies for Greek](#). In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 102–106, Gothenburg, Sweden. Association for Computational Linguistics.
- Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. [XLWiC: A multilingual benchmark for evaluating semantic contextualization](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. The Association for Computational Linguistics.
- Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. [Revisiting the Primacy of English in Zero-shot Cross-lingual Transfer](#). *CoRR*, abs/2106.16171.
- Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. [Multilingual is not enough: BERT for Finnish](#). *arXiv preprint arXiv:1912.07076*.
- Zirui Wang, Zachary C Lipton, and Yulia Tsvetkov. 2020. [On negative interference in multilingual models: Findings and a meta-learning treatment](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450.
- Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. [Neural network acceptability judgments](#). *Transactions of the Association for Computational Linguistics*, 7:625–641.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *NAACL-HLT*.

Reassessing Evaluation Practices in Visual Question Answering: A Case Study on Out-of-Distribution Generalization

Aishwarya Agrawal^{*,‡,◇} Ivana Kajić^{*,◇} Emanuele Bugliarello^{*,△}
Elnaz Davoodi^{†,◇} Anita Gergely^{†,◇} Phil Blunsom[‡] Aida Nematzadeh^{*,‡,◇}

◇DeepMind △University of Copenhagen ‡University of Oxford

{aiagrawal,kivana,nematzadeh}@deepmind.com emanuele@di.ku.dk

Abstract

Vision-and-language (V&L) models pre-trained on large-scale multimodal data have demonstrated strong performance on various tasks such as image captioning and visual question answering (VQA). The quality of such models is commonly assessed by measuring their performance on unseen data that typically comes from the same distribution as the training data. However, when evaluated under out-of-distribution (out-of-dataset) settings for VQA, we observe that these models exhibit poor generalization. We comprehensively evaluate two pretrained V&L models under different settings (i.e. classification and open-ended text generation) by conducting cross-dataset evaluations. We find that these models tend to learn to solve the benchmark, rather than learning the high-level skills required by the VQA task. We also find that in most cases generative models are less susceptible to shifts in data distribution compared to discriminative ones, and that multimodal pretraining is generally helpful for OOD generalization. Finally, we revisit assumptions underlying the use of automatic VQA evaluation metrics, and empirically show that their stringent nature repeatedly penalizes models for correct responses.

1 Introduction

Visual Question Answering (VQA) is the task of automatically answering natural language open-ended questions about images. Tackling VQA involves multiple skills, such as language and visual understanding, integrating information between the two (vision and language) modalities, and commonsense and knowledge based reasoning. One of the goals of the VQA research has been fostering the development of systems that are able to answer *any open-ended question about any image*. This

motivation has inspired a fruitful line of research in designing VQA benchmarks (e.g., Malinowski and Fritz, 2014; Antol et al., 2015; Krishna et al., 2017; Goyal et al., 2017; Hudson and Manning, 2019) and models (e.g., Yang et al., 2015; Anderson et al., 2018; Lu et al., 2019; Cho et al., 2021).

In this work, we investigate if recent pretrained VQA models can indeed answer any open-ended question about images or if they are mostly suitable for answering questions from the VQA benchmarks they are optimized for. In other words, *are models learning to solve the task or learning to solve the datasets?* We believe the former is more aligned with the goal of building real-world VQA systems.

To measure whether models learn to solve the task of VQA, we believe we need to examine their *out-of-distribution (OOD)* generalization capabilities: how they perform on examples drawn from a distribution other than that of the training set. In this work, we extensively evaluate OOD generalization of current pretrained V&L models by conducting cross-dataset evaluations (without any adaptation to the test domain).

Through our extensive experiments, we provide in-depth discussion on the following questions:

- *How well do recent models generalize under OOD settings?* We observe a notable drop in performance from IID to OOD settings across models and benchmarks, demonstrating that models mostly learn to solve specific benchmarks as opposed to learning general skills for answering questions about images. This result is not simply due to a mismatch between the set of answers between the training and test VQA datasets, nor due to poor representation of test answers in VQA training data.
- *Is multimodal pretraining beneficial for OOD generalization?* We find that while image-text pretraining is helpful in most OOD settings, it is not always more useful than in IID ones. Moreover, it is least useful for OOD evalua-

*denotes equal first author contribution. † denotes equal contribution. ‡ denotes equal senior contribution. Detailed contributions follow at the end of the paper.

tion on the VIZWIZ benchmark, highlighting the challenges of a real-world benchmark.

- *Is generative modeling more robust to distribution shifts?* In most cases, we observe that generative models—which are not bound to predictions over a fixed set of answers curated from the training data—are more robust to OOD evaluation than discriminative (*i.e.*, classification-based) ones. Moreover, we quantify what the limitations of discriminative models are for real-world VQA applications (*e.g.*, answering questions of visually-impaired users), where the answers a deployed model needs to produce cannot be predetermined.
- *Are current automatic VQA metrics too stringent for OOD evaluation?* We examine if the performance of our pretrained models is negatively impacted by the current standard VQA accuracy metrics, which match predicted answer strings to a limited number of ground-truth answers. Human evaluation reveals the stringent nature of such accuracy metrics, which is especially pronounced in the OOD settings. Nevertheless, while the IID-to-OOO performance gap is reduced after human evaluation, models still exhibit poor generalization to OOD VQA benchmarks.

We believe our OOD evaluations and supporting analyses expose the shortcomings of current models, and recommend future work to adopt these evaluation practices to provide real-world, robust assessment of VQA systems.

2 Related Work

Beyond IID evaluation in VQA. Previous work has evaluated VQA models beyond the IID setting for robustness to *specific and controlled* aspects – novel compositions of seen concepts (Agrawal et al., 2017; Johnson et al., 2017; Hudson and Manning, 2019), change in prior distributions of answers per question type (Agrawal et al., 2018; Gokhale et al., 2020; Niu et al., 2021), adversarial examples provided by humans (Sheng et al., 2021; Li et al., 2021b), consistency, negation, and simple perturbation in questions (Jimenez et al., 2022), counter-examples (Dancette et al., 2021), and controlled shifts in language and vision modalities (Akula et al., 2021). Our focus, however, is to evaluate for *overall* robustness to OOD data without controlling for specific aspects, by testing our

models on different OOD benchmarks. We believe our experimental setting more closely emulates the expected experience of deployed VQA systems. Moreover, when the exact nature of distribution shift between train and test splits is known (such as in (Agrawal et al., 2018)), approaches developed to tackle such shifts tend to rely on the explicit knowledge of construction of such OOD splits resulting in inflated sense of progress (Teney et al., 2020).

Similar to us, Zhang et al. (2021); Hudson and Manning (2019) also present some experimental results on VQA OOD evaluation, however they do it in limited manner (*e.g.*, do not consider all pairs of datasets, do not evaluate the effect of multimodal pretraining, etc.). To our best knowledge, ours is the first work to extensively quantifying the extent of IID to OOD performance drops in current VQA models and study the effect of several factors: answer overlap, multimodal pretraining, generative vs. discriminative modeling, and stringent evaluation metric.

Domain adaptation in VQA. Some studies (Jabri et al., 2016; Chao et al., 2018; Zhang et al., 2021) have explored domain adaptation of VQA models from one VQA benchmark to another. Our focus, instead, is on evaluating *zero-shot* cross-benchmark generalization *without* any adaptation. This allows us to assess the robustness of current models towards unforeseen distribution shifts. Our work is similar to that of Torralba and Efros (2011) and Hendrycks et al. (2020), who study OOD generalization in vision and text.

Zero-shot VQA with pretrained models. In an emerging line of research (Tsimpoukelli et al., 2021; Alayrac et al., 2022; Song et al., 2022; Piergiovanni et al., 2022), large-scale pretrained unimodal models (Brown et al., 2020; Radford et al., 2021) are repurposed to tackle VQA in zero-shot or few-shot fashion. While such zero-shot VQA evaluations are a better test of generalization than IID evaluations, our focus, differently, is on investigating whether models can generalize to unseen datasets *upon being taught the task by showing examples from one dataset*. Moreover, this line of work does not focus on a thorough analysis of models in OOD settings (which is hard to define for these models due to the massive amount of data they are pretrained on).

3 Experimental Setup

In this section, we present our framework to examine OOD generalization in VQA. We examine two pretrained Transformers across five benchmarks.

3.1 Models

We evaluate the performance of two representative, widely-used pretrained models that have achieved strong performance in various V&L tasks in the last few years: ViLBERT (Lu et al., 2019) and ALBEF (Li et al., 2021a). We evaluate these models in a broad range of settings (generative/discriminative, w/wo pretraining, and multiple benchmarks), resulting in 128 experiments. We chose these models as they include components shown to be important in the literature: cross-attention (ViLBERT and ALBEF), and contrastive learning (ALBEF). We note that our goal is to study trends that hold across different models, and we leave for future work controlled comparisons across architectures.

ViLBERT is one of the first, yet strong models in the recent pretrain–fine-tune paradigm in V&L. Its inputs are a sequence of sub-word tokens (Wu et al., 2016), and a set of regions of interest given by a Faster R-CNN (Ren et al., 2015; Anderson et al., 2018). The authors fine-tune it on VQAV2 by learning a classifier over the most frequent answers. We first re-implement this model successfully, and then extend it to a generative setting by pretraining and fine-tuning a Transformer decoder (more details in App. A). We denote the discriminative/generative version as ViLBERT_{DISC}/ViLBERT_{GEN}. Unless otherwise specified, results for ViLBERT_{DISC} are from our code base for direct comparison with ViLBERT_{GEN}.

ALBEF is a state-of-the-art V&L encoder whose visual inputs are image patches encoded by a vision Transformer (Dosovitskiy et al., 2021; Touvron et al., 2021) that is jointly trained with the rest of the model. Li et al. (2021a) fine-tune ALBEF on VQAV2 by adding a 6-layer Transformer decoder to generate answers (ALBEF_{GEN}). We use the official implementation,¹ and furthermore train a discriminative variant (ALBEF_{DISC}) by learning a multi-answer classifier, as in ViLBERT_{DISC}.

In our analysis, we investigate the role of multi-modal pretraining. ViLBERT was pretrained on 3M image–text pairs from Conceptual Captions (CC; Sharma et al. 2018). Li et al. (2021a) re-

¹<https://github.com/salesforce/ALBEF>.

Dataset	# Train (imgs / qns)	# Val (imgs / qns)	# Classes	Coverage [%]
VQAV2	82,783 / 443,757	40,504 / 214,354	3,129	98.07 / 98.07
GQA	72,140 / 943,000	10,234 / 132,062	1,533	99.78 / 99.79
VG	59,635 / 868,259	39,645 / 577,063	3,449	76.55 / 76.55
VIZWIZ	20,523 / 20,523	4,319 / 4,319	3,112	96.76 / 97.01

Table 1: Datasets statistics. #classes is the number of classes we use for the discriminative models; coverage is the percentage of questions that can be answered with our selected classes in train/validation splits.

leased two checkpoints for ALBEF: one pretrained on 4M images from CC, MS-COCO (Lin et al., 2014), SBU (Ordonez et al., 2011) and Visual Genome (Krishna et al., 2017); the other one is further pretrained on Conceptual 12M (Changpinoy et al., 2021) for a total of 14M images.²

3.2 Datasets and Evaluation Metrics

Datasets. We ground our analysis on five diverse VQA datasets: VQAV2 (Goyal et al., 2017), GQA (Hudson and Manning, 2019), VISUAL GENOME (VG; Krishna et al. 2017), VIZWIZ (Gurari et al., 2018) and VQA-CP (Agrawal et al., 2018). VQAV2 is the most commonly used VQA dataset to date. VQA-CP re-splits it such that, for every question type, train and test sets have different prior distributions of answers. VG includes questions centered around either the full image or a specific region. GQA is a large-scale dataset that focuses on compositionality of template-generated questions. Finally, VIZWIZ is the only real-world VQA dataset, collected from visually-impaired people. VG and GQA have one answer per question, while the other datasets include 10 answers per question. See Tab. 1 and App. A for more details.

There are several differences among these datasets. Both VQAV2 and GQA mostly have one-word answers (89% and 81%, respectively) whilst there are fewer in VG (57%) and VIZWIZ (67%). The type of questions also varies: VG does not contain binary ‘yes/no’ questions, but rather spans 6 WH-questions. By design, GQA questions require more compositional skills but do not test for counting; while VIZWIZ questions are more conversational as they were collected through a speech interface and has a significant proportion of OCR questions (21%). Moreover, a significant number of VIZWIZ questions (28%) are *unanswerable* due to the challenges faced by the visually-impaired users in taking pictures, resulting in poor focus,

²We also conducted experiments with ViLBERT pretrained on same datasets as the 4M ALBEF checkpoint. We found no significant difference compared to the results presented throughout this paper.

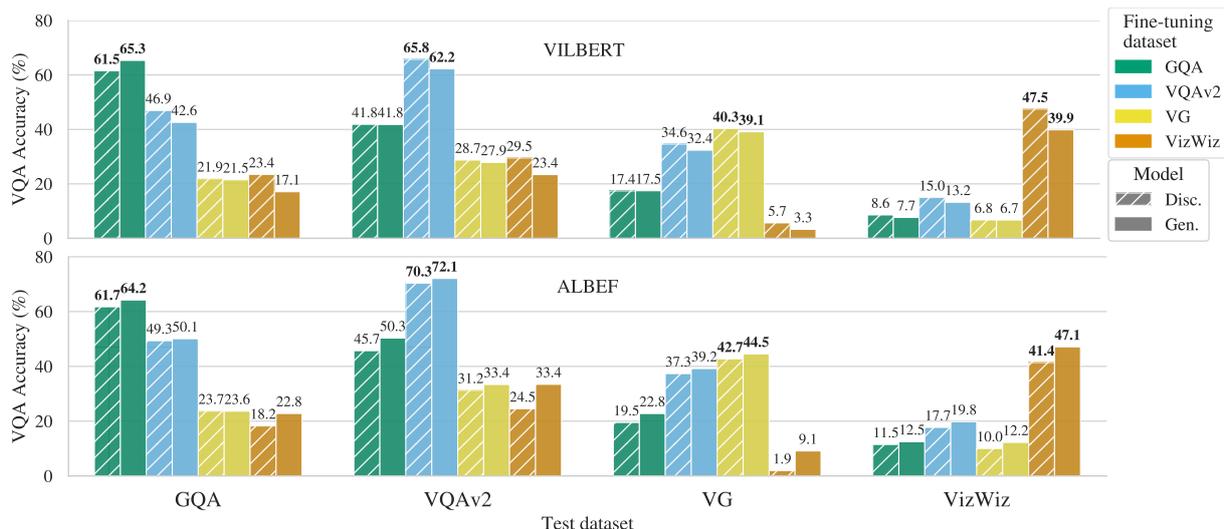


Figure 1: IID (highlighted in bold) vs. OOD performance. Top: ViLBERT pretrained on CC. Bottom: ALBEF pretrained on CC, VG, SBU, MS-COCO and C12M datasets. All models are initialized with BERT weights.

poor lighting or entirely missing the entity of interest. As such, the distribution of images in VIZWIZ is different from other datasets.

Evaluation metrics. These VQA benchmarks compute model accuracy between its prediction and the ground-truth answer(s) by string matching (after simple pre-processing). VQAV2 and VIZWIZ, each with 10 answers per question, account for diversity in ground-truth answers by scoring a given model answer as $\min\{1.0, 0.3 \times \text{count}\}$, where count is the number of annotators that used that answer. For GQA and VG, both with one answer per question, we use top-1 accuracy.³

3.3 Training Details

Following common practice, for discriminative models, we select the top- k most frequent answers as the set of answer classes to perform classification over. Here k is a dataset-dependent variable, chosen to cover most of the questions (see Tab. 1). All models are trained on the respective training sets and evaluated on the validation sets. For VG, we randomly split the data into training and validation (60%/40%) with no image is in both splits.

4 Out-of-Distribution Generalization

We examine to what extent our models learn to solve a specific VQA benchmark by latching on dataset-specific correlations, as opposed to learning

³We note that GQA and VG propose top-5 accuracy. We, instead, opt for top-1 accuracy to keep a consistent setup with VQAV2 and VIZWIZ. And we believe top-5 accuracy is impractical for many applications, such as answering questions for visually-impaired users.

more general skills required in VQA. We fine-tune a pretrained model on the train split of one benchmark (e.g., GQA) and evaluate it on the validation split of a different one (e.g., VG). Overall, we evaluate models by fine-tuning them on each benchmark and testing them against all benchmarks. If pretrained models are indeed learning the VQA skill, we expect to see a small drop in performance between the IID and OOD settings.

The results are presented in Fig. 1, with different evaluation benchmarks grouped on the x -axis. First, across all models and for each benchmark, we see a notable drop in the VQA accuracy from the IID to the OOD setting. While such a drop might be anticipated, we found the extent of the drop surprising given the impressive performance of current pretrained VL models. For all models shown, the largest drops are observed when evaluating models on the VIZWIZ benchmark. Moreover, even the smallest performance drop, which happens when fine-tuning models on VQAV2 and evaluating them on VG, remains relatively large (i.e., 5.3 points for ALBEF_{GEN}). These results show that *pretrained models are largely learning the fine-tuning benchmark without learning to solve the VQA task.*

Second, we observe that fine-tuning on VQAV2 results in the lowest drop in IID to OOD performance across all conditions—the VQAV2 bar (shown in blue in Fig. 1) is the closest to the IID one for GQA, VG, and VIZWIZ. We conclude that fine-tuning on VQAV2 yields a model that best generalizes to OOD settings in our benchmarks. This result is not simply due to the size of the fine-tuning benchmark as VG is larger than VQAV2.

Similarly, all the models achieve highest OOD performance on VQAV2. We conjecture that VQAV2 is the most diverse benchmark of our selection.

4.1 Evaluating on Shared Answer Sets

Discriminative models treat VQA as a multi-answer classification task over the set of top- k most frequent answers in the fine-tuning data. This limits their performance: if a certain answer is not frequent in the fine-tuning data, a discriminative model will perform poorly for such an answer during test time. While this limitation also affects IID evaluation, we expect it to have a stronger effect in OOD generalization (due to potentially different answer distributions between the fine-tuning and test sets). We next examine to what extent this limitation affects OOD performance by controlling for the mismatch in answer sets between the fine-tuning and test sets. We do so by considering only the test questions whose answers are included in the top- k answers of a given fine-tuning dataset (for more details, see App. B).

Fig. 2 shows the improvement in the VQA accuracy over the IID and OOD evaluation accuracy (in Fig. 1) when controlling for the shared answer set. For IID evaluation, only one intersection of answer sets is reported, corresponding to the smallest gap between IID and OOD evaluation, with remaining numbers reported in Tab. 10 (App. B). Thus, the difference between the height of the IID bar (#) and the OOD bar (*) with respect to which answer intersection between IID and OOD is computed, represents the best case scenario for OOD generalization, *i.e.*, the least drop from IID to OOD.

We observe a similar pattern across the models: in most cases, using a shared answer set improves the performance. *Overall, we still observe a notable gap between the OOD and IID settings for the best case OOD generalization scenario, showing that a shared answer set does not circumvent the difficulty of OOD generalization for these models.* A few cases where IID evaluations with a shared answer set hurt performance are discussed in App. B. When evaluating on the shared answer set, we further examine if the drop in accuracy from IID to OOD is due to the low frequency of the test answer classes in the OOD fine-tuning set. The details of the correlation computation and the results are explained in App. B and Tab. 9, respectively. This result indicates that frequency of the answer class is a contributing factor to the weak OOD generalization, but we also explore other causes in Sec. 7.

	VQAV2	GQA	VG	VIZWIZ
VQAV2	92.9	96.7	65.1	43.6
GQA	73.5	99.9	44.8	36.6
VG	52.7	62.4	74.2	32.3
VIZWIZ	79.4	82.5	40.9	86.2

Table 2: Maximum achievable accuracy for all test answers based on the top- k answers present in the respective fine-tuning sets. Rows correspond to fine-tuning datasets, columns correspond to the test benchmarks

4.2 The Case for the Generative Evaluation

A discriminative model cannot correctly answer questions for which the answers lie outside the predefined top- k classes; therefore, by treating VQA as a classification task, we can define the upper-bound performance of discriminative models on VQA by computing the accuracy given all answers in the test set being answered correctly. The upper-bound VQA accuracy is shown in Tab. 2; we observe a large drop from IID to OOD evaluations for most conditions. VIZWIZ has the lowest achievable accuracies in OOD evaluation.

However, our ALBEF_{DISC} and ViLBERT_{DISC} models still perform notably worse than maximum achievable accuracy in all settings (smallest gap of 21.5% across all conditions, see Fig. 7 in App. B); *as a result, the poor OOD performance in the discriminative setting is not simply due to the low maximum achievable accuracy.* We conclude that the common practice of modeling VQA as a classification task severely limits the generalization capability of models to new datasets. On the other hand, generative models do not suffer from a fixed class set. They can generate a larger set of answers—all words for which the tokens occur in the pretraining data, including those that are out-of-vocabulary for the given VQA fine-tune datasets. We argue that generative modeling is a more promising solution for real-world application of VQA; similarly, recent work has identified text generation as a way to unify various V&L tasks (*e.g.*, Cho et al., 2021; Wang et al., 2022; Alayrac et al., 2022).

We next ask *whether our ViLBERT_{GEN} and ALBEF_{GEN} models are more successful in OOD generalization compared to their discriminative counterparts.* For each model (*i.e.*, generative/discriminative ALBEF/ViLBERT), we first calculate the gap between the IID setting and each OOD setting (*i.e.*, Δ OOD), resulting in three values per benchmark. For instance, for the VQAV2 benchmark, Δ OOD numbers are calculated between the model fine-tuned on VQAV2 and those

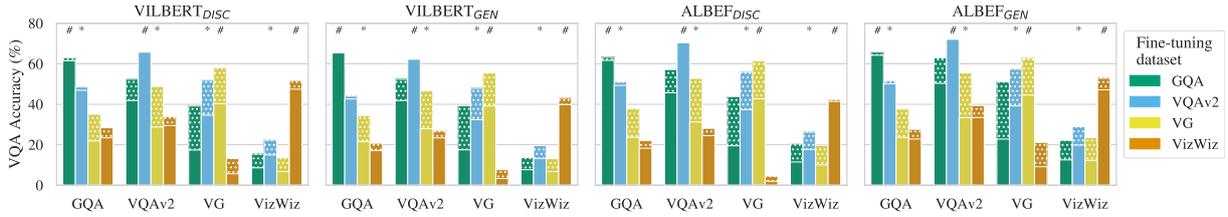


Figure 2: IID (#) vs OOD performance when controlling for the shared answer set. Solid bars are as in Fig. 1; stacked dotted bars are improvements when evaluating on questions with shared answer sets between IID and OOD settings. For IID, the shared answer set is computed with respect to a dataset denoted with *.

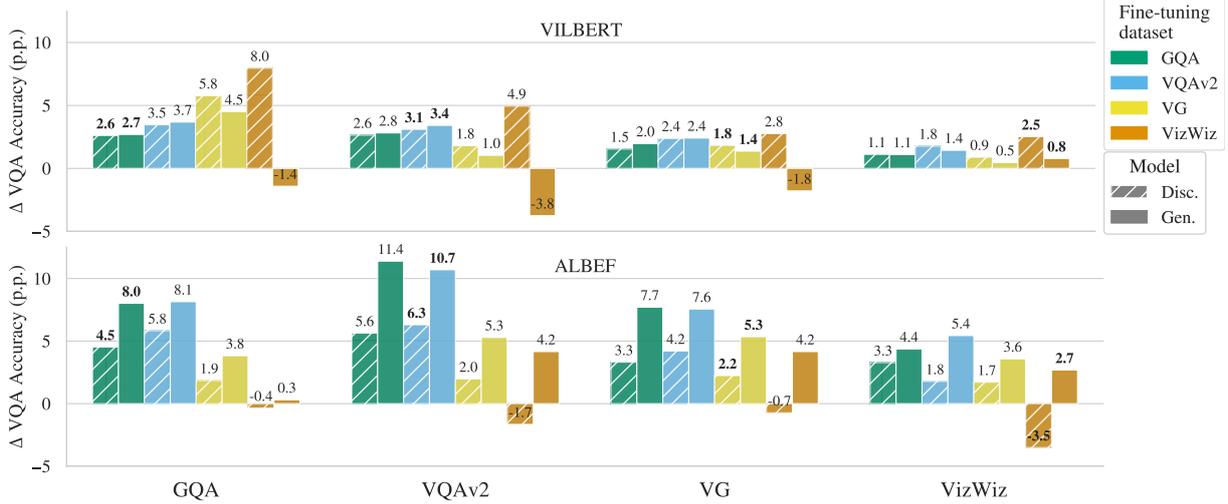


Figure 3: Percentage point difference in VQA accuracy between models with and without multimodal pretraining, for OOD and IID (highlighted in bold) evaluations. All models are initialized with BERT weights.

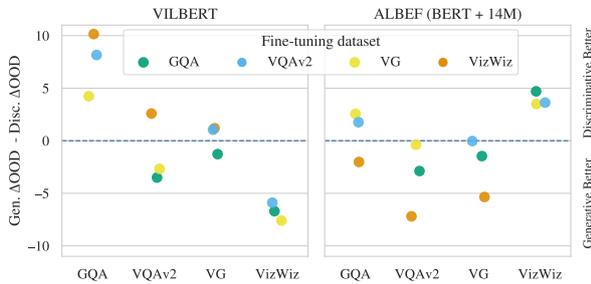


Figure 4: Difference in Δ OOD values between discriminative and generative models.

finetuned on VG, GQA, and VIZWIZ. Note that the higher the Δ OOD value, the poorer a model is in OOD generalization. We then compute the difference between the Δ OOD values of the generative and discriminative models. Fig. 4 visualizes this result; the benchmarks are shown on the x -axis and each circle represents the difference in Δ OOD values between the generative and discriminative model for a given fine-tuning dataset. If a generative model is more robust to OOD evaluation, we expect to see smaller Δ OOD value for that model compared to its discriminative counterpart: when the circles are below the x -axis (depicting negative values), the generative model is more robust than the discriminative one. We observe ALBEF_{GEN}

models often outperform their discriminative counterparts with respect to OOD generalization.

5 The Effect of Multimodal Pretraining

Previous work has shown that pretraining on multimodal (*i.e.*, image–text) data improves IID performance (*e.g.*, Lu et al., 2019; Li et al., 2021a); here, we ask if multimodal pretraining can help in OOD settings as well. We repeat the experiments in Sec. 4 without pretraining our models on multimodal data; instead we train the models on the train split of one benchmark and test it on the validation split of another. Fig. 3 shows the difference between the VQA accuracy of models with and without multimodal pretraining: each bar shows the gap between a bar in Fig. 1 and the equivalent experiment without multimodal pretraining.

We observe that multimodal pretraining is helpful in almost all conditions, since the majority of values displayed in Fig. 3 are positive. Pretraining is improving OOD performance likely because it can reduce the gap between the train and OOD test data by potentially exposing the model to a more diverse set of data points during pretraining. In our

Model	MM PT	VQAv2	VQA-CP	Drop
CF-VQA	–	53.6	63.5	9.9
VILBERT _{DISC}	no	66.7	42.5	24.2
VILBERT _{DISC}	yes	67.0	42.9	24.1
ALBEF _{DISC}	no	64.0	40.1	23.9
ALBEF _{DISC}	yes (4M)	70.0	44.4	25.6
ALBEF _{DISC}	yes (14M)	70.3	45.2	25.1
ALBEF _{GEN}	no	61.4	36.6	24.8
ALBEF _{GEN}	yes (4M)	71.0	49.2	21.8
ALBEF _{GEN}	yes (14M)	72.1	49.6	22.5

Table 3: Performance of models on VQAv2 (IID) and VQA-CP (OOD). The last column shows drop in performance from VQAv2 to VQA-CP. MM PT: Multimodal Pretraining.

experiments, the maximum gain from multimodal pretraining is indeed observed in OOD settings for both VILBERT (fine-tune on VIZWIZ; test on GQA) and ALBEF (fine-tune on GQA; test on VQAv2); however, *multimodal pretraining is not always more useful in OOD settings compared to IID ones*. For example, when evaluating VILBERT on VQAv2, pretraining helps the IID setting more than some of the OOD ones. Lastly, multimodal pretraining is detrimental for some cases where models are fine-tuned on VIZWIZ.

We observe that multimodal pretraining is more effective for the generative ALBEF compared to the discriminative ALBEF (cf. the shaded and solid bar with the same color in Fig. 3 bottom). For the VILBERT model, we generally do not observe such a pattern—discriminative and generative models mostly show comparable improvements due to multimodal pretraining. We observe only small improvements when increasing the size of the multimodal pretraining dataset for the ALBEF model (see Fig. 8 in App. B for more details).

6 Evaluation on VQA-CP

In this section, we evaluate the models⁴ on the VQA under Changing Priors dataset (VQA-CP; Agrawal et al. 2018). This dataset is designed such that, for every question type, train and test splits have different prior distributions of answers. Thus, models that overfit to answer priors in training data and lack sufficient visual grounding show poor generalization on the VQA-CP test set. For comparison, we also evaluate the performance of Counterfactual VQA (CF-VQA; Niu et al. 2021), a state-of-art method on VQA-CP, which does not use either the Transformer architecture nor multi-

⁴ALBEF and VILBERT_{DISC} (using the official codebase).

modal pretraining. However, it explicitly tackles the language (*i.e.*, question and answer) biases in VQA-CP.

Tab. 3 shows that for all the Transformer-based models, there is a large drop in the performance (at least 22%) from VQAv2 to VQA-CP. Thus, in spite of advances in the Transformer architecture and pretraining on diverse datasets, models are still overfitting to answer priors in the training data and lack sufficient visual grounding (Agrawal et al., 2018). However, the drop is much less for CF-VQA (10%), suggesting that *incorporating inductive biases specific to the generalization problem (i.e., modeling language bias) helps more than the Transformer architecture or scaling up the amount of pretraining data*. We also observe that the drop from VQAv2 to VQA-CP is often lower for the generative ALBEF than the discriminative ALBEF (except for ALBEF without any multimodal pretraining). Thus, *generative models are more robust than the discriminative ones*, especially when they are pretrained (similarly to the observations made in Sec. 4.2). As for the effect of pretraining, for generative ALBEF, pretraining helps reduce the drop from VQAv2 to VQA-CP. However, for discriminative models, pretraining does not seem to help generalization (in fact, it worsen ALBEF).

7 Qualitative Analysis

To dig deeper into the potential causes of the poor OOD generalization of our pretrained models, we perform a qualitative study. To this end, we randomly sample and manually examine failure cases in top-30 answer classes with the highest performance drop when moving from IID to OOD evaluation. We only focus on answer classes that are present in both the train and test splits, ensuring that performance drop is not due to the absence of answer classes in the training set. We report the top-5 classes that contribute the most to the drop in performance for each OOD setting in Tab. 11 (App. C). We notice that the following answer classes appear frequently across different OOD settings: yes/no answers, directions (left/right), colors, and numbers. In the following, we discuss a few major potential causes for the poor OOD generalization, and mention VILBERT_{DISC} responses as examples in the discussion, although similar observations hold for other models.

Poor reasoning skills. Models evaluated on GQA, but fine-tuned on another dataset, show the

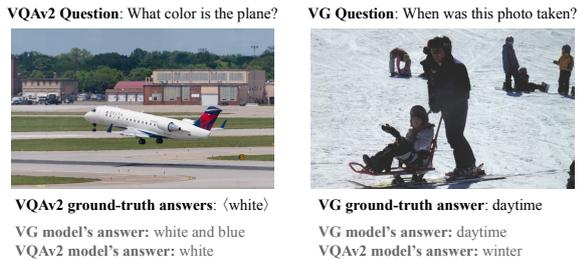


Figure 5: Examples where models’ prediction are correct but not accounted for in the ground-truth set. ⟨ ⟩ denotes a list of unique (out of 10) ground-truth answers. VG (VQAV2) model refers to a ViLBERT_{DISC} fine-tuned on VG (VQAV2).

highest performance drop on classes such as “yes”, “no”, “right”, “left”, “top”, and “bottom”. For instance, ViLBERT_{DISC} fine-tuned on VQAV2, and evaluated on GQA underperforms ViLBERT_{DISC} that has been both fine-tuned and evaluated on GQA by 24% for the answer class “no.” Upon qualitative examination, we find that for many of such failure cases, the GQA questions are more compositional and hence require more complex reasoning (e.g., “Are there both bison and zebras in the image?”, “Is the cheese to the right or to the left of the empty plate?”) than the questions for the same answer classes in other datasets (e.g., from VQAV2 train set: “Is the TV turned on?”, “Which hand is the man holding up?”). This study re-affirms previous findings that VQA models lack sufficient logical, spatial, and compositional reasoning skills (Johnson et al., 2017; Hudson and Manning, 2019) but for the more recent, pretrained Transformer models.

Overfitting to the answer priors. Previous studies have shown that VQA models tend to be biased towards the prior distribution of answers in the training set (per question type) (Agrawal et al., 2018). We find that this limitation exists in the more recent pretrained models as well, and it is especially hurtful in the OOD settings because the priors need not be the same across train and test sets, unlike in the IID settings. For instance, ViLBERT_{DISC} fine-tuned on VQAV2 predicts “2” for a lot of questions with target answer “1” in the VG test set. Similarly, sometimes ViLBERT_{DISC} fine-tuned on VG incorrectly predicts “helmet” for VQAV2 test questions such as “What is the skateboarder wearing to protect his head?”, “What protective gear is he wearing?” when the skateboarder is not wearing anything. This indicates that the model is relying on answer priors rather than visual grounding. Our experimental results on VQA-CP

(Sec. 6) directly quantify the extent of such limitations in current models.

Overfitting to the question format. We observe instances of models failing to correctly answer questions when the format of the questions changes between the fine-tuning and test sets. For instance, questions about “chair” in the VQAV2 fine-tuning set are mostly of the form “What is ... sitting on?” whereas in the GQA test set, they are mostly of the form “What kind of furniture is ...?”. Thus, the “chair” class accuracy of ViLBERT_{DISC} fine-tuned on VQAV2 drops from 48% when tested on VQAV2 to 38% on the GQA test set. Similarly, ViLBERT_{DISC} fine-tuned on GQA fails terribly for “dog” and “cat” classes on the VG test set (accuracy drops of 47% and 43% respectively between GQA–GQA (fine-tuned on GQA, tested on GQA) and GQA–VG). GQA questions are mostly of the form “What animal ...?” or “What kind of animal ...?” whereas VG questions often do not mention the word “animal” and are of the form “Who is ...?” or “What is ...?” (e.g., “Who is holding the Frisbee?”, “What is on the leash?”). To the best of our knowledge, no previous work has reported such behavior of VQA models (i.e., they tend to overfit to the question format).

Finally, we observe cases where correct model responses are evaluated as incorrect by the VQA evaluation metric, as such responses differ from the ground-truth answers. In the next section, we provide examples of such cases and examine the impact of **stringent evaluation metric** on poor OOD generalization by engaging human raters to evaluate responses.

8 Human Evaluation

In our qualitative study, we observed that the stringent nature of the standard VQA evaluation metrics (i.e., performing string matching of model responses with a small set of ground-truth answers) repeatedly penalizes models for correct responses because those responses do not exist in the set of ground-truth answers (Fig. 5). For example, the evaluation metric fails to take into account differences (between model response and ground-truth) due to specificity of the answers (e.g., “on table” vs. “table”, “pizza slices” vs. “pizza”), synonyms, and different interpretations of the question (e.g., Fig. 5 right).

In this section, we aim to quantify how robust standard VQA metrics are by performing human

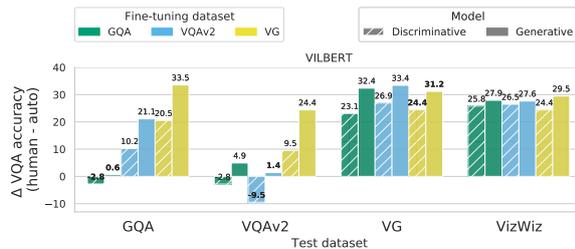


Figure 6: Difference in human and automatic accuracy of $VILBERT_{DISC}$ (shaded bars) and $VILBERT_{GEN}$ (plain bars) for GQA, VQAV2, VG and VIZWIZ. Accuracies in bold denote the IID settings.

evaluation of our models for both IID and OOD settings. The details of the setup and in-depth results are provided in App. D. Below we present our main findings.

Human evaluation yields notably higher accuracies than the automatic evaluation. This is shown in Fig. 6, where the increase can be up to 33.5% when moving from automatic to human evaluation.⁵ This implies the current automatic metrics miss out on a lot of correct responses due to their stringent nature. Interestingly, this increase in model accuracy from automatic to human evaluation is higher for $VILBERT_{GEN}$ than $VILBERT_{DISC}$ for all the benchmarks. This is expected because the generative model is more likely to produce longer, more varied answers, which might not be awarded using automatic metric but are still correct responses. Moreover, human evaluation helps OOD settings more than the IID settings for most of the benchmarks (*e.g.*, GQA, VQAV2). This is also expected, because in the OOD settings, a model might not learn the format of the test answer (“on table” vs. “table”, “clear vs. sunny”) from the train set (unlike in the IID settings) and hence it is more likely to be penalized by the automatic accuracy metric. Thus, we conclude that the currently used accuracy metrics for VQA are not robust, especially for generative models and OOD evaluation settings. Hence, to more accurately evaluate the goodness of our models, *we need to develop better evaluation metrics for VQA.*

Even after human evaluation, models still exhibit poor OOD generalization. Although human evaluation improves the models’ accuracies and more so for the OOD than the IID settings, we observe that the models’ performance in OOD settings is still worse compared to that of IID set-

⁵In some cases, human evaluation yields lower accuracy than the automatic evaluation. We discuss this in App. D.

tings, albeit with reduced margin (see App. D for quantitative results). We also note that while $VILBERT_{DISC}$ usually outperformed $VILBERT_{GEN}$ with the automatic evaluation, $VILBERT_{GEN}$ outperforms $VILBERT_{DISC}$ for all the test sets under human evaluation. This reinforces the observations in Sec. 4.2 regarding stronger OOD generalization of generative models over discriminative ones.

9 Conclusion

In this study, we show that, despite their impressive performance when evaluated on test data drawn from the same distribution as the training data, recent V&L models perform poorly in out-of-distribution (OOD) settings. We conclude that these models learn to solve specific benchmarks as opposed to the skill of visual question answering (VQA). Interestingly, in most cases, we observe that the generative models are more robust to OOD generalization compared to the discriminative ones. Moreover, pretraining the models on large image-text data often helps in OOD generalization. Our results also highlight the importance of human evaluation for a more accurate assessment of model performance: we find that the current VQA automatic metrics miss out on a notable number of correct model responses. Human evaluation is especially important as the community is shifting towards generative VQA models which, unlike discriminative ones, can produce answers that go beyond those seen in a training/fine-tuning dataset. Finally, to make progress towards more capable models, we need more rigorous evaluation protocols that shed light on models’ strengths and short-comings. We believe testing models in OOD settings is a step towards this direction as it helps evaluate models for general skills required to solve the task as opposed to benchmark-specific correlations.

Limitations

We list some limitations of our work which could benefit from future investigations.

First, when exploring potential factors for poor OOD generalization, our quantitative analysis focused only on differences in answer distributions between fine-tuning and test datasets. However, future work should investigate differences in question distribution, image distribution and combinations of these three variables.

Second, it would be interesting to conduct further investigation to understand why multimodal

pretraining does not help in certain cases. A correlation analysis between improvement in accuracy (due to multimodal pretraining), and between pretraining and fine-tuning/test data could be useful.

Third, the models investigated in our study (ViLBERT and ALBEF) are pretrained on a relatively small number (millions) of data points compared to language-only pretrained transformers, such as BERT, trained on billions of tokens. Such large-scale pretraining has been shown to improve OOD robustness for language-only models (Hendrycks et al., 2020). Hence, we leave for future work to investigate multimodal models trained on billions of image–text pairs (for instance, LAION-5B; Schuhmann et al. 2021).

Lastly, in this study, we only focus on standard VQA evaluation metrics for each benchmark. However, it would be interesting to also evaluate the robustness of metrics such as WUPS (Malinowski and Fritz, 2014) that compute answer similarities based on the distance between them in the WordNet (Miller, 1995) tree and are expected to be more robust than the standard metrics.

Ethics Statement

Below we present some considerations related to the ethical and broader impact of our work.

First, all datasets used in our study are from published work and are publicly available, including the VIZWIZ data (Gurari et al., 2018) which has been curated from visually impaired users and released publicly after proper filtering to preserve the privacy of the users.

Second, for human evaluation of our models, we collected human data via the Amazon Mechanical Turk platform. We detail the data collection process and measures taken to control the quality of collected data in App. D. As for the ethical considerations related to collecting data from human subjects, our data collection campaign was approved by an ethics review board in our institution. Human subjects were paid at the rate of 0.15 USD per HIT (Human Intelligence Task) resulting in an hourly payment well above minimum wage.

Third, by testing models on a data distribution different from the training one, the OOD evaluation setting studied in our work has the following broader impacts: it highlights (1) the challenges of generalizing to real-world VQA datasets such as VIZWIZ, and (2) the kind of biases learned (and also potentially amplified) by the models.

Lastly, we discuss both potentially beneficial and harmful applications of the task of Visual Question Answering studied in our work. VQA has many potential applications beneficial for society:

- Aiding visually impaired users in understanding their surroundings (Human: What is on the shelf above the microwave? AI: Canned containers.)
- Teaching children through interactive demos (Kid: What animal is that? AI: That is Dall Sheep. You can find those in Alaska.)
- Aiding analysts in processing large quantities of visual surveillance data (Analyst: What kind of car did the man in red shirt leave in? AI: Blue Toyota Prius.)
- Interacting with in-home physical robots (Human: Is my laptop in my bedroom upstairs? AI: Yes. Human: Is the charger plugged in?)
- Making visual social media content more accessible (AI: Your friend Bob just uploaded a picture from his Hawaii trip. Human: Great, is he at the beach? AI: No, on a mountain.)

But like most other technology, VQA could also be used for potentially harmful applications such as:

- Invasion of individual’s privacy by using VQA to query streams of video data being recorded by CCTV cameras at public places.
- Visually impaired users often need assistance with parsing data containing personal information (Ahmed et al., 2015), such as credit cards, personal mails, etc. Such VQA systems could be configured to leak/retain personally identifiable information.

Contributions

Aishwarya Agrawal initiated and designed the project, ran experiments and analyses on the official codebase of ViLBERT, contributed significantly to paper writing, and provided project support and advice. *Ivana Kajić* was responsible for the project’s technical infrastructure for the re-implementation of ViLBERT, ran experiments and analyses, and contributed significantly to paper writing. *Emanuele Bugliarello* co-led the data preparation, ran experiments and analyses on ALBEF, and contributed significantly to paper writing.

Elnaz Davoodi co-lead the data preparation, and helped setting up and running re-implementation experiments. *Anita Gergely* led the human evaluation experiments, and contributed to paper writing. *Phil Blunsom* provided project advice. *Aida Nematzadeh* provided significant project support and advice, helped running experiments, and contributed significantly to paper writing.

Acknowledgements

We are grateful to Antoine Miech, Lisa Anne Hendricks and Chris Dyer for their constructive feedback. ■ During this project, *Emanuele Bugliarello* was supported by the funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199.

References

- Abien Fred Agarap. 2018. [Deep learning using rectified linear units \(relu\)](#). *arXiv preprint arXiv:1803.08375*.
- Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. [Don’t just assume; look and answer: Overcoming priors for visual question answering](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. 2017. [C-VQA: A compositional split of the visual question answering \(vqa\) v1.0 dataset](#). *ArXiv*, abs/1704.08243.
- Tousif Ahmed, Roberto Hoyle, Kay Connelly, David Crandall, and Apu Kapadia. 2015. [Privacy concerns and behaviors of people with visual impairments](#). In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3523–3532.
- Arjun Akula, Soravit Changpinyo, Boqing Gong, Piyush Sharma, Song-Chun Zhu, and Radu Soricut. 2021. [CrossVQA: Scalably generating benchmarks for systematically testing VQA generalization](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2148–2166, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Saahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. [Flamingo: a visual language model for few-shot learning](#). In *Advances in Neural Information Processing Systems*.
- Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. [Bottom-up and top-down attention for image captioning and visual question answering](#). In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. [VQA: Visual question answering](#). In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. 2021. [Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language BERTs](#). *Transactions of the Association for Computational Linguistics*, 9:978–994.
- Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. [Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3558–3568.
- Wei-Lun Chao, Hexiang Hu, and Fei Sha. 2018. [Cross-dataset adaptation for visual question answering](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. [UNITER: Universal image-text representation learning](#). In *European Conference on Computer Vision (ECCV)*.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. [Unifying vision-and-language tasks via text generation](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1931–1942. PMLR.

- Corentin Dancette, Rémi Cadène, Damien Teney, and Matthieu Cord. 2021. [Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1574–1583.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. [Imagenet: A large-scale hierarchical image database](#). In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *International Conference on Learning Representations*.
- Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. 2020. [MUTANT: A training paradigm for out-of-distribution generalization in visual question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 878–892, Online. Association for Computational Linguistics.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. [Making the v in VQA matter: Elevating the role of image understanding in visual question answering](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. [VizWiz grand challenge: Answering visual questions from blind people](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzić, Rishabh Krishnan, and Dawn Song. 2020. [Pretrained transformers improve out-of-distribution robustness](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.
- Drew A. Hudson and Christopher D. Manning. 2019. [GQA: A new dataset for real-world visual reasoning and compositional question answering](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Allan Jabri, Armand Joulin, and Laurens van der Maaten. 2016. [Revisiting visual question answering baselines](#). In *Computer Vision – ECCV 2016*, pages 727–739, Cham. Springer International Publishing.
- Carlos E. Jimenez, Olga Russakovsky, and Karthik Narasimhan. 2022. [CARETS: A consistency and robustness evaluative test suite for VQA](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6392–6405, Dublin, Ireland. Association for Computational Linguistics.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. [CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. [Visual Genome: Connecting language and vision using crowdsourced dense image annotations](#). *International journal of computer vision*, 123(1):32–73.
- Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021a. [Align before fuse: Vision and language representation learning with momentum distillation](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 9694–9705. Curran Associates, Inc.
- Linjie Li, Jie Lei, Zhe Gan, and Jingjing Liu. 2021b. [Adversarial VQA: A new benchmark for evaluating the robustness of vqa models](#). In *International Conference on Computer Vision (ICCV)*.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. [Microsoft COCO: Common objects in context](#). In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. [ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Mateusz Malinowski and Mario Fritz. 2014. [A multi-world approach to question answering about real-world scenes based on uncertain input](#). In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

- George A. Miller. 1995. [Wordnet: A lexical database for english](#). *Commun. ACM*, 38(11):39–41.
- Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. [Counterfactual VQA: A cause-effect look at language bias](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12700–12710.
- Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. [Im2text: Describing images using 1 million captioned photographs](#). In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.
- AJ Piergiovanni, Wei Li, Weicheng Kuo, Mohammad Saffar, Fred Bertsch, and Anelia Angelova. 2022. [Answer-Me: Multi-task open-vocabulary visual question answering](#). *arXiv preprint arXiv:2205.00949*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. [Faster R-CNN: Towards real-time object detection with region proposal networks](#). In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. [LAION-400M: Open dataset of clip-filtered 400 million image-text pairs](#). *arXiv preprint arXiv:2111.02114*.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. [Conceptual Captions: A cleaned, hypernamed, image alt-text dataset for automatic image captioning](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.
- Sasha Sheng, Amanpreet Singh, Vedanuj Goswami, Jose Magana, Tristan Thrush, Wojciech Galuba, Devi Parikh, and Douwe Kiela. 2021. [Human-adversarial visual question answering](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 20346–20359. Curran Associates, Inc.
- Haoyu Song, Li Dong, Weinan Zhang, Ting Liu, and Furu Wei. 2022. [CLIP models are few-shot learners: Empirical studies on VQA and visual entailment](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6088–6100, Dublin, Ireland. Association for Computational Linguistics.
- Damien Teney, Ehsan Abbasnejad, Kushal Kafle, Robik Shrestha, Christopher Kanan, and Anton van den Hengel. 2020. [On the value of out-of-distribution testing: An example of goodhart's law](#). 33:407–417.
- Antonio Torralba and Alexei A. Efros. 2011. [Unbiased look at dataset bias](#). In *CVPR 2011*, pages 1521–1528.
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. 2021. [Training data-efficient image transformers & distillation through attention](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR.
- Maria Tsimpoukelli, Jacob L. Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. [Multimodal few-shot learning with frozen language models](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 200–212. Curran Associates, Inc.
- Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2022. [SimVLM: Simple visual language model pretraining with weak supervision](#). In *International Conference on Learning Representations*.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. [Google’s neural machine translation system: Bridging the gap between human and machine translation](#). *arXiv preprint arXiv:1609.08144*.
- Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. [Visual entailment: A novel task for fine-grained image understanding](#). *arXiv preprint arXiv:1901.06706*.
- Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. 2017. [Aggregated residual transformations for deep neural networks](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. 2015. [Stacked attention networks for image question answering](#). *CoRR*, abs/1511.02274.
- Mingda Zhang, Tristan D. Maidment, Ahmad Diab, Adriana Kovashka, and Rebecca Hwa. 2021. [Domain-robust VQA with diverse datasets and methods but no target labels](#). *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7042–7052.

A Experimental Setup Details

In this section, we report additional details regarding our experimental setup.

A.1 Models

We evaluate the performance of two strong models, ViLBERT (Lu et al., 2019) and ALBEF (Li et al., 2021a). These models belong to the family of pretrained Transformer that has recently achieved state-of-the-art performance on several V&L tasks, and are specifically instances of dual-stream architectures (Bugliarello et al., 2021). In this paradigm, models are first pretrained on a large collection of image–caption pairs, and then fine-tuned to solve specific downstream tasks. ViLBERT is pretrained using three objectives, masked language modeling (MLM; Devlin et al. 2019), masked region modeling and image–text matching (ITM; Chen et al. 2020). ALBEF is pretrained using MLM, ITM and an image–text contrastive loss (Li et al., 2021a). We refer the reader to Sec. 3 for an overall description of these models. Tab. 4 lists pretraining and architecture details for both models. All the models were fine-tuned using the AdamW (Loshchilov and Hutter, 2019) optimizer, with model-specific hyperparameters in Tab. 6.

ViLBERT. In this model, the textual inputs are first processed through 6 Transformer layers, before being combined with visual inputs through inter- and intra-modal attention layers. We re-implement this architecture, and confirm comparable performance by reproducing its results with the ones obtained through the official codebase⁶ (see Tab. 5 for IID performance of both implementations). A key difference between the two implementations is in the image features: while the official model uses 10–100 regions of interest (RoIs) from a ResNeXt-152 (Xie et al., 2017), our re-implementation relies on 100 RoIs extracted by Faster R-CNN (Ren et al., 2015) trained on VG.

We then extend our codebase to implement a generative version of ViLBERT by replacing the discriminative decoder with an autoregressive decoding head.⁷ The decoder is trained with teacher-forcing, and in datasets with several responses, the

⁶<https://github.com/facebookresearch/vilbert-multi-task/>.

⁷Our implementation of the generative decoder follows that of TransformerDecoder available at <https://github.com/pytorch/pytorch/blob/master/torch/nn/modules/transformer.py>.

most frequent response is used as the ground truth response. Pretraining on CC, when used, is done by generating text. We also examine the effect of pretraining on both the encoder and decoder, and find the learning to be more stable when using only pretrained encoder, although further hyperparameter exploration could mitigate this difference. We study the effects of multimodal pretraining on the Conceptual Captions dataset (Sharma et al., 2018) with 3M images.

ALBEF. Like ViLBERT, ALBEF is a dual-stream encoder but with two main differences: first, the visual inputs are image patches that are processed through a vision Transformer (Dosovitskiy et al., 2021); and second, the cross-modal interactions happen through standard Transformer cross-attention at each layer (whereas ViLBERT uses co-attention layers specifically designed for intra- and inter-modal interactions). In addition, the model is trained with pseudo-targets that are generated from a moving-average version of its weights. We run experiments using the official codebase.⁸ The visual backbone is a DeiT-B/16 (Touvron et al., 2021) pretrained on ImageNet-1k (Deng et al., 2009) at resolution 224×224 , and further trained during the multimodal pretraining phase. For the downstream VQA benchmarks, we follow the authors and resize input images to 384×384 and apply random augmentation during fine-tuning. Li et al. (2021a) formulated the VQA task as generative by adding a 6-layer Transformer decoder initialized from the pretrained encoder. We follow this approach and also evaluate a discriminative version by learning a two-layer MLP with ReLU (Agarap, 2018) non-linearity in between, following the authors’ setup for the Visual Entailment benchmark (Xie et al., 2019). We found the hyperparameters proposed by Bugliarello et al. (2021) to work better. During inference, we evaluate ALBEF in two ways: first, following the authors, we rank the in-domain candidate answers based on their likelihood; second, we let the model generate any possible answer in an open-ended fashion through greedy decoding. We found these two approaches to minimally affect final performance (see Tab. 7). Unless otherwise specified, we report results given by generation as it reflects open-ended question answering.

⁸<https://github.com/salesforce/ALBEF>.

Model	# Params	Pretrain data	# Images	# Captions
VILBERT	240M	CC	3.3M	3.3M
ALBEF (4M)	450M	+COCO+SBU+VG	4M	5M
ALBEF (14M)	450M	+C12M	14M	15M

Table 4: Pretrained models details. ALBEF size includes both the main model and its moving average. Pretraining data: CC (Sharma et al., 2018), COCO (Lin et al., 2014), SBU (Ordonez et al., 2011), VG (Krishna et al., 2017), C12M (Changpinyo et al., 2021).

Model	VQAv2	GQA	VG	VIZWIZ
Official codebase	67.04	66.78	40.69	44.46
Re-implementation	65.75	61.51	40.31	47.46

Table 5: Comparison between the official and our codebases for VILBERT_{DISC} in the IID setting.

A.2 Datasets

Tab. 1 lists statistics for each dataset in our study. VQAv2 (Goyal et al., 2017) is the most commonly used VQA dataset to date, it consists of 265K images and 1.1M question-image pairs, each with 10 ground-truth answers. VQA-CP (Agrawal et al., 2018) re-splits the VQAv2 dataset such that, for every question type, train and test sets have different prior distributions of answers. VG (Krishna et al., 2017) includes 108K images and 1.7M questions, each paired with a single answer, centered around either the full image or a specific region within it. GQA (Hudson and Manning, 2019) is another large-scale effort (22M questions, each with one answer) that focuses on compositionality of template-generated questions for real-world images (from VG). Following prior work, we use the GQA *balanced* subset (1.5M questions). Finally, VIZWIZ (Gurari et al., 2018) is the only real-world VQA dataset as it was collected from visually impaired people. It consists of 31K image-question pairs, each paired with 10 answers.

A.3 Training Details

Following common practice, for discriminative models, we select the top- k most frequent answers from the fine-tuning dataset, as the set of answer classes to perform classification over. Here k is a dataset-dependent variable. For VQAv2 and GQA, we use the same answer sets as VILBERT (3,129 and 1,533, respectively). For VIZWIZ, we select the answers that appear at least 8 times in training and validation sets, for a total of 3,112 answers that cover 97% of the data. For VG, we select the answers that appear at least 29 times in the dataset, for a total of 3,449 answers that cover 76.5% of the data. Importantly, combined with the VQA accu-

Benchmark	Qn len	# Classes	BS	LR	# Epochs
VQAv2	16	3,129	256	4e-5	20
GQA	26	1,533	256	4e-5	20
VG	16	3,449	256	4e-5	20
VIZWIZ	40	3,112	256	4e-5	20

(a) Parameters used for VILBERT models. The internal codebase uses LAMB optimizer with the initial LR of 1e-3, with the best checkpoint selected on eval dataset.

Benchmark	Qn len	# Classes	BS	LR	# Epochs
VQAv2	16	3,129	256	1e-4	20
GQA	26	1,533	256	1e-4	20
VG	16	3,449	256	1e-4	20
VIZWIZ	40	3,129	256	1e-4	20

(b) Discriminative ALBEF.

Benchmark	Qn len	Answer len	BS	LR	# Epochs
VQAv2	16	6	256	2e-5	20
GQA	26	5	256	2e-5	20
VG	16	8	256	2e-5	20
VIZWIZ	40	11	256	2e-5	20

(c) Generative ALBEF.

Table 6: Hyperparameters used in our experiments. Question and answer lengths are in tokens, BS is the batch size, LR is the learning rate.

racy metric defined above, this results in an upper-bound to the accuracy that discriminative models can achieve in each dataset (see Tab. 2).

All models are trained on the respective training sets and evaluated on the validation sets, which lets us conduct in-depth analyses that would otherwise be impossible to carry out on private test sets. As there is no official split of VG, we randomly sample the data into training (60%) and validation (40%) such that no image appears in both splits.

B Additional Results

Evaluation with Shared Answer Sets While different answer sets are an apparent issue for discriminative models, they also impact the performance of generative models, as the number of data points for each answer class seen by the generative model during fine-tuning varies: data-points in top- k answer set are more frequent than others (by definition of top- k). In other words, even though a tokenizer used to produce an answer could generate it, it is unlikely (or less likely) to do so if it has not seen (or seen rarely) that combination of tokens during fine-tuning. Thus, even for generative models, we consider performance on top- k most frequent classes for each benchmark.

Thus, we report the accuracy on the subset of

	VQAV2	GQA	VG	VIZWIZ
VQAV2	72.37	50.56	38.94	19.81
GQA	50.32	64.26	22.80	12.51
VG	33.40	24.99	43.35	12.60
VIZWIZ	34.17	22.73	8.89	48.44

(a) Ranking-based evaluation of ALBEF_{GEN}.

	VQAV2	GQA	VG	VIZWIZ
VQAV2	72.09	50.10	39.20	19.81
GQA	50.33	64.24	22.79	12.50
VG	33.39	23.64	44.55	12.25
VIZWIZ	34.44	22.80	9.13	47.14

(b) Generation-based evaluation of ALBEF_{GEN}.

Table 7: Performance of ALBEF_{GEN} (14M) when tested via ranking (top) and generation (bottom). Rows correspond to the fine-tuning datasets, columns correspond to the test benchmarks. The model performs similarly in both setups. We found similar results for ALBEF_{GEN} (4M).

test questions whose answers are shared between *both* the IID and the OOD models. For instance, when comparing the performance of the VQAV2 and VG fine-tuned models on the VQAV2 test set, we compute the average accuracy on those VQAV2 questions whose ground truth answers are present in the top- k answers from VQAV2 as well as the top- k answers from VG: we extract the common answer labels (between VQAV2 and VG top- k answers) and compute performance on test questions belonging to these shared answer labels only.

For IID evaluations, there are several possible ways to define shared answer sets based on OOD vocabs. While a subset is shown in Fig. 2, Tab. 10 lists the VQA accuracy of each model in the IID settings when evaluated on the questions in the test sets whose answers are shared between the top- k answers in both the IID and the OOD settings (see Sec. 4.1 for more details).

In some IID cases, restricting the answer set to common answers hurts the performance (indicated as a lack of dotted bar in Fig. 2). Interestingly, this pattern is observed across all models for some IID evaluations where the shared answer set is computed with respect to the VG benchmark only: VQAV2 for ViLBERT_{DISC} (-7.65 pp drop), VQAV2 (-8.65 pp drop) and GQA (-8.00 pp drop) for ViLBERT_{GEN}, VQAV2 (-6.86 pp drop) for ALBEF_{DISC}, and VQAV2 (-6.89 pp drop) for ALBEF_{GEN}. This seems to indicate that the GQA and VQAV2 questions corresponding to shared answer set with VG are more difficult than the average difficulty of these test sets.

	VQAV2	GQA	VG	VIZWIZ
VQAV2	–	0.41	0.51	0.11 [^]
GQA	0.25	–	0.44	0.14 [^]
VG	0.28	0.38	–	0.03 [^]
VIZWIZ	0.46	0.54	0.48	–

(a) Discriminative ALBEF.

	VQAV2	GQA	VG	VIZWIZ
VQAV2	–	0.45	0.48	0.26
GQA	0.26	–	0.45	0.22
VG	0.27	0.35	–	0.09 [^]
VIZWIZ	0.48	0.56	0.52	–

(b) Generative ViLBERT.

	VQAV2	GQA	VG	VIZWIZ
VQAV2	–	0.49	0.50	0.27
GQA	0.30	–	0.45	0.34
VG	0.25	0.38	–	0.16
VIZWIZ	0.50	0.57	0.51	–

(c) Discriminative ViLBERT.

Table 8: Spearman’s rank correlation between drops in test accuracy (from IID to OOD) and the differences in proportion of answer classes between IID and OOD fine-tune sets. Unless otherwise specified with a [^] character, ρ values are significant with $p < .05$. Rows correspond to the fine-tuning datasets, columns correspond to the test benchmarks.

	VQAV2	GQA	VG	VIZWIZ
VQAV2	–	0.43	0.51	0.25
GQA	0.27	–	0.43	0.19
VG	0.26	0.36	–	0.13
VIZWIZ	0.47	0.55	0.48	–

Table 9: Spearman’s rank correlation between drops in test accuracy (from IID to OOD) and the differences in proportion of answer classes between IID and OOD fine-tuning sets for ALBEF_{GEN}. $p < .05$ for all ρ . Rows correspond to the fine-tuning datasets, columns correspond to the test benchmarks.

Answer Frequency Correlation In order to examine the relationship between accuracy drop for less frequent classes, we first compute per answer-class accuracy (average accuracy of all test questions belonging to the same answer class) for answers in shared answer set. We then sort the shared answer classes based on their weighted drop in per-class accuracy from IID to OOD (IID accuracy - OOD accuracy), *i.e.* absolute drop in per-class accuracy weighted by number of data points belonging to that class in the test set. We then compute the Spearman’s rank correlation of these weighted drop in per-class accuracies with difference in percentage frequencies of the answer classes between IID

and OOD fine-tuning sets (percentage frequency of an answer class in IID minus its percentage frequency in OOD).

Tab. 8 list Spearman’s rank correlations of IID-to-OOD drops in test accuracy vs. proportion of answer classes in respective (IID and OOD) fine-tuning sets for ALBEF_{DISC}, ViLBERT_{GEN} and ViLBERT_{DISC} (see Sec. 4.1 for more details). As a simple baseline test, we also compute correlations and p-values for a permuted dataset to confirm their lack of significance, or correlation values close to zero.

Maximum Achievable Scores Table 2 lists maximum achievable accuracies, and Figure 7 shows the difference between those scores and bar values shown in Figure 1. In our analyses, we also noted that differences in answer pre-processing strategies can result in slightly different numbers than those reported in Tab. 2. However, those differences did not change the conclusion of our findings.

Effect of pretraining data size on ALBEF For the ALBEF model, while we often observe improvements by increasing the size of the multi-modal pretraining dataset (4M vs. 14M), the improvements are small. When pretraining on the smaller dataset (4M, see Fig. 8), we observe a median improvement (over no pretraining) of 1.9% for the discriminative and 4.9% for the generative ALBEF, while the median additional improvements due to larger pretraining dataset (14M) are 0.1% and 0.6% respectively (refer to Fig. 3). Surprisingly, there are also dataset pairs for which larger pretraining has a negative effect when compared to the performance with a smaller pretraining set (e.g., ALBEF model fine-tuned on VIZWIZ and tested on VQAV2).

C Potential Causes of Poor OOD Generalization: A Qualitative Study

In section 4, we observe that our pretrained models exhibit poor OOD generalization for the task of VQA. We also noted that this poor generalization is not entirely explained by the absence or poor representation of test answer classes in the training data. Here, we perform a qualitative study to dig deeper into the potential causes of the poor OOD generalization. We manually examine 20 randomly-sampled qualitative examples of failure cases on top-30 answer classes contributing the most to the drop in performance from IID to OOD.

We only focus on answer classes that are shared between the train and test splits to make sure the performance drop is not due to the absence of answer classes in the training dataset. We report the top-5 classes that contribute the most to the drop in performance for each OOD setting in Tab. 11. Below, we describe four major potential causes⁹ for the poor OOD generalization that we can infer from our qualitative study on ViLBERT_{DISC}¹⁰ and ALBEF_{GEN}. The specific examples reported below are for ViLBERT_{DISC}.

Poor reasoning skills. In Tab. 11, we can see that a model fine-tuned on VQAV2, VG, or VIZWIZ and evaluated on GQA shows the highest performance drop on classes such as “yes”, “no”, “right”, “left”, “top”, and “bottom”. For instance, VQAV2–GQA (fine-tuned on VQAV2, evaluated on GQA) model underperforms GQA–GQA model by 24% for “no.” Upon qualitative examination, we find that for many of such failure cases, the GQA questions are more compositional and hence require more complex reasoning (e.g., “Are there both bison and zebras in the image?”, “Is the cheese to the right or to the left of the empty plate?”) than the questions for the same answer classes in other datasets (e.g., from VQAV2 train set: “Is the TV turned on?”, “Which hand is the man holding up?”). This study re-affirms previous findings (Johnson et al., 2017; Hudson and Manning, 2019) – VQA models lack sufficient logical, spatial, and compositional reasoning skills – for the more recent, pretrained Transformer models.

Overfitting to the answer priors. Previous studies have shown that VQA models tend to be biased towards the prior distribution of answers in the training set (per question type) (Agrawal et al., 2018). We find that this limitation exists in the more recent pretrained models as well, and it is especially hurtful in the OOD settings because the priors need not be the same across train and test sets, unlike in the IID settings. For instance, ViLBERT_{DISC} fine-tuned on VQAV2 predicts “2” for a lot questions with target answer “1” in the VG test set. Similarly, sometimes ViLBERT_{DISC} fine-tuned on VG incorrectly predicts “helmet” for

⁹For poor OOD generalization on the VIZWIZ benchmark, one of the reasons could be difference in image distributions between VIZWIZ (that contains many blurry pictures, or pictures with poor lighting conditions) and other three datasets (that contain clear pictures).

¹⁰We use the model trained with the official codebase.

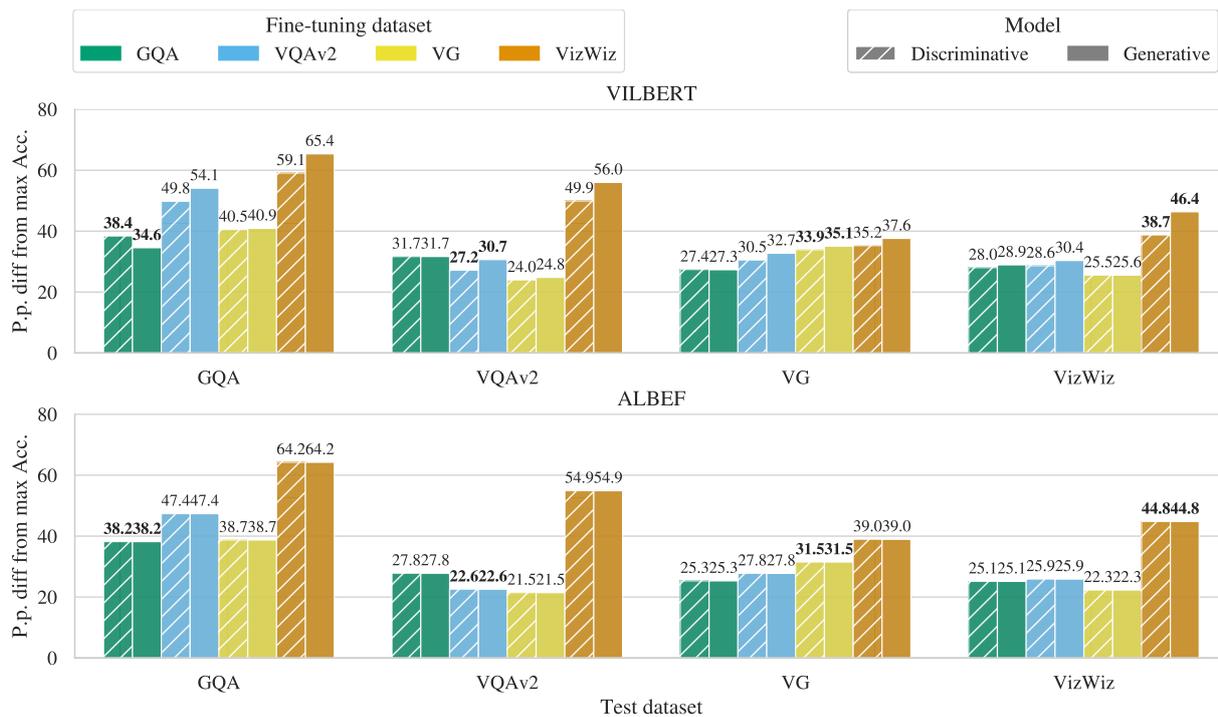


Figure 7: Percentage point difference between maximum achievable accuracies in Table 2 and accuracies in Figure 1. Results for VILBERT pretrained on the same data as ALBEF 4M are also shown.

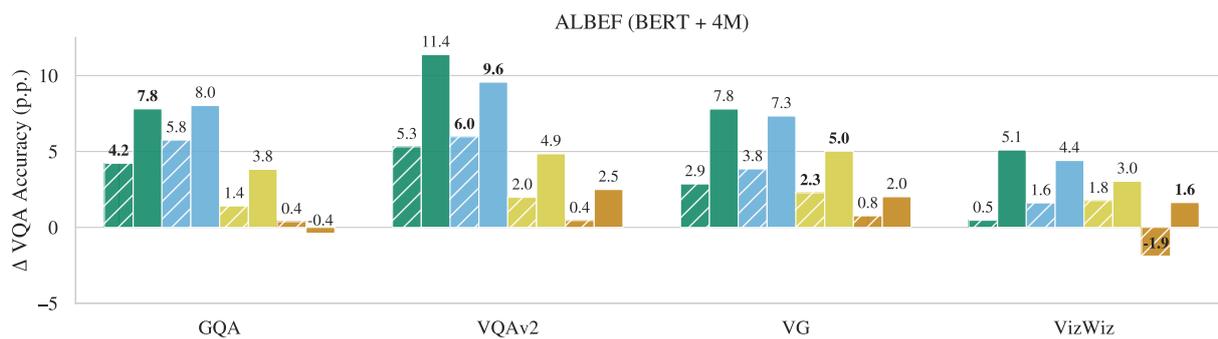


Figure 8: Difference in VQA accuracy (p.p.) for ALBEF that has and has not been pretrained on the 4M dataset.

VQAv2 test questions such as “What is the skateboarder wearing to protect his head?”, “What protective gear is he wearing?” when the skateboarder is not wearing anything. This indicates that the model is relying on answer priors rather than visual grounding. Our experimental results on VQA-CP (Sec. 6) directly quantify the extent of such limitations in current models.

Overfitting to the question format. For each answer class, there is usually a limited variation in the format of questions in the fine-tuning set. For some of the answer classes showing poor OOD generalization, we found that certain question formats are quite dominant in the fine-tuning set, and that these dominant formats are different between the OOD fine-tuning and test sets. Thus, we conjecture

that models are likely overfitting to such dominant formats in fine-tuning data and hence fail to generalize at test time when the format changes. For instance, questions about “chair” in the VQAv2 fine-tuning set are mostly of the form “What is ... sitting on?” whereas in the GQA test set, they are mostly of the form “What kind of furniture is ...?”. Thus, the “chair” class accuracy of VILBERT_{DISC} fine-tuned on VQAv2 drops from 48% when tested on VQAv2 to 38% on the GQA test set. As another example, VILBERT_{DISC} trained on GQA fails terribly for “dog” and “cat” classes on VG test set (accuracy drops of 47% and 43% respectively, where drop is between GQA–GQA and GQA–VG). GQA questions are mostly of the form “What animal ...?” or “What kind of animal

Model	Test	Fine-tune	Answer Set	VQA Acc. (IID)	VQA Acc. (OOD)	Difference
ViLBERT _{DISC}	GQA	GQA	GQA \cap VQAv2	63.05	48.53	14.52
ViLBERT _{DISC}	GQA	GQA	GQA \cap VG	52.32	35.08	17.24
ViLBERT _{DISC}	GQA	GQA	GQA \cap VizWiz	64.91	28.39	36.52
ViLBERT _{DISC}	VG	VG	VG \cap VQAv2	58.18	51.97	6.21
ViLBERT _{DISC}	VG	VG	VG \cap GQA	59.57	39.15	20.42
ViLBERT _{DISC}	VG	VG	VG \cap VizWiz	60.65	13.23	47.42
ViLBERT _{DISC}	VizWiz	VizWiz	VizWiz \cap VQAv2	51.78	22.37	29.41
ViLBERT _{DISC}	VizWiz	VizWiz	VizWiz \cap GQA	51.05	15.40	35.65
ViLBERT _{DISC}	VizWiz	VizWiz	VizWiz \cap VG	50.07	13.59	36.48
ViLBERT _{DISC}	VQAv2	VQAv2	VQAv2 \cap GQA	71.74	52.42	19.32
ViLBERT _{DISC}	VQAv2	VQAv2	VQAv2 \cap VG	58.10	48.84	9.26
ViLBERT _{DISC}	VQAv2	VQAv2	VQAv2 \cap VizWiz	68.20	33.87	34.33
ViLBERT _{GEN}	GQA	GQA	GQA \cap VQAv2	67.01	44.03	22.98
ViLBERT _{GEN}	GQA	GQA	GQA \cap VG	57.35	34.40	22.95
ViLBERT _{GEN}	GQA	GQA	GQA \cap VizWiz	69.63	20.73	48.90
ViLBERT _{GEN}	VG	VG	VG \cap VQAv2	55.52	47.93	7.59
ViLBERT _{GEN}	VG	VG	VG \cap GQA	57.65	39.26	18.39
ViLBERT _{GEN}	VG	VG	VG \cap VizWiz	58.84	7.67	51.17
ViLBERT _{GEN}	VizWiz	VizWiz	VizWiz \cap VQAv2	43.06	19.58	23.48
ViLBERT _{GEN}	VizWiz	VizWiz	VizWiz \cap GQA	42.57	13.71	28.86
ViLBERT _{GEN}	VizWiz	VizWiz	VizWiz \cap VG	41.11	13.20	27.91
ViLBERT _{GEN}	VQAv2	VQAv2	VQAv2 \cap GQA	68.01	52.35	15.66
ViLBERT _{GEN}	VQAv2	VQAv2	VQAv2 \cap VG	53.59	46.78	6.81
ViLBERT _{GEN}	VQAv2	VQAv2	VQAv2 \cap VizWiz	64.69	26.82	37.87
ALBEF _{DISC}	GQA	GQA	GQA \cap VQAv2	63.24	51.06	12.18
ALBEF _{DISC}	GQA	GQA	GQA \cap VG	53.09	37.97	15.12
ALBEF _{DISC}	GQA	GQA	GQA \cap VizWiz	64.85	22.14	42.71
ALBEF _{DISC}	VG	VG	VG \cap VQAv2	61.38	55.83	5.55
ALBEF _{DISC}	VG	VG	VG \cap GQA	63.76	43.82	19.94
ALBEF _{DISC}	VG	VG	VG \cap VizWiz	64.12	4.52	59.60
ALBEF _{DISC}	VizWiz	VizWiz	VizWiz \cap VQAv2	42.42	26.23	16.19
ALBEF _{DISC}	VizWiz	VizWiz	VizWiz \cap GQA	40.49	20.44	20.05
ALBEF _{DISC}	VizWiz	VizWiz	VizWiz \cap VG	38.15	19.68	18.47
ALBEF _{DISC}	VQAv2	VQAv2	VQAv2 \cap GQA	76.64	57.18	19.46
ALBEF _{DISC}	VQAv2	VQAv2	VQAv2 \cap VG	63.47	52.78	10.69
ALBEF _{DISC}	VQAv2	VQAv2	VQAv2 \cap VizWiz	72.84	28.08	44.76
ALBEF _{GEN}	GQA	GQA	GQA \cap VQAv2	65.81	51.72	14.09
ALBEF _{GEN}	GQA	GQA	GQA \cap VG	56.08	37.71	18.37
ALBEF _{GEN}	GQA	GQA	GQA \cap VizWiz	67.05	27.61	39.44
ALBEF _{GEN}	VG	VG	VG \cap VQAv2	62.71	57.33	5.38
ALBEF _{GEN}	VG	VG	VG \cap GQA	65.48	51.17	14.31
ALBEF _{GEN}	VG	VG	VG \cap VizWiz	66.20	21.13	45.07
ALBEF _{GEN}	VizWiz	VizWiz	VizWiz \cap VQAv2	52.85	28.96	23.89
ALBEF _{GEN}	VizWiz	VizWiz	VizWiz \cap GQA	52.58	22.21	30.37
ALBEF _{GEN}	VizWiz	VizWiz	VizWiz \cap VG	51.94	23.56	28.38
ALBEF _{GEN}	VQAv2	VQAv2	VQAv2 \cap GQA	78.03	62.93	15.10
ALBEF _{GEN}	VQAv2	VQAv2	VQAv2 \cap VG	65.20	55.64	9.56
ALBEF _{GEN}	VQAv2	VQAv2	VQAv2 \cap VizWiz	74.38	39.37	35.01

Table 10: VQA accuracy of each model in the IID settings (see column VQA Acc. (IID)) when evaluated on the questions in the test sets whose answers are shared between the top- k answers in both the IID and the OOD settings. Please refer to Sec. 4.1 for more details. Answer Set: OOD benchmarks with respect to which IID shared answer set accuracy is computed. VQA Acc. (OOD): OOD accuracy on questions corresponding to the shared answer set, *i.e.* when fine-tuned on the OOD dataset mentioned in Answer Set column and tested on the benchmark mentioned in the Test column. Difference: VQA Acc. (IID) - VQA Acc. (OOD). Gray bands highlight the OOD benchmarks with respect to which IID shared answer set accuracy is computed in Fig. 2.

...?” whereas VG questions often do not mention the word “animal” and are of the form “Who is ... ?” or “What is ... ?” (*e.g.*, “Who is holding the Frisbee?”, “What is on the leash?”). Similarly, for the answer class “pizza”, ViLBERT_{DISC} fine-tuned

on VG has mostly seen questions of the format “What food is this?”, “What is the man eating?”, “What is on the plate?”, “What’s in the box?” in VG fine-tuning set. However, when evaluated on the VQAv2 test set, the model fails to respond cor-

rectly for questions about “pizza” such as, “What snack is this?” (model response: “pineapple”), “What recipe this will become?” (model response: “cheese”), “What’s in the bowl” (model response: “tomato sauce”). For the last example, perhaps the model is not expecting pizza to be in a bowl.

Related to above, we observed that sometimes $\text{ViLBERT}_{\text{DISC}}$ fails to produce the correct answer type for a given question. For instance, $\text{ViLBERT}_{\text{DISC}}$ fine-tuned on VG responds with “woman” to the question “Is the person who is cutting these carrots right handed or left handed?”. So it appears as if the VG model does not understand the question structure in this example, *i.e.* the response is expected to be either “right” or “left”. Similarly, for the question “Are there more blue or black shirts?”, VG model responds with “rolled up”. Similarly, it answers “1 on right” to the question “What type of apple is shown?”, instead of describing some attribute of apple such as “green”.

Stringent evaluation metric. We notice that sometimes the models’ responses are correct but they are evaluated as incorrect because those responses do not exist in the ground-truth answers. For instance, VQAV2–VG model gets penalized for answering “table” instead of “on table”¹¹ (Q: “Where is . . . ?”) or “sunny” instead of “clear” (Q: “How is the weather?”). More examples in Fig. 5. This effect is expected to be more pronounced for the OOD evaluation than IID, because in IID a model can learn the format of the test answer (“on table” vs. “table”, “clear vs. sunny”) from the train set, whereas in OOD the format in the train set can be different from the test set. Also, such stringent evaluation (*i.e.*, performing string matching with a small set of ground-truth answers) is expected to hurt generative models more than discriminative ones because they show more variations in the form of the answers as they are not limited by a fixed answer vocabulary (*e.g.*, “pizza slices” instead of “pizza” (Q: “What are these?”), “pizzeria” instead of “pizza” (Q: “What kind of restaurant is this?”)). We observed that, VG model (model fine-tuned on VG) evaluated on GQA answers questions about “man” with “snowboarder”, “man on left” (*i.e.* more descriptive referring expressions) than just saying “man” but it does not get any credit

¹¹Note that before computing the accuracy, both the predicted and the ground truth answers are pre-processed for answer normalization but such pre-processing is very basic. More details of the pre-processing can be found at <https://visualqa.org/evaluation.html>

because GQA ground truth is “man”. To quantify the extent of this issue and measure its effect on discriminative vs. generative models, IID vs. OOD settings, we perform human evaluation of machine generated answers and provide additional insights in Sec. 8.

Poor performance of GQA model on color questions (both IID and OOD): $\text{ViLBERT}_{\text{DISC}}$ fine-tuned on GQA does not seem to be transferring well to color questions in the VQAV2 and VG test sets (and even in IID GQA test set). In Tab. 11, we can see that the top-5 answer classes with highest drop in IID-to-OOO performance for GQA model have quite a few colors. For instance, for the answer class “red” in the VG test set, GQA model fails to correctly answer simple questions (given the kind of questions GQA model is fine-tuned on) such as “What is the primary color of the sign on the right?”, “What is the main color of the strawberry?”, “What color is the pull luggage of the woman?”, “What color are the pepperonis?”. It is not clear why GQA model does not perform well on color questions.

D Human Evaluation

Method. We used Amazon Mechanical Turk to collect human judgment about model responses on a random subset of 10K questions for each of the test sets—VQAV2, GQA and VG. Since the size of VIZWIZ test set is less than 10K, we collected human judgment on all the VIZWIZ test questions. However, we dropped the questions that were tagged as “unanswerable” or “unsuitable” (more details are provided below under “Filtering VizWiz data”). The total number of VIZWIZ test questions for which we collected human judgment is 1440 (per model). We performed human evaluation of the responses from the following models – $\text{ViLBERT}_{\text{DISC}}$ ¹² and $\text{ViLBERT}_{\text{GEN}}$ trained on the VQAV2, GQA, VG datasets. We did not collect human judgements for models *fine-tuned* on VIZWIZ, because a significant proportion of the responses from these models tend to be “unanswerable” or “unsuitable” (35% on VQAV2, 39% on GQA, 65% on VG, and 64% on VIZWIZ). Collecting human feedback about such responses would

¹²For $\text{ViLBERT}_{\text{DISC}}$, we had initially collected human judgements for the version trained using the official codebase, and we did not collect annotations again for our reimplementation due to time constraints. Given our results above, we do not expect significant differences between the two versions.

Train data	Test data	Model	Answer classes	
VQAv2	GQA	Discriminative ViLBERT	no, yes, left, right, top	
		Discriminative ViLBERT (in-house)	no, yes, right, bottom, color	
		Generative ViLBERT (in-house)	no, yes, right, left, bottom	
		Discriminative ALBEF	no, left, yes, bottom, chair	
	VG	Generative ALBEF	left, no, yes, bottom, top	
		Discriminative ViLBERT	1, no 1, daytime, on table, in sky	
		Discriminative ViLBERT (in-house)	daytime, 1, white, 2, black	
		Generative ViLBERT (in-house)	daytime, white, 2, black, 1	
	VizWiz	Discriminative ALBEF	1, daytime, in sky, on table, white	
		Generative ALBEF	1, daytime, black, in sky, clear	
		Discriminative ViLBERT	no, blue, yes, white, black	
		Discriminative ViLBERT (in-house)	yes, black, water bottle, corn, soup	
GQA	VQAv2	Generative ViLBERT (in-house)	pink, brown, corn, wine, keys	
		Discriminative ALBEF	keyboard, no, soup, cake, samsung	
		Generative ALBEF	soup, lotion, black, brown, corn	
		Discriminative ViLBERT	yes, no, white, red, black	
	VG	Discriminative ViLBERT (in-house)	no, yes, white, red, tennis	
		Generative ViLBERT (in-house)	no, yes, white, red, tennis	
		Discriminative ALBEF	no, yes, white, red, right	
		Generative ALBEF	no, yes, right, red, black and white	
	VizWiz	Discriminative ViLBERT	white, trees, green, black, black and white	
		Discriminative ViLBERT (in-house)	white, black, trees, green, blue	
		Generative ViLBERT (in-house)	white, trees, black, green, brown	
		Discriminative ALBEF	white, trees, black and white, grass, green	
VG	VQAv2	Generative ALBEF	trees, green, black and white, black, grass	
		Discriminative ViLBERT	no, blue, yes, white, laptop	
		Discriminative ViLBERT (in-house)	blue, white, black, dog, laptop	
		Generative ViLBERT (in-house)	white, blue, laptop, black, dog	
	GQA	Discriminative ALBEF	no, keyboard, soup, red, cake	
		Generative ALBEF	no, dog, keyboard, laptop, blue	
		Discriminative ViLBERT	0, white, nothing, gray, red	
		Discriminative ViLBERT (in-house)	0, 3, left, nothing, brown	
	VizWiz	Generative ViLBERT (in-house)	0, 1, gray, left, wii	
		Discriminative ALBEF	0, nothing, left, brown, 2	
		Generative ALBEF	0, 3, nothing, right, gray	
		Discriminative ViLBERT	right, left, bottom, top, gray	
VizWiz	GQA	Discriminative ViLBERT (in-house)	bottom, left, top, color, large	
		Generative ViLBERT (in-house)	left, bottom, color, top, gray	
		Discriminative ALBEF	left, bottom, top, black, chair	
		Generative ALBEF	left, bottom, color, top, gray	
	VG	Discriminative ViLBERT	blue, black, grey, red, soup	
		Discriminative ViLBERT (in-house)	grey, black, blue, white, computer screen	
		Generative ViLBERT (in-house)	grey, blue, black, pink, computer screen	
		Discriminative ALBEF	grey, soup, remote, cake, samsung	
	VQAv2	GQA	Generative ALBEF	grey, blue, soup, wine, pink
			Discriminative ViLBERT	no, yes, 1, 2, white
			Discriminative ViLBERT (in-house)	no, 1, 2, 0, white
			Generative ViLBERT (in-house)	no, yes, 1, 2, white
VG		Discriminative ALBEF	no, 1, 2, yes, blue	
		Generative ALBEF	yes, no, 1, 2, 0	
		Discriminative ViLBERT	no, right, left, man, bottom	
		Discriminative ViLBERT (in-house)	no, right, bottom, man, top	
VizWiz		Generative ViLBERT (in-house)	no, yes, left, right, man	
		Discriminative ALBEF	no, left, bottom, top, man	
		Generative ALBEF	yes, left, no, bottom, top	
		Discriminative ViLBERT	1, white, green, 2, black	
VG	Discriminative ViLBERT (in-house)	1, 2, white, green, man		
	Generative ViLBERT (in-house)	1, 2, white, green, black		
	Discriminative ALBEF	1, green, white, 1, 2, blue		
	Generative ALBEF	1, 2, black, man, green		

Table 11: Top-5 answer classes with highest performance drop from IID to OOD (for the same test set) for all OOD configurations. The answer classes are sorted by drop in weighted (wtd) accuracy, i.e. drop in absolute (abs) accuracy weighted by the # test questions for that answer class.

not provide useful insights, because all questions in VQAv2, GQA and VG should be answerable, therefore all cases of “unanswerable” should be incorrect. Such responses are just a side effect of a model’s priors caused by all the unanswerable training points in the VIZWIZ fine-tuning set.

For each response, we asked 5 raters to evaluate the question, image, and a given model response, and indicate through a binary choice whether they considered the model response a correct answer to the question or not. To control the quality of the data, we filtered out low quality data using different heuristics such as distribution of yes/no answers for each worker, their mean submission times, average agreement with their fellow workers, or average alignment with the automatic accuracy.¹³ In each of these cases, we looked at random samples from the outliers to qualitatively confirm our hypothesis. More details about the human evaluation interface are presented in the next paragraph.

To compute human accuracy of a model response (for a given question and image), we considered a response correct if at least 4 raters voted it is correct, and incorrect otherwise. We decided so in order to decrease noise introduced by cases where there was low agreement between raters.

Data collection interface. Fig. 10 shows a sample of the interface the MTurk raters used to submit their responses. The workers were shown some examples, but in order not to bias them, we did not give them detailed guidance as to what should be considered correct for not - rather we asked them to rely on common sense, and consider an answer correct if it seems both factually accurate and natural in the context. See Fig. 11 for details.

Filtering VizWiz data. For human evaluation we filtered out all image-question pairs for which the ground truth answer indicates it is *unanswerable*. That is, we have not collected human feedback for questions for which the ground truth answer appears in the following list:

- "unanswerable", "unsuitable"
- "insufficient image"
- "unknown", "unsure", "not clear"
- "blurry", "too blurry"

¹³How frequently a worker’s response (yes/no) aligns with the automatic accuracy computed (100.0/0.0) More specifically, we equate the worker’s *yes* response with 100.0 and *no* with 0.0 and look at the average difference between the worker’s response and the automatic accuracy

- "i don’t know", "don’t know", "i don no", "no idea"
- "unusable image", "unsuitable image", "unstable image", "insufficient image quality", "unreadable"
- "i can’t tell", "can’t tell", "can’t see"
- "0" ¹⁴

In particular, this left us with 1440 questions for the VIZWIZ dataset.

Results. We report the human accuracies for VILBERT_{DISC} and VILBERT_{GEN} in Fig. 9 (bottom). We also report the accuracies obtained using automatic metrics (please see Sec. 3.2 for description of automatic metrics for each dataset) computed over the same random subset of test questions as that used for human evaluation in Fig. 9 (top). Please refer to Sec. 8 for discussion of results.

Qualitative examples of questions being incorrectly penalized by automatic evaluation Tab. 12 shows some examples for responses which were awarded 0.0 accuracy using automatic metrics but were marked as correct by all 5 raters during human evaluation.

Discussion on VQA data quality For the collected human judgement data, we find that for a significant number of questions (32%) there was low agreement between the 5 raters, i.e. either 3/5 answered *correct* while the remaining 2/5 answered *incorrect*, or vice-versa. Note that this is after we already filtered out low quality judgements. We have to recognize that, despite our best efforts to control data quality using our heuristics, there might still be low quality data in our dataset. Low quality annotations can be misleading and might distort the results of our analysis. Yet, we believe that we have collected a large enough sample to dampen the effect of these on the reported results. Upon examining some examples from such low agreement questions, we find that many such cases highlight the quality of the VQA data. For instance, questions not being sufficiently objective but up for interpretation, questions phrased poorly that make it difficult to understand what the question is asking about, *etc.* We discuss these further below.

- **Low agreement due to ambiguity.** One reason why human raters might give different feedback stems from ambiguity and subjectiv-

¹⁴Qualitative examples have suggested that often this was used to indicate *unanswerable*.



Figure 9: Human evaluation of ViLBERT_{DISC} (shaded bars) and ViLBERT_{GEN} (plain bars) and comparison with automatic evaluation on a random subset (10K) of test questions for each test dataset – GQA, VQAV2, VG and VIZWIZ. Accuracies in bold denote the IID settings. Top: accuracies obtained using automatic evaluation. Bottom: accuracies obtained using human evaluation.

ity around the question and the contents of the image. In these cases it is up to the raters subjective opinion to judge whether the answer is acceptable or not. Find some qualitative examples in Tab. 13.

- **Low agreement due to poor quality question.** Some questions in the original dataset are of rather poor quality which makes it near impossible for the rater to provide a valuable response. Find some qualitative examples of such questions in Tab. 14.

Also surprisingly, from Fig. 6, we see that for models fine-tuned on VQAV2 or GQA and tested on VQAV2, and models fine-tuned on GQA and tested on GQA, human evaluation yields lower accuracy than automatic evaluation! This is not as expected. Upon examining some examples for responses with 100.0 automatic accuracy but marked as *incorrect* by at least 4 human raters, we again find some noise in the ground-truth answers. Tab. 15 shows some examples. Below we report the number of questions where at least 4 human raters voted incorrect even though the automatic metric indicated ≥ 90.0 accuracy. Generative case: {GQA \rightarrow GQA (fine-tuned on GQA, tested on GQA): 86, GQA \rightarrow VQAV2: 49, VQAV2 \rightarrow VQAV2: 48}, Discriminative case: {GQA \rightarrow GQA: 128, GQA \rightarrow VQAV2: 52, VQAV2 \rightarrow VQAV2: 76}.



dataset: VG
img_id: 2413078
q_id: 151766
Q: What are they wearing on their heads?
A: helmet
GT: helmets
accuracy: 0.0
votes: 5 yes, 0 no



dataset: VQAv2
img_id: 546983
q_id: 546983002
Q: What is flying in the sky?
A: kite
GT: kites
accuracy: 0.0
votes: 5 yes, 0 no



dataset: GQA
img_id: 2413903
q_id: 5199731
Q: Which kind of device is on the table?
A: laptop
GT: cell phone
accuracy: 0.0
votes: 5 yes, 0 no

Table 12: A few examples of questions to which the model gave a response that was objectively correct, yet the automatic evaluation metric has marked these data points as 0% accurate. (*votes* here refers to how many raters selected *yes* (i.e. correct) or *no* (i.e. incorrect) when asked about this data point, while GT stands for *ground truth*.)



dataset: VG
img_id: 2358330
q_id: 700783
Q: Where is he riding?
A: park
GT: in street
accuracy: 0.0
votes: 3 yes, 2 no



dataset: VQAv2
img_id: 254750
q_id: 254750003
Q: Where is the toilet paper?
A: bathroom
GT: on sink
accuracy: 0.0
votes: 3 yes, 2 no



dataset: GQA
img_id: 2338989
q_id: 17319928
Q: What is on the green sign?
A: word
GT: flag
accuracy: 0.0
votes: 3 yes, 2 no

Table 13: Low agreement due to ambiguity. In many cases, whether an answer is correct could be up to interpretation. (*votes* here refers to how many raters selected *yes* (i.e. correct) or *no* (i.e. incorrect) when asked about this data point, while GT stands for *ground truth*)



dataset: VG
img_id: 2396675
q_id: 1453804
Q: What is the kitchen dresser?
A: cabinet
GT: brown
accuracy: 0.0
votes: 2 yes, 3 no



dataset: VQAv2
img_id: 503518
q_id: 503518006
Q: What is happening?
A: phone
GT: watching videos, showing phone
accuracy: 0.0
votes: 2 yes, 3 no



dataset: GQA
img_id: 2346071
q_id: 5863992
Q: What kind of furniture is playing a game?
A: table
GT: couch
accuracy: 0.0
votes: 2 yes, 3 no

Table 14: Low agreement due poor question quality. Some questions have poor phrasing that make it difficult to understand what exactly is being asked. In these cases even the humans are not sure what to answer. (*votes* here refers to how many raters selected *yes* (i.e. correct) or *no* (i.e. incorrect) when asked about this data point, while GT stands for *ground truth*)



dataset: VQAv2
img_id: 367228
q_id: 367228001
Q: Is the kite flying high enough?
A: yes
GT: [no, yes, yes, no, no, no, yes, no, no, yes]
accuracy: 100.0
votes: 1 yes, 4 no



dataset: VQAv2
img_id: 197745
q_id: 197745007
Q: How many spots are on this animal?
A: 100
GT: [70, 100, 100, numerous, 200, 100, 100, 100, 20, lots]
accuracy: 100.0
votes: 1 yes, 4 no



dataset: VQAv2
img_id: 264737
q_id: 264737002
Q: How many animals are in the picture?
A: 6
GT: [7, 6, 6, 9, 6, 6, 6, 7, 7, 6]
accuracy: 100.0
votes: 1 yes, 4 no

Table 15: Examples of the few cases where humans considered the response incorrect despite 100.0 automatic accuracy. (*votes* here refers to how many raters selected *yes* (i.e. correct) or *no* (i.e. incorrect) when asked about this data point, while GT stands for *ground truth*)



Question: What are the people standing on?

Response: sand

Is the displayed **response** a correct answer to the above **question** about the image?

Correct

Incorrect

Figure 10: Sample of the MTurk interface the raters used to annotate data.

Instructions

You will see a question and a corresponding, short response displayed next to the image. The question and response both concern the contents of the image. Based on the image, is the provided response a correct answer to the question?

Your task: Please indicate whether the response is correct by selecting **Correct** or **Incorrect**.

A response should be considered **correct** if:

1. It is relevant to the question and answers the question in a direct, grammatically correct way, AND
2. It accurately reflects the contents of the displayed image.

A response should be considered **incorrect** if:

1. It is not a coherent, natural response to the question, OR
2. It does not accurately reflect the contents of the displayed image.

Note: We expect the responses to be short and concise, not full sentences. Thus a response should **not** be considered incorrect solely on the basis that it is brief.

Please see the examples below to understand the task better:

- **Example 1:** This response is correct.



Question: What colour is the flip-flop?

Response: red

Is the displayed **response** a correct answer to the above **question** about the image?

Correct

Incorrect
- **Example 2:** This response is correct.



Question: Is this a cat?

Response: no

Is the displayed **response** a correct answer to the above **question** about the image?

Correct

Incorrect
- **Example 3:** This response is incorrect. Although the response addresses the question, it does not accurately reflect the image's contents. A correct response would be 'baseball'.



Question: What are they playing?

Response: basketball

Is the displayed **response** a correct answer to the above **question** about the image?

Correct

Incorrect
- **Example 4:** This response is incorrect. Although it is true that the scene is at the zoo, the response does not directly address the question. A correct response would be 'no'.



Question: Is this at a museum?

Response: at zoo

Is the displayed **response** a correct answer to the above **question** about the image?

Correct

Incorrect

Figure 11: Instructions given to MTurk raters.

Our kind of people? Detecting populist references in political debates

Christopher Klamm

Ines Rehbein

Simone Paolo Ponzetto

Data and Web Science Group

University of Mannheim, Germany

{christopher, ines, simone}@informatik.uni-mannheim.de

Abstract

This paper investigates the identification of populist rhetoric in text and presents a novel cross-lingual dataset for this task. Our work is based on the definition of populism as a "communication style of political actors that refers to the people" but also includes anti-elitism as another core feature of populism. Accordingly, we annotate references to *The People* and *The Elite* in German and English parliamentary debates with a hierarchical scheme. The paper describes our dataset and annotation procedure and reports inter-annotator agreement for this task. Next, we compare and evaluate different transformer-based model architectures on a German dataset and report results for zero-shot learning on a smaller English data. We then show that semi-supervised tri-training can improve results in the cross-lingual setting. Our dataset can be used to investigate how political actors talk about *The Elite* and *The People* and to study how populist rhetoric is used as a strategic device.

1 Introduction

The rise of populism in Europe and throughout the world has been noted not only in politics and the media but also has been the subject of many studies in political science and related areas (see, among others, [Mudde \(2007\)](#)). The concept of populism, however, is complex and vague and eludes a strict definition. So far, only limited agreement exists on the exact properties of the construct, despite numerous efforts to provide a clear definition.

In the literature, populism has been described as an ideology ([McRae, 1969](#); [Mudde, 2004](#)), a rhetoric ([Abts and Rummens, 2007](#)) or style ([Moffitt, 2016](#)), as a political strategy ([Weyland, 2001, 2021](#); [Hawkins and Kaltwasser, 2017](#)) and as a discourse ([Laclau, 1977](#); [Aslanidis, 2016](#)), amongst others (see [Aslanidis \(2018\)](#) for a short overview). The *Oxford Handbook on Populism* ([Rovira Kaltwasser et al., 2017](#)) groups existing work into three

dominant approaches to analyzing populism, i.e., (i) the ideational approach of [Mudde \(2004\)](#), (ii) the socio-cultural approach ([Ostiguy, 2017](#)), and (iii) the political-strategic approach ([Hawkins and Kaltwasser, 2017](#)), each one capturing a different view on populism.

Nevertheless, most studies agree that *anti-elitism* and *people-centrism* are amongst the core dimensions of populist rhetoric, and the two dimensions are therefore included as features in most survey tools used to measure the degree of populism of political parties and actors ([Polk et al., 2017](#); [Rooduijn et al., 2019a](#); [Meijers and Zaslove, 2020](#)). One major drawback of surveys, however, is that they only provide us with one score for each party or actor and can not be used to study how populist rhetoric is used as a strategic tool in different contextual settings.

As a result, more and more efforts have been made recently to measure populist and anti-elitist attitudes directly from text ([Rooduijn and Pauwels, 2011](#); [Dai, 2018](#); [Aslanidis, 2018](#); [Ernst et al., 2019](#); [Hawkins et al., 2019](#); [di Cocco and Monechi, 2021](#); [Vaughan and Heft, 2022](#)). This has the advantage of providing us with more fine-grained and context-dependent measures that enable us to investigate when and how anti-elitist rhetoric is used as a strategic tool in party competition ([Vaughan and Heft, 2022](#)). In addition, it has been suggested that populist rhetoric targeting political elites might function "as a form of ethnoracial dog-whistle politics" ([Bonikowski and Zhang, 2023, p.2](#)). Evidence for this claim comes from the frequent co-occurrence of right-wing populism with nativist messages, as shown in Example 1.1 below, taken from a parliamentary speech of a far-right politician in the German Bundestag.

Ex. 1.1 *Because the Merkel government has lied to the people about how long refugees and illegal migrants will actually be with us [...] (N. Kleinwächter, AfD, 15/11/2019)*

This example illustrates the different dimensions of populist rhetoric where anti-elitism is combined with a Manichean worldview that separates society into two antagonistic camps, *the corrupt elite* and *the pure people* (Mudde, 2004). This divide into *Us-versus-Them*, also known as *Othering*, is a well-known strategy for creating in- and outgroups, used to conceptualize specific groups as outsiders and to depict them as inferior or even as dangerous. Example 1.1 uses *Othering* to transfer the message that “refugees and illegal migrants” are not part of *The People* and that an immoral political elite is acting against *The People*’s general interest (“the Merkel government has lied to the people”).

While there is no shortage of studies on various aspects of populism, only a few works have tried to develop robust and reliable measures of populism that can be used for empirical research at scale to quantify the degree of populism expressed by political actors, such as politicians and parties. Being able to assess populism from a quantitative standpoint using large amounts of data, e.g., text, has the potential, in turn, to help us understand the causes and consequences of populism by allowing us to track its spatial and temporal distribution.

In the paper, we provide a methodology to detect and quantify references to *The People* and *The Elite* in large amounts of text. We present a novel dataset of German and English political debates where instances of *The People* and *The Elite* have been manually annotated and use this data to learn to predict those references in monolingual and cross-lingual settings. We then show that these predictions align with the results of expert surveys for measuring populism but, crucially, provide us with *more fine-grained and context-sensitive* information that can be used to study left- and right-wing populism in parliamentary debates at large scale. We make all data and models available at <https://github.com/umanlp/mope.git>.

2 Related Work

2.1 Defining Populism

Defining populism is an intellectual challenge *per se*. Most scholars, however, agree that populism is a multi-dimensional construct and that *anti-elitism* and *people-centrism* are two of the core characteristics of populist discourse (Mudde, 2004; Hawkins, 2009; Dai, 2018; Schulz et al., 2017). Many studies have adapted Mudde’s view of populism as “a thin-centered ideology that considers society to be

ultimately separated into two homogeneous and antagonistic camps, ‘the pure people’ versus ‘the corrupt elite’” (Mudde, 2004, p. 543).

Another influential view distinguishes between *thin* and *thick* populism, where the former is considered as a “communication style of political actors that refers to the people” (Jagers and Walgrave, 2007, pp.322). *Thick* populism, on the other hand, is similar to Mudde’s definition and combines people-centrist references with anti-elitism and the exclusion of certain minority groups from *The People*. Our operationalization of populist rhetoric is most similar to Jagers and Walgrave (2007)’s *thin populism*. Still, it can also be used within other conceptual frameworks that rely on people-centrism and anti-elitism as defining features of populism.

So far, a variety of approaches have been proposed for analyzing populism. Some works rely on **expert opinions and surveys** (Rooduijn et al., 2019b; Meijers and Zaslove, 2021a) to obtain theoretically grounded measurements of populism. This approach, however, only yields scores on the level of parties or organizations but defies a more fine-grained or graded analysis on the text or sub-text level (Aslanidis, 2018). **Text-based approaches**, on the other hand, have the potential to identify context-sensitive manifestations of populism and its characteristics and, in turn, profile political actors along multiple dimensions.

2.2 Measuring populism in text

Text-based methods for measuring populism can be classified into four main approaches. The first is based on **manual content analysis** where a larger text is segmented into smaller units, and trained human coders inspect each unit and search for populist cues (Jagers and Walgrave, 2007; Hawkins, 2009, *inter alia*). While this approach can obtain high content validity, it is also extremely time-consuming and, depending on the categories in the codebook, does not necessarily generalize well across different topics, geographical and cultural specificities, or time periods.

A second approach, called **holistic coding**, also involves human annotation where trained coders read the document and, based on the comparison to a small set of anchor texts, decide whether the text as a whole should be considered as populist or not (Hawkins and Castanho Silva, 2018; Hawkins et al., 2019, *inter alia*). Document-level analysis is less fine-grained, and often it is not evident why a

Level 1	<i>Elite E</i>				<i>People P</i>	
Level 2	<i>Person P</i>		<i>Organisation O</i>		–	
Level 3	Domain:	Label:	Domain:	Label:	Domain:	Label:
	Politics	EPPOL	Politics	EOPOL	Nation	PNAT
	Economy	EPECON	Economy	EOECON	Ethnicity/religion	PETH
	Finance	EPFIN	Finance	EOFIN	Profession/function	PFUN
	Media	EPMED	Media	EOMED	Age	PAGE
	Science	EPSCI	Science	EOSCI	Social variables	PSOC
	Religion	EPREL	Religion	EOREL	(gender/class/...)	
	Culture	EPCULT	Culture	EOCULT	Generic	PGEN
	Military	EPMIL	Military	EOMIL		
	NGOs	EPNGO	NGOs	EONGO		
	Movements	EPMOV	Movements	EOMOV		
Other:	references to own person EPOWN		geo-political entity GPE			

Table 1: Hierarchical annotation of references to *The People* and *The Elite*.

particular text has been coded as populist. Furthermore, assigning scores to documents offers limited interpretability for analysis.

The third approach for measuring populism applies **computer-assisted content analysis**, based on dictionaries that contain cue words related to populist rhetoric, such as *people*, *elite*, *establishment*, *corrupt*, etc. (e.g. Jagers and Walgrave (2007); Caiani and della Porta (2011); Vasilopoulou et al. (2014); March (2017); Pauwels (2011); Rooduijn and Pauwels (2011); Bonikowski and Gidron (2016)). While dictionary-based approaches are fast and scale easily, they are less valid and reliable than manual content analysis (Grimmer and Stewart, 2013). This is partly due to the arbitrariness in the selection of the dictionary entries or keywords, where (potentially biased) choices made in the creation of the dictionary can impact the analysis. Another reason for the often low content validity is that dictionary-based methods are not context-sensitive. For instance, Rooduijn and Pauwels (2011) have tried to capture notions of *people-centrism* and *anti-elitism* in text using a dictionary-based approach, and found a reduced content validity compared to manual coding, especially for people-centrism.

The fourth approach uses **supervised machine learning (ML)** for populism detection. First steps in this direction have been taken by Dai (2018); di Cocco and Monechi (2021) and Huguet Cabot et al. (2021). Dai (2018) presents an approach based on document embeddings and SVMs to predict

whether a text is populist or not. The reported performance is quite high (95% acc.), but merely due to the choice of evaluation metric and the highly skewed class distribution (i.e., only 4% of the instances in the dataset are labeled as populist).

In contrast, di Cocco and Monechi (2021) do not rely on manual annotations but approximate populism by party affiliation. They consider all sentences uttered by members of a populist party as populist and show that their measure of populism, based on the predictions of a classifier trained on the weakly supervised data, correlates with party membership and, thus, with the experts’ ratings of populism. However, the approach does not capture the defining features of the construct, and it is unclear what has been learned by the classifier.

Huguet Cabot et al. (2021) present a dataset of Reddit comments annotated for stance (Discriminatory, Critical, Neutral, Supportive) and emotions towards six social groups (Conservatives, Liberals, Immigrants, Refugees, Jews, Muslims). While they also aim at detecting *Us vs. Them* rhetoric, in their work, the groups are given. In contrast, we explicitly model the building blocks of populism, i.e., references to *The People* and *The Elite*, and detect all mentions of either group in text. The advantage of our approach is threefold. First, our representations are contextualized, thus overcoming the shortcomings of dictionary-based approaches. Second, by manually coding all mentions to *The People* and *The Elite* in text, we can overcome the problem of incomplete or biased keyword lists, which is

party	speeches	speakers	tokens
CDU/CSU	76	57	72,113
SPD	58	44	48,988
AfD	39	30	29,301
FDP	34	25	22,736
Left	29	21	20,266
Greens	27	18	18,756
cross-bencher	3	1	1,457
total	267	196	213,617

Table 2: Some statistics for our new data set (CDU/CSU: Christian Democratic Union and Christian Social Union; SPD: Social Democratic Party; AfD: Alternative for Germany; FDP: Free Democratic Party; Left: The Left; Greens: The Greens).

another weakness of dictionary-based approaches (Grimmer and Stewart, 2013). Finally, our approach yields more fine-grained results that allow us to study differences in populist rhetoric, e.g., for actors from different ideological backgrounds.

3 MoPE: Annotating Mentions of the People and the Elite

We now present MoPE, our new data set with annotated mentions of *The People* and *The Elite*.

The People versus The Elite. According to Mudde (2017), the difference between the two camps in populist rhetoric is not based on issues of class or nationality, but rather on *morality*. *The People* are an artificial construct of a (non-existing) homogeneous community whose defining criteria are self-ascribed and depend on the specific ideology that serves as the carrier for the *thin-centered ideology*, i.e., populism (see §2.1). *The Elite*, on the other hand, can be seen as the anti-thesis of the *The People* and also obtains its defining features based on the situational context.

To operationalize the two concepts, we use a hierarchical schema where we encode instances of the two classes on the first level (Table 1). Level 2 then distinguishes individuals and groups of persons from elite organizations, while Level 3 encodes fine-grained information about the individual actors. Our schema builds upon and extends the categories in the codebook of Wirth et al. (2019, p.12)¹. Additionally, Level 3 encodes geo-political entities (GPE) as they provide important information for many applications. Following Jagers and Walgrave (2007) and Wirth et al. (2019), we use the

¹<https://osf.io/2z3dk/>

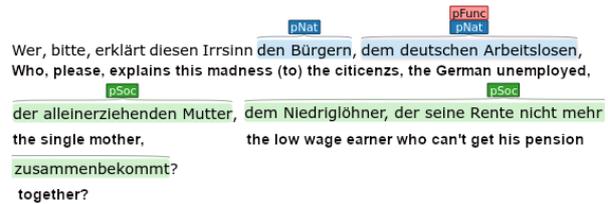


Figure 1: Annotations of references to *The People* (PNAT: people by nationality; PFunc: people by function; PSOC: social variables like gender, class).

term *Elite* in a broad sense as referring to persons, groups, organizations or institutions with a disproportionate amount of power, wealth, privilege or skills through which they can have an impact on politics and society. As instances of *The People*, we consider (a) unspecified groups of people and (b) individuals that denote common members of the public, such as John Q. Public.

German Bundestag data. We extracted a dataset of German parliamentary debates for the 19th legislative term (2017–2021), controlled for topic and party membership of the speakers.² The time frame was selected because of its relevance for the rise and consolidation of populist rhetoric in German politics. Our data set includes 267 speeches by 196 different speakers from 6 German parties (Table 2). Figure 1 shows an example annotation from our data, with references to different mentions of *The People*. Please note that while our task has some similarities to Named Entity Recognition (NER), there are also crucial differences. Most importantly, only some of our mentions are proper names, while many of them are noun phrases that include subordinated clauses like relative clauses (e.g., “the low wage earner who can’t get his pension together” in Figure 1). This means that the average span length of our mentions is considerably longer than for NER, which introduces additional ambiguity for annotation and prediction.³ We will come back to this issue in §3.2. Annotations can (and often do) include embedded mentions. Entities can belong to more than one class (see, e.g., *the German unemployed* in Figure 1, which belongs to the classes “People by Nation” and “People by Function”).

²We follow best practices and provide a datasheet (Bender and Friedman, 2018; Gebru et al., 2021) with details on corpus creation and sampling in the supplementary materials.

³For example, some of the ambiguities arise from PP attachment ambiguities for longer mention spans.

	Label Domain	<i>exact</i> F1	<i>overlap</i> F1	<i>mentions</i> avg. #
Elite (Person)	Politics	0.73	0.84	2,017.5
	Science	0.37	0.37	40.5
	Culture	0.59	0.65	17.0
	Economy	0.11	0.11	9.5
	Finance	0.11	0.11	9.0
	Movements	0	0	7.5
	NGO	0.18	0.18	5.5
	Media	0.22	0.55	4.5
	Military	0	0.25	4.0
	Religion	1.00	1.00	1.0
	avg.	70.6	81.3	2,116.0
Elite (Organisation)	Politics	0.76	0.84	2,443.0
	Finance	0.64	0.79	147.0
	Military	0.72	0.77	132.0
	Economy	0.32	0.56	97.5
	NGO	0.40	0.42	42.5
	Media	0.54	0.77	26.0
	Science	0.46	0.57	17.5
	Movements	0.59	0.59	8.5
	Culture	0	0	2.5
	Religion	0	0	2.0
	avg.	72.8	81.2	2,918.5
People	Function	0.58	0.76	1,572.0
	Age	0.73	0.87	487.5
	Social	0.49	0.61	426.5
	Nation	0.56	0.70	258.5
	Generic	0.42	0.42	187.0
	Ethnicity	0.41	0.51	128.0
		avg.	57.2	71.9

Table 3: Average F1 (micro) for exact match and span overlap for the two coders on the full German data.

English Europarl-UdS data. We additionally compile an English data set to enable testing for the generalization capabilities of our models not only across languages but also beyond recent debates and topics. The English data was extracted from the EuroParl-UdS corpus (Karakanta et al., 2018), a multilingual (En, De, Es) parallel corpus of parliamentary debates from the European parliament, with speeches from 1999–2018. We randomly selected speeches from three different years (1999, 2014, 2015), with 70 different speakers from 18 countries (for details, see Appendix, Tables 12, 10).

Annotation process. The data was double annotated by two student assistants with background in political/social science. During the annotation process, we had weekly meetings to discuss ambiguous cases. The final version was adjudicated by one of the authors (a linguist by training), who also corrected inconsistent span annotations: it includes 9,297 annotated mentions (German subcorpus). In our experiments, we ignore all mentions where the speakers refer to themselves (Label EPOWN) using

	Label Domain	<i>exact</i> F1	<i>overlap</i> F1	<i>mentions</i> avg. #
Elite (Person)	Politics	0.76	0.83	241.0
	Movements	0.29	0.57	3.5
	Science	0	0	1.0
	avg.	0.75	0.82	245.5
Elite (Organisation)	Politics	0.75	0.82	410.0
	Movements	0.15	0.15	6.5
	Economy	0.65	0.69	24.5
	NGO	0.55	0.73	5.5
	Science	0.67	0.67	1.5
	Media	0.86	0.86	3.5
	Finance	0	0	1.0
	Military	0	0	0.5
	avg.	0.73	0.80	453.0
People	Social	0.71	0.87	151.5
	Function	0.28	0.38	29.0
	Nation	0.67	0.78	18.0
	Generic	0	0	5.0
	Age	0.67	0.67	7.5
	Ethnicity	0	0	1.5
		avg.	0.62	0.76

Table 4: Average F1 (micro) for exact match and span overlap for the two coders on the English data.

the pronouns *I/me*, since this label can be assigned based on a simple string match. This results in a set of 7,422 mentions with 22,479 annotated tokens that we divide into training, dev and test set (see Appendix B, Table 11 for more details on the size and distribution of the different splits).

The English data set includes 29,584 tokens with 1,423 annotated mentions (1,074 w/o EPOWN) and 3,567 annotated tokens (3,218 w/o EPOWN).

3.1 Inter-annotator agreement (IAA)

Since our data includes multi-label annotations, we cannot report Cohen’s κ . We follow Hripcsak and Rothschild (2005) and compute F1, treating the annotations of one annotator as the ground truth and the other as the predicted annotations. We then switch roles and report averaged micro F1 on the mention level for the fine-grained labels (level 3).⁴ Table 3 reports micro F1 on the mention level for German, using a strict measure that only considers a mention as correct when all tokens that belong to that mention have been identified correctly. The last column shows the average number of tokens annotated by our two coders (i.e., the number of in-

⁴Also see the discussion in Hripcsak and Rothschild (2005) why chance-corrected measures are not optimal for NER and other sequence-level tasks where the number of negative entities is unknown.

stances *before* adjudication). As the exact mention metric is rather strict and punishes spans that have been identified correctly by both coders but where the span boundaries slightly disagree, we also report a measure based on token overlap that has been introduced for the evaluation of opinion role spans (Katiyar and Cardie, 2016). Here we consider a mention as correct if the annotations overlap and both annotators have assigned the same label. Micro F1 for exact match is 0.69, while the overlap measure is much higher with an F1 of 0.80.

Table 4 shows IAA for the English data from the EU parliament. As for German, references to the people seem to be the most difficult class.

3.2 Error analysis

We notice a high variance in F1 for the different classes. In particular, we can see that F1 for the frequent label types is much higher than IAA for the low-frequency labels. Looking at the data, we see that our domain expert annotators often disagree on the exact span of the mentions. In particular, one annotator often failed to include complement clauses which strongly impacts exact IAA.

The F1 scores for overlapping annotation spans (Table 3) show a substantial increase for many classes, confirming our assumption that the annotators did not so much disagree on the *class labels* but on the *span boundaries* of the mentions. As mentioned above, at times, the domain experts also struggled with PP attachment decisions, as illustrated in Example 3.1 where “at age 63” should not be included in the mention span.

Ex. 3.1 *So why should professional soldiers at age 63 no longer be able to meet the physical demands of service [...] (E. Brecht, SPD, 9/6/2021)*

In addition, the confusion matrix (Appendix B, Table 7) suggests that recall is a problem, showing a considerable number of instances that have been coded by one annotator only. We confirm this problem by looking at individual classes. Especially generic mentions of *The People* have been annotated mostly by one of the two annotators (263 instances have been identified as PGEN by A1 while A2 annotated 111 instances only). This recall problem has been discussed by Beigman Klebanov et al. (2008) for the metaphor detection task where the authors distinguish between *genuine disagreements* and *slips of attention*, which is a common phenomenon, especially for rare classes where the units of analysis are not given, and the annotators

first have to detect them in longer texts before they can assign the labels.

We also notice some systematic disagreements for the classes in our schema. Examples are, for instance, the classes PEOPLE BY NATION and PEOPLE BY ETHNICITY, where A1 shows a bias for the first label while A2 preferred the second. This happened for mentions like *the population of X*, which can be interpreted as ‘citizens of X’ (PNAT) or as referring to all people who live in the country and thus share the same cultural background (PETH). Another systematic disagreement concerns PEOPLE BY FUNCTION and GENERIC mentions, illustrated in Example 3.2. Here, A1 interpreted the mention (“the people who...”) as a generic reference (PGEN) while A2 focused on the function of the people (rebuilding the country) and assigned the label PFUNC.

Ex. 3.2 *I am proud of our country and of [the people who, through the economic miracle, have made it a country that is treated with respect and appreciation _{pFunc/pGen}]. (J. Juratovic, SPD, 28/5/2020)*

In general, we notice that IAA for mentions of *The Elite* is higher than for references to *The People*. We suggest that this is due to two reasons. First, mentions to *The People* are, per definition, more abstract and vague, and second, the average mention length for instances of *The People* is longer than for *The Elite* (elite person: 2.3, elite organization: 2.7, people: 3.1 tokens).

4 Experiments

We use our data set from §3 to benchmark the task of predicting mentions of *The Elite* and *The People* from text sentences. Our task can be decomposed into two separate sub-tasks: (i) mention *detection* (MD) and (ii) mention *classification* (MC). We present experiments where we compare different transformer-based model architectures (Vaswani et al., 2017; Devlin et al., 2019) for those tasks. Specifically, we compare (i) a pipeline approach (MD→MC) with (ii) an end-to-end token classification model (E2E-Tok) and (iii) semi-supervised tri-training (TRI) (Zhou and Li, 2005).

Mention detection. Our MD model is a token classification model, similar to the NER model of Devlin et al. (2019), and predicts the span boundaries for mentions of *The People* and *The Elite* on the token level. We use the BIO schema to encode

Task & model architecture			dev set			test set		
			Prec	Rec	F1	Prec	Rec	F1
	<i>span detect.</i>	MD	82.0 ± 1.00	83.0 ± 0.80	82.4 ± 0.86	79.5 ± 1.21	80.4 ± 1.91	80.0 ± 1.34
Level 1	<i>label predict.</i>	MC	97.6 ± 0.10	97.5 ± 0.10	97.6 ± 0.10	96.8 ± 0.03	96.8 ± 0.03	96.8 ± 0.03
Level 2	<i>upper bound</i>		96.5 ± 0.10	96.4 ± 0.10	96.4 ± 0.10	95.9 ± 0.37	95.9 ± 0.37	95.9 ± 0.37
Level 3	<i>on gold spans</i>		92.5 ± 0.46	92.4 ± 0.47	92.4 ± 0.47	88.1 ± 1.76	88.1 ± 1.76	88.1 ± 1.76
Level 1	Pipeline	MD→MC	74.5 ± 1.0	81.1 ± 1.07	77.7 ± 1.03	72.6 ± 1.13	79.6 ± 1.24	75.9 ± 1.18
	End-to-end	E2E-Tok	82.6 ± 1.09	83.1 ± 1.41	82.8 ± 0.20	77.1 ± 2.84	79.6 ± 1.29	78.3 ± 1.63
Level 2	Pipeline	MD→MC	72.7 ± 0.2	78.9 ± 0.22	75.7 ± 0.21	70.9 ± 0.22	77.6 ± 0.24	74.1 ± 0.23
	End-to-end	E2E-Tok	83.0 ± 0.31	80.7 ± 0.80	81.9 ± 0.55	79.2 ± 0.89	78.3 ± 0.74	78.7 ± 0.39
Level 3	Pipeline	MD→MC	68.7 ± 3.0	72.3 ± 3.16	70.4 ± 3.08	63.8 ± 3.85	67.9 ± 4.10	65.8 ± 3.97
	End-to-end	E2E-Tok	80.6 ± 1.38	79.6 ± 0.88	80.1 ± 0.49	73.6 ± 2.00	74.8 ± 1.21	74.2 ± 0.48

Table 5: F1 (micro), precision and recall for the different models on the German dev and test sets. **Bold** indicates the best performing end-to-end scores for each annotation level and \pm shows stdev over the three runs.

the span boundaries and, for each token, predict whether it belongs to a specific mention.

Mention classification. Our next model architecture tries to predict the label for a given mention using sequence classification. For this, we concatenate the input sentence with the respective mention, separated by a [SEP] token, and input the sequence to the model, which then predicts a label for the entire sequence. Please note that this model relies on gold spans as input and provides an upper bound for determining the correct class of a mention.

Pipeline. When performing mention classification, the span-based MC model needs to know the span boundaries to predict a mention’s label. Therefore, we test a pipeline approach where we first use the MD model to detect the spans of the mentions and then predict the label, using the MD output as input to the MC model.

End-to-end token classification. We compare the pipeline results to an end-to-end token classification model. The architecture is similar to the MD model, but in addition to span boundary detection, we also predict the labels of the mentions on the token level. We use the BIO schema as prefixes to the class labels to encode the span boundaries *and* class for each mention and, for each token, predict whether it belongs to a specific span *and* class (including the None class).

Cross-lingual tri-training with disagreement. Semi-supervised approaches have successfully improved model performance, especially in low-resource scenarios. We, therefore, test the potential of *tri-training* (Zhou and Li, 2005) in a cross-

lingual setting to improve results for knowledge transfer from German to English. Tri-training is an iterative process where we use the predictions of two classifiers c_1, c_2 to assign labels to unlabeled instances and expand the training set of a third classifier. Previous work has shown that *tri-training with disagreement*, i.e., adding only those instances to the training data of c_3 where c_1 and c_2 agree with each other’s predictions but disagree with the prediction of c_3 , can filter out uninformative instances and improve the efficiency of the training process (Chen et al., 2006; Zhou, 2008; Sjøgaard, 2010).

Specifically, we use the end-to-end architecture (E2E-Tok) to train three multilingual classifiers based on bert-base-multilingual-cased with different seeds on the German train set. For each seed, we select the model that performed best on the dev set. We then use the three classifiers to predict labels for new, unlabeled data points from the English part of the EuroParl-UdS corpus and, for each classifier c_i , select new instances based on *disagreement* and add them to c_i ’s training set. Please note that this results in different training sets for each classifier. We then continue fine-tuning the classifiers on the expanded training data for m iterations, followed by n iterations of supervised training on gold data. We repeat this process until the results on the dev set stop improving. Then we use the three semi-supervised classifiers to predict labels for the test set based on majority voting.

In contrast to previous work (Ruder and Plank, 2018), we do not share parameters between learners but encourage the diversity of the models by keeping them separate. For efficiency, we do not fully retrain the models on the expanded data but

Level	Model	German test _{de}			English test _{en}			
		prec	rec	F1	Model	prec	rec	F1
Level 1	mBERT	78.7 ± 1.59	76.3± 0.68	77.5 ± 0.96	ZERO	71.9 ± 2.33	74.7± 1.00	73.3± 0.75
	TRI	77.2± 0.22	77.7 ± 0.28	77.4± 0.25	TRI	70.6± 1.14	79.6 ± 1.06	74.8 ± 1.11
Level 2	mBERT	77.0± 1.07	75.0± 0.15	76.0± 0.60	ZERO	69.6 ± 2.00	74.0± 1.63	71.7± 1.81
	TRI	78.2 ± 0.84	77.2 ± 0.44	77.7 ± 0.19	TRI	70.1 ± 1.62	79.4 ± 1.20	74.4 ± 0.41
Level 3	mBERT	70.9± 0.92	72.6± 0.40	71.7± 0.42	ZERO	68.3± 1.20	74.8± 0.66	71.4± 0.96
	TRI	75.3 ± 0.03	72.7 ± 1.34	74.0 ± 0.70	TRI	69.8 ± 1.50	75.5 ± 0.42	72.5 ± 0.87

Table 6: Results for zero-shot learning and tri-training for the mBERT E2E-Tok model on the German test set and on the English benchmark data.

simply add $m + n$ epochs of fine-tuning in each iteration. For details on model setup and parameter settings, see Appendix B.1, B.4 and B.2.

4.1 Results for German

In all experiments, we report results averaged over three runs with different initializations. All models are implemented with the Huggingface transformers library (Wolf et al., 2020) and PyTorch (Paszke et al., 2017). For evaluation, we use seqeval (Nakayama, 2018), a python implementation of the well-known CoNLL 2000 evaluation script for sequence tagging tasks (Tjong Kim Sang and Buchholz, 2000), and report precision, recall and F1 (micro) in *strict* mode on the mention level for the different levels of our hierarchical annotations (see Appendix B.3 for details).

We first report results for the token-based **mention detection** task (Table 5). F1 on the development and test set are close with around 80%. The upper bound for **mention classification** of gold mention spans is very high for the coarse-grained levels where we distinguish between mentions of *The People* and *The Elite* (Level1/2), with an F1 of around 96%. For the fine-grained classes, the upper bound is around 92% for dev and 88% for test (Table 5, MC, Level3).

We now turn to the end-to-end architectures (MD→MC, E2E-Tok) where we predict the span boundaries *and* the class labels. While the MC model performs well on gold mentions, it visibly struggles to predict labels for automatically determined spans, and F1 decreases by around 20% for all levels (Table 5). On the other hand, our end-to-end token-based model is much better suited for this task, with an F1 over 74% for L3 and around 80% for the coarse-grained prediction of mentions of *The People* and *The Elite*.

4.2 Cross-lingual transfer to English

Zero-shot transfer. Lauscher et al. (2020) have shown that results for *zero-shot cross-lingual transfer* do not decrease much for lower-level tasks like PoS and NER if source and target language are typologically close. This observation encourages us to try zero-shot transfer learning for our task, which is closely related to NER. We use the E2E-Tok architecture from our previous experiments and initialize it with a pretrained multilingual transformer (mBERT). We then train mBERT on the German data and use it to detect instances of *The People* and *The Elite* in the English debates. The experiments are meant to investigate how well we can transfer information from German to English without annotating *any* English data.

Table 6 shows results for the mBERT model on the German test set and zero-shot learning, using the same model to predict labels for the English benchmark data. We can see that F1 for the fine-grained Level-3 predictions on the English test set is only slightly lower than for German (71.7% vs. 71.4% F1). However, the gap between precision and recall is more substantial than in the monolingual setting, and the trend is reversed, showing higher recall with much lower precision. Not surprisingly, results for mBERT on the German test set are lower than the ones for the German BERT model (cf. Table 5).

Looking at the tri-training results, we observe another increase of around 1% for the English data. Interestingly, training the classifier on unlabeled English data also yields an improvement of >2% F1 on the German test set (L3) for mBERT, closing the gap between the mBERT and German BERT results. Overall, the results indicate a successful transfer, considering that the model did not see *any* hand-labeled English data during training.

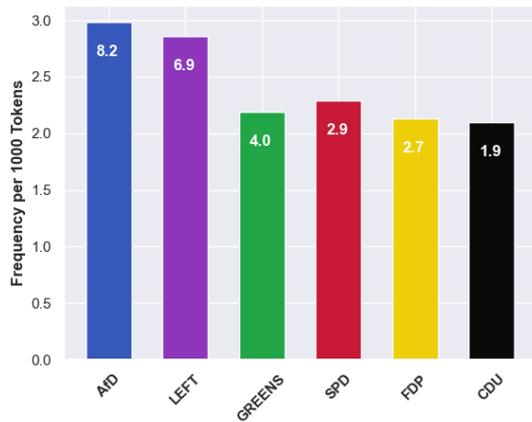


Figure 2: Distribution of references to *The People* in the German Bundestag (2017-2021). Numbers in the bar show POPPA scores for people-centrism.

5 Measuring *thin populism* from text

We are now able to investigate Jagers and Walgrave (2007)’s concept of *thin populism* by looking at how often political actors refer to different subsets of *The People*. For that, we use our three monolingual classifiers described in §4 and predict labels for all debates from the German Bundestag from the 19th legislative term (2017–2021) (> 16 million tokens). We take the majority vote of the three classifiers to determine the final predictions. Figure 2 shows the distribution of the aggregated counts for all references to *The People* for each party. ⁵

We can now validate how well our operationalization of *thin populism* in text correlates with expert ratings. For that, we compute Spearman’s rank correlation between the normalized counts for each party and the party’s score for people-centrism in the Populism and Political Parties Expert Survey (POPPA) (Meijers and Zaslove, 2021b) (also see Table 9 in the Appendix, C). We observe a very strong positive correlation ($\rho = .94$, $p = .005$) between the expert ratings for people-centrism and our predicted counts (Level 1), where both left and right-wing populist parties show a substantially higher amount of people mentions.

However, when looking at the fine-grained predictions for different subgroups of *The People*

⁵We excluded the CSU from the analysis. While the party is forming a joint parliamentary group with the CDU in the Bundestag, it is only running for election in a single German province, Bavaria. This results in a conflict between the party’s “Bavaria first!” policy on the province level and the need to accommodate their sister party’s policies on the federal level (Frymark, 2018, pp.2-3). We, therefore, expect that the governing faction is not representative of the party as a whole.

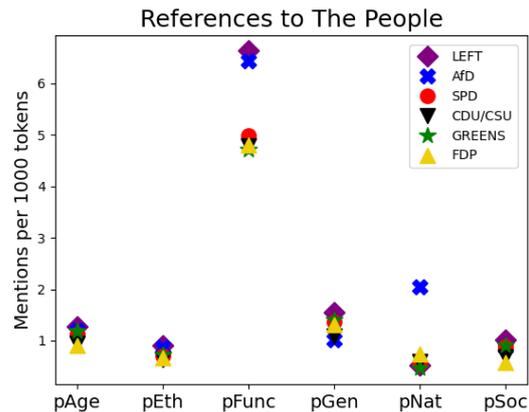


Figure 3: Distribution of group mentions in the 19th legislative term of the German Bundestag (2017-2021).

(Level 3, Figure 3), we also notice interesting differences. For example, both populist parties use a higher amount of references to PEOPLE BY FUNCTION than the mainstream parties. At the same time, only the far-right AfD shows excessive use of PEOPLE BY NATION, often as a dog-whistle to send the message that some people are not “our kind of people”.⁶

Overall, our approach of predicting references to *The People* is able to successfully identify populist rhetoric in large amounts of text and agrees well with expert ratings. However, our results also highlight the importance of a more fine-grained operationalization of *thin populism* that distinguishes between different subgroups of *The People*.

6 Conclusions

In this paper, we presented MOPE, a novel data set for detecting mentions of *The People* and *The Elite* in political text. Our data set includes more than 9,000 annotated mentions for German and an English benchmark set with around 1,600 mentions for cross-lingual transfer learning. We evaluated different transformer-based model architectures on our new data set and explored zero-shot cross-lingual transfer and cross-lingual tri-training.

In future work, we will combine references to *The Elite* with stance detection, which will allow us to model and quantify the different dimensions of populism separately, i.e., *people-centrism* and *anti-elitism*, thus enabling large-scale studies of populism from left- and right-wing political actors in different contextual settings.

⁶This observation is consistent with the AfD’s high POPPA score for nativism (9.7 of 10).

7 Limitations

We would like to point out some limitations of our work. First, in this paper, we do not yet provide measures of populist rhetoric but release a data set and method for detecting instances of *The People* and *The Elite* in text, which we see as a prerequisite for a theoretically grounded, multi-dimensional model of populism that captures the core features of the construct, i.e., *anti-elitism* and *people-centrism*. While our results correlate with expert ratings from survey tools for German, the validity of the English annotations still needs to be tested, and the accuracy for infrequent classes needs to be improved. In addition, further work needs to investigate the robustness of our models on data from different domains and text types.

Acknowledgements

We are grateful to Laura Schmitt and Marlene App for their annotation work. This work is supported by a research grant from the Ministry of Science, Research and the Arts (MWK) Baden-Württemberg.

References

- Koen Abts and Stefan Rummens. 2007. [Populism versus democracy](#). *Political Studies*, 55(2):405–424.
- Paris Aslanidis. 2016. [Is Populism an Ideology? A Refutation and a New Perspective](#). *Political Studies*, 64(1_suppl):88–104.
- Paris Aslanidis. 2018. [Measuring populist discourse with semantic text analysis: an application on grassroots populist mobilization](#). *Quality & Quantity*, 52:1241–1263.
- Beata Beigman Klebanov, Eyal Beigman, and Daniel Diermeier. 2008. [Analyzing disagreements](#). In *Coling 2008: Proceedings of the workshop on Human Judgements in Computational Linguistics*, pages 2–7, Manchester, UK. Coling 2008 Organizing Committee.
- Emily M. Bender and Batya Friedman. 2018. [Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science](#). *Transactions of the Association for Computational Linguistics*, 6:587–604.
- Bart Bonikowski and Noam Gidron. 2016. [The populist style in American politics: presidential Campaign discourse, 1952–1996](#). *Social Forces*, 94(4):1593–1621.
- Bart Bonikowski and Yueran Zhang. 2023. [Populism as Dog-Whistle Politics: Anti-Elite Discourse and Sentiments Toward Minority Groups](#). *Social Forces*. Soac147.
- Manuela Caiani and Donatella della Porta. 2011. [The Elitist Populism of the Extreme Right: A Frame Analysis of Extreme Right Wing Discourses in Italy and Germany](#). *Acta Politica*, 46(2):180–202.
- Wenliang Chen, Yujie Zhang, and Hitoshi Isahara. 2006. [Chinese chunking with tri-training learning](#). In *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages: Beyond the Orient: The Research Challenges Ahead*, ICCPOL’06, page 466–473, Berlin, Heidelberg. Springer-Verlag.
- Yaoyao Dai. 2018. [Measuring Populism in Contexts: A Supervised Approach with Word Embedding Models](#). Working paper.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Jessica di Cocco and Bernarddo Monechi. 2021. [How populist are parties?: Measuring degrees of populism in party manifestos using supervised machine learning](#). *Political analysis*, pages 1–17.
- Nicole Ernst, Frank Esser, Sina Blassnig, and Sven Engesser. 2019. [Favorable opportunity structures for populist communication: Comparing different types of politicians and issues in social media, television and the press](#). *The International Journal of Press/Politics*, 24(2):165–188.
- Kamil Frymark. 2018. [The Free State of Bavaria and its party: the CSU faces an electoral test](#). OSW Commentary NUMBER 288 | 10.10.201.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. [Datasheets for datasets](#). *Commun. ACM*, 64(12):86–92.
- Justin Grimmer and Brandon M. Stewart. 2013. [Text as data: The promise and pitfalls of automatic content analysis methods for political texts](#). *Political Analysis*, 21:267–297.
- Kirk A. Hawkins. 2009. [Is Chávez Populist? Measuring Populist Discourse in Comparative Perspective](#). *Comparative Political Studies*, 42(8):1040–1067.
- Kirk A. Hawkins, R. Aguilar, B. C. Silva, E. K. Jenne, B. Kocijan, and Cristóbal Rovira Kaltwasser. 2019. [Measuring populist discourse: The global populism database](#). In *The EPSA Annual Conference*, pages 20–22.

- Kirk A. Hawkins and B. Castanho Silva. 2018. Text analysis: Big data approaches. In *The ideational approach to populism: Theory, method & analysis*. Routledge.
- Kirk A. Hawkins and Cristóbal Rovira Kaltwasser. 2017. What the (Ideational) Study of Populism Can Teach Us, and What It Can't. *Swiss Political Science Review*, 23(4):526–542.
- G. Hripcsak and A.S. Rothschild. 2005. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 3(12):296–298.
- Pere-Lluís Huguet Cabot, David Abadi, Agneta Fischer, and Ekaterina Shutova. 2021. Us vs. them: A dataset of populist attitudes, news bias and emotions. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1921–1945, Online. Association for Computational Linguistics.
- Jan Jagers and Stefaan Walgrave. 2007. Populism as Political Communication Style: An Empirical Study of Political Parties' Discourse in Belgium. *European Journal of Political Research*, 46(3):319–345.
- Alina Karakanta, Mihaela Vela, and Elke Teich. 2018. EuroParl-UdS: Preserving and extending metadata in parliamentary debates. In *ParlaCLARIN workshop, 11th Language Resources and Evaluation Conference (LREC2018)*, Miyazaki, Japan.
- Arzoo Katiyar and Claire Cardie. 2016. Investigating LSTMs for joint extraction of opinion entities and relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 919–929, Berlin, Germany. Association for Computational Linguistics.
- Ernesto Laclau. 1977. Towards a theory of populism. In E. Laclau, editor, *Politics and Ideology in Marxist Theory*, page 143. London: New Left Books.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.
- Constantine Lignos and Marjan Kamyab. 2020. If you build your own NER scorer, non-replicable results will come. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 94–99, Online. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Luke March. 2017. Left and right populism compared: The british case. *The British Journal of Politics and International Relations*, 19(2):282–303.
- Donald McRae. 1969. Populism as an ideology. In Ghița Ionescu and Ernest Gellner, editors, *Populism: Its Meanings and National Characteristics*, pages 153–65. London: Weidenfeld and Nicolson.
- Maurits Meijers and Andrej Zaslove. 2020. Populism and Political Parties Expert Survey 2018 (POPPA).
- Maurits J. Meijers and Andrej Zaslove. 2021a. Measuring populism in political parties: Appraisal of a new approach. *Comparative Political Studies*, 54(2):372–407.
- Maurits J. Meijers and Andrej Zaslove. 2021b. Measuring populism in political parties: Appraisal of a new approach. *Comparative Political Studies*, 54(2):372–407.
- Benjamin Moffitt. 2016. *The global rise of populism: Performance, political style, and representation*. Stanford University Press, Stanford.
- Cas Mudde. 2004. The Populist Zeitgeist. *Government and Opposition*, 39(4):541–563.
- Cas Mudde. 2007. *Populist Radical Right Parties in Europe*. Cambridge University Press, Cambridge.
- Cas Mudde. 2017. Populism: An Ideational Approach. In C. Rovira Kaltwasser, P. Taggart, and et al. Ochoa Espejo, P., editors, *The Oxford Handbook of Populism*, pages 27–47. Oxford: Oxford University Press.
- Hiroki Nakayama. 2018. sequeval: A python framework for sequence labeling evaluation. Software available from <https://github.com/chakki-works/sequeval>.
- Pierre Ostiguy. 2017. Populism: A Socio-cultural Approach. In C. Rovira Kaltwasser, P. Taggart, and et al. Ochoa Espejo, P., editors, *The Oxford Handbook of Populism*, pages 74–97. Oxford: Oxford University Press.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.
- Teun Pauwels. 2011. Measuring populism: a quantitative text analysis of party literature in Belgium. *Journal of Elections, Public Opinion, and Parties*, 21(1):97–119.
- Jonathan Polk, Jan Rovny, Ryan Bakker, Erica Edwards, Liesbet Hooghe, Seth Jolly, Jelle Koedam, Filip Kostelka, Gary Marks, Gijs Schumacher, Marco Steenbergen, Milada Vachudova, and Marko Zilovic. 2017. Explaining the salience of anti-elitism and reducing political corruption for political parties in europe with the 2014 chapel hill expert survey data. *Research & Politics*, 4(1):2053168016686915.

- M. Rooduijn, S. Van Kessel, C. Froio, A. Pirro, S. De Lange, D. Halikiopoulou, P. Lewis, C. Mudde, and P. Taggart. 2019a. [The populist: An overview of populist, far right, far left and eurosceptic parties in europe.](#)
- Matthijs Rooduijn and Taun Pauwels. 2011. Measuring populism: Comparing two methods of content analysis. *West European Politics*, 34:1272–1283.
- Matthijs Rooduijn, Stijn van Kessel, Caterina Froio, Sarah de Lange, Daphne Halikiopoulou, Paul Lewis, Cas Mudde, and Paul Taggart. 2019b. [The popuList: An overview of populist, far right, far left and Eurosceptic parties in Europe.](#)
- Cristóbal Rovira Kaltwasser, Paul Taggart, Paulina Ochoa Espejo, and Pierre Ostiguy. 2017. *The Oxford Handbook of Populism*. Oxford: Oxford University Press.
- Sebastian Ruder and Barbara Plank. 2018. [Strong baselines for neural semi-supervised learning under domain shift.](#) In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054, Melbourne, Australia. Association for Computational Linguistics.
- Anne Schulz, Philipp Müller, Christian Schemer, Dominique Stefanie Wirz, Martin Wettstein, and Werner Wirth. 2017. [Measuring Populist Attitudes on Three Dimensions.](#) *International Journal of Public Opinion Research*, 30(2):316–326.
- Anders Søgaard. 2010. [Simple semi-supervised training of part-of-speech taggers.](#) In *Proceedings of the ACL 2010 Conference Short Papers*, pages 205–208, Uppsala, Sweden. Association for Computational Linguistics.
- Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. [Introduction to the CoNLL-2000 shared task chunking.](#) In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.
- Sofia Vasilopoulou, Daphne Halikiopoulou, and Theofanis Exadaktylos. 2014. Greece in crisis: Austerity, populism and the politics of blame. *Journal of Common Market Studies*, 52:388–402.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.
- Michael Vaughan and Annett Heft. 2022. [Anti-elitism in the european radical right in comparative perspective.](#) *JCMS: Journal of Common Market Studies*, 61(1):76–94.
- Kurt Weyland. 2001. Clarifying a Contested Concept: Populism in the Study of Latin American Politics. *Comparative Politics*, 34(1):1–22.
- Kurt Weyland. 2021. [Populism as a political strategy: An approach’s enduring – and increasing – advantages.](#) *Political Studies*, 69(2):185–189.
- Werner Wirth, Martin Wettstein, Dominique Wirz, Nicole Ernst, Florin Büchel, Anne Schulz, Frank Esser, and et al. 2019. *Codebook: NCCR democracy Module II: The Appeal of populist Ideas and Messages*. Unpublished paper.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing.](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.
- Zhi-Hua Zhou. 2008. [Semi-supervised learning by disagreement.](#) In *2008 IEEE International Conference on Granular Computing*, pages 93–93.
- Zhi-Hua Zhou and Ming Li. 2005. [Tri-training: exploiting unlabeled data using three classifiers.](#) *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541.

Supplementary Material

A Inter-annotator agreement (IAA)

Table 7 shows the confusion matrix for our two human annotators (A1, A2) for the fine-grained classes (Level 3) in the German Bundestag debates. Due to space limitations, only the most frequent classes are shown. The **prefixes** of the labels are EP: Elite-Person, EO: Elite-Organisation, P: People. The **domains of the labels** are FIN: finance, MIL: military; POL: politics; ECO: economy; AGE: people by age; ETH: people by ethnicity; FUN: people by profession/function; GEN: Generic mentions; NAT: people by nation; SOC: social variables (gender, class); GPE: geo-political entities.

B Training details

B.1 Setup and parameters

For all experiments, we report results averaged over three runs. In each run, we initialise the model with a different seed: {18, 23, 44}. As optimizer, we use AdamW (Loshchilov and Hutter, 2019). The initial learning rate was set to 2.69^{-05} , with a weight decay of 0.0198. We did not freeze any layers but fine-tuned the whole model in all experiments. For tri-training, we experimented with $m = \{3, 5\}$ and $n = \{1, 5\}$ and found that $n=3$ and $m=1$ were robust across different levels. A more principled hyperparameter search might further improve results.

B.2 Training/dev/test splits

Table 11 shows the distribution of labels in the different data splits (train/development/test) for each level in our hierarchical annotation schema. We ensure that none of the agenda items in the test set are included in the training set. This results in a much more challenging and realistic setting compared to distributing speeches from the same agenda item into training and test sets.

B.3 Sequence tagging evaluation

As noted by Lignos and Kamyab (2020), many evaluation scripts for sequence tagging tasks will produce non-replicable results due to inconsistent handling of “improper label sequences”, i.e., mentions that have been labeled with the correct class but have been assigned an incorrect prefix. This results in an inconsistent number of entities in the gold standard and thus produces results that are not

comparable. To avoid this problem, we report results for the *strict* mode where prefixes are included in the evaluation.

For illustration, consider the following two sequences:

- GOLD: ['B-ELI', 'O', 'B-ELI', 'I-ELI', 'O', 'B-ELI']
- PRED: ['B-ELI', 'O', 'O', 'I-ELI', 'O', 'B-ELI']

In *strict* mode, the seqeval evaluation script would consider only proper mentions starting with 'B' for calculation (precision $\frac{2}{2} = 1.00$):

- GOLD: ['**B-ELI**', 'O', 'B-ELI', 'I-ELI', 'O', '**B-ELI**']
- PRED: ['**B-ELI**', 'O', 'O', 'I-ELI', 'O', '**B-ELI**']

However, in *default* mode, the seqeval evaluation first "repairs" the improper label sequences:

- PRED: ['B-ELI', 'O', 'O', '**B-ELI**', 'O', 'B-ELI']

After that, in *default* mode, all three mentions are used for calculation, even if they do not start in the original sequence with a starting token (precision $\frac{2}{3} = 0.67$):

- GOLD: ['**B-ELI**', 'O', '**B-ELI**', 'I-ELI', 'O', '**B-ELI**']
- PRED: ['**B-ELI**', 'O', 'O', '**B-ELI**', 'O', '**B-ELI**']

B.4 Tri-training with disagreement

We use a sample of 20,000 instances (sentences) from the EuroParl-UdS corpus as unlabelled data for tri-training. The data size was determined to extract a sufficient number of data points for tri-training while keeping the additional time for training and prediction low. From the 20,000 instances, between 950 to 1,500 instances have been selected for each classifier during tri-training (see Table 8 for exact numbers).

We loaded the checkpoints for the three best baseline classifiers (E2E) and continued training for 5 epochs on the newly extracted instances. Finally, we trained each classifier for another 5 epochs on the original training set. Then we used the three classifiers to predict labels for the test instances based on a majority vote.

A1 A2	eoFin	eoMil	eoPol	eoEco	epPol	pAge	pEth	pFun	pGen	pNat	pSoc	GPE	None
eoFin	93	0	6	7	0	0	0	1	0	0	0	0	44
eoMil	0	100	0	0	0	0	0	2	0	1	0	0	42
eoPol	5	8	1,641	1	46	0	1	1	0	1	0	17	583
eoEco	1	0	1	33	0	0	0	11	0	0	0	0	59
epPol	1	0	43	0	1,273	0	3	54	1	26	3	2	293
pAge	0	0	0	0	2	330	0	5	1	1	32	0	50
pEth	0	0	0	0	1	3	54	5	7	6	7	0	25
pFun	0	1	0	0	1	10	2	912	40	15	124	5	314
pGen	0	0	1	0	0	0	8	0	78	1	0	0	23
pNat	0	0	0	0	0	0	30	3	12	144	2	0	26
pSoc	0	0	0	0	1	2	2	12	3	1	194	0	35
GPE	0	0	13	0	0	0	2	0	1	0	1	1,008	188
None	16	5	203	18	93	62	33	341	121	43	110	102	198,211

Table 7: Confusion matrix for two human annotators A1, A2 for the fine-grained classes (Level 3) in the German Bundestag debates (most frequent classes only).

	Level1	Level2	Level3
Clf 1	1,142	1192	947
Clf 2	969	946	1024
Clf 3	1,066	1236	1518

Table 8: Unlabelled training instances extracted for each level and classifier during tri-training.

party	people-centrism	populism
AfD	8.2	9.4
LEFT	6.9	5.6
GREENS	4.0	1.4
CSU	3.9	3.2
SPD	2.9	1.5
FDP	2.7	2.5
CDU	1.9	0.8

Table 9: POPPA-2018 expert ratings for people-centrism and populism for the parties in the German Bundestag.

C Populism and Political Parties Expert Survey (POPPA)

Table 9 shows expert ratings from the 2018 Populism and Political Parties Expert Survey (POPPA) (Meijers and Zaslove, 2021b) for all six German parties that participated in government in the 19th legislative term (2017–2021). The first column lists scores for people-centrism, a core feature of populism strongly related to Jagers and Walgrave (2007)’s concept of *thin populism*, and the second column shows the mean populism score for each party, aggregated over all relevant dimensions of populism in the survey. The ratings were collected between April 2018 and July 2018 from 294 country experts and include survey items for populism, political style, party ideology, and party organization in 28 European countries.⁷

D Dataset details

⁷<http://poppa-data.eu/>

Id	Country	# toks
AT	Austria	260
BE	Belgium	2,161
BG	Bulgaria	114
CZ	Czech Republic	31
DE	Germany	358
DK	Denmark	757
EE	Estonia	655
ES	Spain	1,188
FR	France	2,111
GB	United Kingdom	6,918
IE	Ireland	1,063
IT	Italy	2,166
LV	Latvia	256
MT	Malta	214
NA	no information available	7,235
NL	Netherlands	1,492
PL	Poland	474
RO	Romania	895
SE	Sweden	1,525

Table 10: No. of tokens per country for the English data set from the EU parliament (1999-2015). NA indicates that no country information was specified in the meta-data.

		Dataset distribution							
Label		train		dev		test		total	
		#ment.	#token	#ment.	#token	#ment.	#token	#ment.	#token
Level 1									
<i>Elite</i>	ELITE	2603	8028	438	1342	1049	3302	4090	12672
<i>People</i>	PEOPLE	1510	5093	134	501	656	2503	2300	8097
Level 2									
<i>Person</i>	ELITE-PERSON	1033	3607	172	573	402	1408	1607	5588
<i>Organisation</i>	ELITE-ORGAN	1571	4421	267	769	656	2503	2488	7084
<i>People</i>	PEOPLE	1510	5093	134	501	650	1894	2300	8097
Level 3 <i>Elite-Person</i>									
Domain:									
<i>politics</i>	EPOL	969	3293	157	493	370	1316	1496	5102
<i>science</i>	EPSCI	31	150	3	9	32	146	46	204
<i>culture</i>	EP CULT	8	50	2	3	8	17	15	77
<i>military</i>	EP MIL	4	44	6	37	67	149	5	46
<i>finance</i>	EP FIN	2	5	None	None	1	8	7	41
<i>economy</i>	EP ECON	4	14	9	35	12	31	13	37
<i>movement</i>	EP MOV	5	19	None	None	None	None	13	36
<i>NGOs</i>	EP NGO	4	19	3	11	9	24	5	24
<i>media</i>	EP MED	5	11	5	36	6	53	6	19
<i>religion</i>	EP REL	1	2	None	None	None	None	1	2
Level 3 <i>Elite-Organisation</i>									
Domain:									
<i>politics</i>	EO POL	1318	3612	121	183	125	368	2031	5524
<i>finance</i>	EO FIN	76	279	1	3	1	2	117	441
<i>military</i>	EO MIL	70	192	6	30	21	156	148	414
<i>economy</i>	EO ECON	50	148	11	48	68	319	90	346
<i>NGOs</i>	EO NGO	25	82	4	13	74	209	40	124
<i>media</i>	EO MED	15	37	40	160	1	2	33	97
<i>science</i>	EO SCI	9	36	1	5	3	4	17	93
<i>movement</i>	EO MOV	7	33	None	None	None	None	11	40
<i>religion</i>	EO REL	1	2	None	None	None	None	3	5
Level 3 <i>People</i>									
Domain:									
<i>function</i>	P FUN	736	2771	202	491	4	18	1125	4354
<i>age</i>	P AGE	252	720	16	43	9	23	388	1136
<i>social</i>	P SOC	201	652	7	32	164	231	228	845
<i>ethnicity</i>	P ETH	72	266	2	4	11	28	149	620
<i>national</i>	P NAT	113	348	77	292	511	1421	194	611
<i>generic</i>	P GEN	138	336	8	52	65	220	221	531
<i>geo-pol.ent.</i>	G PE	725	1296	16	46	312	1291	1010	1710

Table 11: Label distribution (per annotated token and per mention) for the train/dev/test splits for different levels of annotation.

Id	Name	Party	# toks
1	Mauro NOBILIA	Union for Europe of the Nations Group	562
2	Ole KRARUP	Group for a Europe of Democracies and Diversities	327
3	Carl LANG	Technical Group of Independent Members	360
4	Philip BUSHILL-MATTHEWS	Europ. People's Party (Christian Democrats) and Europ. Democrats	336
5	Alejandro CERCAS	Party of Europ. Socialists	583
6	Daniel DUCARME	Europ. Liberal, Democrat and Reform Party	235
7	Maj Britt THEORIN	Party of Europ. Socialists	412
8	Bartho PRONK	Europ. People's Party (Christian Democrats) and Europ. Democrats	508
9	Anne VAN LANCKER	Party of Europ. Socialists	866
10	Anne E. JENSEN	Europ. Liberal, Democrat and Reform Party	430
11	Hélène FLAUTRE	Greens/Europ. Free Alliance	1,141
12	Herman SCHMID	Confederal Europ. United Left/Nordic Green Left	507
13	Liam HYLAND	Union for Europe of the Nations Group	556
14	Rijk van DAM	Group for a Europe of Democracies and Diversities	375
15	Marco CAPPATO	Technical Group of Independent Members	472
16	Renzo IMBENI	Party of Europ. Socialists	309
17	Maurizio TURCO	Technical Group of Independent Members	74
18	Vytėnė Povilaitis ANDRIUKAITIS	Party of Europ. Socialists	1,362
19	Julie GIRLING	Europ. Conservatives and Reformists Group	519
20	Lynn BOYLAN	Confederal Europ. United Left	268
21	Pavel POC	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	31
22	Anthea McINTYRE	Europ. Conservatives and Reformists Group	185
23	Nessa CHILDERS	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	224
24	Štefan FÜLE	Party of Europ. Socialists	3,017
25	Jacek SARYUSZ-WOLSKI	Europ. People's Party (Christian Democrats)	254
26	Johannes Cornelis van BAALEN	Alliance of Liberals and Democrats for Europe	317
27	Sandra KALNIETE	Europ. People's Party (Christian Democrats)	71
28	Marju LAURISTIN	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	127
29	Victor BOȘTINARU	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	152
30	Paul NUTTALL	Europe of Freedom and Direct Democracy Group	103
31	Mike HOOKEM	Europe of Freedom and Direct Democracy Group	394
32	Ioan Mircea PAȘCU	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	216
33	Richard HOWITT	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	244
34	Georgi PIRINSKI	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	114
35	Andrus ANSIP	Alliance of Liberals and Democrats for Europe	83
36	Tatjana ŽDANOKA	Greens/Europ. Free Alliance	185
37	Jean-Claude JUNCKER	Europ. People's Party (Christian Democrats)	551
38	Syed KAMALL	Europ. Conservatives and Reformists Group	1,011
39	Guy VERHOFSTADT	Alliance of Liberals and Democrats for Europe	1,060
40	Nigel FARAGE	Europe of Freedom and Direct Democracy Group	1,042
41	Gerard BATTEN	Europe of Freedom and Direct Democracy Group	208
42	Theodor Dumitru STOLOJAN	Europ. People's Party (Christian Democrats)	123
43	Věra JOUROVÁ	Alliance of Liberals and Democrats for Europe	1,046
44	Janice ATKINSON	Europe of Freedom and Direct Democracy Group	197
45	Louise BOURS	Europe of Freedom and Direct Democracy Group	249
46	Mairead McGuinness	Europ. People's Party (Christian Democrats)	15
47	Terry REINTKE	Greens/Europ. Free Alliance	358
48	Sophia in 't VELD	Alliance of Liberals and Democrats for Europe	292
49	Mary HONEYBALL	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	239
50	Ulrike LUNACEK	Greens/Europ. Free Alliance	260
51	Jonathan ARNOTT	Europe of Freedom and Direct Democracy Group	104
52	Julie WARD	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	193
53	Clare MOODY	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	223
54	Theresa GRIFFIN	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	375
55	Bill ETHERIDGE	Europe of Freedom and Direct Democracy Group	225
56	Diane DODDS	Non-attached Members	196
57	Doru-Claudian FRUNZULICĂ	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	404
58	Julia PITERA	Europ. People's Party (Christian Democrats)	220
59	Yana TOOM	Alliance of Liberals and Democrats for Europe	158
60	Luigi COCILOVO	Europ. People's Party (Christian Democrats) and Europ. Democrats	515
61	Jan ANDERSSON	Party of Europ. Socialists	606
62	Luciana SBARBATI	Europ. Liberal, Democrat and Reform Party	234
63	Alain LIPIETZ	Greens/Europ. Free Alliance	245
64	Sylviane H. AINARDI	Confederal Europ. United Left/Nordic Green Left	365
65	Margrethe Vestager	Alliance of Liberals and Democrats for Europe	1,259
66	Kaja KALLAS	Alliance of Liberals and Democrats for Europe	287
67	Ramon TREMOSA i BALCELLS	Alliance of Liberals and Democrats for Europe	605
68	Steven WOOLFE	Europe of Freedom and Direct Democracy Group	430
69	Anneliese DODDS	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	445
70	Alfred SANT	Progressive Alliance of Socialists and Democrats in the Europ. Parliament	214
total			29,584

Table 12: Speakers and party affiliation for the English data set from the EU parliament (1999-2015).

SharPT: Shared Latent Space Prompt Tuning

Bo Pang Semih Yavuz Caiming Xiong Yingbo Zhou
Salesforce Research

{b.pang, syavuz, cxiong, yingbo.zhou}@salesforce.com

Abstract

Prompt tuning is an efficient method for adapting large language models, and Soft Prompt Transfer (SPoT) further narrows the gap between prompt tuning and full model tuning by transferring prompts learned from source tasks to target tasks. It is nevertheless difficult and expensive to identify the source task that provides optimal prompts. In this work, we propose to learn a shared latent space which captures a set of basis skills from a mixture of source tasks. Given an instance, its embedding queries the latent space, yielding a basis skill vector. This vector generates soft prompts, via a lightweight prompt generator, which modulates a frozen model. The latent space and prompt transformation are learned end-to-end by training on source tasks. Transfer learning from source tasks to a target task simply amounts to finetuning the prompt generator, accounting for roughly 0.3% parameters of the frozen backbone model, while the shared latent space is also frozen in finetuning. Our approach outperforms prior soft prompt methods by a significant margin on a variety of tasks such as NLI, sentence completion, QA, conference resolution, word sense disambiguation. We also find, on various model scales, our method achieves competitive performance compared to finetuning the full model.

1 Introduction

Adapting pre-trained large language models (LLMs) has advanced the progress in many NLP areas (Devlin et al., 2019; Raffel et al., 2020). This is typically done by finetuning all parameters of a model on a downstream task (i.e., MODEL TUNING). This approach is however expensive, especially given the growing sizes of SOTA LLMs.

This limitation motivates recent research on parameter-efficient methods which only tune a small amount of parameters (Houlsby et al., 2019; Brown et al., 2020; Karimi Mahabadi et al., 2021;

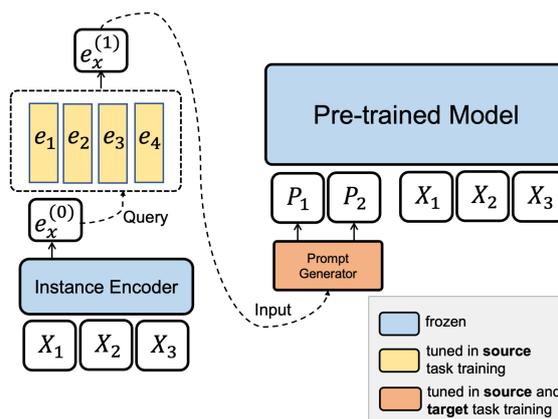


Figure 1: An illustration of SharPT. An instance, as illustrated by with three tokens $\{X_1, X_2, X_3\}$, is encoded by the instance encoder, giving $e_x^{(0)}$, and then queries the skill latent space, resulting in a skill vector $e_x^{(1)}$. The skill vector is transformed by a simple and lightweight prompt generator, outputting prompt tokens (e.g., $\{P_1, P_2\}$). They are prepended to the instance tokens and modulate the pre-trained frozen model. The instance encoder and the pre-trained model are frozen in all scenarios. The skill vectors are tuned in source task training and frozen in target task training. The prompt generator is tuned in both source task and target task training.

Lester et al., 2021; Li and Liang, 2021; Hambarzumyan et al., 2021). Among them, a line of research focus on the methods that modulate a frozen LLM via prompts (Liu et al., 2021). Brown et al. (2020) showed that prepending an input text with a prompt, which typically consists of a task description and/or several examples, can effectively adapt a frozen GPT-3. This approach nevertheless underperforms MODEL TUNING and is sensitive to the choice of prompt wordings. Instead of actual text (or hard prompt), Lester et al. (2021) proposed PROMPT TUNING, which prepends a soft prompt, consisting of k tunable tokens, to input text. The soft prompt can be optimized with gradient-based methods. PROMPT TUNING achieves competitive performance to MODEL TUNING when the model size is large (e.g., over 10B parameters) but still underperforms with smaller models.

SPoT (Vu et al., 2022) improves over PROMPT-

TUNING by leveraging knowledge from source tasks. They first learn a task-specific soft prompt for each task in a set of source tasks. Given a target task, they search over the set of source prompts and use the best one or some weighted combination to initialize the prompt for the target task and then tune the prompt. It further narrows the performance gap to MODEL TUNING on smaller models. But it is complicated and expensive to identify the source task that provides optimal prompts.

In this work, we propose a novel prompt-based transfer learning method, SHARPT (**Shared Latent Space Prompt Tuning**). Figure 1 illustrates the general idea. SHARPT assumes a shared (discrete) latent space by all source and target tasks. We call each vector in the latent space as a *skill vector*, since we assume each one captures a basis NLP capacity or skill after training on the source tasks. Given an instance (from either a source task or a target task), an instance encoder embeds it into an instance vector, which is then used to query the latent space to find the nearest neighbor, yielding a skill vector for this instance. A lightweight prompt generator then generates soft prompts as a function of the selected skill vector. The soft prompts condition a frozen LLM. The latent space and prompt generator are learned end-to-end on a mixture of source tasks. In target task training, the latent space is frozen and only the prompt generator is tuned.

SHARPT retains the key advantage of prior prompt methods, parameter-efficiency. It only updates approximately 0.1% to 0.3% parameters compared to MODEL TUNING. Different from prior methods, we add an instance encoder to encode each instance. The instance encoder is lightweight and frozen in all scenarios.

SHARPT and SPOT both exploit a generic idea, *leveraging knowledge shared across tasks*. The approaches to achieve this are however distinctly different. SPOT assumes *task-to-task transfer* based on *task-level prompts* and the knowledge is encoded in task prompts. It is not straightforward to identify a source prompt for a target task. They illustrated two approaches: (1) SPOT-Oracle and (2) SPOT-Retrieval. SPOT-Oracle involves using oracle test labels and expensive search (e.g., 48 times more expensive than regular prompt tuning in their experiments). In SPOT-Retrieval, they first tuned a task prompt for each source and target task independently and retrieved a prompt based on prompt similarity. Note that the retrieval tun-

ing is only for searching a source prompt, which is in addition to final prompt tuning on the target task. In contrast, SHARPT assumes the knowledge is encoded in a *shared latent space* and utilizes *instance-level prompts*, which are generated based on latent vectors from the shared space. These designs make source-to-target transfer simple. We learn the shared latent space with all source tasks in a single training run. Also, the tuning on the target task only requires a single run. Given an instance from a target task, we use the instance embedding to identify a skill vector, learned from all source tasks, which is then transformed to soft prompts.

In summary, we design an instance-prompt-based method by learning a shared skill latent space. We apply SHARPT to a diverse set of tasks covering diverse domains and task categories. We find that our method outperforms prior prompt-based methods and matches full-model-tuning across model scales.

2 Method

Suppose we have a task with data $T = \{(\mathbf{x}, \mathbf{y})\}$ and a pre-trained LLM P_θ . MODEL TUNING updates θ to minimize $\mathcal{L}(\theta) = -\log P_\theta(\mathbf{y}|\mathbf{x})$ ¹. PROMPT TUNING prepends to \mathbf{x} a soft prompt, $\mathbf{p} \in \mathbb{R}^{L \times d}$, which has L vectors of size d . It then optimizes \mathbf{p} by minimizing $\mathcal{L}(\mathbf{p}) = -\log P_\theta(\mathbf{y}|\mathbf{p}, \mathbf{x})$.

SHARPT assumes there exists a discrete latent space, consisting of a set of skill vectors $\mathbf{E} = \{\mathbf{e}_i \in \mathbb{R}^m\}_{i=1}^K$ with K vectors in total. The soft prompt is a simple transformation of one of the skill vectors \mathbf{e}_i , that is, $\mathbf{p} = f_\alpha(\mathbf{e}_i)$. The transformation or prompt generator (f_α) is a light-weight MLP.

$$\mathbf{e}'_i = \text{Tanh}(W_1 \mathbf{e}_i + b_1), \mathbf{p}_l = W_2(z_l + \mathbf{e}'_i) + b_2 \quad (1)$$

where $z_l \in \mathbb{R}^d$ is the position embedding for the l th token (and randomly initialized in training) and $W_1 \in \mathbb{R}^{d \times m}$, $W_2 \in \mathbb{R}^{d \times d}$. Then we have the soft prompt $\mathbf{p} = \{\mathbf{p}_l\}_{l=1}^L$.

Given \mathbf{x} , we infer its skill vector by (1) embedding it via a frozen instance encoder (e.g., SimCSE BERT-base), which yields $\mathbf{e}_x^{(0)}$; (2) querying \mathbf{E} to find the nearest neighbour. Formally, that is,

$$\mathbf{e}_x^{(1)} = \mathbf{e}_k, \quad k = \arg \min_{i \in [K]} \left\| \mathbf{e}_x^{(0)} - \mathbf{e}_i \right\|_2. \quad (2)$$

For a target task, our method is then trained with the following loss,

$$\mathcal{L}(\alpha) = -\log P_\theta(\mathbf{y}|f_\alpha(\mathbf{e}_k), \mathbf{x}). \quad (3)$$

¹Summation over the data is omitted for notation clarity.

In target task training aforementioned, \mathbf{E} is known and fixed. We next specify how to learn it from source tasks. Suppose we have N source tasks, $\{T_j^{(s)}\}_{j=1}^N$. We simply mix all tasks together, $T^{(s)} = \bigcup_{j=1}^N T_j^{(s)}$. Given $x \in T^{(s)}$ and its embedding $e_x^{(0)}$. \mathbf{E} is learned with the following loss,

$$\mathcal{L}(\mathbf{E}) = \left\| \text{sg}(e_x^{(0)}) - e_k \right\|_2, \quad (4)$$

where $\text{sg}()$ is a stop gradient operator and e_k is defined in Equation (2). The overall loss in source task learning is,

$$\mathcal{L}(\alpha, \mathbf{E}) = \mathcal{L}(\alpha) + \mathcal{L}(\mathbf{E}) \quad (5)$$

In summary, the forward pass for training on source and target tasks are exactly the same (also see Figure 1). The only difference is the loss function, Equation 5 (source) versus Equation 3 (target).

3 Experiments

High-to-Low Resource Transfer In this setting, the target tasks are low-resource tasks (less than 10K training examples), while the source tasks are high-resource tasks. It consists of 25 tasks in total. There are 15 source tasks (e.g., DocNLI, DROP) and 10 target asks (e.g., BoolQ, ColA). Please see Appendix A for the complete list or Table 1 for the target tasks. We keep the setting to be almost the same as a major experiment in Vu et al. (2022) for a fair comparison, with the exception that we exclude C4 from the source task since it is a much larger dataset than other tasks. Excluding C4 does not affect SPOT performance since it does not provide an optimal source prompt for any target task.

Transfer across Different Task Categories We here investigate the transferability from datasets in some task categories to datasets in other held-out task categories. Following Sanh et al. (2022), we assume datasets in each category measures a general NLP ability, and use the same taxonomy defined in Sanh et al. (2022). The source tasks include (1) QA tasks: ReCoRD, SQuAD, DROP, MultiRC, and RACE; (2) sentiment analysis tasks: Yelp-2 and SST-2; (3) a paraphrase detection task: QQP; (4) a semantic similarity task: CXC. The target tasks include (1) a sentence completion task: COPA; (2) NLI tasks: CB and RTE; (3) a coreference resolution tasks: WSC; (4) a word sense disambiguation task: WiC.

Training Details As in prior works (Raffel et al., 2020; Lester et al., 2021), all datasets are converted to a text-to-text format. All experiments are conducted with T5-base-LM-adapted as the backbone unless stated otherwise. We use a SimCSE (Gao et al., 2021) model (BERT-base) as the instance encoder. Since the instance encoder is always frozen, we can pre-compute the embeddings of all instances and only keep the embeddings. However, we find that memory and time saved in this approach is negligible². In source task training, the model (skill latent space and prompt generator) is simply tuned on the mixture of all source tasks for each setting. The model is tuned for 80K steps. In learning and testing on target tasks, we closely follow the procedure in Vu et al. (2022). The model is tuned for 100K on each target task. We save a checkpoint every 500 steps and report results on the checkpoint with the highest validation performance. The prompt generator generates 64 soft tokens. The following hyperparameters are shared in all target and source task training: learning rate (0.3), the number of warmup steps (4000), optimizer (Adam).

4 Results

High-to-Low Resource Transfer The results are shown in Table 1. We first compare our method, SHARPT, to methods with comparable compute- and parameter-efficiency, PROMPTTUNING and SPOT-Retrieval. Our method has a clear improvement over the two methods across most tasks and on the average performance. We next compare SHARPT with much more expensive methods, SPOT-Oracle and MODEL TUNING. Note that SPOT-Oracle is significantly more expensive than our method since it tunes on each target task with each possible task prompt (e.g., it requires roughly 48 times more training time), and utilizes oracle labels. While being much more efficient, SHARPT matches or outperforms SPOT-Oracle. Also, our method performance is on par with the MODEL TUNING performance which requires to tune the entire model. These results indicate SHARPT is an efficient and competitive approach.

Transfer across Different Task Categories The results are shown in Table 2. Our method outperforms both PROMPTTUNING and SPOT methods.

²For instance, removing the instance encoder in training (by pre-computing the instance embeddings) does not allow a larger batch size compared to including the instance encoder.

	BoolQ	CB	CoLA	COPA	CR	MRPC	RTE	STS-B	WiC	WSC	Average
ModelTuning	81.4	94.0	51.1	71.2	<u>94.1</u>	87.5	81.5	89.4	68.3	<u>80.8</u>	<u>79.9</u>
SPoT-Oracle	77.6	97.0	<u>55.6</u>	<u>69.3</u>	93.9	<u>88.7</u>	74.7	90.0	70.2	77.2	79.4
PromptTuning	73.0	92.7	52.9	56.7	93.5	86.1	68.7	88.1	63.6	71.5	74.7
SPoT-Retrieval	74.2	95.4	54.8	58.3	93.6	88.4	71.6	90.0	66.7	72.9	76.6
SHARPT	<u>78.9</u>	<u>94.6</u>	58.2	67.0	94.5	89.7	<u>79.4</u>	89.1	<u>68.8</u>	81.6	80.2

Table 1: Results on the high-to-low transfer learning setting. Methods in the upper panel are significantly more expensive than those in the lower panel. The best performance is in **bold**, and the second best is underlined.

	COPA	CB	RTE	WSC	WiC
ModelTuning	71.2	<u>94.0</u>	81.5	80.8	68.3
SPoT-Oracle	63.0	92.9	72.0	77.2	70.2
PromptTuning	56.7	92.7	68.7	71.5	63.6
SPoT-Retrieval	61.2	89.4	71.4	73.6	66.7
SHARPT	<u>65.0</u>	94.6	<u>79.4</u>	<u>79.0</u>	<u>69.8</u>

Table 2: Results on transferring across task categories.

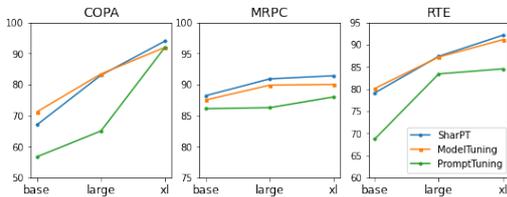


Figure 2: Results on models of different sizes.

The improvement over SPoT methods is larger in this setting than in the high-to-low transfer setting. This might be because SPoT relies more on knowledge shared by tasks in the same category, while SHARPT learns a *shared latent space across all source tasks* and is more suitable to leverage knowledge shared across datasets of different categories.

Across Model Scales In the experiments above, we show that our method can close the performance gap between full model tuning and prompt-based methods on a mid-sized model, T5-base (220M). Here conducts experiments with larger models, T5-large (800M) and T5-xl (3B), and compare SHARPT to MODEL TUNING and PROMPT TUNING. As shown in Figure 2, SHARPT matches or slightly outperforms MODEL TUNING under the three model scales. Our method also shows considerable improvements over PROMPT TUNING.

Ablations We ablate two key components of SHARPT: (1) training on source tasks; (2) skill latent space that captures shared knowledge. See the results in Table 3. Clearly, knowledge learned from source tasks and encoded in the latent space is critical for target task performance.

	BoolQ	CB	CoLA	COPA
SHARPT	78.9	94.6	58.2	67.0
No Source Task Training	64.3	89.3	10.3	58.0
No Latent Space	67.9	82.4	17.6	61.0

Table 3: Ablation results.

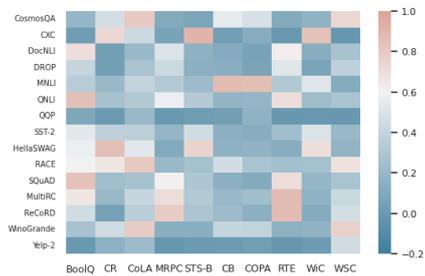


Figure 3: A heatmap of task relations based on skill vector usage of each task.

Task Relations We investigate if the latent space captures source and target task relations to allow knowledge transfer. Each instance queries the latent space and selects one latent skill. We convert this selection to a one-hot vector and treat it as an instance encoding. A task representation is the average of instance encodings in the task. The cosine similarity between two task representations is computed as their relation. The relations between source and target tasks are visualized in Figure 3. It seems that more complicated source tasks such as QA and NLI tasks transfer more knowledge to target tasks via the skill latent space.

5 Conclusion

We introduce SHARPT, which learns a shared latent space which captures a set of basis NLP capacities from a mixture of source tasks. Target instance queries this space to retrieve a skill vector, which then generates prompt tokens to condition a frozen LLM. Our approach outperforms prior soft prompt methods by a significant margin on a variety of tasks. Our method also matches full-model-tuning across model scales.

Limitations

Although our method is much simpler than SPOT, PROMPTTUNING is still arguably the simplest method for adapting LLMs to downstream tasks. It would be a fruitful research direction to design transfer learning approaches that retain (or even improve) our method’s performance and meanwhile further simplify our method, getting closer to the simplicity of PROMPTTUNING.

References

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017)*, pages 1–14.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. [BoolQ: Exploring the surprising difficulty of natural yes/no questions](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. [The pascal recognising textual entailment challenge](#). In *Proceedings of the 1st International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment (MLCW 2005)*, page 177–190.
- Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. [The CommitmentBank: Investigating projection in naturally occurring discourse](#). In *Proceedings of Sinn und Bedeutung 23 (SuB 2018)*, volume 23, pages 107–124.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- William B. Dolan and Chris Brockett. 2005. [Automatically constructing a corpus of sentential paraphrases](#). In *Proceedings of the Third International Workshop on Paraphrasing (IWP 2005)*.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. [DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Karen Hambardzumyan, Hrant Khachatryan, and Jonathan May. 2021. [WARP: Word-level Adversarial ReProgramming](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for nlp](#). In *International Conference on Machine Learning*, pages 2790–2799. PMLR.
- Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. [Cosmos QA: Machine reading comprehension with contextual commonsense reasoning](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.
- Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. [First Quora Dataset Release: Question pairs](#).
- Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. [Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks](#). In *Annual Meeting of the Association for Computational Linguistics*.

- Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. [Looking beyond the surface: A challenge set for reading comprehension over multiple sentences](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. [RACE: Large-scale ReAding comprehension dataset from examinations](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. [The winograd schema challenge](#). In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*, page 552–561.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *arXiv preprint arXiv:2107.13586*.
- Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. 2021. [Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for MS-COCO](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2855–2870, Online. Association for Computational Linguistics.
- Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. [WiC: the word-in-context dataset for evaluating context-sensitive meaning representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019)*, pages 1267–1273.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. [Choice of plausible alternatives: An evaluation of commonsense causal reasoning](#). In *Proceedings of the 25th AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning (AAAI Spring Symposium 2011)*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. [Winogrande: An adversarial winograd schema challenge at scale](#). *Communications of the ACM*, 64(9):99–106.
- Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. [Multi-task prompted training enables zero-shot task generalization](#). In *International Conference on Learning Representations*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.
- Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou, and Daniel Cer. 2022. [SPoT: Better frozen model adaptation through soft prompt transfer](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. [Neural network acceptability judgments](#). *Transactions of the Association for Computational Linguistics (TACL 2019)*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. [DocNLI: A large-scale dataset for document-level natural language inference](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [HellaSwag: Can a machine really finish your sentence?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. [Record: Bridging the gap between human and machine commonsense reading comprehension](#). *arXiv preprint arXiv:1810.12885*.

A Source and Target Tasks in the High-to-Low Resource Transfer Setting

The source tasks include DocNLI (Yin et al., 2021), Yelp-2 (Wang et al., 2018), MNLI (Williams et al., 2018), QQP (Iyer et al., 2017), QNLI (Wang et al., 2018), ReCoRD (Zhang et al., 2018), CXC (Parekh et al., 2021), SQuAD (Rajpurkar et al., 2016), DROP (Dua et al., 2019), SST-2 (Socher et al., 2013), WinoGrande (Sakaguchi et al., 2021), HellaSWAG (Zellers et al., 2019), MultiRC (Khashabi et al., 2018), CosmosQA (Huang et al., 2019), RACE (Lai et al., 2017).

The target tasks include BoolQ (Clark et al., 2019), CB (De Marneffe et al., 2019), CoLA (Warstadt et al., 2019), COPA (Roemmele et al., 2011), CR (De Marneffe et al., 2019), MRPC (Dolan and Brockett, 2005), RTE (Dagan et al., 2005), STS-B (Cer et al., 2017), WiC (Pilehvar and Camacho-Collados, 2019), WSC (Levesque et al., 2012).

Mini But Mighty: Efficient Multilingual Pretraining with Linguistically-Informed Data Selection

Tolulopé Ògúnremí
Stanford University
tolulope@cs.stanford.edu

Dan Jurafsky
Stanford University
jurafsky@stanford.edu

Christopher D. Manning
Stanford University
manning@cs.stanford.edu

Abstract

With the prominence of large pretrained language models, low-resource languages are rarely modelled monolingually and become victims of the “curse of multilinguality” in massively multilingual models. Recently, AfriBERTa showed that training transformer models from scratch on 1GB of data from many unrelated African languages outperforms massively multilingual models on downstream NLP tasks. Here we extend this direction, focusing on the use of related languages. We propose that training on smaller amounts of data but from related languages could match the performance of models trained on large, unrelated data. We test our hypothesis on the Niger-Congo family and its Bantu and Volta-Niger sub-families, pretraining models with data solely from Niger-Congo languages and finetuning on 4 downstream tasks: NER, part-of-speech tagging, sentiment analysis and text classification. We find that models trained on genetically related languages achieve equal performance on downstream tasks in low-resource languages despite using less training data. We recommend selecting training data based on language-relatedness when pretraining language models for low-resource languages.

1 Introduction

Since the introduction of the large pretrained language models (Devlin et al., 2019), low-resource languages have not had the opportunity to be treated in the same way as high-resource languages such as English, French or Mandarin Chinese. Massively multilingual models trained using a mixture of high and low-resource languages such as mBERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020) or mT5, (Xue et al., 2021) have been proposed as a solution. Yet these do not work as well on low-resource languages as they do on high-resource languages due to the “curse of multilinguality” (Conneau et al., 2020), where an

increase of languages in a model leads to capacity dilution, negatively affecting performance for all languages. This makes massively multilingual models sub-optimal solutions for such languages.

The quality of the training data for low-resource languages seems to differ greatly to that of high-resource languages (Kreutzer et al., 2022). The AfriBERTa models (Ogueji et al., 2021) demonstrate the considerable success of pretrained representations when trained with a ‘small’ (1GB), high-quality dataset focused on eleven languages of a single continent – Africa. AfriBERTa Large outperforms the larger, massively multilingual models on named-entity-recognition (NER) and text classification for various African languages. While this *continental* approach is promising, it uses a mixture of different language families that are not genetically related.

Here, we propose using language relatedness in lieu of general geographic proximity of languages to pretrain transformer models. We test this hypothesis by grouping training data by language family and then testing on four tasks: NER, Part-of-Speech Tagging (POS Tagging), Sentiment Analysis and Text Classification. New models trained range from 100 to 600 MB of training data, in contrast to 1GB of data for AfriBERTa and 2395 GB for XLM-R. We find that the smallest models trained on the most closely-related languages perform as well as models trained with up to 10 times the amount of data (AfriBERTa).

In this paper we:

- Train and release¹ pretrained models on genetically grouped African languages
- Finetune and release models for NER, POS tagging, sentiment analysis and text classification on various African languages

¹Models are available to download at <https://github.com/Tolulope/mini-but-mighty>

- Find that training on genetically grouped languages performs equally to larger models, despite training on much less data.

2 Related Work

Despite a long history of work on individual NLP tasks on African languages (Adedjouma Sèmiyou et al., 2012; Dibitso et al., 2019; Schlunz et al., 2016; Pauw et al., 2006; Onyenwe et al., 2014; Hunegnaw et al., 2021; Orimaye et al., 2012; Eisenlen, 2016; Alabi et al., 2020), the lack of freely available and aggregated models made it difficult for languages to build off of each other.

The lack of adequate training data in low-resource languages, including African languages, led to multilingual pretraining transformer models such as mBERT (Devlin et al., 2019), XLM-RoBERTa (Conneau et al., 2020) and mT5 (Xue et al., 2021) using multilingual resources such as Wikipedia and the Common Crawl corpus.

In contrast, the “small data” approach, introduced with the release of the AfriBERTa models (Ogueji et al., 2021) advocates for pretraining models with small amounts of data solely in low-resource languages. The AfriBERTa Large model outperforms XLM-R and mBERT on text classification and NER for a few African languages. This is likely due to the lack of inclusion of a range of African language data and the use of unclean, crawled datasets in the original training data for the large models.

Our proposal to use small, high-quality data draws on the finding that small data perform competitively given the right quality of data (Kreutzer et al., 2022). Our work asks how far we can extend this small data approach by seeing whether large uncurated datasets can be outperformed or at least equalled by small, carefully selected high-quality datasets.

3 Method

3.1 Languages

In our work, we train models with a wide variety of African languages. To compare with AfriBERTa, we use the Afro-Asiatic languages (Amharic, Hausa, Somali, Tigrinya, and Afaan Oromoo) and when focussing on linguistic typology, we work on Niger-Congo Languages. The Niger-Congo family, introduced by Greenberg in 1949, is a genetic family of languages merging the

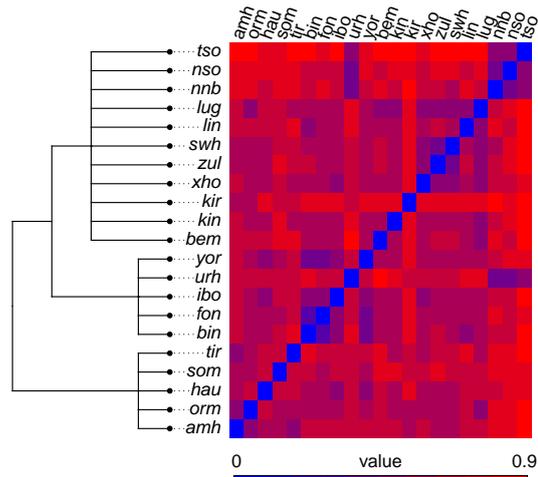


Figure 1: Heatmap displaying the average of syntactic and phonological distances queried from lang2vec between languages used to train the models along with the phylogenetic tree of the languages. Blue represents very close languages and red very distant languages. Clusters are visible for Volta-Niger languages (urh, yor, ibo, fon, and bin) and Bantu languages (nnb, nso, and tso amongst others).

Bantu and ‘Semi-Bantu’ families, due to the similarities found between both (Greenberg, 1949). It spans sub-Saharan African and is a genetic grouping. Figure 1 displays a heatmap of the average of the syntactic and phonological distances between the languages used extracted from WALS using lang2vec (Littell et al., 2017). We see clusters of similarity for the genetically grouped Volta-Niger and Bantu languages, and so our groups, while designed genetically, also are typologically coherent.

The African languages used to train models in this work are summarised with their language families in Table 1.

3.2 Training Data

When training the pretrained models, we add to the AfriBERTa corpus (Ogueji et al., 2021) by collecting various data sources online. See the list of data sources in Appendix A.1. We prioritise datasets produced solely by or in partnership with members of their communities.

3.3 Model Architecture and Training Details

We train all new models with the same architecture as AfriBERTa Large, with 6 attention heads, 768 hidden units, 3072 feedforward size, and a maximum length of 512 (Ogueji et al., 2021). Models trained from scratch are trained for 460,000

Language	ISO Code	Language Family	Branch
Afaan Oromoo	orm	Afro-Asiatic	-
Amharic	amh	Afro-Asiatic	-
Hausa	hau	Afro-Asiatic	-
Somali	som	Afro-Asiatic	-
Tigrinya	tir	Afro-Asiatic	-
Bemba	bem	Niger-Congo	Bantu
Gahuza	kir+kin	Niger-Congo	Bantu
isiXhosa	xho	Niger-Congo	Bantu
isiZulu	zul	Niger-Congo	Bantu
Kiswahili	swa	Niger-Congo	Bantu
Lingala	lin	Niger-Congo	Bantu
Luganda	lug	Niger-Congo	Bantu
Nande	nnb	Niger-Congo	Bantu
Sepedi	nso	Niger-Congo	Bantu
Setswana	ssw	Niger-Congo	Bantu
Xitsonga	tso	Niger-Congo	Bantu
Èdó	bin	Niger-Congo	Volta-Niger
Fon	fon	Niger-Congo	Volta-Niger
Igbo	ibo	Niger-Congo	Volta-Niger
Urhobo	urh	Niger-Congo	Volta-Niger
Yorùbá	yor	Niger-Congo	Volta-Niger
Nigerian Pidgin	pcm	English Creole	

Table 1: Summary of languages used for training language models with their language family, branch and ISO 639-3 code used to refer to languages in Section 4.

steps with a learning rate of $1e-4$. To compare pretrained to continued pretraining, we continue pretraining of the AfriBERTa model by 180,000 steps with all the data from the Niger-Congo family. We also compare newly trained models to monolingual and massively multilingual models trained with much more data: BERT Cased (Devlin et al., 2019), BERT Uncased, RoBERTa (Liu et al., 2019), and XLM-RoBERTa (Conneau et al., 2019).

To initially compare genetics with geography, we train two models with different subsets of the AfriBERTa corpus. *AfriBERTa (Niger-Congo)* is trained with data from the Niger-Congo languages in AfriBERTa (Gahuza, Igbo, Kiswahili and Yorùbá) and *AfriBERTa (Afro-Asiatic)* is trained with the Afro-Asiatic languages in AfriBERTa (Afaan Oromoo, Amharic, Hausa, Somali, and Tigrinya).

The Niger-Congo family has many branches. Due to data availability, we focus on the Volta-Niger and Bantu branches. We supplement the existing data in the AfriBERTa corpus with data in Bemba, Edo, Fon, isiXhosa, isiZulu, Kiswahili (Congolese variant), Lingala, Luganda, Nande, Sepedi, Setswana, Urhobo and Xistonga). Data from these languages totals roughly 364 MB of data. We call the model trained with all of these languages *Niger-Congo BERTa*. We then divide the data by language family and pretrain *BantuBERTa* and

VoltaBERTa.

To test the effects of tokenisation, we train a custom tokenizer with the training data from the Niger-Congo family with the same vocabulary size as AfriBERTa, namely 70,000. The training data for the tokenizer was sampled using the method introduced in XLM (Conneau and Lample, 2019), using an $\alpha = 0.3$.

3.3.1 Size comparison models

To test whether the data selection for the Niger-Congo BERTa models results in the models’ performance downstream, we train AfriBERTa models with the same amount of training data in Niger-Congo BERTa (364 MB), BantuBERTa (260 MB) and VoltaBERTa (107 MB). The resulting models are Afriberta 107, AfriBERTa 260 and AfriBERTa 364, which will be finetuned and directly compared to a model of the same size. To achieve this we proportionally reduce the amount of training data for each language in the AfriBERTa corpus to create three pretraining corpora with 107MB, 260MB and 364MB accordingly each with data from the eleven languages used to train AfriBERTa. The results are averaged across relevant languages for each sized model: Volta-Niger languages for AfriBERTa 107, Bantu languages for AfriBERTa 260 and all Niger-Congo languages for AfriBERTa 364.

The newly trained models along with AfriBERTa are summarised in Table 2.

3.4 Evaluation Data

We evaluate our models on four downstream tasks: named-entity recognition, part-of-speech tagging, sentiment analysis and text classification.

NER: For NER, we use the MasakaNER dataset (Adelani et al., 2021b), covering 10 African Languages covering Afro-Asiatic (Amharic, Hausa, Luo) and Niger-Congo languages. The Niger-Congo branches represented are Bantu (Kinyarwanda, Luganda, Kiswahili), Volta-Niger (Igbo, Yorùbá) and West Atlantic (Wolof).

POS Tagging: For POS Tagging, we use high-quality POS tagging data provided by Masakhane (which is not yet publicly available) covering Bambara, Hausa, Igbo, Kinyarwanda, Nyanja (or Chichewa), Nigerian Pidgin English, Kiswahili, isiXhosa, isiZulu and data from the DHASA-SACAIR 1st Joint Task on Part-of-Speech Tagging for African Languages covering isiNdebele, isiXhosa, isiZulu and Setswana.

Model	Languages	Training Data (MB)	Evaluation Data (MB)	Time to train (hrs)
<i>AfriBERTa (Ogueji et al., 2021)</i>	orm, amh, kin, kir, hau, ibo, pcm, som, swa, tir, yor	939	80	–
<i>AfriBERTa (Niger-Congo)</i>	kin, kir, ibo, swa and yor	279	23	57
<i>AfriBERTa (Afro-Asiatic)</i>	All Afro-Asiatic languages	611	57	60
<i>AfriBERTa Continued</i>	All Niger-Congo languages	364	41	75
<i>Niger-Congo BERTa</i>	All Niger-Congo languages	364	41	75
<i>BantuBERTa</i>	All Bantu languages	260	36	57
<i>VoltaBERTa</i>	All Volta-Niger languages only	107	12	57
<i>AfriBERTa 107</i>	orm, amh, kin, kir, hau, ibo, pcm, som, swa, tir, yor	107	12	57
<i>AfriBERTa 260</i>	orm, amh, kin, kir, hau, ibo, pcm, som, swa, tir, yor	260	36	57
<i>AfriBERTa 364</i>	orm, amh, kin, kir, hau, ibo, pcm, som, swa, tir, yor	364	41	75

Table 2: Summary of models trained and/or used in experiments. Models trained on NVIDIA TITAN RTX GPUs

Sentiment Analysis: For Sentiment Analysis, we use YOSM (Shode et al., 2022) and NaijaSenti (Muhammad et al., 2022). YOSM is a sentiment corpus of film reviews in Yorùbá. NaijaSenti is a Twitter sentiment analysis corpus covering the Nigerian languages Hausa, Igbo, Nigerian Pidgin English and Yorùbá.

Text classification: For text classification, we use a Hausa and Yorùbá news topic classification dataset (Hedderich et al., 2020) and the KINNEWS and KIRNEWS dataset (Niyongabo et al., 2020) covering Kinyarwanda and Kirundi.

4 Results

Results for our experiments are listed in Figure 2 and Tables 3 to 8. Given that datasets have data for languages in different families and branches, we select relevant models for comparison here and leave the full set of the results in the Appendix.

4.1 NER

We finetune the pretrained language models for NER using the Masakhane NER dataset. The results for the AfriBERTa model are taken from the paper (Ogueji et al., 2021). The results for our NER experiments are in Figure 2.

For Niger-Congo languages, shown in Figure 2a, *Niger-Congo BERTa* performs almost as well as AfriBERTa and AfriBERTa with continued pre-training. The difference in results is not statistically significant, but the slight increase may suggest that more training data results in better performance for the NER task.

For Afro-Asiatic languages, shown in Figure 2b, the *AfriBERTa (Afro-Asiatic)* model performs almost as well as AfriBERTa with differences in

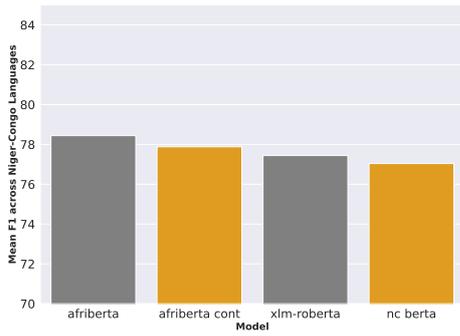
F1 that are not statistically different (less than 0.1 F1). This suggests that training data selection based on genetic grouping results in downstream performance that is not significantly different, despite the reduction in data used. XLM-RoBERTa performs best for Luo and Nigerian Pidgin. Nigerian Pidgin is an English Creole, so we can assume the abundance of English training data in XLM-RoBERTa’s training data helps performance. Luo, a language not present in the training data of any of the models has the best performance with XLM-RoBERTa. This suggests that for unseen languages and English Creoles, it may still be best to finetune massively multilingual models.

4.2 POS Tagging

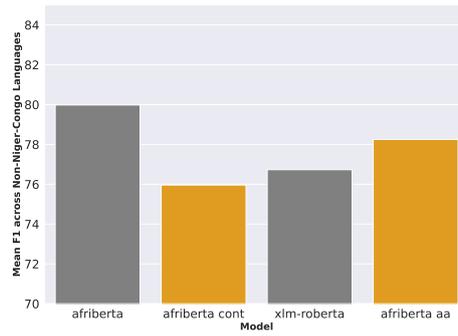
We finetune the models trained on the Part-of-Speech Tagging task, using our two datasets. With languages that have multiple datasets, we train separate models and report the mean per language.

In Table 3 we can see that *BantuBERTa* performs best on most Bantu languages, with an improvement on AfriBERTa of 1.8 F1 for isiZulu, 1.5 F1 for isiXhosa and 1.51 F1 for Chichewa, despite using roughly 25% of the training data of AfriBERTa. Despite the results not being significantly different, we see that training smaller models with higher quality data and a criterion of genetic relatedness leads to performance that is as good as larger models.

For Hausa, an Afro-Asiatic language, we see in Table 4 that *AfriBERTa (Afro-Asiatic)* does not perform significantly differently from AfriBERTa, with only a slight difference in F1 score (0.08 F1 less than AfriBERTa). This suggests that for POS Tagging, linguistically-informed data selection leads to performance that is as good as that



(a) Mean F1 of Niger-Congo languages



(b) Mean F1 of non-Niger-Congo languages

Figure 2: Plots of the mean F1 scores across languages for NER. Plot (a) shows the mean F1 for Niger-Congo languages and Plot (b) shows the mean across non-Niger-Congo languages.

lang	family	afriberta	bantu	nc berta
bam	bantu	86.65	86.70	86.8
ibo	volta-niger	80.47	78.53	80.03
kin	bantu	94.44	94.45	94.38
nbl	bantu	80.91	80.88	81.05
nya	bantu	81.14	82.65	81.67
ssw	bantu	84.50	85.44	85.54
swa	bantu	92.06	91.79	91.51
xho	bantu	84.96	86.46	86.40
zul	bantu	82.35	84.16	83.50
mean		86.04	85.86	85.87

Table 3: F1 scores for POS Tagging models of languages that are in the Niger-Congo family. **BantuBERTa** and **Niger-Congo BERTa** perform as well as AfriBERTa across languages.

lang	afriberta	afriberta cont	afriberta aa	xlm roberta
hau	91.34	89.55	91.26	89.89
pcm	87.57	86.31	86.19	89.76
mean	89.46	87.93	88.73	89.83

Table 4: F1 scores for POS Tagging models of languages that are not in the Niger-Congo family. **AfriBERTa (Afro-Asiatic)** is performing almost as well as AfriBERTa for Hausa, despite being trained with much less data.

of larger models outside the Niger-Congo family. Nigerian Pidgin performs best with XLM-RoBERTa, an expected result given that Nigerian Pidgin is an English Creole.

4.3 Sentiment Analysis

The results for Yorùbá presented are the mean F1 scores from the YOSM and NaijaSenti models.

lang	afriberta	nc berta	afribera nc	volta niger
ibo	86.78	87.53	86.96	88.48
yor	86.09	85.92	85.93	86.42
mean	86.44	86.72	86.44	87.45

Table 5: F1 scores for Sentiment Analysis models of languages that are in the Volta-Niger family. **VoltaBERTa** performs as well as AfriBERTa, despite being trained with 10% of the data.

lang	afriberta	nc berta tok	xlm roberta	afriberta aa
hau	87.42	85.54	85.85	87.43
pcm	72.94	74.83	79.06	70.95
mean	80.18	80.19	82.46	79.19

Table 6: F1 scores for Sentiment Analysis models of languages that are not in the Volta-Niger family. **AfriBERTa (Afro-Asiatic)** is performing almost as well as AfriBERTa for Hausa, despite being trained with much less data.

For Volta-Niger languages, the model trained on only 100MB of data, **VoltaBERTa** has the best performance for both Igbo and Yorùbá, outperforming AfriBERTa by 1.7 and 0.33 F1 despite being trained on 10% of the data. Here we see the advantages of a model being trained on a smaller, yet distinct branch of the Niger-Congo family. The results imply that a smaller linguistically-selected model is as good as a larger non-linguistically-selected model, and has the advantage of being smaller and

therefore more widely usable. It is possible that the high similarity of these languages leads to the model’s increased ability learn about the languages and perform better downstream.

Hausa has the best performance with *AfriBERTa (Afro-Asiatic)* and Nigerian Pidgin English with XLM-Roberta. We also see that the English Creole performs best when finetuned on a model trained on English data, supporting our language-relatedness claim with a different set of languages. Training data from similar languages suffices for competitive performance downstream.

4.4 Text Classification

For text classification, we continue to see the trend that models trained on much less data do not have significantly different performance downstream. *AfriBERTa (Afro-Asiatic)*’s performance is almost as good as AfriBERTa’s for Hausa, *BantuBERTa* with a Niger-Congo tokenizer performs almost as well for Kinyarwanda and outperforms AfriBERTa for Kirundi for Bantu languages and *VoltaBERTa* does not perform significantly differently for Yorùbá. In yet another task, we demonstrate that linguistically-informed data selection trumps data quantity.

4.5 Is quality still relevant if we hold size constant?

In addition to comparing model performance with different amounts of training data, we also directly compare models trained with the same amount of data but with different sets of languages with varying levels of genetic similarity below.

Table 8 summarises the training data experiments with the mean F1 score for each model across languages for each task. AfriBERTa 107 is compared to the *VoltaBERTa* model as they both use 107 MB of training data, AfriBERTa 260 is compared to the *BantuBERTa* as both models use 260 MB of data and AfriBERTa 364 is compared to *Niger-Congo BERTa* as they both use 364 MB of training data. We train the *Niger-Congo BERTa* models with and without a custom tokenizer. The results from models trained with a custom tokenizer have an asterisk. We see that when size is held constant the models trained with high-quality data from closely-related languages perform at least as well as models train with data from a wider range of languages. These results highlight the importance of data selection when resources are limited and support our claim that pretraining with genetically-

related languages doesn’t result in significantly different performance downstream.

Overall, we see that across tasks and languages, models trained with data from genetically related languages alone work as well as models trained with up to 10 times the amount of data.

5 Model Visualisation

5.1 Model Visualisation

To visualise the models, we extract sentence embeddings by concatenating the weights of the last four layers of the model for 1,000 sentences in each language’s evaluation set. We use 1000 sentences for each language to ensure an even distribution across languages. For dimensionality reduction, we use Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) and visualise each sentence in two dimensions. We present UMAP plots as they are not as sensitive to parameters as t-SNE.

Visualisations of models grouped by language family (specific branches when part of the Niger-Congo family) are below. All visualisations show evidence of language-specific and family-specific clustering in the models.

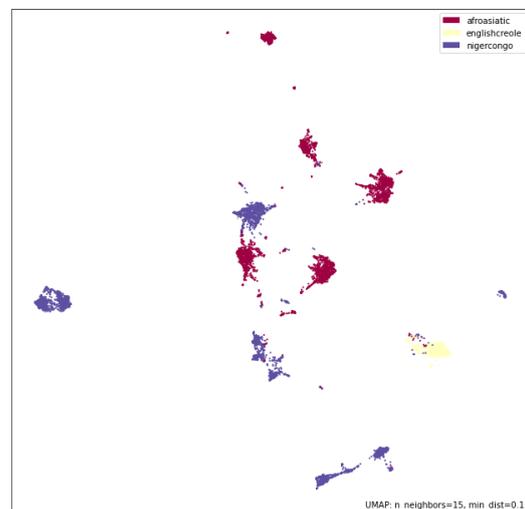


Figure 3: AfriBERTa visualised with languages in the training, coloured by language family (Afro-asiatic in pink, English Creole in yellow and Niger-Congo languages in purple). There appear to be language specific clusters.

When reduced by UMAP, AfriBERTa does not seem to cluster languages by family. Nigerian Pidgin English, is situated away from most of other languages, apart from Yorùbá (bottom right). This

lang	family	afriberta	afriberta cont	afriberta aa	bantu tok	nc berta tok	volta niger
hau	afro-asiatic	90.13	88.18	89.84	84.1	84.22	71.77
kin	bantu	73.87	74.41	70.45	73.69	73.46	68.26
kir	bantu	82.37	84.18	81.38	84.72	83.59	80.91
yor	volta-niger	79.88	80.63	70.88	70.52	78.98	79.70
mean		81.56	81.85	78.14	78.24	80.06	76.77

Table 7: F1 scores for Text Classification models

	afriberta 107	volta niger	afriberta 260	bantu	afriberta 364	nc berta
NER	79.61	82.46	78.10	79.45	76.66	77.04
POS Tagging	79.32	80.40	85.51	86.56	84.78	85.67*
Sentiment Analysis	85.30	87.45				
Text Classification	76.68	78.15	77.50	79.21*	78.11	78.68*
mean per model	80.23	82.12	80.23	81.74	79.85	80.46

Table 8: Table showing the mean F1 across languages in each sub-family compared to an AfriBERTa model trained on the same amount of data for each task. Results with an asterisk (*) are from models trained with a custom tokenizer.

is could be due to borrowing of Yorùbá words into Nigerian Pidgin English.

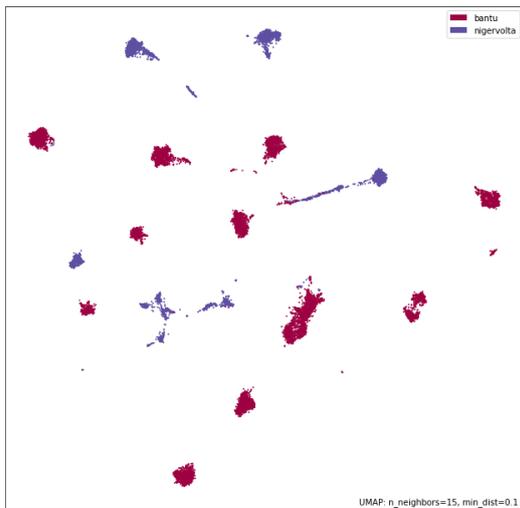


Figure 4: Niger-Congo BERTa visualised with languages in the training, coloured by language family (Bantu languages in pink and Volta-Niger languages in purple). We see language-specific clusters, but no branch-specific separation of the language clusters.

The *Niger-Congo BERTa* model does not seem to cluster languages by sub-family. This may be because all the languages are in the same larger family already.

The *VoltaBERTa* model completely splits Bantu and Volta-Niger Languages, possibly helped by the absence of Bantu languages in the training data. This could be due to the scripts of Igbo and Yorùbá

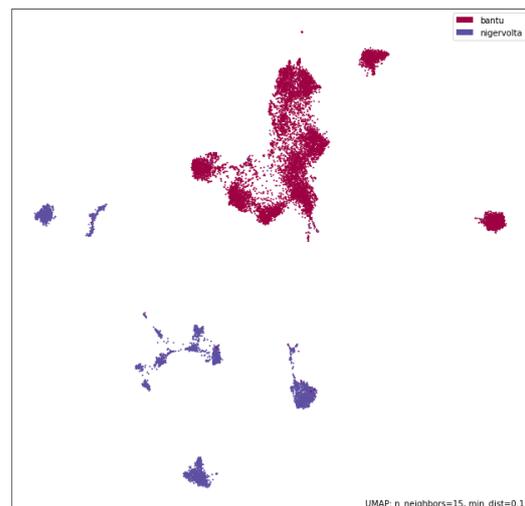


Figure 5: *VoltaBERTa* visualised with languages in the training, coloured by language family (Bantu languages in pink and Volta-Niger languages in purple). Here, Bantu languages are clearly separated from Volta-Niger languages.

(use of diacritics) and Fon (use of different characters), leading the model to internally distinguish between languages part of the Volta-Niger family and those that are not.

Overall, we see that the more closely related the languages used to train the pretrained model, the more distinct the representations of different language families or branches in the UMAP visualisations. This is most likely due to the other languages not being present in the training data, but

the results for POS Tagging and Sentiment Analysis show that this focus on closely related languages leads to improvements in performance with much less data.

6 Discussion

In this work we see that when pretraining multilingual models with closely related languages, the resulting finetuned models work just as well as models finetuned on a wider variety of languages. Sentence embeddings show that the more closely related the languages in the training data, the better the model’s ability to differentiate language families.

We do not see one model consistently outperforming others. However, we do see multilingual models of closely related languages work for those languages downstream and generalise better to unseen languages within the family. **BantuBERTa** works very well for POS Tagging of Bantu languages and **VoltaBERTa** for sentiment analysis of Volta-Niger languages. Continued pretraining of AfriBERTa with closely related languages gives the best text classification result on average. This “small data” combined with language similarity approach demonstrates that it is possible to maintain performance with fewer resources, possibly at the expense of using different models for different downstream tasks.

7 Conclusion

In this paper, we have pretrained several multilingual transformer models exclusively with low-resource languages. We have shown that the grouping of closely-related languages in training data can match or improve performance across several downstream tasks despite the reduction in training data used. We have also demonstrated that for very low-resource languages, we can exploit language similarity to improve performance of NLP tasks on these languages with models trained on similar languages only.

8 Limitations

In this work we did not have an exact overlap of downstream tasks to training data and therefore could not exactly match pretrained models to general task performance. We did not have Bantu language data for Sentiment Analysis, preventing us from making conclusions on this task with BantuBERTa. We also note that we only have data from

two branches of the Niger-Congo family. Data from a wider variety of branches would have helped us make more general conclusions.

We did not compare any of our models to finetuned large language models, nor did we fine-tune our pretrained models before finetuning them for the downstream tasks. It is possible that language-adaptive finetuning of Niger-Congo languages on these models trained exclusively on Niger-Congo languages may lead to even better performance. Given the lack of resources in these languages, one would have to determine guidelines on which data would be used for pretraining or finetuning in this case.

9 Acknowledgements

We would like to thank the anonymous reviewers, Alex Tamkin, Kaitlyn Zhou and Mirac Suzgun for their comments.

This work was supported by Award IIS-2128145 from the NSF and the Stanford School of Engineering Fellowship.

References

- A Adedjouma Sèmiyou, John OR Aoga, and Mamoud A Igue. 2012. Part-of-speech tagging of Yoruba standard, language of Niger-Congo family. *Research Journal of Computer and Information Technology Sciences*, 1:2–5.
- David Adelani, Dana Ruiters, Jesujoba Alabi, Damilola Adebajo, Adesina Ayeni, Mofe Adeyemi, Ayodele Esther Awokoya, and Cristina España-Bonet. 2021a. [The effect of domain and diacritics in Yoruba-English neural machine translation](#). In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 61–75, Virtual. Association for Machine Translation in the Americas.
- David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D’souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiou Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verah Otiende, Iroko Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwunke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Ge-

- breyohannes, Henok Tilaye, Kelechi Nwaike, De-gaga Wolde, Abdoulaye Faye, Blessing Sibanda, Ore-vaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021b. [MasakhaNER: Named entity recognition for African languages](#). *Transactions of the Association for Computational Linguistics*, 9:1116–1131.
- Željko Agić and Ivan Vulić. 2019. [JW300: A wide-coverage parallel corpus for low-resource languages](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.
- Jesujoba Alabi, Kwabena Amponsah-Kaakyire, David Adelani, and Cristina España-Bonet. 2020. [Massive vs. curated embeddings for low-resourced languages: the case of Yorùbá and Twi](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2754–2762, Marseille, France. European Language Resources Association.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Unsupervised cross-lingual representation learning at scale](#).
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Alexis Conneau and Guillaume Lample. 2019. [Cross-lingual language model pretraining](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- MA Dibitso, PA Owolawi, and SO Ojo. 2019. Part of speech tagging for Setswana African language. In *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, pages 1–6. IEEE.
- Roald Eiselen. 2016. [Government domain named entity recognition for South African languages](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 3344–3348, Portorož, Slovenia. European Language Resources Association (ELRA).
- Chris Chinenye Emezue and Femi Pancrace Bonaventure Dossou. 2020. [FFR v1.1: Fon-French neural machine translation](#). In *Proceedings of the The Fourth Widening Natural Language Processing Workshop*, pages 83–87, Seattle, USA. Association for Computational Linguistics.
- Ignatius Ezeani, Paul Rayson, Ikechukwu Onyenwe, Chinedu Uchechukwu, and Mark Hepple. 2020. [Igbo-english machine translation: An evaluation benchmark](#).
- Joseph H. Greenberg. 1949. [Studies in African linguistic classification: I. the Niger-Congo family](#). *Southwestern Journal of Anthropology*, 5(2):79–100.
- Michael A. Hedderich, David Adelani, Dawei Zhu, Jesujoba Alabi, Udia Markus, and Dietrich Klakow. 2020. [Transfer learning and distant supervision for multilingual transformer models: A study on African languages](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2580–2591, Online. Association for Computational Linguistics.
- Ashebir Hunegnaw et al. 2021. Sentiment analysis model for Afaan Oromoo short message service text: A machine learning approach. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(13):332–342.
- Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroko Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhlov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. [Quality at a glance: An audit of web-crawled multilingual datasets](#). *Transactions of the Association for Computational Linguistics*, 10:50–72.
- Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. [URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Rooweither Mabuya, Jade Abbott, and Vukosi Marivate. 2021. [Umsuka english - isizulu parallel corpus](#).
- Vukosi Marivate and Tshephisho Sefara. 2020. [South african news data](#).
- Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.
- Cindy McKellar. 2018. [Autshumato Setswana Monolingual Corpora](#).
- Shamsuddeen Hassan Muhammad, David Ifeoluwa Adelani, Ibrahim Said Ahmad, Idris Abdulmumin, Bello Shehu Bello, Monojit Choudhury, Chris Chinenye Emezue, Anuoluwapo Aremu, Saheed Abdul, and Pavel Brazdil. 2022. Naijasenti: A Nigerian twitter sentiment corpus for multilingual sentiment analysis. *arXiv preprint arXiv:2201.08277*.
- Jonathan Mukiibi, Babirye Claire, and Nakatumba-Nabende Joyce. 2021. [The Makerere MT Corpus: English to Luganda parallel corpus](#).
- Rubungo Andre Niyongabo, Qu Hong, Julia Kreutzer, and Li Huang. 2020. [KINNEWS and KIRNEWS: Benchmarking cross-lingual text classification for Kinyarwanda and Kirundi](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5507–5521, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. [Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages](#). In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ikechukwu Onyenwe, Chinedu Uchechukwu, and Mark Hepple. 2014. [Part-of-speech tagset and corpus development for Igbo, an African language](#). In *Proceedings of LAW VIII - The 8th Linguistic Annotation Workshop*, pages 93–98, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.
- Sylvester Olubolu Orimaye, Saadat M Alhashmi, and Siew Eu-gene. 2012. Sentiment analysis amidst ambiguities in YouTube comments on Yoruba language (Nollywood) movies. In *Proceedings of the 21st International Conference on World Wide Web*, pages 583–584.
- Guy De Pauw, Gilles-Maurice de Schryver, and Peter W Wagacha. 2006. Data-driven part-of-speech tagging of Kiswahili. In *International Conference on Text, Speech and Dialogue*, pages 197–204. Springer.
- Wikus Pienaar, Wildrich Fourie, and Cindy McKellar. 2018. [Autshumato English-Xitsonga Manually Translated Parallel Corpora](#).
- Georg I Schlunz, Nkosikhona Dlamini, and Rynhardt P Kruger. 2016. Part-of-speech tagging and chunking in text-to-speech synthesis for South African languages. In *Interspeech 2016*. Curran Associates, Inc.
- Shivachi Casper Shikali and Mokhosi Refuoe. 2019. [Language modeling data for swahili](#).
- Iyanuoluwa Shode, David Ifeoluwa Adelani, and Anna Feldman. 2022. [YOSM: A new Yorùbá sentiment corpus for movie reviews](#). In *3rd Workshop on African Natural Language Processing*.
- Claytone Sikasote and Antonios Anastasopoulos. 2022. [Bembaspeech: A speech recognition corpus for the bemba language](#). In *Proceedings of the Language Resources and Evaluation Conference*, pages 7277–7283, Marseille, France. European Language Resources Association.
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Alp Öktem, Muhannad Albayk Jaam, Eric DeLuca, and Grace Tang. 2020. [Gamayun - language technology for humanitarian response](#). In *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, pages 1–4.

A Appendix

A.1 List of data sources

A.6 Training data comparisons

Language	Data Sources
Afaan Oromoo	AfriBERTa Corpus
Amharic	AfriBERTa Corpus
Hausa	AfriBERTa Corpus
Somali	AfriBERTa Corpus
Tigrinya	AfriBERTa Corpus
Bemba	Text from Bemba Speech Corpus (Sikasote and Anastopoulos, 2022)
Gahuza	AfriBERTa Corpus
isiXhosa	Xhosa Navy Parallel Corpus (Tiedemann, 2012)
isiZulu	Umsuka English - isiZulu Parallel Corpus (Mabuya et al., 2021)
Kiswahili	AfriBERTa Corpus, Language modeling data for Swahili (Shikali and Refuoe, 2019) and Gamayun (Öktem et al., 2020) Congolese Kiswahili Medium kit
Lingala	Gamayun (Öktem et al., 2020) Lingala Kit
Luganda	Makerere MT Corpus (Mukiibi et al., 2021)
Nande	Gamayun (Öktem et al., 2020) Nande kit
Sepedi	South African News Data (Marivate and Sefara, 2020)
Setswana	Autshumato Setswana Monolingual Corpora (McKellar, 2018) and South African News Data (Marivate and Sefara, 2020)
Xitsonga	Autshumato English-Xitsonga Manually Translated Parallel Corpora (Pienaar et al., 2018)
Èdó	JW300 (Agić and Vulić, 2019)
Fon	FFR Translate Corpus (Emezue and Dossou, 2020)
Igbo	AfriBERTa Corpus and Igbo Monolingual Dataset (Ezeani et al., 2020)
Urhobo	JW300 (Agić and Vulić, 2019)
Yorùbá	AfriBERTa Corpus and MENYO-20k dataset (Adelani et al., 2021a)
Nigerian Pidgin	AfriBERTa Corpus

Table 9: List of sources for language data used to train the models in Table 2.

A.2 Full NER Results

lang	afri berta nc	xlm roberta	bantu tok	volta niger tok	afri berta aa	bert cased	nc berta	nc berta tok	volta niger	bantu	afri berta cont
amh	37.9 ± 7.11	55.85 ± 2.45	0.0 ± 0.0	0.0 ± 0.0	72.73 ± 5.64	0.0 ± 0.0	40.09 ± 5.68	0.0 ± 0.0	7.95 ± 69.67	39.68 ± 3.1	63.11 ± 9.69
hau	85.0 ± 3.01	89.35 ± 3.0	84.26 ± 1.28	82.34 ± 3.9	90.11 ± 2.49	85.89 ± 3.09	84.43 ± 2.89	84.83 ± 1.25	82.31 ± 3.94	83.72 ± 2.26	87.64 ± 1.73
ibo	87.16 ± 1.86	83.96 ± 2.16	75.99 ± 2.11	86.65 ± 2.01	83.19 ± 1.44	83.13 ± 2.45	86.97 ± 3.88	86.03 ± 3.25	86.59 ± 2.4	77.5 ± 3.99	87.44 ± 2.61
kin	71.78 ± 4.26	72.36 ± 3.56	72.27 ± 3.99	62.71 ± 3.9	65.34 ± 3.11	71.35 ± 3.28	71.77 ± 2.8	71.17 ± 4.67	63.01 ± 2.65	72.43 ± 5.88	72.47 ± 5.77
lug	78.42 ± 2.7	80.0 ± 4.58	77.85 ± 1.41	70.48 ± 3.49	75.17 ± 2.75	77.82 ± 4.46	79.3 ± 3.2	78.21 ± 3.32	70.46 ± 3.39	78.28 ± 6.65	78.97 ± 4.79
luo	68.96 ± 3.09	74.73 ± 5.19	70.1 ± 5.87	58.63 ± 5.57	68.62 ± 5.93	73.05 ± 5.66	69.86 ± 2.2	70.06 ± 8.19	59.29 ± 9.08	67.93 ± 4.38	69.71 ± 4.81
pcm	81.18 ± 1.81	86.97 ± 3.12	76.05 ± 3.23	75.91 ± 5.04	81.54 ± 1.77	86.8 ± 5.5	80.92 ± 4.05	79.09 ± 5.46	76.17 ± 5.44	76.26 ± 4.57	83.38 ± 5.99
swa	87.3 ± 2.1	87.16 ± 2.0	87.62 ± 2.4	77.28 ± 4.18	81.49 ± 3.19	83.73 ± 2.53	86.83 ± 2.81	85.94 ± 2.27	77.33 ± 4.12	87.64 ± 1.51	87.87 ± 1.51
wol	58.37 ± 5.33	64.87 ± 3.95	59.15 ± 4.5	51.81 ± 9.09	59.16 ± 10.79	62.77 ± 8.6	59.54 ± 10.09	57.65 ± 10.21	52.66 ± 10.73	59.84 ± 7.7	61.43 ± 3.2
yor	79.04 ± 5.42	76.28 ± 6.12	68.75 ± 4.91	77.81 ± 3.17	69.79 ± 6.45	73.2 ± 4.29	77.85 ± 3.98	78.21 ± 6.41	78.32 ± 3.97	68.57 ± 7.66	79.07 ± 4.3

Table 10: Full set of NER Tagging Results. Models are finetuned five times with the mean and 95% confidence interval displayed.

A.3 Full Text Classification Results

	nc berta	volta niger	afriberta	bantu tok	nc berta tok	afriberta aa	afriberta cont	bantu
hau	81.08 ± 2.24	73.85 ± 7.79	90.13 ± 2.75	84.10 ± 2.0	84.22 ± 4.85	89.84 ± 1.21	88.18 ± 2.59	78.26 ± 2.31
kin	73.2 ± 1.29	67.56 ± 3.26	73.87 ± 2.42	73.69 ± 2.26	73.46 ± 2.5	70.45 ± 1.94	74.41 ± 1.77	74.07 ± 2.1
kir	81.28 ± 5.04	80.52 ± 1.36	82.37 ± 9.38	84.72 ± 3.26	83.59 ± 6.31	81.38 ± 2.34	84.18 ± 2.33	82.47 ± 6.16
yor	79.69 ± 6.17	79.70 ± 3.53	79.88 ± 5.41	70.52 ± 4.56	78.98 ± 4.43	70.88 ± 8.5	80.63 ± 3.08	69.15 ± 4.23

Table 11: Full set of the Text Classification results. Models are finetuned five times with the mean and 95% confidence interval displayed.

A.4 Full POS Tagging Results

	afri berta nc	xlm roberta	afri berta	bantu tok	volta niger tok	afri berta aa	bert cased	nc berta	nc berta tok	volta niger	bantu	afri berta cont
bam	86.66± 2.03	88.23± 0.4	86.97± 1.63	87.1± 2.72	86.71± 0.85	86.6± 0.95	87.79± 1.63	86.8± 0.67	87.01± 1.45	86.97± 1.45	86.7± 1.8	86.62± 1.49
hau	87.77± 1.28	90.44± 1.61	91.13± 1.0	88.39± 2.07	87.54± 1.02	91.26± 1.37	89.12± 2.6	87.78± 2.08	88.71± 1.38	87.23± 0.9	87.89± 0.92	89.74± 1.99
ibo	79.7± 1.71	79.99± 2.3	80.26± 3.33	77.68± 2.02	79.75± 5.21	77.73± 3.07	79.19± 1.64	80.03± 2.47	80.29± 2.55	79.88± 1.87	78.53± 2.67	80.80± 2.36
kin	93.91± 1.17	93.15± 1.07	94.28± 0.7	94.41± 0.51	83.8± 2.26	89.3± 2.76	93.36± 1.29	94.38± 1.18	93.97± 0.2	85.73± 2.55	94.45± 0.68	93.96± 1.04
nbl	80.38± 1.52	81.83± 0.48	80.74± 0.48	80.67± 0.58	79.34± 0.99	79.97± 1.94	81.65± 1.38	81.05± 0.98	80.74± 0.57	79.92± 1.42	80.88± 0.36	80.53± 0.96
nya	80.52± 1.65	82.03± 1.83	80.98± 1.15	81.33± 1.41	77.51± 2.27	79.71± 2.2	80.91± 3.65	81.67± 2.68	82.04± 2.78	78.39± 2.86	82.65± 2.92	80.96± 0.99
pcm	85.86± 0.99	89.76± 1.5	87.64± 1.55	85.43± 1.55	84.44± 1.45	86.19± 1.06	89.68± 1.53	85.87± 1.11	85.95± 1.47	84.55± 1.35	85.56± 1.47	86.42± 0.58
ssw	84.14± 2.4	84.89± 1.21	85.0± 1.54	85.09± 1.73	82.64± 1.06	84.25± 0.56	84.29± 1.82	85.54± 0.75	85.14± 1.71	83.25± 2.29	85.44± 0.74	85.54± 1.64
swa	92.03± 1.36	91.73± 0.99	91.59± 1.12	91.74± 1.01	84.25± 1.07	87.08± 1.55	89.77± 1.91	91.51± 0.95	91.71± 1.34	84.71± 1.31	91.79± 0.86	91.77± 0.81
xho	92.7± 1.49	94.52± 1.07	93.52± 1.75	94.84± 0.51	90.38± 1.24	92.85± 0.96	93.39± 0.44	94.84± 0.5	94.57± 1.01	91.79± 0.58	95.05± 0.39	94.5± 0.64
xhol	75.77± 2.76	77.5± 2.28	76.62± 1.89	77.82± 1.1	67.03± 3.65	74.98± 2.71	74.83± 2.91	77.97± 1.49	78.08± 3.4	72.92± 1.95	77.76± 2.15	78.03± 3.34
zul	84.7± 1.12	85.46± 0.18	85.21± 0.59	86.28± 0.87	83.49± 1.34	84.68± 1.51	84.65± 1.35	85.5± 1.37	85.8± 0.81	84.29± 1.9	85.95± 1.42	85.26± 0.79
zull	79.73± 1.75	82.2± 2.86	79.35± 2.05	81.78± 1.35	73.1± 3.79	77.57± 3.32	80.96± 2.34	81.49± 2.62	81.79± 1.54	76.61± 2.91	82.36± 1.52	82.25± 1.91

Table 12: Full set of POS Tagging Results. Models are finetuned five times with the mean and 95% confidence interval displayed.

A.5 Full Sentiment Analysis Results

lang	afri berta nc	afri berta	bert un-cased	bantu tok	xlm roberta	volta niger tok	bert cased	mbert	nc berta	nc berta tok	volta niger	bantu
hau	84.15± 1.75	87.42± 1.13	81.74± 2.41	85.98± 2.44	85.85± 1.64	85.38± 1.4	84.02± 3.24	83.25± 3.63	84.35± 3.55	85.54± 1.49	82.47± 2.96	83.1± 2.31
ibo	86.96± 3.23	86.78± 1.46	80.44± 4.32	84.07± 2.22	84.62± 13.16	87.11± 1.54	85.03± 4.74	84.99± 3.92	87.53± 2.31	86.58± 3.08	88.48± 2.38	83.81± 2.85
pcm	70.73± 3.71	72.94± 5.13	75.02± 9.45	77.47± 8.6	79.06± 1.38	72.21± 1.3	73.73± 13.96	71.95± 15.06	66.59± 5.06	74.83± 4.62	63.55± 10.43	71.0± 11.9
yor	84.49± 2.02	85.18± 2.55	79.01± 3.71	82.03± 2.72	55.29± 0.0	86.38± 2.14	82.98± 2.57	80.96± 19.38	85.64± 0.82	85.11± 1.44	85.77± 1.18	82.48± 0.9
yosm	87.36± 5.48	87.0± 4.57	72.83± 4.25	80.29± 4.52	82.83± 2.35	85.79± 5.38	82.43± 2.23	83.59± 5.45	86.19± 5.11	85.99± 4.72	87.07± 4.22	76.98± 2.54

Table 13: Full set of Sentiment Analysis Results. Yorùbá data from NaijaSenti (*yor*) and Yorùbá data from YOSM (*yosm*) were finetuned separately. Models are finetuned five times with the mean and 95% confidence interval displayed.

	afriberta 107	volta niger	volta niger tok
ibo	84.45	86.59	86.65
yor	74.76	78.32	77.81
mean	79.61	82.46	82.23

Table 14: F1 scores for models of the same size finetuned for NER on Volta-Niger languages. *VoltaBERTa* performs best overall

	afriberta 260	bantu	bantu tok
kin	70.29	72.43	72.27
lug	77.27	78.28	77.85
swa	86.75	87.64	87.62
mean	78.10	79.45	79.25

Table 15: F1 scores for models of the same size finetuned for NER on Bantu languages. The *BantuBERTa* model outperforms the AfriBERTa model of the same size on Bantu languages by 1.35 F1 on average for NER.

	afriberta 364	nc berta	nc berta tok
ibo	86.45	86.97	86.03
kin	72.43	71.77	71.17
lug	76.09	79.3	78.21
swa	87.37	86.83	85.94
wol	60.38	59.54	57.65
yor	77.22	77.85	78.21
mean	76.66	77.04	76.20

Table 16: F1 scores for models of the same size finetuned for NER on Niger-Congo languages. *Niger-Congo BERTa* performs best on average.

	afriberta 107	volta niger
yor	76.68	78.15

Table 17: F1 scores for models of the same size finetuned for Text Classification on Yorùbá, a Volta-Niger languages with *VoltaBERTa* outperforming the AfriBERTa model trained on the same amount of data.

	afriberta 260	bantu	bantu tok
kin	73.17	74.07	73.69
kir	81.83	82.47	84.72
mean	77.5	78.27	79.21

Table 18: F1 scores for models of the same size finetuned for Text Classification on Bantu languages. The *BantuBERTa* model, both with and without a custom tokenizer outperforms the AfriBERTa model of the same size on Bantu languages.

	afriberta 364	nc berta	nc berta tok
kin	72.96	73.20	73.46
kir	82.29	81.28	83.59
yor	79.08	79.69	78.98
mean	78.11	78.06	78.68

Table 19: F1 scores for models of the same size finetuned for Text Classification on Niger-Congo languages. *Niger-Congo BERTa* with and without a custom tokenizer perform better than the AfriBERTa model of the same size.

	afriberta 107	volta niger	volta niger tok
ibo	79.32	80.40	79.75

Table 20: F1 scores for models of the same size finetuned for POS Tagging on Volta-Niger languages with *VoltaBERTa* outperforming the AfriBERTa model trained on the same amount of data.

	afriberta 260	bantu	bantu tok
bam	87.26	86.7	87.1
kin	93.64	94.45	94.41
nbl	80.25	80.88	80.67
nya	80.83	82.65	81.33
ssw	84.45	85.44	85.09
swa	91.59	91.79	91.74
xho	84.42	86.41	86.33
zul	81.64	84.16	84.03
mean	85.51	86.56	86.34

Table 21: F1 scores for models of the same size finetuned for POS Tagging on Bantu languages with *BantuBERTa* almost always outperforms the AfriBERTa model trained on the same amount of data.

	afriberta 364	nc berta	nc berta tok
bam	86.84	86.8	87.01
ibo	80.28	80.03	80.29
kin	93.8	94.38	93.97
nbl	79.89	81.05	80.74
nya	81.00	81.67	82.04
ssw	83.74	85.54	85.14
swa	91.74	91.51	91.71
xho	84.03	86.41	86.33
zul	81.74	83.50	83.80
mean	84.78	85.65	85.67

Table 22: F1 scores for models of the same size finetuned for POS Tagging on Niger-Congo languages. *Niger-Congo BERTa* with and without a custom tokenizer perform better than the AfriBERTa model of the same size.

	afriberta 107	volta niger	volta niger tok
ibo	85.85	88.48	87.11
yor	84.74	86.42	86.09
mean	85.30	87.45	86.60

Table 23: F1 scores for models of the same size fine-tuned for Sentiment Analysis on Volta-Niger languages with *VoltaBERTa* consistently outperforming the AfriBERTa model trained on the same amount of data.

Long Document Summarization with Top-down and Bottom-up Inference

Bo Pang Erik Nijkamp Wojciech Kryscinski
Silvio Savarese Yingbo Zhou Caiming Xiong

Salesforce Research

{b.pang, erik.nijkamp, wojciech.kryscinski}@salesforce.com

{ssavarese, yingbo.zhou, cxiong}@salesforce.com

Abstract

Text summarization aims to condense long documents and retain key information. Critical to the success of a summarization model is the faithful inference of latent representations of words or tokens in the source documents. Most recent models infer the latent representations with a transformer encoder, which is purely bottom-up and thus does not capture long-distance context well. Also, self-attention-based models face the challenge of quadratic complexity with respect to sequence length. We propose a method to improve summarization models on these two aspects. Our method assumes a hierarchical latent structure of a document where the top-level captures the long range dependency at a coarser time scale and the bottom token level preserves the details. *Critically, our method enables token representations to be updated in both a bottom-up and top-down manner.* In the bottom-up pass, token representations are inferred with local self-attention to leverage its efficiency. Top-down correction is then applied to allow tokens to capture global context. We demonstrate the effectiveness on a diverse set of summarization datasets, including narrative, conversational, scientific documents and news. Our model achieves state-of-the-art performance on a wide range of long document summarization benchmarks, compared to recent efficient transformers. We show that our model can summarize an entire book and achieve competitive performance using 0.27% parameters and much less training data, compared to a recent GPT-3-based model. These results indicate the general applicability and benefits of the framework.

1 Introduction

An abstractive summarization system aims to generate a semantically coherent and linguistically fluent summary by conditioning on the document. The dominant approach for abstractive summarization is to use a Seq2Seq model (Sutskever et al., 2014) with an encoder-decoder architecture instantiated

with either RNNs (Hochreiter and Schmidhuber, 1997) or transformers (Vaswani et al., 2017). In such a model, an encoder computes or infers ¹ latent representations of observed tokens (words or subwords) in a document, conditioning on which a decoder generates a summary. This paper studies the problem of how to compute informative latent representations, which in turn would improve summarization.

We propose a method which *synergizes bottom-up computation with top-down computation* while assuming a multi-scale latent structure of a document. In a multi-scale structure, higher-level variables (like those representing sentences, segments) model the document at a coarser time-scale and abstract away details, and are suitable for capturing long range dependency of the document; in contrast, lower-level variables (like those representing tokens) preserve details, and prevent the summary from losing key details (such as the name of an entity). In our method, the summary is generated by conditioning on token representations (low-level variables), similar to recent abstractive summarization models (Zaheer et al., 2020; Beltagy et al., 2020). There is however a critical difference. In our method, *token representations are first bottom-up inferred and then top-down updated with high level representations, hence rendering low-level representations aware of global context.* See Figure 1 for an overview of our method.

Multi-level models have been widely studied in modeling for images (Sønderby et al., 2016), speech (Mehri et al., 2016), and language (Chung et al., 2016). It is also not new in the summarization literature. Prior summarization research has explored hierarchical models (Cheng and Lapata, 2016; Nallapati et al., 2016; Zhang et al., 2019; Xu et al., 2020; Cohan et al., 2018; Ruan et al., 2022). These works focus on the bottom-up computation

¹In this paper, "compute" and "infer" (and "computation" and "inference") are used interchangeably.

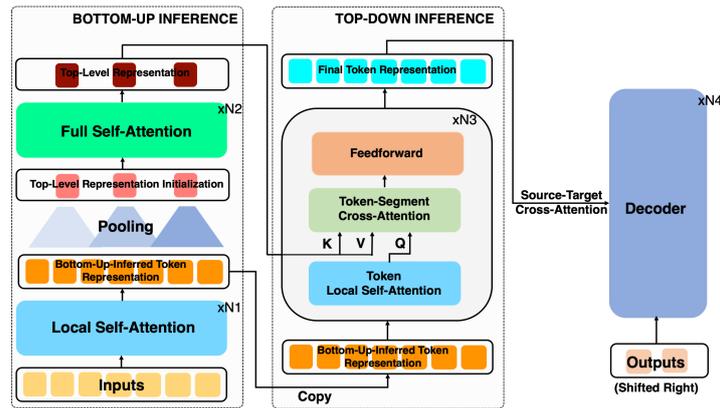


Figure 1: An overview of the top-down transformer. Suppose a document with 7 tokens is the inputs to the model, as shown on the bottom left. The bottom-up inference is achieved with local self-attention (N_1 layers) as shown in the left panel. To initialize the top-level representations, we pool bottom-up-inferred token representations with either equal weights or adaptive weights (see Section 2.3 for details). Top-level representations are then updated with full self-attention (N_2 layers) to capture global context. They are then used to update bottom-up-inferred token representations, accounting for the top-down update for token representations, as shown in the middle panel. The final token representations are attended by the decoder to generate a summary. Note that inference is used in the sense of statistical inference for latent variables and does not imply no training.

in a hierarchical model, computing higher-level representations (e.g., sentences, paragraphs) based on lower-level representations (e.g., words). In contrast, our method emphasizes the combination of bottom-up, as done in prior works, and top-down where lower-level representations are updated and enriched with higher-level representations (see the middle panel in Figure 1). This design is critical for summarization which requires global context. As shown in our ablations, removing the top-down update undermines the summarization performance.

The proposed method is agnostic to the model architecture. Due to the dominance of transformer models in NLP (Chen et al., 2018; Zhang et al., 2020; Sun et al., 2019; Martin et al., 2020), we instantiate our method with a transformer-based model. There is a bottleneck of applying transformers to long documents, because its computational and memory cost has a quadratic dependency on the sequence length. This issue is especially critical for summarization since we are more interested in summarizing long documents since short ones can be quickly read through by humans. To address this issue, a large amount of prior works have been devoted to develop efficient transformers with sub-quadratic complexity (Wang et al., 2020; Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020; Kitaev et al., 2020; Roy et al., 2021).

Our method provides a natural way to diminish this quadratic complexity issue. In the bottom-up computation, we use local self-attention where each token only attends the tokens within a local fixed-length window, and thus the complexity does

not grow as a function of the input sequence length. The top-down correction for (local) token representations enables them to capture more global context, reducing the limitation of local attention. In prior works like Longformer (Beltagy et al., 2020), Bigbird (Beltagy et al., 2020), local attention is also used. Our method is different from these models in terms of how to inject global information to locally computed representations. Longformer and BigBird utilize a few global tokens which attend and are attended by all local tokens, whereas we use top-down correction. Our approach can better capture global information compared to prior models, as demonstrated by clear performance improvements over these models in our experiments.

In summary, our methods have two key components: (1) *local attention in bottom-up computation* and (2) *top-down correction for locally-computed-token-representations by high level representations*. The first component alleviates the computational and memory cost and allows our model to process long documents, and the second component injects global information to local tokens and improves summarization performance. We call our model as *top-down transformer*, to emphasize the importance of the top-down update. We evaluate the model on a diverse set of summarization benchmarks. They cover documents from a variety of domains, including news articles and scientific, conversational, and narrative documents, and of various lengths ranging from hundreds of words (e.g., a news article), several thousands to over ten thousands of words (e.g., a scientific paper, a book chap-

ter), to even over hundred thousands of words (e.g., an entire book). Across all long document datasets, our models achieve competitive or state-of-the-art performance. We also show that our model is able to summarize a whole book. Compared to Wu et al. (2021) using GPT-3 and requiring humans to extensively label data, our model achieves competitive performance on book summary with only 0.27% parameters and a small amount of publicly available data. The diverse and strong empirical results support the effectiveness and wide applicability of the proposed model.

Our contributions are summarized as follows: (1) we propose a method which combines bottom-up computation and top-down update for long document summarization; (2) we conduct extensive evaluations and achieve strong performance on various long document benchmarks; and (3) we adapt our method to the challenging task of summarizing an entire book and achieve GPT-3-level performance with only 0.27% parameters.

2 Methods

Figure 1 gives a graphical overview of the top-down transformer. We introduce its details in this section. Suppose a document has N tokens, $\mathbf{t} = \{t_i\}_{i=1}^N$. In our method, token representations are computed by combining top-down and bottom-up processes. This leads to effective and efficient inference for token representations. They are then attended by a decoder to generate a summary, as in a regular encoder-decoder transformer.

2.1 Bottom-Up Computation

In the bottom-up path, contextual embeddings of the tokens, $\{e_i \mid e_i \in \mathbb{R}^d\}_{i=1}^N$, are computed with N_1 layers of local self-attention. In particular, each token t_i only attends to nearby tokens within a window of size of w . The complexity is hence $O(Nw)$, in contrast to $O(N^2)$ for full self-attention models.

2.2 Top-Down Computation

The efficiency with local self-attention in the bottom-up path nevertheless comes with a limitation, that is, each e_i only captures the context within a local window instead of that of the whole document. To mitigate this issue, we propose a top-down update for token representations.

Consider a two-level multi-scale latent structure for a document. The lower level consists of token representations, $\{e_i\}_{i=1}^N$, computed by the bottom-

up computation. The top level consists of units at a coarser level. It is affordable to apply full self-attention at the top level due to its coarser granularity, allowing these top-level units to capture global document context. The self-attention mechanism for the top-level representations is the original multi-head self-attention proposed in Vaswani et al. (2017).

Denote the top level representations after self-attention update as $\{s_j \mid s_j \in \mathbb{R}^d\}_{j=1}^M$ (see Section 2.3 for details on top-level representation initialization methods). We can then update the bottom-up-inferred token representations with the top-level representations. This is achieved with N_3 top-down computation layers, as illustrated by the middle panel in Figure 1. Each layer contains three transformations on $\{e_i\}$: (1) token self-attention, (2) token-segment cross-attention, (3) feed-forward. (1) and (3) are the same as those in the bottom-up layers or regular self-attention layer with local attention. (2) implementing the cross-attention between the top and bottom levels is the critical operation. In particular, each e_i is updated with cross-attention,

$$\tilde{e}_i = e_i + \text{LayerNorm}\left(\sum_{j=1}^M \alpha_{ij} f_v(s_j)\right), \quad (1)$$

$$\alpha_{ij} = \frac{\exp(f_q(e_i)^T f_k(s_j))}{\sqrt{d} \sum_{l=1}^M \exp(f_q(e_i)^T f_k(s_l))} \quad (2)$$

where f_q , f_k , and f_v indicate query, key, and value linear mappings, respectively. For notational clarity, Equation 1 only illustrates the case with a single attention head. In practice, we use multi-heads. The cross-attention operation injects global contextual information into bottom-up-inferred token representations, e_i , and yields global-context-aware token representations, \tilde{e}_i , conditioning on which a summary can be generated by a decoder.

To instantiate the top-down computation, we need to make two choices: (1) the number of top-levels above the token level and (2) the unit representation for each top-level. We choose to use one top level since it is sufficiently coarser to apply full self-attention for a wide range of long document benchmarks we experimented on. A natural choice for top level units is sentence, paragraph, and chapter, depending on the number top level considered. Such a choice however leads to complicated implementations and reduced scalability due to the varying length of these units. We hence choose a

simpler approach, where the top level consists of fixed-length segments of the documents. While we use a single top level, multiple top levels can be simply achieved with segments with increasingly coarser granularity.

In the top-down computation, segment-level self-attention has a complexity of $O(M^2)$, and token-segment cross-attention has a complexity of $O(NM)$. Thus, together with bottom-up inference, the complexity is $O(Nw + M^2 + NM)$. In practice, we use relatively small w (window size) and M (number of segments).

2.3 Pooling Methods

As aforementioned, we use a single top level, consisting of fixed-length segments. The segment representations are initialized by pooling token representations. Following the notation above, suppose a document is divided into M segments, and the embedding of the j th segment is initialized as,

$$s_j^{(0)} = \sum_{n=1}^k p_n e_{j \times d + n} \quad (3)$$

where k is the kernel size and d is the stride. p_n is the weight for the n th token. We introduce two approaches to compute the weights. The first method is average pooling (AvgPool) and hence $p_n = \frac{1}{k}$, which is simple and convenient. In the second approach, we leverage the reference summary to define the importance of each token to assign adaptive weights (AdaPool). Particularly, we learn an importance tagger with labels constructed with the reference summaries, which involves three steps:

1. Construct training labels for the importance tagger: (1) word lemmatization for document and reference words; (2) label a document word as important if it appears in the reference word list and is a non-stopword
2. Train a top-down transformer encoder with constructed labels as the importance tagger
3. Train the summarization model with oracle weights (i.e., constructed labels from Step 1.) and test it with the adaptive importance weight assigned by the learned tagger

In our experiments, we also used OracleAdaPool where the weights are obtained from Step 1 with the reference summaries. Note that if $\{p_n\}_{n=1}^k$ does not form a valid probability distribution, s_j can be

computed with a normalized weight distribution within each pooling window as follows,

$$s_j^{(0)} = \frac{\sum_{n=1}^k \exp(p_n) e_{j \times d + n}}{\sum_{n=1}^k \exp(p_n)}. \quad (4)$$

$\{s_j^{(0)}\}_{j=1}^M$ are updated with self-attention, yielding $\{s_j\}_{j=1}^M$, which are then used in top-down inference for token representations, as discussed in Section 2.2.

3 Experiments

3.1 Overview

We thoroughly evaluate the proposed method on various summarization datasets. See Table 7 in the appendix for a summary of datasets used in the current work. Our model is first evaluated on two standard long document summarization benchmarks, PubMed and arXiv (Cohan et al., 2018). It outperforms various efficient transformers and other approaches and achieves state-of-the-art performance. Although we focus on long document summarization, models under our framework is also applicable to shorter documents. We test our model on CNN-Dailymail (See et al., 2017), the most widely used short summarization dataset. Compared to a full self-attention model, our model achieves competitive or better performance. Recently, a more challenging benchmark, SummScreen (Chen et al., 2021), is proposed, where summarization systems need to summarize TV show scripts. These documents convey plot events often indirectly and implicitly in dialogues, in contrast to news and scientific articles where statements follow a logical order and facts are offered explicitly. Moreover, a typical episode contains multiple subplots that proceed in parallel. Solving this benchmark thus requires a system to draw information from utterances spreading out through the entirety of the input and integrate them to a concise description. Our model outperforms strong baselines on this challenging benchmark by a significant margin. Another challenging dataset, BookSum (Kryściński et al., 2021), is also recently released. It covers books from the literature domain, including stories, plays, and novels. Similar to ScreenSum, it requires integrating plot events from indirectly expressed descriptions. A further challenge is to process long-form texts up to hundreds of pages or over 100,000 words. Our method does well on this challenge, achieving competitive or superior performance compared to

	PubMed			arXiv		
	R-1	R-2	R-L	R-1	R-2	R-L
Pegasus (568M)	44.21	16.95	38.83	44.21	16.95	38.83
Dancer	46.34	19.97	42.42	45.01	17.60	40.56
TLM-I+E	42.13	16.27	39.21	41.62	14.69	38.03
SSN-DM	46.73	21.00	34.10	44.90	19.06	32.77
BigBird (577M)	46.32	20.65	42.33	46.63	19.02	41.77
Longformer (460M)	46.97	20.23	42.88	46.63	19.62	41.83
LSH	48.12	21.06	42.72	-	-	-
TopDownFormer (AvgPool) (464M)	<u>48.34</u>	<u>21.40</u>	<u>44.22</u>	<u>48.67</u>	<u>20.70</u>	<u>43.91</u>
TopDownFormer (AdaPool) (464M)	51.05	23.26	46.47	50.95	21.93	45.61
TopDownFormer (OracleAdaPool)	55.15	26.55	50.25	64.16	33.39	56.88

Table 1: Results on Scientific Articles. Best performance (no oracle) is in bold, and the second best is underlined.

a GPT-3-based model (Wu et al., 2021). While the GPT-3-based model has 175 billion parameters and requires human labelers to extensively write summaries and provide reward information, our model with 464 million parameters is 380 times smaller and merely requires training on relatively minimal data. These results suggest our framework is a generally effectively for documents of various lengths, domains.

3.2 Implementation Details

We use the same encoder-decoder architecture for all datasets. The encoder has 8 bottom-up layers and 4 top-down layers for tokens, and 2 self-attention layers for segments. The decoder has 12 layers. The encoder layers for tokens (12 layers) and the decoder layers are all initialized from BART (Lewis et al., 2020) except the parameters for token-segment cross-attention in the top-down layers, which are randomly initialized. The self-attention parameters for segments are also randomly initialized. The window size is 1024 unless otherwise specified. Our settings closely follow Longformer (Beltagy et al., 2020) which has 12 layers for the encoder and decoder, is initialized from BART, and uses a local window size of 1024. Thus, comparison with Longformer is a test of the effect of top-down correction for token representations. The segment-pooling has a kernel size of 32 and a stride size of 24. The maximum number of segments is 512. The maximum document lengths for PubMed, arXiv, CNN-DM, TVMegaSite, ForeverDreaming, BookSum are 8192, 16384, 1024, 12288, 12288, 12288, respectively. The optimizer for all models is Adam with an learning rate of $5e-5$. Model performance is evaluated with ROUGE scores (Lin, 2004). Reported performance is based on the checkpoint with the best validation R-2 score. Summary samples for each dataset generated by our models are provided in the Appendix.

3.3 Scientific Documents

We first test the effectiveness of our framework on two widely used datasets based on scientific documents, PubMed and arXiv. They consist of long documents of length ranging from several thousands of words to over ten thousands words. Three variants of our model with various pooling weights are presented. AvgPool, AdaPool, and OracleAdaPool in Table 1 indicate average pooling, pooling with adaptive weights, pooling with adaptive weights determined by references, respectively (see Section 2.3 for more details).

The experiment results are displayed in Table 1. Pegasus (Zhang et al., 2020) is pretrained on a large-scale of dataset with a pretraining objective specifically designed for summarization. It uses a full self-attention encoder and thus has to truncate the source document due to the quadratic memory complexity. The summarization-oriented large-scale pre-training makes it a strong baseline. Dancer (Gidiotis and Tsoumakas, 2020) takes a divide-and-conquer approach in which the summary is divided into sections and each section is paired to the appropriate section of the document and the model is trained on short sequences and has a low memory requirement. This is a straightforward approach achieving strong performance.

TLM-I+E (Pilault et al., 2020) first extracts salient sentences and then uses a GPT-style model to generate a summary by conditioning on the introduction section and extracted sentences (instead of the whole document), thus reducing memory requirement. SSN-DM (Cui and Hu, 2021) is an extractive model and uses a sliding encoder to process segments of a document and a memory module to capture autoregressive dependency between segments. These two models bear similarities to our model in that they use a multi-scale structure. The extracted salient sentences in TLM-I+E can be considered a representation of the document at a coarser granularity since salient information

is retained. Instead of keeping the coarser representations in the latent space, TLM-I+E reads out them to the observed word space. In SSN-DM, the fixed-size memory module pooling information from each segments can also be considered a high level representation of the document. Despite these similarities, our model, synergizing bottom-up and top-down inference, clearly outperforms these prior models.

BigBird (Zaheer et al., 2020), Longformer (Beltagy et al., 2020), and LSH (Kitaev et al., 2020; Huang et al., 2021) are efficient transformers. BigBird based on Pegasus pre-training combines local attention, random attention tokens, and global attention tokens. LSH uses content-dependent sparse attention based on local sensitivity hashing. Longformer is closely related to our models. It uses the same local attention as in our bottom-up computation except it has an extra [CLS] token which is a global attention token. Longformer is also initialized from BART. The only difference is that our models compute token representations with both top-down and bottom-up processes, in contrary to pure bottom-up in Longformer. The clear performance improvement over Longformer and other efficient transformers indicates the effectiveness of the synergy of bottom-up and top-down computation.

3.4 Short Documents

CNN-DailyMail			
	R-1	R-2	R-L
BART (Reported)	44.15	21.28	40.90
BART (Re-eval)	43.93	20.81	40.79
TopDownFormer (AvgPool)	<u>44.32</u>	21.03	41.40
TopDownFormer (AdaPool)	44.85	21.31	41.15
TopDownFormer (OracleAdaPool)	63.87	38.42	59.10

Table 2: Results on CNN-DailyMail. Best performance (no oracle) is in bold, and the second best is underlined.

To demonstrate the general applicability of the proposed framework, we show its effectiveness on short document summarization and compare it to full self-attention model. We hypothesize that although the bottom-up computation uses local self-attention, our method with the top-down correction would lead to competitive or better summarization performance.

Our model parameters are initialized from BART. Hence, BART with full self-attention forms a natural baseline, allowing for direct comparison. In the bottom-up inference, the local attention window size of our models is 256. As shown in Table 2,

our models achieve slightly better performance, especially in terms of R-1 and R-L, than BART. It confirms our hypothesis that a synergy of bottom-up with local attention and top-down inference with global attention is effective and achieves on-par or better performance as full self-attention.

3.5 SummScreen

Scientific and news articles often require that facts are offered explicitly and statements follow a logical order, which might allow summarization models to exploit layout and stylistic biases. We next test the proposed method on a more challenging dataset, SummScreen, which requires a model to draw and integrate information from indirect expressions across a wide range of the document. SummScreen (Chen et al., 2021) provides two datasets, TVMegaSite and ForeverDreaming, collecting from two different TV show transcript websites. Each document is the transcript of a TV show episode and the summary is an associated recap.

Table 3 summarizes the results. Extractive oracle is an extractive method by extracting nearest neighbors based on Rouge scores. Longformer is an abstractive method and takes the whole document as input. Hybrid models first select salient sentences and then input them to BART. Our models outperform these strong baselines and even achieves comparable or superior performance than prior models having access to oracle information.

3.6 BookSum

BookSum (Kryściński et al., 2021) is another challenging dataset, consisting of books from the literature domain including stories, plays and novels. It includes examples on three levels of granularity with increasing difficulty: (1) paragraph-level with inputs with hundreds of words, (2) chapter-level, with inputs with several thousands or over ten thousands of words, (3) book-level, with inputs spanning up to hundreds of pages and over hundred thousands of words. The chapter-level examples have comparable lengths to other popular long-form summarization datasets such as PubMed, arXiv. We first test our models on the chapter level. The book-level summarization is extremely challenging. First, the number of examples (313 books) is limited. Second, a book is too long to fit in current models. We train our model in a curriculum and recursive way to address the two issues.

	TVMegaSite			ForeverDreaming		
	R-1	R-2	R-L	R-1	R-2	R-L
Extractive Oracle	49.0	11.6	46.9	38.8	11.5	33.9
Longformer	42.9	11.9	41.6	25.9	4.2	23.8
Hybrid (BART + Content Selection)	38.8	10.2	36.9	25.3	3.9	23.1
Hybrid (BART + Oracle Content Selection)	42.1	11.9	40.9	26.4	5.0	23.3
TopDownFormer (AvgPool)	<u>49.30</u>	<u>14.35</u>	<u>47.45</u>	<u>35.84</u>	<u>8.86</u>	<u>30.62</u>
TopDownFormer (AdaPool)	51.02	14.66	49.01	36.84	9.19	31.12
TopDownFormer (OracleAdaPool)	53.55	15.63	51.29	39.54	10.08	33.59

Table 3: Results on SummScreen. Best performance (no oracle) is in bold, and second best is underlined.

BookSum Chapter Level			
	R-1	R-2	R-L
Extractive Oracle	42.68	9.66	21.33
BART (406M)	37.09	8.23	15.37
T5 (738M)	37.38	8.42	16.77
Pegasus (568M)	36.17	7.79	16.09
Longformer (460M)	32.84	7.45	14.59
BigBird (577M)	31.78	6.50	14.17
TopDownFormer (AvgPool) (464M)	<u>37.99</u>	<u>9.10</u>	<u>18.02</u>
TopDownFormer (AdaPool) (464M)	38.34	9.19	18.08
TopDownFormer (OracleAdaPool)	41.10	9.49	19.19

Table 4: Results on BookSum Chapter Level. Best performance (no oracle) is in bold, and second best is underlined.

3.6.1 Chapter Level

Table 4 displays the results. Kryściński et al. (2021) takes a divide-and-conquer approach to summarize chapters. They finetune BART, T5, and Pegasus on the paragraph level data and the chapter summary is obtained by concatenating the paragraph summary. This might miss the intra-paragraph context. Our models directly summarize the whole chapters and outperform these divide-and-conquer models. Efficient transformers, Longformer and BigBird, are also able to take in the whole chapters as inputs. But these bottom-up approaches clearly underperform our models.

3.6.2 Book Level

We first train a top-down transformer on chapter-level and then fine-tune it on book-level data. The inputs to the book-level model are (1) the concatenated chapter reference summaries in training or (2) the concatenated chapter summaries generated by the chapter-level model in testing. The chapter-to-book curriculum training is to mitigate the scarcity of book-level data. The recursive summarization of chapters and then books can be considered abstractive content selection applied to book data.

Table 5 summarizes the book-level results. The middle section shows the performance for the models with the divide-and-conquer approach (Kryściński et al., 2021), same as those for the chapter-level data. Wu et al. (2021) also attempts to summarize books using GPT-3 with reinforcement learning (RL) finetuning. The results are shown in third

BookSum Book Level			
	R-1	R-2	R-L
Extractive Oracle	46.62	9.17	18.31
BART	29.97	6.02	10.97
T5	39.46	7.69	13.77
Pegasus	35.29	6.79	12.71
GPT3-175B full tree RL	41.51	10.46	<u>16.88</u>
GPT3-175B first subtree RL	<u>43.19</u>	<u>10.63</u>	17.10
GPT3-6B full tree RL	36.79	7.22	14.84
TopDownFormer (464M)	44.19	10.89	16.13

Table 5: Results on BookSum Book Level. Best performance (no oracle) is in bold, and second best is underlined.

section in Table 5. Their method shares similarity with ours in that they decompose books into shorter sequences and train the model and summarize the text segments recursively. There are three differences between our approach and theirs. First, we train our model with the limited data from BookSum, while (Wu et al., 2021) requires human labelers to write summaries, which is highly costly. Second, our model has lower complexity, allowing it to take in longer input. Thus, we only need to decompose the book one time (into chapters), in contrast to multiple recursive decomposition steps. Multiple recursive summarization steps is prone to accumulating errors. Third, GPT-3 uses bottom-up inference to infer token representations, in contrast to the synergy of bottom-up and top-down inference in our approach. The last two differences might account for our competitive performance using a much smaller model (0.46B vs. 175B) and less data.

3.7 Ablation Studies

Our method has two key components: (1) local attention in bottom-up computation, and (2) top-down update to inject global context. We conduct ablation studies on these two factors. All ablation experiments are performed with PubMed.

We first ablate top-down update (TDU). The results are summarized in Table 6. The first row shows the performance of the top-down transformer with top-down update via cross-attention and window size 1024, which is our final model. The second row shows the performance for a vari-

ant of top-down update. In this variant, to update the bottom-up inferred token representations, we concatenate the token representations with the corresponding top-level segment representations, in contrast to the cross-attention approach used in the final model. We can see a clear performance degradation, indicating the importance of the cross-attention-based top-down update. The third row displays the results without top-down update, and the decoder attends the bottom-up-inferred token representations to generate summaries. Compared to our final model, the performance is also degraded, suggesting the effectiveness of the top-down update.

The lower panel of Table 6 presents ablations on window size (WS) of local attention. As the window size increases, the performance on all metrics enhances. The effect is quite large when the window size is increased from 32 to 256. The effect becomes smaller after 256, but the model performance can still benefit from larger window size.

		R-1	R-2	R-L
TDU via cross-attention	WS - 1024	48.34	21.40	44.22
TDU via concat	WS - 1024	47.04	20.36	43.03
No TDU	WS - 1024	46.97	20.23	42.88
TDU via cross-attention	WS - 32	46.30	19.55	42.21
TDU via cross-attention	WS - 64	47.25	20.37	43.12
TDU via cross-attention	WS - 128	47.44	20.56	43.35
TDU via cross-attention	WS - 256	47.89	21.06	43.77
TDU via cross-attention	WS - 512	48.08	21.16	44.05

Table 6: Ablation studies of Top-Down Transformer. TDU: top-down update. WS: window size.

4 Related Work

Summarization Models Prior works have proposed extractive models (Nallapati et al., 2017; Cui and Hu, 2021), abstractive models (Nallapati et al., 2016; Zhang et al., 2020), and hybrid models combining extractive and abstractive methods (Gehrmann et al., 2018; Pilault et al., 2020), for text summarization. Although our model mostly follows the abstractive approach, it also has connections to the hybrid models. These models usually first extract salient sentences from the source document and then summarize the extracted sentences with an abstractive model. Extracted sentences can be viewed a high level representation of the document, although it is the observed space but not in the latent space as in our framework. A continuous representations in the latent space facilitates end-to-end learning. Moreover, assigning importance weight with the importance tagger in our method resembles an extractive step in a hybrid model, and

thus top down transformer with learned importance tagger can be considered a hybrid model.

Efficient Transformers Despite the effectiveness of transformers on a variety of tasks, its quadratic complexity with respect to the sequence length has limited its application to problems with long sequences. A large amount of works have attempted to address this limitation. A major line of work focuses on designing various sparse attention mechanisms. These works can be roughly categorized into two groups, depending on whether the sparsity pattern is content-dependent (Kitaev et al., 2020; Roy et al., 2021; Wang et al., 2021; Liu et al., 2021) or content-independent (Child et al., 2019; Beltagy et al., 2020; Ainslie et al., 2020; Zaheer et al., 2020). Our work is mostly related to content-independent sparse attention. A main assumption of content-independent sparse attention is that the context temporally and/or spatially proximate to the query token is more important, which is intuitively sensible and supported by empirical attention analysis (Child et al., 2019). Thus, a common sparse attention pattern is local attention, where each query token only attends to a neighborhood within a fixed temporal and/or spatial window. While this reduces the complexity to be linear, a model with only local attention cannot model long-range dependency. Prior works combine local attention with other attention patterns with wider or global receptive field such as dilated attention, random attention tokens, and global attention tokens (Beltagy et al., 2020; Zaheer et al., 2020). Our models also use local attention for its efficiency and leverage top-down inference to enable global-context awareness.

5 Conclusion

In this work, we propose a summarization method which combines bottom-up computation with top-down computation to improve token representation inference. In the bottom-up pass, token representations are inferred with local self-attention to exploit its efficiency. Top-down correction is then applied to allow tokens to capture global context. Our model achieves (1) state-of-the-art performance on a wide range of long document summarization benchmarks, and (2) competitive performance on summarizing whole books using 0.27% parameters and much less training data, compared to a recent GPT-3-based model. These results indicate the general applicability and benefits of the proposed

framework.

Limitations

In the current work, we only explore a model with a single top-level layer. It would be a fruitful research direction to study models with multiple layers, with growing level of abstraction. This might improve both the efficiency and performance of the current model, since long range dependency is mostly captured by higher-level layers and the window size at the low-level can be small.

References

- Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. [ETC: Encoding long and structured inputs in transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online. Association for Computational Linguistics.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. [The best of both worlds: Combining recent advances in neural machine translation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia. Association for Computational Linguistics.
- Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2021. Summscreen: A dataset for abstractive screenplay summarization. *arXiv preprint arXiv:2104.07091*.
- Jianpeng Cheng and Mirella Lapata. 2016. [Neural summarization by extracting sentences and words](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.
- R. Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *ArXiv*, abs/1904.10509.
- Junyong Chung, Sungjin Ahn, and Yoshua Bengio. 2016. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*.
- Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. [A discourse-aware attention model for abstractive summarization of long documents](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.
- Peng Cui and Le Hu. 2021. [Sliding selector network with dynamic memory for extractive summarization of long documents](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5881–5891, Online. Association for Computational Linguistics.
- Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. [Bottom-up abstractive summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.
- Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. [Efficient attentions for long document summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.
- Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. [Reformer: The efficient transformer](#). In *International Conference on Learning Representations*.
- Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2021. Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Ye Liu, Jianguo Zhang, Yao Wan, Congying Xia, Lifang He, and Philip Yu. 2021. [HETFORMER: Heterogeneous transformer with sparse attention for long-text](#)

- extractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 146–154, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. [CamemBERT: a tasty French language model](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.
- Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. 2016. [Samplernn: An unconditional end-to-end neural audio generation model](#). *arXiv preprint arXiv:1612.07837*.
- Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. [Summarunner: A recurrent neural network based sequence model for extractive summarization of documents](#). In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. [Abstractive text summarization using sequence-to-sequence rnns and beyond](#). *arXiv preprint arXiv:1602.06023*.
- Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. [On extractive and abstractive neural document summarization with transformer language models](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. Association for Computational Linguistics.
- Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. [Efficient content-based sparse attention with routing transformers](#). *Transactions of the Association for Computational Linguistics*, 9:53–68.
- Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. [Histruct+: Improving extractive text summarization with hierarchical structure information](#). *arXiv preprint arXiv:2203.09629*.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.
- Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. [Ladder variational autoencoders](#). *Advances in neural information processing systems*, 29:3738–3746.
- Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. [Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. [Sequence to sequence learning with neural networks](#). In *Advances in neural information processing systems*, pages 3104–3112.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in neural information processing systems*, pages 5998–6008.
- Shuohang Wang, Luwei Zhou, Zhe Gan, Yen-Chun Chen, Yuwei Fang, Siqi Sun, Yu Cheng, and Jingjing Liu. 2021. [Cluster-former: Clustering-based sparse transformer for question answering](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3958–3968, Online. Association for Computational Linguistics.
- Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. [Linformer: Self-attention with linear complexity](#). *arXiv preprint arXiv:2006.04768*.
- Jeff Wu, Long Ouyang, Daniel M Ziegler, Nissan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. [Recursively summarizing books with human feedback](#). *arXiv preprint arXiv:2109.10862*.
- Shusheng Xu, Xingxing Zhang, Yi Wu, Furu Wei, and Ming Zhou. 2020. [Unsupervised extractive summarization by pre-training hierarchical transformers](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1784–1795, Online. Association for Computational Linguistics.
- Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. [Big bird: Transformers for longer sequences](#). In *NeurIPS*.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. [Pegasus: Pre-training with extracted gap-sentences for abstractive summarization](#). In *International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. [HiBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

A Data Statistics

Dataset	# Docs.	# Input Words	# Summary Words	Domain
PubMed	133K	3,224	214	Scientific
arXiv	215K	6,913	292	Scientific
TVMegaSite	22.5K	6,420	380	Conversational
ForeverDreaming	4.3K	7,605	113	Conversational
BookSum-Chapter-Level	12K	5,102	505	Narrative
BookSum-Book-Level	436	112,885	1,167	Narrative
CNN-DM	311K	906	63	News

Table 7: Summarization Datasets. It shows the total number of documents, the average number of input words, the average number of summary words, and the domain for each dataset.

B Qualitative Examples

PubMed Example #1: Reference

a new class of water - soluble c60 transfecting agents has been prepared using hirschbingel chemistry and assessed for their ability to act as gene - delivery vectors in vitro. in an effort to elucidate the relationship between the hydrophobicity of the fullerene core, the hydrophilicity of the water - solubilizing groups, and the overall charge state of the c60 vectors in gene delivery and expression, several different c60 derivatives were synthesized to yield either positively charged, negatively charged, or neutral chemical functionalities under physiological conditions. these fullerene derivatives were then tested for their ability to transfect cells grown in culture with dna carrying the green fluorescent protein (gfp) reporter gene. statistically significant expression of gfp was observed for all forms of the c60 derivatives when used as dna vectors and compared to the ability of naked dna alone to transfect cells. however, efficient in vitro transfection was only achieved with the two positively charged c60 derivatives, namely, an octa - amino derivatized c60 and a dodeca - amino derivatized c60 vector. all c60 vectors showed an increase in toxicity in a dose - dependent manner. increased levels of cellular toxicity were observed for positively charged c60 vectors relative to the negatively charged and neutral vectors. structural analyses using dynamic light scattering and optical microscopy offered further insights into possible correlations between the various derivatized c60 compounds, the c60 vector / dna complexes, their physical attributes (aggregation, charge) and their transfection efficiencies. recently, similar gd@c60 - based compounds have demonstrated potential as advanced contrast agents for magnetic resonance imaging (mri). thus, the successful demonstration of intracellular dna uptake, intracellular transport, and gene expression from dna using c60 vectors suggests the possibility of developing analogous gd@c60 - based vectors to serve simultaneously as both therapeutic and diagnostic agents.

PubMed Example #1: Model Output

a new class of water - soluble c60 transfecting agents with positively charged, negatively charged, or neutral chemical functionalities under physiological conditions was prepared using hirschbingel chemistry. transfection, cytotoxicity, and biophysical structure / activity studies were performed in an effort to elucidate the relationship between the hydrophobicity of the fullerene core, hydrophilicity, and the overall charge state of the c60 vectors in gene delivery / expression. only two positively charged c60 derivatives, namely, an octa - amino derivatized c60 and a dodeca - amino derived c60 vector, showed efficient in vitro dna delivery into cells. increased levels of cellular toxicity were observed for positively charged compounds relative to the negatively charged and neutral vectors, and aggregation was the major factor that negatively affected the cytotoxic profiles of these c60 - vector / dna complexes. the successful demonstration of intracellular dna uptake and gene expression from dna dna conjugates formed from c60 compounds suggests the possibility of developing analogous gd@c60 vectors to serve as both therapeutic and diagnostic agents.

PubMed Example #2: Reference

context : isolated involvement of the appendix in crohn's disease is reported to be 0.2 % to 1.8 %, and is usually associated with ileocaecal crohn's disease in 25 % of ileal and 50 % of caecal disease. while appendicitis in a patient who was previously diagnosed to have ileocaecal crohn's may be managed with appendicectomy and ileocaecal resection, appendicectomy alone when performed for appendicitis in a patient with unsuspected ileocaecal crohn's disease could lead to postoperative complications including enterocutaneous fistula.case report : a young female patient who underwent appendicectomy elsewhere for acute appendicitis presented to us with a persistent enterocutaneous fistula of 6 weeks duration. she had complained of general ill health and occasional altered bowel habits for 6 months prior to the acute appendicitis presentation. our investigations, including a ct scan, suggested the possibility of ileocaecal crohn's disease. she underwent excision of the enterocutaneous fistula and ileocaecal resection, and histopathology of the resected specimen confirmed crohn's disease. in the postoperative period she received mesasalazine. when last seen 2 years later during her regular follow - up, she was found to be in good health.conclusion : the possibility of ileocaecal crohn's disease should be considered in patients presenting with unexplained postoperative enterocutaneous fistula following appendicectomy. a high index of clinical suspicion is required to make a prompt diagnosis and institute appropriate further treatment in form of ileocaecal resection.

PubMed Example #2: Model Output

introduction : isolated involvement of the appendix in crohn's disease is reported to be 0.2 % to 1.8 %, and is usually associated with ileocaecal crohn's disease in 25 % of ileal and 50 % of caecal disease. while appendicitis in a patient who was previously diagnosed to have ileocaecal crohn's may be managed with appendicectomies, appendicectomy alone when performed for acute appendicectomy could lead to postoperative complications including enterocutaneous fistula.case report : a young female patient who underwent appendicectomy elsewhere presented to us with a persistent enterococcal fistula of 6 weeks duration. on further questioning, she informed us that she had presented to this hospital with a 2 - week history of abdominal pain, confined to the right lower abdomen. she had complained of general ill health and occasional altered bowel habits for 6 months prior to the acute appendicectomy presentation. our investigations, including a computerized tomography scan, suggested the possibility of crohns disease. she underwent excision of the enterococcus fistula and resection of the caecum, and histopathology of the resected specimen confirmed the diagnosis. in the postoperative period she received mesasalazine. when last seen 2 years later during her regular follow - up, she was found to be in good health.conclusion : appendicectomy alone is a safe procedure in patients with suspected crohn ileocaecal involvement. while appendiceal resection will suffice in those who present subacutely and are diagnosed preoperatively by clinical signs, a high index of clinical suspicion is required to make a prompt diagnosis.

Table 8: Summary Samples for PubMed

arXiv Example #1: Reference

we study experimentally the thermal fluctuations of energy input and dissipation in a harmonic oscillator driven out of equilibrium, and search for fluctuation relations. we study transient evolution from the equilibrium state, together with non equilibrium steady states. fluctuations relations are obtained experimentally for both the work and the heat, for the stationary and transient evolutions. a stationary state fluctuation theorem is verified for the two time prescriptions of the torque. but a transient fluctuation theorem is satisfied for the work given to the system but not for the heat dissipated by the system in the case of linear forcing. experimental observations on the statistical and dynamical properties of the fluctuation of the angle, we derive analytical expressions for the probability density function of the work and the heat. we obtain for the first time an analytic expression of the probability density function of the heat. agreement between experiments and our modeling is excellent.

arXiv Example #1: Model Output

we study the fluctuations of energy and dissipation in a thermostated harmonic oscillator driven out of equilibrium by an external deterministic forcing. this oscillator is very well described by a second order langevin equation. using the fluctuation dissipation theorem, we obtain fluctuation relations for the work given to the system by the external torque and for the heat dissipated by the system, by writing the first principle of thermodynamics between the two steady states. from the experimental results, we derive analytical expressions for the probability density functions of the work and the dissipated heat, and for their finite time corrections, both for a forcing linear in time and a forcing sinusoidal in time. from these expressions, we present some analytical derivations of fluctuation theorems based on experimental observations.

arXiv Example #2: Reference

the pairing hamiltonian constitutes an important approximation in many- body systems, it is exactly soluble and quantum integrable. on the other hand, the continuum single particle level density (cspld) contains information about the continuum energy spectrum. the question whether one can use the hamiltonian with constant pairing strength for correlations in the continuum is still unanswered. in this paper we generalize the richardson exact solution for the pairing hamiltonian including correlations in the continuum. the resonant and non - resonant continuum are included through the cspld. the resonant correlations are made explicit by using the cauchy theorem. low lying states with seniority zero and two are calculated for the even carbon isotopes. we conclude that energy levels can indeed be calculated with constant pairing in the continuum using the cspld. it is found that the nucleus @xmath{0} is unbound. the real and complex energy representation of the continuum is developed and their differences are shown. the trajectory of the pair energies in the continuum for the nucleus @xmath{1} is shown.

arXiv Example #2: Model Output

the exact solution of the richardson pairing hamiltonian is obtained by including the resonant and non resonant continuum through the continuum single particle level density (cspld). the gamow states, which appear in the complex energy representation, provide the main contribution from the continuum. the inclusion of the continuum has been used to study the unbound isotope @xmath{0} and beyond. it was found that the continuum pairs (pair energies with positive real components) converge to the real part of the uncorrelated pair energy and they do not appear in complex conjugate partners. as a result the total energy of the system may be complex. from the exact solutions of the pairing and pairing - like hamiltonians the cspld can be used to investigate many - body correlations.

Table 9: Summary Samples for arXiv

CNN-DailyMail Example #1: Reference

Michelle MacLaren is no longer set to direct the first " Wonder Woman " theatrical movie. MacLaren left the project over " creative differences " Movie is currently set for 2017.

CNN-DailyMail Example #1: Model Output

CNN confirms that director Michelle MacLaren is leaving the " Wonder Woman " movie. The movie, starring Gal Gadot, is still set for release on June 23, 2017.

CNN-DailyMail Example #2: Reference

Andrew Mogni, 20, from Glen Ellyn, Illinois, had only just arrived for a semester program when the incident happened in January. He was flown back to Chicago via air on March 20 but he died on Sunday. Initial police reports indicated the fall was an accident but authorities are investigating the possibility that Mogni was robbed. His cousin claims he was attacked and thrown 40 ft from a bridge.

CNN-DailyMail Example #2: Model Output

Andrew Mogni, 20, from Glen Ellyn, Illinois, had only just arrived for a semester program in Italy when the incident happened in January. He was flown back to Chicago via air ambulance on March 20, but he died on Sunday after falling off a 40 ft bridge in Rome in a suspected robbery attack in Rome, police reports indicated the fall was an accident but authorities are investigating the possibility he was robbed.

Table 10: Summary Samples for CNN-DailyMail

TVMegaSite Example #1: Reference

Jake meets Tad at ConFusion where Tad is enjoying a salad. Tad doesn't believe David's story as to why he is in Gloucester. Liza joins them and serves Tad with a restraining order to stay away from the bar in Gloucester. Amanda takes Trevor for an exam at the hospital and joins Angie. David also joins them. Erica sits alone in her hotel room when Opal comes to ask if she's seen the documentary on Pine Valley. Ryan stares at a blank television when Emma comes downstairs with Corinna. Emma asks Ryan if he is going to watch the documentary. In Gloucester, Gayle comes in to check on Greenlee. Greenlee tells Gayle that she has to get home. Greenlee clinches her fist as she imagines how it would be if she were home. Erica panics as to what Ryan might have said on the documentary. Erica receives another copy of the documentary that Hayley did of Pine Valley. David asks Angie if read the gift that he gave her. Amanda tells David that she knew that he had lied to her about having patients to see at the hospital and instead had gone to Gloucester. Tad reprimands Jake for wanting to go to Gloucester without telling Amanda. Jake gets up to go back to work and sees Amanda and Trevor. Jake asks her how long she had been standing there. Amanda answers, " Long enough." Madison and Angie discuss Madison's mom. Erica finally gives in and agrees to watch the DVD. Opal is thrilled, but Erica insists that she wants to watch it alone. Ryan visits with DVD in hand and suggests that they watch it together. David meets with Dr. Clayton and tells him about Greenlee. Greenlee dreams of her family and friends back in Pine Valley. Erica and Ryan watch the DVD. Liza lets Madison know that her father had gotten jail time, but would be out of jail within a year. Amanda and Jake discuss David and how Jake doesn't believe that he is really sick. Amanda tells Jake that if her persists in accusing David then she doesn't know how much longer they can go on. Greenlee meets with Dr. Clayton about her surgery. Ryan and Erica kiss. Liza and Tad kiss in his apartment.

TVMegaSite Example #1: Model Output

Tad and Jake are at Krystal's. Jake lets Tad know that David is going to Gloucester. Liza walks in and gives Tad a restraining order against him. At the hospital, Angie tells Amanda that she had seen her on Hayley's documentary. David walks up and listens to their conversation. At home, Opal questions Erica if she had watched the documentary on television about Pine Valley. Erica tells Opal that she doesn't want to see the documentary. Ryan gets ready for Emma and Corinna's sleepover. Ryan lets Emma know that he hadn't been able to watch the documentary that Hayley had shown on television. Greenlee dreams that she is back in Pine Valley with her family. Erica awakens from her coma and finds out that Jackson is alive. Jackson tells Greenlee how much he had missed her. Erica lets Opal know that she would like to fall in love again. Opal asks her if she is insecure about Ryan and what is going on with him. Erica gets a visit from a man, who gives her a DVD of the documentary from Hayley. David and Angie argue over the fact that he isn't as sick as she claims to be. Amanda accuses David of lying to her about where he had been the night that she was stabbed. Tad tries to talk Liza out of breaking into David's bar in Gloucester, but she refuses to listen to him. Madison comes into the hospital and tells Angie that she can not go to her father's sentencing. Madison lets Angie know that her father is being sentenced today and she was going to go, but Angie encourages her to go. Jake and Tad try to talk her out of going to the hearing, but Liza insists on going. Jake tries to get Tad to promise that he will not go into Gloucester without Liza's permission. Erica and Opal watch the video of Ryan's confession. Ryan comes to visit Erica and asks her to watch a little TV. David meets with Dr. Clayton about Greenlee's condition. David introduces Greenlee to Clayton. Tad visits Liza at the bar and apologizes to her for putting her in this position. Tad and Liza begin to argue over his interference in her life. David calls Greenlee and tells her that they are going to take her on a tour of the medical facilities in Gloucestershire.

TVMegaSite Example #2: Reference

Erica and Ryan are in her office at Fusion kissing when Greenlee walks up to the door and starts to turn the doorknob. David clasps his hand over her mouth to keep her from screaming. At ConFusion, Liza and Tad kiss before they start to dance. Krystal watches then asks Rob to take her back to his place. Jake and Amanda spend a quiet evening at home when there is an incessant knocking on the door. Jake opens the door and Opal lets them know that she read her tea leaves and knows that someone is headed back into their lives. She fears that it is David. Jake and Amanda don't seem to be too concerned by Opal's anxiety attack over her tea leaves. David reminds Greenlee that Dr. Coleman said she could face another surgery. In talking to David, Greenlee realizes that Ryan is with another woman and comes to the conclusion that it is Kendall. At ConFusion, Tad sees David rushing out to his car. Ryan and Erica come clean to the press that they are involved. Liza visits Amanda and tries to soothe her fears that David is back in town. Tad and Jake visit David at Wildwind. Jake promises David that he will be watching him. Ryan and Erica come home to his penthouse and finds things completely out of order. Erica and Ryan make love in front of the fireplace. Greenlee lets herself into Ryan's place and sees him and Erica making love.

TVMegaSite Example #2: Model Output

At the hospital, Liza kisses Tad. Krystal walks in and sees them kissing. Liza asks Tad if she can steal him. At Wildwind, Jake and Amanda are in bed with the baby when there is a knock on the door. Jake answers it and it is Opal. Opal tells Jake that she knows that David is coming back to town. Jake assures her that he doesn't know where David is. At Fusion, Greenlee questions David as to what he is doing here. Greenlee demands to see Ryan, but David refuses to let her see Ryan. David tries to get Greenlee to calm down and let him examine her, but Greenlee insists on going up on the roof to talk to Ryan. At Ryan's home, Erica tells Ryan that she is not used to things going so smoothly in their relationship. Jake tells Opal that Wildwind is being sold and a real estate agent is showing it to the public. Jake lets Opal know that he hadn't heard from David in a while. Amanda comes downstairs and tells Jake about Opal's tea leaves giving her a strong feeling that something from their past is coming to town and that David could be there already. Jake asks Amanda if she is all right. Amanda lets Jake know that David had disappeared. At the Confusion bar, Tad tells Liza that he thinks that there is something going on between him and Krystal, but Liza denies it. David examines Greenlee and lets her know that Ryan is with another woman. David offers to take Greenlee back to his place, but she insists on knowing who the woman is before she changes her mind. Erica asks Ryan to take her home for a romantic dinner and a fire in the fireplace. Ryan and Erica arrive home to find a picture of the moon on the table. David tells Greenlee that he is taking her back to Gloucester for another surgery. David lets Greenlee know that Zach and Kendall had left town with Spike and had taken Spike with them. Jake and Tad burst into David's hotel room and find him hanging upside down in the attic. Jake orders David to leave town. David insists that he will leave town, but Jake tells him that they will be watching for him. David leaves the hotel room, but when he opens the door to leave, he finds that Greenlee is gone. Jake calls Amanda and tells her that they have to file a lawsuit against David because he is back in town.

Table 11: Summary Samples for TVMegaSite

ForeverDreaming Example #1: Reference

Sydney races to find a cure for Vaughn, but in order to find the antidote, Sydney must make a deal with Sark that could endanger Sloane's life. Meanwhile, Will continues his research for Vaughn and discovers some disturbing inconsistencies involving 20-year-old standardized IQ tests. Sydney finds out that Vaughn has a girlfriend.

ForeverDreaming Example #1: Model Output

Sydney must find a way to save Vaughn's life while trying to track down Sark's base of operations in order to obtain the antidote to a deadly virus that Sloane has been infected with. Meanwhile, Vaughn is in critical condition after being exposed to Rambaldi's toxin, and Sydney is forced to make a dangerous deal with Sark in exchange for his aid in getting rid of him. Will meets with a professor to find out if any of the CIA's standardized tests were ever used to recruit American children in the 1980's, and discovers that one in ten thousand first graders could pass the test.

ForeverDreaming Example #2: Reference

Half a universe from Earth, the Doctor and Martha answer a distress call from the cargo ship SS Pentallian, which will impact a star in 42 minutes. The ship scooped out part of the star, which is alive, for cheap fuel. The star uses crew members Korwin and Ashton as host bodies, and begins taking out everyone on board. Martha and crew member Riley, while trying to reach the front of the ship, are jettisoned into space in an escape pod by Ashton. The Doctor remagnetises the pod to bring Martha and Riley back. The Doctor begins being taken over by the star, and tells Martha to vent the engines, getting rid of the "sun particles" in the fuel. This causes the engines to start working again, and frees the ship from the star's pull.

ForeverDreaming Example #2: Model Output

The Doctor and Martha are trapped on board a space station in the Torajji system, where the crew are trying to prevent the ship from colliding with the sun. The Doctor uses the sonic screwdriver on Martha's mobile phone to activate Universal Roaming Activation, which allows him to travel anywhere in space and time without interference from the ship's control centre. However, the device malfunctions and the ship begins to fall towards the sun, and the Doctor is forced to use the emergency escape pod to escape. The pod, which contains the Doctor, Martha and two other crewmembers, is destroyed by the impact, but the Doctor manages to return to the control centre to try and stop the ship hitting the sun before it does so.

Table 12: Summary Samples for ForeverDreaming

BookSum Book-Level Example #1: Reference

At the opening of Act I, it is a cloudy autumn day on a Russian country estate. In the garden, the old nurse Marina stands at the samovar and offers Doctor Astrov something to eat, but he refuses. He complains about the difficulty of his job. Telegin, an impoverished local landowner, sits with them. Voynitsky, known as Vanya, comes out of the house and joins them. He is almost fifty and is weary and irritable. He complains about his brother-in-law, Serebryakov, Serebryakov's young second wife, Helen, and about how their visit has turned the place upside down. Serebryakov, Helen, and Serebryakov's daughter, Sonya, join them for a moment. After they depart, Vanya sighs about Helen's beauty and then complains about how he has toiled his whole life on this estate for the professor and it has come to naught. After Vanya's sister's death, he and Sonya worked here so the professor could continue his studies and his writings, but Vanya has come to see that work as foolish and irrelevant. When Astrov suggests that Vanya is jealous, Vanya laughs that he obviously is, especially as the old, gout-and-rheumatism-ridden man seems to attract beautiful women. Helen ventures outside and tells Astrov his services are not needed for her husband. Mrs. Voynitsky, Vanya's mother and Sonya's grandmother, tells them about a new pamphlet written by a friend in Kharkov. When Vanya sneers that all they do is read pamphlets, she becomes distressed and claims he hates her. Vanya merely says he is old, tired, and frustrated. A laborer arrives and tells Astrov he is wanted at the factory; the doctor bitterly departs, but not before they all discuss how he is very interested in forestry work. Sonya speaks up cheerfully about how Astrov is trying to save the old forest from destruction because forests make people happier. Astrov speaks of how Russians have torn down the forests and destroyed the wildlife; they no longer create, but rather destroy. After Sonya walks Astrov out, Vanya tries to seduce Helen, but she pushes him away. She muses about how Sonya clearly seems to love the doctor but he does not love her back. Helen sighs that she is simply bored and life is too much for her. In Act II, Serebryakov complains to Helen of how he is old and no one respects him. His querulous behavior only annoys Helen, who begs him to stop it. Serebryakov ignores her and bemoans how his life of scholarship seems to be nothing now. Sonya joins them and tells them Serebryakov must see Astrov now; she wants her father to stop behaving like a child. The elderly nurse Marina comforts Serebryakov and leads him out. Helen tells Vanya, who entered the room, that her husband wears her. Vanya can only lament that everything is over for him and his life was wasted on trivial things. Helen is annoyed and moves to leave, but he bars her way. She accuses him of being drunk, and he admits to it. After Helen sweeps out of the room, Vanya ruminates on what a fool he was not to fall in love with her when she was younger; he once admired the professor, but now he does not. When Astrov returns, he mocks Vanya for having feelings for Helen, but Vanya will not admit it. Astrov leaves to get a drink; Sonya pulls him aside and makes him promise to stop drinking and stop getting her uncle drunk. He agrees. They continue to talk for a moment. He comments that Helen is beautiful but idle and useless. This country life makes people like that, and he despises it; he has been beaten down and sees no light at the end for himself. The peasants are all the same, and educated people are ridiculous. He only likes forests. Sonya compliments him and tries to cheer him up. As he prepares to leave, she asks how he might feel if he were to out that a friend of hers has feelings for him, and he drolly says he cannot love anyone. After he leaves, Sonya feels a surge of happiness though she is not sure why. In Act III, Sonya confesses to Helen that she loves Astrov, and Helen suggests that she say something to see if the doctor loves Sonya too. Sonya gives her permission for Helen to do this. Astrov and Helen meet to ostensibly look at his forestry maps. He discourses volubly on the patterns of deforestation until he sees that Helen is uninterested. Helen insists she is interested but says they should talk about something else. She point-blank asks if he likes Sonya, and he says no. He then moves in to seduce Helen, but she wants none of it. As he tries to kiss her, Vanya enters the room with flowers. Helen is horrified by the situation and begs Vanya to tell her husband that they must leave today. A moment later, Serebryakov and the others enter and Serebryakov announces that he has an idea to sell the estate because he and Helen need to afford a place in the city. This announcement angers Vanya tremendously, and he begins to complain violently about how Serebryakov is a fraud, is uninspired, is thankless, and how he, Vanya, has labored for Serebryakov his whole life and for no reason. He insists this is Sonya's estate. Serebryakov is startled by Vanya's outburst. He insists he cannot stay here anymore. Sonya implores him to talk to her uncle and he agrees. He departs, and those in the room hear a gunshot, then another. Helen and Vanya struggle over a revolver as Vanya screeches that he missed the professor. Cursing, he sinks into a chair. In Act IV, Telegin and Marina discuss Serebryakov and Helen's planned departure for that day, then exit the room. Vanya and Astrov come in. Astrov mocks Vanya for his behavior and asks him to return what he stole. Vanya maintains that he is innocent of theft. Astrov laments how this parochial existence crushes people: it is stultifying and useless. When Sonya enters Astrov tells her to tell her uncle to return the bottle of morphia he stole. Sonya turns to her uncle and tearfully asks him for the bottle. He complies. She takes him to make up with her father. Helen enters to tell Astrov goodbye. He tries to seduce her again in a rather lackluster fashion; she kisses him and bids him farewell. Everyone bids goodbye to Serebryakov and Helen, who leave the estate. Sonya and Vanya return to work. Marina suggests Astrov stay for dinner; he refuses and says he must be off. Astrov leaves, and even though Sonya knows he did not love her, she is sad. Vanya, though, claims he is extremely depressed. With both in tears, Sonya comforts her uncle. She tells him that life may be difficult for them now, but the afterlife will be full of peace, love, and rest.

BookSum Book-Level Example #1: Model Output

The scene opens in a country house in the Russian countryside. Ivan, a young man, is sitting with his mother, who is knitting him a Christmas stocking. He tells her that he does not want to have any tea, but she insists that he drink it anyway. She asks him how long they have known each other, and he says that it has been ten years since he first met her. He has not had a day of rest since he has known her, and has toiled without repose or a day's freedom ever since. He had a typhoid epidemic in the third week of Lent, and when he returned home, he had to take care of a man who had died of typhoid fever. He then sat down and closed his eyes for a while and wondered if he had killed the man. He wonders if the future generations will forget him, but he is sure that they will not. Ivan tells them that the house is a mess, because the Professor has been up all night writing and doesn't have time to eat or drink. He asks them to wake him up at two in the morning, but when they do not wake him, he thinks that the Professor wants some tea. They all come in from their walk, and as they come in, they talk about how happy they are to be together, and how they are all living in peace and contentment. Ivan is struck by the sight of a beautiful woman, and says that as he looks at her, his heart swells with unbridled happiness. The work of preserving the forest is thrilling, he says, and Dr. Astroff watches over the old woods and sets out new plantations every year. He wants to show them to Helena and Sonia, and Helena asks him if he finds it interesting. Helena is sitting next to him, and tells him to look for Batushka's works in the library the next day. He is worried that he is getting angina from his old age. Helena tries to tell him to make peace with himself, and to stop being so angry with everyone else. But he says he will not listen to her, because he has spent too much time thinking about the past, and now he cannot bear to watch the success of others and to fear death. Helena gets up and walks away from him, sitting down at a distance. She tries to get him to tell her why he doesn't love anyone. He says he is not an idiot, that he can understand her, but that he cannot understand her because she is young and healthy and beautiful and longing for life and he is an old dotard, almost a dead man. She tells him that she can understand him and that he should be able to understand her. Now, she says, she is too old and sick to continue to live in the country, and she has a young daughter and a young wife to care for. She doesn't want to live on the income generated by the estate, so they have to find some way to guarantee that they can live in town on the money generated each year. They decide to sell the woods, because they don't have enough money to do so every year, and they can't afford to buy the woods every year either. They need to find a way to make sure they can afford a certain amount of money to buy a cottage in Finland. They also have to figure out what to do with the rest of the property. The estate is worth ninety-five thousand roubles, and the estate was bought from his uncle, who had a debt of twenty-five. Semyon Lakedemonoff's brother bought the estate from the Lakedemons' uncle. The family ties are what got the estate to be worth so much money. Everyone agrees that the estate is for Sonia's good, and that she should get to keep it. But now that he has gotten old and ill, the time has come for him to dispose of his property in regard to the interests of his VOITSKI, ASTROFF, SEREBRAKOFF, Vanya, and SONIA arrive at the house to say goodbye to Tommo and Marina. They are to move to Kharkov to live with the professor and his wife. They have been frightened by what they have just witnessed, and decide to go to the city to see if they can find a place to live there. They will not be staying in the village any longer. Except for Vanya and Alexander, who stay to say good-bye to his wife and son-in-law. The Professor kisses them all three times, and then goes out to see them off. He gives them one last kiss to each of them before he leaves. They say they will always remember each other with pleasure, that they are interesting and original, and original. They shall rest

Table 13: Summary Samples for BookSum Book-Level

BookSum Book-Level Example #2: Reference

In his London studio, artist Basil Hallward puts the finishing touches on his latest portrait, that of a young man. Although Lord Henry, who is visiting with Basil, asks about the young man's identity, Basil declines to answer, noting his preference for secrecy. Basil never intends to exhibit the painting, because if he did, it would bare the deepest feelings in his soul. However, Basil lets slip that the subject of the portrait is Dorian Gray, who shortly thereafter pays the two men a house call. Lord Henry immediately begins to influence Dorian, suggesting that he should treasure and guard his youth and beauty while he has them, because they will soon fade. Terrified of aging, Dorian wishes he could trade his soul to stay as young as he looks in the portrait; a short while later, he again wishes that he could stay young while the image in the painting aged. The portrait thus begins to take on a life-like existence; in fact, Basil's threat to burn the portrait is likened to "murder" and Basil prefers the company of the portrait to the real Dorian. Dorian falls in love with a young actress, Sibyl Vane, a woman he barely knows. She plays a different woman at each night's performance, earning the label of "genius" from Dorian, who is as smitten with her acting more than with her personality. They become engaged, much to the surprise of Lord Henry and Basil. The sweet, wholesome Sibyl discusses her engagement with her family. Because her mother is indebted to the theatre manager, Mr. Isaacs, for fifty pounds, she is against the marriage unless Dorian is wealthy; they do not know that he is. Sibyl's angry brother, James, is leaving for Australia, but he vows to kill Dorian if he wrongs his sister in any way. James also confronts his mother about gossip he has heard – that his mother and deceased father never married, which Mrs. Vane admits is true. Dorian attends a performance of Sibyl's with Lord Henry and Basil, but the performance is terrible. Sibyl tells Dorian she can no longer act, because he has shown her a beautiful reality. Dorian is disgusted by her poor acting, because her performances were what drew him to her; he dismisses her and returns home. To his surprise, the portrait shows marks of cruelty around the mouth, lines that do not show on Dorian's face. He begins to suspect that his wish is coming true, so he vows to be good so that both he and the portrait can remain young. He, therefore, intends to apologize to Sibyl the next day and makes to marry her after all. However, he is too late: Sibyl commits suicide at the theatre that night. Dorian first feels responsibility for her death, but then views it both as wonderful entertainment and a selfish act on her part. Lord Henry tries to keep Dorian's name out of the scandal. Dorian and Lord Henry spend the evening at the opera. The next morning, Basil arrives and expresses concern for Dorian, given the events of the previous day. Dorian, however, is completely unconcerned about Sibyl or her family; he wants to talk only of happy subjects. The next day, he covers his portrait and moves it to the attic, to which Dorian has the only key. He then settles in to read a yellow book sent by Lord Henry; the book becomes Dorian's blackprint for life. Several years pass, and Dorian lives a hedonistic life according to the guidelines established by Lord Henry and the yellow book. While the face in the portrait has turned ugly, Dorian remains young, beautiful, and innocent. People talk about Dorian's "madness of pleasure" and his dreadful influence on the people around him, but that is of no consequence to him. Finally, when he is thirty-eight years old, Dorian shows the portrait to Basil, who begs Dorian to repent of his sin and ask that the wish be revoked. Instead, Dorian kills Basil and hides his body. Blackmailing his old friend Alan Campbell, Dorian is able to dispose of Basil's body. An hour later, Dorian attends a party, but is bored and distracted. He then heads for an opium den and, out on the street, meets Sibyl's younger brother, who has been waiting for an opportunity to harm Dorian for nearly twenty years. Dorian makes a case for mistaken identity when he claims to have the face of a twenty-year-old and cannot be the man James is looking for. A woman in the street reveals that Dorian "sold himself to the devil for a pretty face," so James again pursues Dorian. At his country estate one week later, Dorian entertains guests but believes James in hunting him. Dorian soon learns, however, that a man accidentally killed in a hunting accident is James, and so he feels safe. The novel concludes six months later. Dorian and Lord Henry dine, and talk turns serious – Dorian talks of Basil, and Lord Henry reflects on a sermon he heard the previous Sunday while walking in the park. Lord Henry also inquires about the secret of Dorian's youth, which Dorian dismisses. Dorian then asks Lord Henry never to give the yellow book to anyone else. That evening, while Dorian examines the portrait, he decides to destroy it with the knife used to murder Basil. Soon after, Dorian's servants and a police officer find an old, ugly man lying dead on the ground in front of a portrait of a young and innocent Dorian.

BookSum Book-Level Example #2: Model Output

Dorian Gray is sitting in the garden of his uncle's house in London, playing the piano. In the center of the room is a portrait of a young man of extraordinary beauty, and in front of it is Basil Hallward's portrait of Dorian Gray. The portrait belongs to Hallward, and he refuses to show it at the Grosvenor Academy, because it contains too much of himself in it. He tells Dorian's uncle, Lord Henry Wotton, that he is in love with Dorian and wants to marry him. Dorian is sitting at the piano and finishes a painting that he has been working on. He looks at the portrait and realizes that it is of him. He realizes that he will never be able to look at it again, but he cannot stop looking at it because he wants to remember the beauty of his own face. He asks his uncle for some information about Dorian, and his uncle tells him that Dorian has a beautiful mother who was married to a poor man who was killed in a duel. She left him a son, who is very good-looking and who has inherited all of her property. Lord Henry tells him to write to her and ask for some advice, and Dorian agrees. One day, Dorian meets Sibyl Vane, a beautiful young woman who works as a governess for a rich family in the East End of London. She is in the employ of Lord Henry's friend, Mr. Erskine of Treadley, and Lord Henry wants to see her. He also wants to get her out of the hands of the Jew who has her bound to him for three years and eight months. He proposes to her, but she refuses him. She says that she does not think he is good enough for her, and she will never love anyone of his rank. He is disappointed, but does not say anything to his mother about it. The next day, he meets the Duchess of Monmouth, who tells him he should find a wife and marry her. She wants him to have a future and not to spend his money frivolously. He agrees, but when he tells her that he does not love her, she laughs at him and refuses to call him by his new name, Prince Charming. He goes to see the play, and is horrified to see that the face on the canvas is that of the portrait of Romeo and Juliet. He cannot believe that he could have done such a terrible thing to Juliet and that she could still be his wife. He leaves the theater and wanders the streets of London until he finds himself in Covent Garden. He finds some women waiting for him, and one of them laughs when he calls her by his nickname, "Prince Charming." She curses him and runs away. He runs into a dark alley and is suddenly grabbed by a man with a gun pointed at his head. It is James Vane. Vane threatens to kill Dorian if he doesn't make peace with God. He gives Dorian one minute to make his peace before he kills him. When Dorian gets to the street, he finds that the man he was trying to kill is not the same man he thought he was. It turns out that Vane is twenty-eight years younger than Dorian. The woman who took his money tells him not to talk to her again. She runs off, and when Dorian looks back, the woman has disappeared. When he wakes up the next morning, he has not had a nightmare. He writes two letters to his assistant, Alan Campbell, telling him that there is a dead man sitting on a table in his house, and that he must destroy the body so that no one will ever know who he is. He then goes to his bedroom and finds a small box of lacquer, which he takes out and puts inside. He puts the box back, gets into a horse-drawn carriage, and gives the driver an address. The driver takes him to the address, and as he is leaving the house, he sees the dead body of a man on the table. When Campbell returns, he tells Alan not to disturb the body, but to come back at seven o'clock in the evening. When the man arrives, he throws the picture over the table, but Dorian does not believe that it has been disturbed. He returns home and finds that Campbell has brought back the chemicals and the irons, and the other things that he needs to do the job. He opens the cabinet where he had hidden Basil's coat and bag, and finds the green paste. At midnight, he gets a hansom and leaves the house with the instructions to meet him at 7 o'clock the next day. He sits in the back of the carriage as the driver drives him through the streets. He wonders if it is possible to cure the soul by means of the senses and the body by way of the soul. He wakes up in the middle of the night to find that the portrait has not changed.

Table 14: Summary Samples for BookSum Book-Level

Open Information Extraction with Entity Focused Constraints

Prajna Upadhyay

BITS Pilani Hyderabad Campus,
Secunderabad, India

prajna.u@hyderabad.bits-pilani.ac.in

Oana Balalau and Ioana Manolescu

Inria and Institut Polytechnique de Paris
Palaiseau, France

first.last@inria.fr

Abstract

Open Information Extraction (OIE) is the task of extracting tuples of the form (subject, predicate, object), without any knowledge of the type and lexical form of the predicate, the subject, or the object. In this work, we focus on improving OIE quality by exploiting domain knowledge about the subject and object. More precisely, knowing that the subjects and objects in sentences are often named entities, we explore how to inject constraints in the extraction through constrained inference and constraint-aware training. Our work leverages the state-of-the-art OpenIE6 platform, which we adapt to our setting. Through a carefully constructed training dataset and constrained training, we obtain a 29.17% F1-score improvement in the CaRB metric and a 24.37% F1-score improvement in the WIRE57 metric. Our technique has important applications – one of them is investigative journalism, where automatically extracting conflict-of-interest between scientists and funding organizations helps understand the type of relations companies engage with the scientists. Our code and data are available at <https://github.com/prajnaupadhyay/openie-with-entities>

1 Introduction

Open Information Extraction (OIE) is the task of extracting triples from unstructured corpora in a domain-independent manner. A triple consists of a subject, a relation, and an object. OIE has important applications, such as question answering (Lu et al., 2019), or automatically creating or extending knowledge bases (Bhutani et al., 2019). OIE is a challenging task, with the performance of state-of-the-art models varying from 88.5% F1 score (Wang

et al., 2021) to 34% (Gashteovski et al., 2021), depending on the difficulty of the benchmark.

When the named entities in a domain are known to be the subject/object of extractions, OIE should also identify relations between these entities. An important use case is automatically creating a knowledge base of relations between scientists and companies, i.e. identifying conflict-of-interest between the scientists and funding bodies, where the named entities are the names of scientists and companies, and the relation describes the conflict of interest between them. Clustering these relation phrases, such as *received a research gift from, received speaker fees or consults for* helps analyze the relationships that companies engage with the scientists. These relations are crucial to understanding scientists’ positions on health issues (Oreskes and Conway, 2010) in investigative journalism. However state-of-the-art OIE models do not always retain named entities in the extractions, for example, given the sentence “*Shahrad Taheri received funding for research through a grant from Cambridge Weight Plan*”, an OIE tool (Kolluru et al., 2020a) returns $\langle \text{Shahrad Taheri, received, funding for research} \rangle$. While this extraction correctly identifies the subject of the triple, the quality of the predicate and object could be improved as follows: the extraction $\langle \text{Shahrad Taheri, received funding for research through a grant from, Cambridge Weight Plan} \rangle$ retains the second important entity (Cambridge Weight Plan) and is precise about the relation. Such sentences are frequent in the declarations of conflict of interest that authors add to articles in PubMed, a dataset of scientific articles on life sciences and biomedical topics.

In this work, we focus on **relation extraction, when the subject and object are named entities**. In particular, we would like to significantly improve the performance of OIE tools, such that triples as $\langle \text{first entity, predicate, second$

The work was done when the first author was at Inria and Institut Polytechnique de Paris.

entity) are not missed or poorly extracted. To achieve this, we leverage deep learning with constraints, i.e. techniques that enforce constraints on the classifier's predictions. Constrained learning is very common in sequence-to-sequence tasks, such as relation or entity extraction, where the output should have a specific form. Constraint learning has also been successfully used in OIE. In our case, we enforce constraints on the subject, object and predicate forms, and we investigate several techniques to achieve the best result, such as constraint-aware training (Nandwani et al., 2019) and constraint inference (Lee et al., 2019). We deployed our technique within OpenIE6 (Kolluru et al., 2020a), a state-of-the-art tool for OIE.

Our salient contributions are: *i*) We extend the OpenIE6 model with entity-centric constraints; *ii*) We implement the constraints as penalties in the loss function, and as hard constraints during inference. *iii*) We show through an extensive evaluation that our method improves over the state-of-the-art; *iv*) We perform a large scale evaluation of the system, on conflict of interest declarations from PubMed bibliographical data.

2 Related Work

In the literature, the extraction of triples of the form $\langle \text{subject}, \text{relation}, \text{object} \rangle$ has been studied in several settings. A relation can be expressed using a surface form, i.e., the tokens present in a sentence, or a canonical form, usually introduced in a knowledge base. In the most general setting, we do not enforce any constraints on the types of the three elements, and the task is referred to as **open information extraction** (OIE). In the most restricted setting, the subject and object are entities, and the relation comes from a predefined set of relations. This task is known as **relation extraction**. Finally, **open relation extraction**, also referred to as **relation discovery**, refers to approaches that use little training (such as distant supervision, few-shot learning, or semi-supervision) or no training (unsupervised) to classify relations between entities. Some inconsistencies arise in the use of the terminology in the literature, e.g., "open relation extraction" has been also used to designate open information extraction, in (Mesquita et al., 2013).

Open Information Extraction. Open information extraction (Kolluru et al., 2020a; Etzioni et al., 2008) extracts triples from unstructured corpora in a domain-independent way. More precisely, the

relations are not known beforehand and the subject and object are not required to be named entities. The state-of-the-art techniques are based on neural networks, which model the problem as a sequence labeling task (Kolluru et al., 2020a; Stanovsky et al., 2018; Cui et al., 2018). OpenIE6 (Kolluru et al., 2020a) is a neural model that achieves state-of-the-art results when compared with several other models (Del Corro and Gemulla, 2013; Gashteovski et al., 2017; Cui et al., 2018; Stanovsky et al., 2018; Roy et al., 2019; Zhan and Zhao, 2020; Kolluru et al., 2020b). Since these tools work without any domain knowledge, they might miss or extract poorly triples containing named entities. We aim to solve this problem, and our technique is trained to improve relation extraction when entities are present in the corpus.

Relation Extraction. In relation extraction (Han et al., 2020), given a sentence containing two entities, the task is to select the relation between the entities from a fixed set of relations. This is achieved via a classifier, and the challenge is in identifying relevant features for classification. Traditionally this has been achieved via hand-crafted features, such as lexical, syntactic, or semantic (Jiang and Zhai, 2007; Nguyen et al., 2007). More recently, neural models such as BERT (Devlin et al., 2018) have been very successful in relation classification (Baldini Soares et al., 2019).

Open Relation Extraction/Relation Discovery. In (Yao et al., 2011), the authors first discover relations between entities using the dependency paths between two tagged entities, and they propose an unsupervised probabilistic generative model for inducing clusters from the surface forms. In (Yu et al., 2017), surface forms of relations are first extracted by taking into account the dependency path between entities, and finally, they are mapped to canonical forms present in a KB. In (Hu et al., 2020), the authors propose a relation encoder based on BERT (Devlin et al., 2018) that computes an embedding representation of the relation based on the sentence where named entities appear, together with an adaptive clustering technique that does not require prior knowledge of the number of clusters. While some approaches (Yao et al., 2011; Yu et al., 2017) extract surface forms of relations when the arguments are entities, similar to our goal in this work, they use for this only dependency path information and do not deal with conjunctive sentences

as OpenIE (Kolluru et al., 2020a). In addition, OpenIE6 has shown better performance than models using dependency parsing such as ClausIE (Kolluru et al., 2020a; Del Corro and Gemulla, 2013).

3 Problem Definition

Our goal is to extract triples from sentences that respect the guidelines detailed by the CaRB metric (Bhardwaj et al., 2019), i.e., they should be *i*) complete: all triples should be extracted from a sentence, *ii*) asserted: the triple should be implied from the sentence *iii*) informative: the triple should contain maximum relevant information from the sentence and *iv*) atomic: extraction cannot be split into multiple extractions.

Given a sentence S containing entities $E = \{e_1, \dots, e_i, \dots, e_n\}$, we denote by $\langle s, r, o \rangle$ a triple that is extracted from S . The CaRB rules can be customized to fit our setting as follows:

- **Complete:** For every e_i , there exists at least a triple $\langle s, r, o \rangle$ where e_i is s or o .
- **Asserted:** Each tuple must be implied by the original sentence.
- **Informative:** The extraction should contain the maximum possible information from S . For instance, from *Joe Biden is the president of the US*, an uninformative extraction is $\langle \text{Joe Biden, is, the president} \rangle$ while the informative extraction is $\langle \text{Joe Biden, is the president of, US} \rangle$.
- **Atomic:** If s or o contains e_i , then it contains only that entity and no additional tokens. If s or o contain e_i and e_j , it is always possible to create two triples $\langle s_1, r, o \rangle$ and $\langle s_2, r, o \rangle$, $s_1 = e_i$ and $s_2 = e_j$, similarly for o .

4 Entity Focused Constraints

OpenIE6 (Kolluru et al., 2020a) receives in input a sentence and outputs a list of extractions of the form $\langle \text{subject, predicate, object} \rangle$. The architecture of the model is a deep neural network that first encodes tokens using BERT (Devlin et al., 2018), and then iteratively identifies at most M extractions, i.e., calls the same architecture for each extraction for M times (Figure 1). The embeddings of the labels generated at the end of the 1st iteration are added to the embeddings of the tokens in the second iteration, and so on. This adds context so

that a new extraction is generated the next time. Each token is assigned a label from $\{S$ (subject), R (relationship), O (object) or N (none) $\}$.

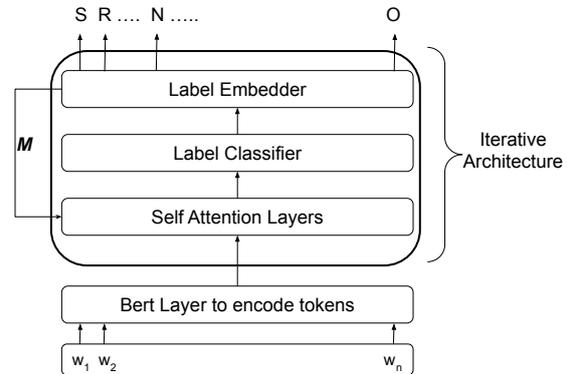


Figure 1: OpenIE6 uses the same architecture to generate embeddings for the words in M extractions, with the output of the previous extraction given as input for the next extraction

OpenIE6 constraint-aware training OpenIE6 uses constraint-aware training to infuse the model with task-related knowledge in the form of constraints. The model learns to satisfy these constraints during training without explicitly enforcing them during the inference, hence these types of constraints are typically referred to in the literature as *soft constraints*. This is achieved by adding additional penalties in the loss function, as follows:

POS Coverage (POSC). Tokens labeled as nouns, verbs, adjectives, or adverbs should be part of at least one extraction.

Head Verb Coverage (HVC). Verbs that are not *light verbs* (e.g., do, give, have, make, etc.), referred to as *head verbs*, should be present in the relation span of *a few but not too many extractions*.

Head Verb Exclusivity (HVE). The relation span of one extraction should contain *at most one head verb*.

Extraction Count (EC). The extractions having head verbs in the relation should be at least equal to the number of head verbs in the sentence.

These are **entity independent constraints**. Their full equations can be found in the OpenIE6 paper (Kolluru et al., 2020a).

Adding entity-specific constraints. We enforce additional constraints to obtain extractions satisfying our problem statement. Let $x_n^{ent} \in \{0, 1\}$ denote whether the n th token w_n belongs to some entity tagged in the sentence, and E be the set of entities. At each extraction level m , the model com-

puts $Y_{mn}(k)$, the probability of assigning to the n th token the label $k \in \{S, R, O, N\}$ (subject, relation, object or none). We introduce the following entity-specific constraints:

1. **Entities as subject or object (ENT-ARG).**

Each entity in the sentence should be present in at least a subject or object of an extraction:

$$J_{ent_so} = \sum_{n=1}^N x_n^{ent} \cdot (1 - \max_{m \in [1, M]} (\max_{k \in \{S, O\}} Y_{mn}(k))) \quad (1)$$

The penalty is 0 when for each token belonging to an entity ($x_n^{ent} = 1$) we have $Y_{mn}(k) = 1$, that is maximum probability of being in the subject or object, for at least one extraction.

2. **Entity exclusivity (ENT-EXCL).** The subject and object should contain at most one entity each. Let $p_e(k)$, with $k \in \{S, R, O, N\}$ be the average token probability of label k in entity e , where e consists of one or more tokens. Then, we express the penalty as follows:

$$J_{ent_exs} = \sum_{m=1}^M \max(0, (\sum_{e \in E} p_e(S) - 1)) \quad (2)$$

$$J_{ent_exo} = \sum_{m=1}^M \max(0, (\sum_{e \in E} p_e(O) - 1)) \quad (3)$$

The penalty is 0 when no entity is labeled as subject/object or when only one entity is labeled as such ($\sum_{e \in E} p_e(O/S)$ is 0 or 1).

3. **Entity in relation penalty (ENT-REL).** A penalty is introduced if an entity appears as a part of a relation of some extraction. This loss is directly proportional to the probability of tokens that are part of some entities and which have been labeled as part of a relation:

$$J_{ent_rel} = \sum_{n=1}^N x_n^{ent} \cdot \sum_{m=1}^M Y_{mn}(R) \quad (4)$$

The penalty is 0 when $Y_{mn}(R)$ is 0 for every token of an entity.

4. **Entity segmentation penalty (ENT-TOG).** A penalty is introduced if tokens describing the same entity are not labeled in the same way, for example, the first token of the entity is part of the predicate, while the rest of the

tokens are part of the object. Let $w(e)$ be the set of tokens in a given entity e . Let $l_p^m(w)$ be the predicted label of a token (the label with the highest probability) at extraction m . As we are concerned with entities described by two or more tokens, the predicted label l_e^m of the entity e is the majority label of its tokens, or the label with the highest total sum of probabilities in case of a tie. For each $w \in w(e)$, we introduce a loss equivalent to $Y_{mw}(l_p)$ if $l_p^m(w) \neq l_e^m$:

$$J_{ent_seg} = \sum_{m=1}^M \sum_{e \in E} \sum_{w \in w(e)} Y_{mw}(l_p) (1 - \delta_{l_p^m(w), l_e^m}) \quad (5)$$

where δ is the Kronecker delta function.

Finally, the total loss can be written as:

$$J_{ent} = J + \lambda_1 J_{ent_so} + \lambda_2 (J_{ent_exs} + J_{ent_exo}) + \lambda_3 J_{ent_rel} + \lambda_4 J_{ent_seg} \quad (6)$$

where λ_* are hyperparameters, while J is the original OpenIE6 loss.

Constraints at inference. We investigate a second type of constrained learning called constraint inference. The constraints applied in this setting are hard constraints, which the model is forced to apply. The constraints are applied in the decoding phase and modify the tokens' labels (S, P, O, N). We propose three constraints inspired by the entity constraints introduced in the constraint-aware training.

1. **Entity exclusivity.** Once we have encountered one entity labeled as a subject or object in the sentence, the following entities are not allowed to receive the same label.
2. **Entity in relation.** We enforce that an entity appearing in the predicate is classified according to its second-best class probability.
3. **Entity segmentation penalty.** We enforce that all the tokens belonging to an entity be labeled with the same label.

We do not transform the constraint *entities as subject or object* in an inference constraint as it cannot be applied at the level of one existing extraction. This constraint can only be a penalty in the loss, such that it rewards sets of extractions in which all the entities are part of the arguments.

5 Experimental Evaluation

5.1 Datasets

We use the OpenIE6 data for training and validation and Pubmed data for testing. The OpenIE6 dataset consists of Wikipedia sentences, while the Pubmed data is a set of conflict of interest statements between authors and various organizations, such as those illustrated in Section 1.

Given that our focus is on improving performance when entities are present in a sentence (Section 3), and in particular, enforcing that entities are the subject or object, we need appropriate training data for the task. We are unaware of a dataset of extractions where arguments are entities, while the extraction also has the surface forms of relation. For example, FewRel (Han et al., 2018) and TA-CRED (Zhang et al., 2017), two standard datasets used in relation extraction, do not contain the surface form of the relation; they only label the entire sentence as containing a particular relation.

Training data. The OpenIE6 training dataset consists of 91K sentences and 190K extractions of the form $\langle \text{subject}, \text{predicate}, \text{object} \rangle$. We tag entities in each sentence using the state-of-the-art named entity recognition tool Flair (Akbiik et al., 2019). We focus on extractions of the following form: *i*) The **subject** of the extraction is exactly one entity; and *ii*) The **object** ends with an entity. We discard the extractions that do not match these constraints. In each extraction, we keep only the entity in the object and move the preceding tokens to the relationship part of the extraction. For example, one of the sentences in the original training set is “*Parmenides had a large influence on Plato, who not only named a dialogue, Parmenides, after Parmenides, but always spoke of Parmenides with veneration.*” and one of the extractions is $\langle \text{Parmenides}, \text{had}, \text{a large influence on Plato} \rangle$. The extraction satisfies both the above conditions, hence we transform it to $\langle \text{Parmenides}, \text{had a large influence on}, \text{Plato} \rangle$. If the object contains only an entity, we apply the identify transformation. We refer to a sentence with at least one transformed extraction as a *clean sentence*.

We create 3 training datasets:

ORIGINAL: The original training set containing 91K sentences.

CLEAN: 7K clean sentences with their modified extractions.

MIXED: We add the remaining sentences and their extractions from the original training set to CLEAN.

Gold data. We created a gold standard dataset from Pubmed conflict-of-interest statements to be used as test data. We tagged and counted the entities with NER Flair and selected 282 sentences with a minimum of 2 entities. The maximum number of entities found in a sentence was 14.

We asked the annotators to find all the triples $\langle s, p, o \rangle$ containing those entities as arguments (in *s* or *o*). In addition, the extractions should follow the guidelines explained in Section 3 on completeness, assertion, informativeness, and atomicity. The total number of extractions obtained after annotations were 1113. One annotator annotated each sentence.

Quality of gold data. To evaluate the dataset’s quality, we sampled 50 sentences from our gold sentences, and one of the authors annotated them so that we had two annotations for this set. We found the agreement by considering one annotation as gold and computing WiRE57 F1. The agreement F1 score obtained was 83, which is a high agreement.

Table 1 shows example annotations of triples for the sentence *Menno Huisman reports grants from and personal fees from Boehringer Ingelheim and Bayer Health Care*. For each CaRB property, we show the correct and incorrect extractions. An extraction of the form $\langle \text{Menno Huisman}, \text{reports grants from}, \text{Bayer Health Care}, \text{Germany} \rangle$ violates the assertion property because it adds extra information to the sentence. $\langle \text{Menno Huisman}, \text{reports}, \text{grants} \rangle$ violates the informativeness property even if it is a valid extraction because it lacks the complete second argument, i.e., Boehringer Ingelheim. The extraction $\langle \text{Menno Huisman}, \text{reports grants from}, \text{Boehringer Ingelheim and Bayer Health Care} \rangle$ is not atomic because the two entities in the second argument should be part of 2 extractions. If any of the four correct extractions adhering to the completeness property are missing, this property is violated.

5.2 Models

We experimented with the following models:

OpenIE6. This is the default OpenIE6 model.

OpenIE6 (ECTR). OpenIE6 model with entity constraint training (ECTR), as in Section 4.

	Correct	Incorrect
Completeness	\langle Menno Huisman, reports grants from, Boehringer Ingelheim), \langle Menno Huisman, reports grants from, Bayer Health Care), \langle Menno Huisman, reports personal fees from, Boehringer Ingelheim), \langle Menno Huisman, reports personal fees from, Bayer Health Care)	If any of the extractions is missing
Assertion	\langle Menno Huisman, reports grants from, Bayer Health Care)	\langle Menno Huisman, reports grants from, Bayer Health Care, Germany)
Informativeness	\langle Menno Huisman, reports grants from, Boehringer Ingelheim)	\langle Menno Huisman, reports, grants)
Atomic	\langle Menno Huisman, reports grants from, Boehringer Ingelheim)	\langle Menno Huisman, reports grants from, Boehringer Ingelheim and Bayer Health Care), \langle Menno Huisman, reports grants from and personal fees from, Boehringer Ingelheim)

Table 1: Examples of correct and incorrect annotations for the 4 CaRB properties

`OpenIE6 (ECTR, ECIN)`. To the trained model `OpenIE6 (ECTR)`, we add constraints at inference in the evaluation of the test data.

`OpenIE6 (ECIN)`. To the trained model `OpenIE6`, we add constraints at inference in the evaluation of the test data.

We note that the models use a different coordinate boundary model than the one in the `OpenIE6` paper. We retrained the coordinate boundary model using a newer Huggingface Transformers library version (Wolf et al., 2020) for compatibility with our code. However, we could not reproduce the accuracy, obtaining 83.3 instead of 85.4. A better coordinate boundary model would positively impact performance, both with and without constraints.

Parameters. The model’s training consists of two phases, a warm-up phase, where the training is done without constraints, and a constrained training part. The warm-up training was done for 30 epochs, and the constrained training was done for 15 epochs. During constrained training, all constraints had equal weights. The learning rate was set to $5e-06$. `BERT-base-cased` model was used with two iterative layers. We repeat the experiments with 6 different random seeds for the network initialization, and we average the results. We run our code on a 32GB GPU.

Baselines We implement four baselines.

`ConnectingPhrase`. This simple technique returns the phrase connecting the two entities in a

sentence as the relation between them. It comprises the following steps:

1. We first use the coordinate boundary detection model (available with `OpenIE6` code). Coordinate boundary detection models (Saha and Mausam, 2018; Kolluru et al., 2020a) split a conjunctive sentence into smaller parts. For example, the sentence “*Adrian Brown and Shahrads Taheri received funding for research through a grant from Cambridge Weight Plan.*” is split into:
 - (a) “*Adrian Brown received funding for research through a grant from Cambridge Weight Plan.*”
 - (b) “*Shahrads Taheri received funding for research through a grant from Cambridge Weight Plan.*”

This is crucial to improve the recall.

2. Next, we label the entities in sentences obtained using `Flair` (Akbi et al., 2019).
3. For each consecutive pair of entities e_i, e_{i+1} in the sentence, we return an extraction containing as subject e_i , as predicate the phrase connecting the entities, and as object e_{i+1} .
4. We filter the extractions by removing the ones whose predicates do not contain a token labeled as a verb by a part-of-speech parser. The final set of extractions is obtained at the end of this step.

`DependencyPath`. We follow the same steps as in `ConnectingPhrase`, except that in 3. above, we return as the predicate the tokens on the dependency path between entities e_i and e_{i+1} .

`PostprocessedOpenIE6`. We run the original OpenIE6 tool and post-process its output as follows: we tag entities in `subject` and `object` of the extractions, and then we modify extractions, in the same manner as when we created the CLEAN dataset (Section 5.1), and leave unchanged the ones not satisfying our conditions.

`FilteredOpenIE6`. We remove the extractions from `PostprocessedOpenIE6` that were not modified according to the procedure used for generating the CLEAN dataset.

Evaluation metrics. Several evaluation metrics have been proposed to evaluate the performance of an OpenIE system. **WiRe57** (Lechelle et al., 2019) is a one-to-one matching metric, in which each system extraction is matched to exactly one gold extraction. Given a sentence, a system extraction matches a gold extraction if they share at least one word from each of the relation, subject, and object. Two extractions are compared by computing the token level recall and precision between the gold subject and system subject, respectively, the predicates and objects. Precision is the percentage of system words found in the gold extraction. The recall is the percentage of gold words in the systems’ predictions. The system extractions are matched one-to-one to gold extraction in decreasing order of $F1$ -score. **CaRB** (Bhardwaj et al., 2019) is a many-to-one matching metric in which several gold extractions can be matched to one system extraction when computing the recall. This avoids penalizing a system if one extraction would better correspond to two or more golden extractions, as is the case, for instance, in `<Adrian Brown; has received travel grants from; Cambridge Weight Plan and Oxford University>` (note that there should have been two triples extracted here, each with a different object). Precision is computed by matching system extractions one-to-one to gold extractions, decreasing order of precision score. Hence, we will penalize the extraction above when computing precision, as one gold extraction will not be matched.

We report both metrics, however, WiRe57 is more in line with our task as it respects the atom-

icity constraint in Section 3, given that it does not reward system triples with several entities in one argument.

6 Results and Discussion

Evaluation. In Table 2 we show the results on the test data, measuring both CaRB and WiRe57. We use the different training datasets that we introduced and the different training constraints. When training without any entity constraints, the training dataset can make a significant difference, as we observe `OpenIE6` trained on CLEAN has a more than 26% increase in CaRB F1 than `OpenIE6` trained on the ORIGINAL dataset. In addition, adding entity constraints further improves the results as shown by the models `OpenIE6 (ECTR)` which has the best WiRe57 score for all CLEAN, MIXED and ORIGINAL models. The smallest improvement is for the model trained with the ORIGINAL dataset, as in this case the training data may be in conflict with the constraints, having for example several entities in one argument. `OpenIE6 (ECIN)` improves upon `OpenIE6`, with a significant increase in the precision of the WiRe57 metric, which is expected given the hard constraints are being forced on the triples. However, `OpenIE6 (ECTR)` has a more significant improvement than `OpenIE6 (ECIN)` according to WiRe57 (the metric aligned with our problem statement, as explained in Section 5.2), showing that it is more important to have soft constraints, which are rewarding good extractions during training and hence obtaining a better extraction model. Combining soft and hard constraints gives the best model, `OpenIE6 (ECTR, ECIN)`. Regarding the baselines, `PostprocessedOpenIE6` and `FilteredOpenIE6` have good precision but lower recall than our top-performing models, showing the importance of the constraint learning and adapted training datasets.

Ablation study. We perform an ablation study to evaluate the importance of the entity constraints added during the training. We take our best performing model, `OpenIE6 (ECTR)` trained on the CLEAN dataset, and we train it with 1, 2, or 3 constraints at a time. Table 3 shows the results obtained on our test set. When we add just one constraint, as expected, the constraint ENT-ARG enforces the highest WiRe57 recall, as it has learned to penalize extractions where entities may be missing from the arguments. However, this model has the lowest precision, due to the fact it allows more than one

Method	Training data	CaRB			WiRe57		
		P	R	F1	P	R	F1
OpenIE6	CLEAN	75.07	62.52	67.97	66.15	45.79	54.04
OpenIE6(ECTR)		<u>80.76</u>	61.77	<u>69.95</u>	73.05	<u>45.37</u>	<u>55.95</u>
OpenIE6(ECIN)		76.05	<u>63.64</u>	69.65	70.91	44.80	54.89
OpenIE6(ECTR, ECIN)		79.85	63.09	70.46	<u>74.88</u>	45.36	56.48
OpenIE6	MIXED	52.23	50.69	51.60	43.30	37.41	40.04
OpenIE6(ECTR)		57.83	55.22	56.38	49.60	38.27	43.20
OpenIE6	ORIGINAL	43.01	39.75	41.29	32.45	31.81	32.11
OpenIE6(ECTR)		41.61	40.90	41.19	33.29	31.78	32.48
PostprocessedOpenIE6		59.25	52.52	55.62	43.82	42.49	43.12
FilteredOpenIE6		85.01	41.44	55.78	81.01	30.77	44.57
DependencyPath	-	58.51	57.54	58.02	59.65	36.94	45.62
ConnectingPhrase	-	58.23	70.63	63.84	58.45	45.31	51.04

Table 2: Model comparison on the test dataset. Best values are in bold and second best are underlined.

Constraints	CaRB			WiRe57			Violations			
	P	R	F1	P	R	F1	ENT-ARG	ENT-EXCL	ENT-REL	ENT-TOG
\emptyset (OpenIE6)	75.07	62.52	67.97	66.15	45.79	54.04	32.44	1.59	16.30	1.11
{ENT-ARG}	64.58	<u>64.02</u>	63.70	58.21	47.52	52.24	25.01	4.90	4.87	0.56
{ENT-EXCL}	78.65	60.99	68.67	68.98	44.83	54.32	33.59	1.30	15.65	1.35
{ENT-REL}	78.18	62.24	69.10	70.50	<u>46.27</u>	55.73	28.72	2.68	8.77	1.18
{ENT-TOG}	75.39	62.77	68.28	67.15	45.51	54.20	32.51	1.48	15.57	1.13
{ENT-ARG, ENT-EXCL}	78.06	61.18	68.28	67.32	45.48	54.22	32.09	1.56	14.59	1.12
{ENT-ARG, ENT-REL}	74.23	59.20	65.79	62.86	46.13	53.04	<u>26.94</u>	4.60	<u>5.45</u>	0.89
{ENT-ARG, ENT-TOG}	67.85	64.24	65.54	62.17	46.17	52.82	27.77	3.98	7.98	<u>0.82</u>
{ENT-EXCL, ENT-REL}	80.22	61.97	69.89	72.93	45.17	55.77	32.27	<u>1.41</u>	10.35	1.36
{ENT-EXCL, ENT-TOG}	78.56	61.45	68.86	69.63	44.93	54.59	33.05	1.43	13.90	1.35
{ENT-REL, ENT-TOG}	74.38	61.53	66.93	66.60	43.21	52.31	32.33	3.34	9.29	1.74
EC \ ENT-ARG	79.12	62.70	69.90	73.20	45.03	55.75	32.15	1.49	9.64	1.32
EC \ ENT-EXCL	75.48	61.28	67.47	65.99	45.46	53.70	27.75	4.49	6.76	0.94
EC \ ENT-REL	80.89	60.48	69.19	68.75	45.74	54.89	31.72	1.58	12.31	1.13
EC \ ENT-TOG	79.98	62.20	<u>69.94</u>	72.76	45.43	<u>55.92</u>	31.26	1.59	9.62	1.22
EC (OpenIE6(ECTR))	<u>80.76</u>	61.77	69.95	<u>73.05</u>	45.37	<u>55.95</u>	31.39	1.63	9.77	1.30

Table 3: Ablation study with models trained on the CLEAN dataset. We report CaRB, WiRe57, and the percentage of entity constraints violations on the test set.

entity in one argument. Removing the constraint from the set, EC \ ENT-ARG, gives us the highest precision. A combination of ENT-EXCL and ENT-REL performs the best among the models that were trained with 2 constraints, which is expected since the models trained with ENT-EXCL and ENT-REL were the top-2 performing models when trained individually. Enforcing only ENT-TOG does not bring important improvements, and training with the whole EC is slightly better than when training with EC \ ENT-TOG. Hence, ENT-TOG could be removed without a significant drop in quality.

For a complete analysis, we also compute the percentage of violations in the extractions (Table 3). For ENT-ARG, we count as a violation every entity that is not found in at least one extraction, and we

divide by the total number of entities in the test set. For ENT-EXCL, we count a violation for each subject or object with more than one entity and normalize by twice the number of extractions. For ENT-REL, a violation is a relation containing an entity, normalized by the number of extractions. Finally, for ENT-TOG, a violation is an entity in extraction with more than one tag (S,O,R,N), normalized by the number of extractions containing an entity. We observe that ENT-ARG is violated the most, followed by ENT-REL. When enforcing ENT-ARG, we obtain the best results for 3 out of 4 constraints. This does not result, however, in the best F1 score, showing the importance of minimizing violations of type ENT-EXCL. ENT-ARG and ENT-EXCL have competing goals: ENT-ARG

enforces the occurrence of entities in arguments, but ENT-EXCL does not allow more than one entity in an argument. So, whenever ENT-ARG is enforced with ENT-EXCL, we see an increase in the number of ENT-ARG or ENT-EXCL violations. Finally, when comparing the model with no entity constraints, `OpenIE6`, with the model enforcing all 4 constraints, `OpenIE6(ECTR)`, we observe a more significant difference in the violations ENT-ARG and ENT-REL, the constraints that are more frequently violated.

Quality of Named Entity Recognition on Pubmed. We sampled and annotated 50 test set sentences, taking care to keep the words together in long named entities, such as “Oregon Health and Science University Center for Embryonic Cell and Gene Therapy”. We obtained 88% F1 score for the NER model Flair (Akbik et al., 2019), in line with the performance of the model on Ontonotes (Weischedel et al., 2017) and CONLL (Tjong Kim Sang and De Meulder, 2003).

Evaluation on the CaRB dataset. `OpenIE6` has been evaluated on the CaRB dataset (Bhardwaj et al., 2019). We evaluate our constrained models to investigate their performance on this standard benchmark, see Table 4. Note that annotating guidelines for CaRB were not the same as for our Pubmed test data: there might be more than one entity in the arguments of a relation. This inherently limits the quality of our results. However, we show that the constrained models trained on the MIXED and ORIGINAL datasets have competitive performance with the original `OpenIE6` model while performing much better on our test data, as shown in Table 2. As expected, the models trained on the CLEAN dataset perform the worst, as they have seen only extractions with entities in the arguments; to achieve the best results, the user should choose a model considering the nature of the dataset. We note that the results for the `OpenIE6` model are slightly lower than those reported in the original paper because of the coordinate boundary model, as mentioned in Section 5. The conjunctive model is a core `OpenIE6` component; gains in its precision would likely improve performance, both with and without constraints.

Conflicts of interest in PubMed. We analyse the extractions by `OpenIE6(ECTR)` and the original `OpenIE6` model on a larger PubMed dataset consisting of 170K sentences. Table 5 shows the

Method	Training data	CaRB	WiRe57
		F1	F1
OpenIE6(ECTR)	CLEAN	24.44	11.59
OpenIE6	CLEAN	27.76	12.42
OpenIE6(ECTR)	MIXED	50.16	38.15
OpenIE6	MIXED	50.27	38.59
OpenIE6(ECTR)	ORIGINAL	50.70	39.28
OpenIE6	ORIGINAL	<u>50.60</u>	<u>39.15</u>

Table 4: Model comparison on the CaRB dataset. Best values are in bold and second best are underlined.

number of extractions (**#ext**), extractions containing one entity in the subject and object (**#ext1**), containing a “Person” entity in subject and “Organization” entity in the object (**#ext2**), and the number of sentences processed by the model per second (**speed**). `OpenIE6(ECTR)` finds more interesting triples where a conflict of interest relation is expressed between a person and an organization entity, compared to the original `OpenIE6`. Also, our model processes more sentences per second compared to the original `OpenIE6`. This is because `OpenIE6` generates more extractions per sentence, however, even with more extractions, the model retrieves fewer conflicts of interest relations between a person and an organization.

	OpenIE6(ECTR)	OpenIE6
#sen	170298	
#ext	233081	564877
#ext1	138188 (59.29%)	117795 (20.85%)
#ext2	106232 (45.58%)	92152 (16.31%)
speed	87.41	56.14

Table 5: Comparison of extractions from a larger dataset of PubMed conflict of interest statements.

Conclusion. We presented an approach that significantly improves OIE when the input sentence contains entities while being competitive on a standard OIE benchmark. Finally, we showed that our method is much better suited for a real use case, as it extracts high-quality triples from PubMed.

Acknowledgments We thank Rémi Goujot for manually annotating the PubMed dataset used to test our model, during his high school internship. This work was performed using HPC resources from GENCI-IDRIS (Grant 2022-AD011011614R2). The authors were partially funded by the ANR-20-CHIA-0015 project and by the Hi!PARIS Center.

7 Limitations

We identify the following limitations affecting our proposed methods:

- The performance of our models is impacted by the quality of the named entity recognition tool, as well as the performance of the conjunctive model.
- Training OpenIE6 with more constraints requires around 3h/epoch, while the model with the original constraints requires half this time.
- Users trying our tool, but also the original OpenIE model, should have the computational possibility of using the BERT-based model, the main component of OpenIE6. We plan to release trained models based of smaller language models.

References

- Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.
- Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. [Matching the blanks: Distributional similarity for relation learning](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.
- Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. [CaRB: A crowdsourced benchmark for open IE](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6262–6267, Hong Kong, China. Association for Computational Linguistics.
- Nikita Bhutani, Yoshihiko Suhara, Wang-Chiew Tan, Alon Halevy, and H. V. Jagadish. 2019. [Open information extraction from question-answer pairs](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2294–2305, Minneapolis, Minnesota. Association for Computational Linguistics.
- Lei Cui, Furu Wei, and Ming Zhou. 2018. [Neural open information extraction](#).
- Luciano Del Corro and Rainer Gemulla. 2013. [Clausie: clause-based open information extraction](#). In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). *arXiv preprint arXiv:1810.04805*.
- Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. [Open information extraction from the web](#). *Communications of the ACM*, 51(12):68–74.
- Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. [MinIE: Minimizing facts in open information extraction](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark. Association for Computational Linguistics.
- Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Goran Glavas, and Mathias Niepert. 2021. [Benchie: Open information extraction evaluation based on facts, not tokens](#). *arXiv preprint arXiv:2109.06850*.
- Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. [More data, more relations, more context and more openness: A review and outlook for relation extraction](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758, Suzhou, China. Association for Computational Linguistics.
- Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. [FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.
- Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip Yu. 2020. [SelfORE: Self-supervised relational feature learning for open relation extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3673–3682, Online. Association for Computational Linguistics.
- Jing Jiang and ChengXiang Zhai. 2007. [A systematic exploration of the feature space for relation extraction](#). In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120, Rochester, New York. Association for Computational Linguistics.

- Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Mausam, and Soumen Chakrabarti. 2020a. [Openie6: Iterative grid labeling and coordination analysis for open information extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3748–3761. Association for Computational Linguistics.
- Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020b. [IMOJIE: Iterative memory-based joint open information extraction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5871–5886, Online. Association for Computational Linguistics.
- William Lechelle, Fabrizio Gotti, and Phillippe Langlais. 2019. [WiRe57 : A fine-grained benchmark for open information extraction](#). In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 6–15, Florence, Italy. Association for Computational Linguistics.
- Jay Yoon Lee, Sanket Vaibhav Mehta, Michael Wick, Jean-Baptiste Tristan, and Jaime Carbonell. 2019. Gradient-based inference for networks with output constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4147–4154.
- Xiaolu Lu, Soumajit Pramanik, Rishiraj Saha Roy, Abdalghani Abujabal, Yafang Wang, and Gerhard Weikum. 2019. [Answering complex questions by joining multi-document evidence with quasi knowledge graphs](#). SIGIR’19, page 105–114, New York, NY, USA. Association for Computing Machinery.
- Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. [Effectiveness and efficiency of open relation extraction](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Seattle, Washington, USA. Association for Computational Linguistics.
- Yatin Nandwani, Abhishek Pathak, Parag Singla, et al. 2019. A primal dual formulation for deep learning with constraints. In *Advances in Neural Information Processing Systems*, pages 12157–12168.
- Dat P. T. Nguyen, Yutaka Matsuo, and Mitsuru Ishizuka. 2007. Relation extraction from wikipedia using subtree mining. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2, AAAI’07*, page 1414–1420. AAAI Press.
- N. Oreskes and E.M. Conway. 2010. *Merchants of Doubt: How a Handful of Scientists Obscured the Truth on Issues from Tobacco Smoke to Global Warming*. Bloomsbury Publishing.
- Arpita Roy, Youngja Park, Taesung Lee, and Shimei Pan. 2019. [Supervising unsupervised open information extraction models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 728–737, Hong Kong, China. Association for Computational Linguistics.
- Swarnadeep Saha and Mausam. 2018. [Open information extraction from conjunctive sentences](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. [Supervised open information extraction](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition](#). In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2021. Zero-shot information extraction as a unified text-to-triple translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1225–1238.
- Ralph M. Weischedel, Eduard H. Hovy, Mitchell P. Marcus, and Martha Palmer. 2017. Ontonotes : A large training corpus for enhanced processing.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. [Structured relation discovery using generative models](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466, Edinburgh, Scotland, UK. Association for Computational Linguistics.
- Dian Yu, Lifu Huang, and Heng Ji. 2017. [Open relation extraction and grounding](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 854–864, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Junlang Zhan and Hai Zhao. 2020. Span model for open information extraction on accurate corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9523–9530.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. [Position-aware attention and supervised data improve slot filling](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Hierarchical3D Adapters for Long Video-to-text Summarization

Pinelopi Papalampidi* Mirella Lapata

Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
p.papalampidi@sms.ed.ac.uk, mlap@inf.ed.ac.uk

Abstract

In this paper, we focus on video-to-text summarization and investigate how to best utilize multimodal information for summarizing long inputs (e.g., an hour-long TV show) into long outputs (e.g., a multi-sentence summary). We extend SummScreen (Chen et al., 2022), a dialogue summarization dataset consisting of transcripts of TV episodes with reference summaries, and create a multimodal variant by collecting corresponding full-length videos. We incorporate multimodal information into a pre-trained textual summarizer efficiently using adapter modules augmented with a hierarchical structure while tuning only 3.8% of model parameters. Our experiments demonstrate that multimodal adapters outperform more memory-heavy and fully fine-tuned textual summarization methods.

1 Introduction

What happens in the very last episode of “Friends”? Anyone who has seen this episode can summarize its key moments: Ross confesses his love for Rachel, they decide to resume their relationship, while Monica and Chandler adopt twins and move to the suburbs. TV viewers can naturally perform this dialogue summarization task having access to multiple modalities: they not only hear the actors speak but also see their expressions, actions, and whereabouts on screen.

Despite recent advances in summarization (Nalapaty et al., 2016; See et al., 2017; Liu and Lapata, 2019b) and increasing interest in different types of dialogue summarization, e.g., meeting transcripts (Gliwa et al., 2019; Zhong et al., 2021) or screenplays (Chen et al., 2022), the contribution of modalities other than text remains relatively understudied. This is not entirely surprising given the challenges associated with the multimodal summarization task

illustrated above (e.g., produce a written summary of a TV episode). Firstly, the input is long, it cannot fit into standard sequence-to-sequence architectures, and the different modalities have to be somehow combined; secondly, the output is also long, summaries consist of multiple sentences and rich vocabulary; and thirdly, it involves complex inference over long-range dependencies between events and characters and common sense reasoning. At the same time, creating large-scale multimodal datasets with long videos and aligned textual data is challenging and time consuming, limiting the research conducted in this domain.

Previous work on video-to-video summarization identifies highlights from YouTube videos, TV shows, or movies (Song et al., 2015; Gygli et al., 2014; De Avila et al., 2011; Papalampidi et al., 2021b). However, in most cases, either the videos are short or the datasets are small with a few hundred examples. There is also limited work on video-to-text summarization. We are only aware of one large-scale multimodal dataset for this task, namely How2 (Sanabria et al., 2018), which again contains short videos (i.e., 2–3 minutes long) with simple semantics, and short, single-sentence summaries.

In this paper, we focus on video-to-text summarization and investigate how to best utilize multimodal information for condensing long inputs (e.g., an hour-long TV show) into long outputs (e.g., a multi-sentence summary). We create a multimodal variant of SummScreen (Chen et al., 2022), a recently released dataset comprising of transcripts of TV episodes and their summaries. We collect full-length videos for 4,575 episodes and multiple reference summaries. We build our model on top of a pre-trained sequence-to-sequence architecture (i.e., BART; Lewis et al. 2020) fine-tuned on summarization and capable of generating fluent long text. We convert its textual encoder to a multimodal one by adding and tuning adapter layers (Rebuffi et al., 2017; Houlisby et al., 2019),

*Now at DeepMind.

	Modality	Input	Output	Datasets
text-to-text	text	short	short	XSum (Narayan et al., 2018), CNN-DailyMail (Nallapati et al., 2016), NYT (Durrett et al., 2016), Gigaword (Napoles et al., 2012)
	text	long	long	SamSum (Gliwa et al., 2019), QMSum (Zhong et al., 2021), SummScreen (Chen et al., 2022)
video-to-video	vision	short	short	OVP (De Avila et al., 2011), YouTube (De Avila et al., 2011), SumMe (Gygli et al., 2014)
	vision/text	short	short	TVSum (Song et al., 2015)
	vision/text(/audio)	long	long	LoL (Fu et al., 2017) TRIPOD+ (Papalampidi et al., 2021b)
video-to-text	vision	long	short	TACoS (Rohrbach et al., 2014)
	vision/text/audio	short	short	How2 (Sanabria et al., 2018)
	vision/text/audio	long	long	SummScreen ^{3D}

Table 1: Datasets used for summarization grouped based on the input/output modalities and input/output length. A more detailed comparison and statistics for video-to-text datasets can be found in Appendix A (Table 10).

which only account for 3.8% of model parameters. We also explore strategies for *content selection*, since the input is too long to fit into standard sequence-to-sequence models. Empirical results across evaluation metrics demonstrate that multimodal information yields superior performance over just text, both in terms of content selection and summarization; this is the case even when our adapter model is compared to fully fine-tuned approaches and more memory-heavy architectures (e.g., Longformer; Beltagy et al. 2020) that can process the entire input.

Our contributions can be summarized as follows: (1) we augment SummScreen (Chen et al., 2022) with multimodal information, providing videos aligned with transcripts and summaries; to the best of our knowledge, this constitutes the largest available resource for long video multimodal summarization; (2) we propose a *parameter efficient* approach to augment a pre-trained textual summarizer with multimodal information; and (3) explore different methods for identifying salient moments in a long video and show that multimodal information also improves content selection.

2 Related Work

Video Summarization Much previous work has focused on text-to-text or video-to-video summarization. We provide a comprehensive categorization of existing datasets according to input/output length and modality in Table 1. *Multimodal abstractive summarization* (video-to-text) has attracted less attention, mainly due to the difficulty of collecting large-scale datasets. How2 (Sanabria et al., 2018) is the only publicly available benchmark for this task, it includes short instructional videos with textual transcripts and one-sentence summaries. We generate multiple-sentence sum-

maries from long videos and their transcripts. While previous approaches have focused on various modality fusion methods with small RNN-based models (Palaskar et al., 2019), we take advantage of large pre-trained LMs (Lewis et al., 2020; Raffel et al., 2020; Radford et al., 2019) for generating fluent text summaries.

Recent years have also witnessed increasing interest in multimodal video captioning, a task related to multimodal summarization, which aims to generate one-sentence descriptions for localized events in short videos (Xu et al., 2016; Rohrbach et al., 2017; Zhou et al., 2018; Lei et al., 2020b). Existing methods employ strong language-and-vision encoders with massive pre-training (Li et al., 2020; Luo et al., 2020; Xu et al., 2021; Lei et al., 2020a; Li et al., 2021), while the decoder is typically shallow and under-trained.

Realizing the importance of large LMs for generation, recent work has focused on how to efficiently render pre-trained LMs multimodal. Notably, Tsimpoukelli et al. (2021) convert a pre-trained LM into an image captioning model, by giving images as prompts and training only a vision encoder. Yu et al. (2021) summarize How2 videos by augmenting BART with visual information via a *new cross-attention block* added to every encoder layer. However, their approach adds a very large number of *new parameters* and requires full fine-tuning, which leads to overfitting in our case when the dataset size is small.

Dialogue Summarization In the context of text-to-text generation, dialogue summarization is challenging due to the difficulty of fitting very long input into pre-trained sequence-to-sequence models. Longformer (Beltagy et al., 2020) alleviates this by employing local self-attention in combination

Episodes	4,575	
Input (transcript + video + audio)		
Shots	1,048,024	
Shots/episode	193.64 (109.09)	
Utterances/episode	322.76 (116.52)	
Tokens/episode	5720.55 (2223.38)	
Output (summaries)		
Summaries/episode	1.53	(0.79)
TVMegaSite/#tokens	4,280	395.69 (275.84)
YouTube/#tokens	334	136.22 (45.12)
IMDb/#tokens	946	111.21 (82.18)
tvdb/#tokens	1,454	126.14 (82.14)
Training (unique input-output pairs)	5,199	
Validation episodes	296	
Testing episodes	296	

Table 2: SummScreen^{3D} statistics. For summaries, we show their provenance, number of summaries per site (second column), and mean number of tokens per summary; standard deviations are shown in parentheses.

with global tokens for reducing the computational overhead. Despite recent attempts to make self-attention more efficient (Kitaev et al., 2020; Tay et al., 2020; Zaheer et al., 2020), it is still unclear whether it has an advantage over content selection with a full-attention mechanism (Zhang et al., 2021; Shaham et al., 2022) for long dialogue summarization. Zhong et al. (2022) incorporate dialogue-specific objectives for pre-training summarization models, while Zhang et al. (2022) hierarchically summarize the input chunk-by-chunk.

Parameter-efficient Tuning Fine-tuning is a common approach for transferring pre-trained models to different tasks or domains (Howard and Ruder, 2018). It is customary to fine-tune all the parameters of the pretrained model which, however, becomes prohibitive as model size and number of tasks grow. Recent work has proposed parameter-efficient transfer learning methods which fine-tune only a small number of *additional* parameters. Two popular approaches include *adapter tuning*, where bottleneck layers are added and tuned at every layer of the model (Rebuffi et al., 2017; Houlsby et al., 2019) and *prompt tuning*, where (soft) prompts are prepended as part of the input (Brown et al., 2020; Li and Liang, 2021). In this work, we utilize the former method for adapting a textual summarizer to our multimodal setting and dialogue input format.

3 The SummScreen^{3D} Dataset

SummScreen (Chen et al., 2022) is a long dialogue summarization dataset¹ containing transcripts from

¹<https://github.com/mingdachen/SummScreen>

TV episodes and human-written abstractive summaries. We extend this dataset to a multimodal setting by also considering the corresponding full-length videos. SummScreen contains two subsets depending on the series genre: SummScreen-FD and SummScreen-TMS. We use the latter subset which mostly covers soap operas from TVMegaSite², as it is easier to obtain full-length videos and each series has hundreds of episodes.

For each episode in SummScreen-TMS, we automatically search for the title and release date in Youtube. If there is a match with large duration (indicating that this is a full episode rather than a segment), we download the video and closed captions (CC). Overall, we collected videos for 4,575 episodes from five different shows in SummScreen-TMS.³ In addition to TVMegaSite summaries (distributed with SummScreen), we further retrieved summaries from YouTube descriptions, IMDb, and tvdb, again using the episode title and release date as search terms. The statistics of our dataset which we call SummScreen^{3D} (3D for language, video, and audio) are in Table 2 and we provide further details in Appendix A. As can be seen, each episode has (on average) multiple references which vary in length (TVMegaSite summaries are longest).

We split SummScreen^{3D} into training, validation, and test sets with the same distribution over different shows per set. We reserved 296 episodes for validation and the same number for testing, and used the rest for training. Since we have multiple reference summaries for some episodes, we increased the size of the training set by adding m episode-summary pairs, matching the same episode with each of its m references. This resulted in 5,199 unique samples for training.

4 Video-to-Text Summarization

Our approach leverages the generation capabilities of large pre-trained sequence-to-sequence models (Lewis et al., 2020; Raffel et al., 2020). As our backbone model, we employ BART-large (Lewis et al., 2020) which has been fine-tuned on CNN-DailyMail (Nallapati et al., 2016; Zhang et al., 2021) and has thus acquired a summarization inductive bias. As TV show transcripts are very long and cannot fit into BART, we select a subset of utterances (i.e., speaker turns) as input via content

²<http://tvmegasite.net>

³https://github.com/ppapalampidi/long_video_summarization

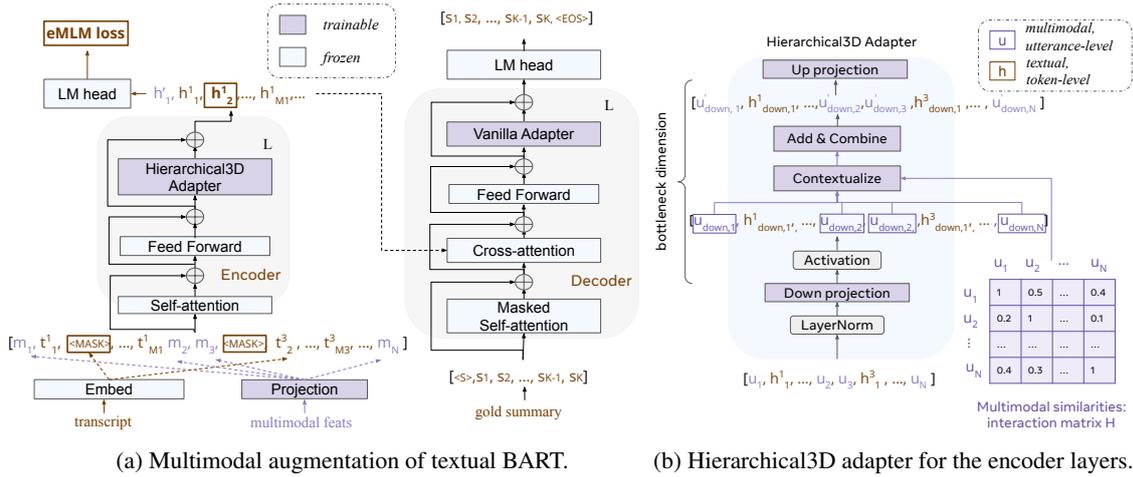


Figure 1: Multimodal augmentation of pre-trained BART. We augment the encoder and decoder layers with adapters which we fine-tune on the target dataset, while the remaining network is frozen. As input, we consider textual tokens and coarse-grained multimodal information which we prepend before each utterance. We also corrupt part of the textual input during training and add an auxiliary MLM loss to the encoder for predicting the corrupted tokens. On the right, we show the hierarchical adapter added to each encoder layer: after down-projecting all representations, we only consider the multimodal ones and further contextualize them via attention. Then, we combine the representations and up-project again to the original model dimension.

selection (see details in Section 5). We transfer this model to our task and domain (i.e., multimodal dialogue summarization), by adding adapter layers (Rebuffi et al., 2017; Hounsby et al., 2019; Sung et al., 2022) in both the encoder and decoder, and tuning them on SummScreen^{3D} while keeping the rest of the network frozen. We briefly discuss below our backbone text-based model and then elaborate on how we incorporate multimodal information.

4.1 Backbone Textual Model

Our summarizer follows a standard sequence-to-sequence Transformer architecture (Vaswani et al., 2017). The encoder maps tokens $[t_1, t_2, \dots, t_N]$ to a sequence of contextualized representations $[h_1, h_2, \dots, h_N]$ which are then fed to the decoder for generating the summary. The encoder consists of L stacked layers, each of which has a self-attention block for contextualizing the token representations, followed by a feed-forward network. The decoder has a similar architecture, it additionally contains a *cross-attention* block for identifying relations between the input and currently generated text and makes use of *masked* self-attention to control access to context for each token. The decoder is followed by a linear layer (i.e., Language Model (LM) head) which projects the output representations onto the vocabulary and a final softmax layer. The model is optimized for predicting the next token s_{t+1} in the summary given $[s_0, s_1, \dots, s_t]$, the context generated so far, and the transcript $[t_1, t_2, \dots, t_N]$.

4.2 Multimodal Augmentation

Our hypothesis is that adding multimodal information to a textual summarizer (i.e., converting the textual encoder to a multimodal one) will increase the quality of its output summaries. We expect that the video/audio will compensate for important non-verbal information typically absent from the transcript (e.g., who is speaking to whom, who is present in the same room, who is crying or yelling). We further expect multimodal information to make up for the loss of context incurred by content selection. We next describe how we compute multimodal representations for an episode and how we augment BART with these representations.

Multimodal Representations We use *utterances* as the unit of representation for multimodal information. We segment episodes into shots (using PySceneDetect⁴) and map these to utterances in the corresponding transcript. Specifically, we align the closed captions in the video which are time-stamped to the utterances in the transcript using Dynamic Time Warping (DTW; Myers and Rabiner 1981; Papalampidi et al. 2021b). We thus create a one-to-many alignment where an utterance corresponds to one or more shots. For each shot, we extract textual, visual, and audio features (see Appendix B.1 for details), and compute an utterance-level representation for each modality by average pooling over all aligned shots.

Given textual x_i , visual v_i , and audio a_i repre-

⁴<https://github.com/Breakthrough/PySceneDetect>

sentations for utterance i , we learn a multimodal representation as part of our network:

$$\begin{aligned} x'_i &= f(W_x x_i) \quad v'_i = f(W_v v_i) \quad a'_i = f(W_a a_i) \\ m_i &= f(W_m [x'_i; v'_i; a'_i]) \end{aligned} \quad (1)$$

where $f(\cdot)$ is the ReLU activation function, $[\cdot; \cdot; \cdot]$ denotes concatenation, $W_x \in \mathbb{R}^{d_x \times d_i}$, $W_v \in \mathbb{R}^{d_v \times d_i}$, $W_a \in \mathbb{R}^{d_a \times d_i}$, and $W_m \in \mathbb{R}^{3d_i \times d_m}$ are learnable matrices; d_i and d_m are the input and model dimensions with $d_i \ll d_m$, and m_i is the final multimodal representation corresponding to the i^{th} utterance in the transcript.

Multimodal Encoder In order to integrate utterance-level multimodal representations with BART, we consider a “global utterance token” inspired by the Longformer architecture (Beltagy et al., 2020). We preprocess the input into utterances and prepend a global token $\langle \text{EOS} \rangle$ per utterance as a placeholder for multimodal representations. The encoder thus receives as input sequence $[\mathbf{m}_1, t_1^1, t_2^1, \dots, t_{M_1}^1, \dots, \mathbf{m}_N, t_1^N, t_2^N, \dots, t_{M_N}^N]$ where, “global” representations \mathbf{m} constitute a rich multimodal space (i.e., they are not learned solely from text via local self-attention; Figure 1a).

4.3 Self-supervised Auxiliary Guidance

Our primary loss for training the model described above is the negative log likelihood of predicting the next token in the summary given episode \mathcal{E} :

$$L_{LM} = \frac{1}{K} \sum_{t \in [1, K]} -\log p(s_t | s < t; \mathcal{E}) \quad (2)$$

We further wish to encourage the model to attend to multimodal information and learn a meaningful projection (Equation (1)). To do this, we corrupt part of the textual input by masking tokens (see bottom left part of Figure 1a) and adding an auxiliary masked language modeling (MLM) loss for the initial training steps only. So as not to disrupt the bias of the decoder, which is already trained on textual summarization, we apply the MLM loss in the outputs of the encoder while the model is trained on the downstream task. Given token-level encoder outputs $[h_1, h_2, \dots, h_N]$, we copy and re-use the LM head of the decoder in order to project them into the vocabulary (see top left part of Figure 1a). And compute the negative log likelihood only for the set of masked tokens \mathcal{M} :

$$\mathcal{L}_{eMLM} = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} -\log p(t | h_{t_i \notin \mathcal{M}}) \quad (3)$$

We refer to this loss as encoder-based MLM loss (eMLM; Baziotis et al. 2021). It trains the encoder to reconstruct input text representations while attending to multimodal information. After X initial training steps, we drop the auxiliary loss and stop corrupting the textual input in order for the model to be optimized on summarization. We use a mixture of whole utterance corruption (Zhang et al., 2020a; Zhong et al., 2022) and content word corruption, masking out named entities, nouns, and verbs excluding auxiliaries (see Section 6).

4.4 Hierarchical3D Adapters

We specialize BART for our multimodal summarization task by inserting adapter modules (Rebuffi et al., 2017; Houlsby et al., 2019) into each encoder and decoder layer (after the feed-forward block). Each adapter adds only a small number of new parameters, which are randomly initialized and tuned on our end task, while the rest of the network is frozen. A vanilla adapter takes as input hidden representations $[\mathbf{u}_1, h_1^1, h_2^1, \dots, \mathbf{u}_N, \dots, h_{M_N}^N]$, where $h_1^1, h_2^1, \dots, h_{M_N}^N$ are textual token-level hidden representations and $\mathbf{u}_1, \dots, \mathbf{u}_N$ are multimodal utterance-level hidden representations (in accordance to the input format presented in Figure 1a), and performs the following transformations:

$$h_{down,i} = f(\text{LN}(W_d h_i + b_d)) \quad (4)$$

$$h_{up,i} = W_u h_{down,i} + b_u \quad h_i = h_i + h_{up,i} \quad (5)$$

where $W_d \in \mathbb{R}^{d_m \times d_B}$, d_m is the model dimension, d_B is the bottleneck dimension of the adapter, $f(\cdot)$ is a non-linearity, LN a trainable layer normalization, $W_u \in \mathbb{R}^{d_B \times d_m}$, b_d , and b_u are the corresponding bias vectors, and $h_{down,i}$ and $h_{up,i}$ are down and up projections of h_i .

In this work, we augment the vanilla adapters of the *encoder* with a hierarchical structure (illustrated in Figure 1b). After computing (low level) self-attention between all input *textual tokens* in an encoder layer, we add a hierarchical adapter to compute *higher-level interactions* between *utterance-level multimodal* representations. By including this interaction block in the adapter, we can better propagate long-range dependencies between utterances and enforce a more global view of the events in an episode and their associations, while keeping the number of trainable parameters low.

Using the scaled dot product, we compute interaction (aka similarity) matrix H between utter-

ances (see Figure 1b) based on their *multimodal representations* $[m_1, m_2, \dots, m_N]$:

$$e_{ij} = (W_i m_i + b_i)(W_j m_j + b_j) / \sqrt{d_m} \quad (6)$$

where W_i, W_j are learnable projection matrices, d_m is the model dimension, and e_{ij} is the degree of similarity between m_i and m_j .

At each adapter layer of the encoder, after down-projecting all vectors to the bottleneck dimension, we further contextualize utterance-level multimodal representations $u_{down,i}$ with respect to each other given the degree of similarity provided by H (“Contextualize” block in Figure 1b):

$$u'_{down,i} = \sum_{k=1}^N r(H_{ik}/\tau) u_{down,k} + u_{down,i}$$

where N is the number of utterances, $r(\cdot)$ is the softmax function, and τ is a low temperature parameter (< 1) for increasing sparsity. After contextualization, we up-project all vectors to the original dimension d_m , as in vanilla adapters (Equation (5)).

5 Content Selection

As explained earlier, episodes in SummScreen^{3D} are very long ($\sim 5,720$ tokens). BART, which has a maximum token length of 1,024, can approximately encode one fifth of the transcript.⁵ We therefore perform content selection, i.e., identify salient utterances and give these as input to BART. We describe below three approaches inspired by information retrieval, summarization (Gehrmann et al., 2018; Liu and Lapata, 2019a), and computational narrative analysis (Papalampidi et al., 2021b,a).

Retrieval-based Selection We follow previous approaches (Zhang et al., 2021) in determining salient content with BM25 (Robertson and Zaragoza, 2009). BM25 is a widely known retrieval model similar to tf*idf. It assigns each utterance a “relevance” score (by comparing it against the entire transcript). Utterances with high scores are deemed salient and the K best ones are selected.

Learning-based Selection Alternatively, we may also model content selection as a binary classification problem. Given a transcript containing N utterances we predict whether each should be selected as input for the downstream summarization task (label 1) or not (label 0). We create noisy

⁵We can extend positional embeddings to 1,536 by applying bilinear interpolation, however, the memory requirements would still be prohibitive for longer sequences.

labels by matching transcript utterances to (reference) summary sentences. Specifically, we encode sentences and utterances via Sentence-BERT (Reimers and Gurevych, 2019), and assign a positive label to the utterances most similar to the reference sentences. A content selector is then trained on these pseudo-labels to identify salient utterances. We can also incorporate multimodal information in this content selection setting, using the same utterance-level representations fed into BART. We first contextualize them via a shallow transformer encoder, and add a classification head for predicting important utterances. The model is optimized with binary cross-entropy loss. During inference we select the top K predicted utterances.

Turning Point Identification We also perform content selection based on a Turning Point (TP) identification model (Papalampidi et al., 2021b,a) pre-trained on the TRIPOD movie dataset (Papalampidi et al., 2019). TPs are key events in narratives; they are distinguished into five different types depending on their functionality (e.g., Opportunity, Change of Plans, Point of No Return, Major Setback, Climax). The TP identification model considers the same multimodal information as the content selector above and identifies utterances that represent each TP. We consider the top $K/5$ predicted utterances per turning point.

6 Experimental Setup

Implementation Details We provide details of the multimodal feature extraction (i.e., utterance-level visual, audio, and textual features) in Appendix B.1. We corrupt the textual input and use the auxiliary eMLM loss (Section 4.3) only for the first $X = 1,500$ training steps; we train our model for a total of 12,000 steps. During corruption, we mask out all content words (i.e., named entities, verbs, and nouns) and a random 10% of the input utterances. For generating summaries during inference, we use beam search with beam = 5 and 3-gram blocking (Paulus et al., 2018). We provide further implementation details in Appendix B.2.

Training vs. Inference Although we experiment with different content selection methods during inference, we randomly sample input utterances during training. Random sampling acts as data augmentation, since the model sees slightly different input-output pairs during training at different iterations. We experimentally verify in Section 7 this

is preferable to a fixed selection of utterances, especially considering the small size of our dataset. We select $K = 60$ utterances to feed into BART models given the input length limit, and order them according to their original position in the transcript.

Evaluation Metrics We evaluate the generated summaries using ROUGE F1 (Lin, 2004) against reference summaries.⁶ Since ROUGE is not always a good indicator of summary quality and does not discriminate between different error types (e.g., factuality vs. fluency), we consider additional metrics based on Question-Answering (QA).⁷ We obtain questions based on gold summaries and evaluate whether the correct answers exist in the generated summaries. We expect factual summaries to answer a high percentage of questions.

As in previous work (Maynez et al., 2020; Kryscinski et al., 2020; Honovich et al., 2021), we automatically generate QA pairs against reference summaries. We identify named entities and nouns using spaCy (Honnibal and Montani, 2017), and feed them as gold answers alongside the summaries to a question generator. We discriminate between named entities and nouns as answer types for measuring factuality in event-entity associations and other attributes pertaining to nouns. We used T5-base (Raffel et al., 2020) as our question generator and RoBERTa-base (Liu et al., 2019) as the QA system for answering questions given system generated summaries as input passages. Both were fine-tuned on SQuAD2.0 (Rajpurkar et al., 2016).

We measure accuracy as the partial overlap between gold and predicted answers for named entities. For nouns, we resort to textual entailment in order to account for synonyms and paraphrases in the generated summaries. We concatenate the question with gold or generated answer and predict a score for the directional relation between them. If the score is above 0.5, we consider the generated answer correct. We used BART-large (Lewis et al., 2020) fine-tuned on the MultiNLI corpus (Williams et al., 2018) as our entailment model.

We created a test suite of gold QA pairs, by retaining only those that can be answered correctly by the QA model given the reference summaries (Honovich et al., 2021). We overall generated 2,513 questions for named entities and 381 questions for

⁶<https://pypi.org/project/py-rouge/>

⁷We also experimented with BERTScore (Zhang et al., 2020b) but observed no discernible performance differences between any pair of models.

Selection	R-2		R-L	
	text	+H-3D	text	+H-3D
Lead	6.51	—	30.72	—
Last	6.41	—	30.59	—
Middle	6.70	—	31.03	—
Random	6.54	7.24	30.91	32.15
Retrieval	6.30	6.89	30.20	31.42
TP identification	6.78	7.36	31.24	32.01
Learned selection	6.74	7.62	31.22	32.64
Pseudo-oracle	7.96	8.42	32.85	33.40

Table 3: Content selection methods for text-only BART and our multimodal Hierarchical3D variant (H-3D).

nouns for the 296 episodes in our test set. On average, we have 8.5 questions per episode for named entities and 2.3 questions for nouns.⁸

7 Results

Content Selection Table 3 compares how different approaches to content selection influence summarization performance according to ROUGE F1. We compare some simple baselines like selecting the Lead, Middle, and Last 60 utterances from the transcript as well as at Random. In addition, we compare a text only summarizer against our Hierarchical3D model. Differences amongst content selection methods are generally small. BM25 performs worse than random whilst a multimodal content selector trained on pseudo-labels performs overall best. As an upper bound, we also report results with oracle labels as input demonstrating that there is still room for improvement.

Regardless of how content is selected, we observe that our Hierarchical3D variant significantly improves performance, and interestingly, the performance gap is larger when the selection method is weaker (e.g., random vs. pseudo-oracle). This indicates that to a certain extent multimodal information makes up for suboptimal content selection.

Text vs. Multiple Modalities In Table 4 we compare our multimodal model (with the best performing content selector) against textual summarizers developed for processing long input or specifically for dialogue summarization. These include Longformer (LED; Beltagy et al. 2020) with full fine-tuning¹⁰, a variant of LED pre-trained on

⁸We release our test suite of gold QA pairs together with the SummScreen^{3D} corpus.

⁹Textual summarizers are initialized with the same checkpoint, while some models are further tuned (e.g., DialogLED).

¹⁰Adding (and tuning) adapter layers in LED led to significantly inferior performance, which in turn suggests that adapting such a network is not straightforward.

Models	R-1	R-2	R-L
HERO FT	21.56	1.74	21.27
Summ ^N FT	24.71	4.42	22.61
LED FT	33.53	7.60	31.77
DialogLED FT	32.66	7.38	31.12
BART FT	32.61	6.94	30.83
BART AT	33.27	6.74	31.22
BART AT + H-3D	34.51	7.62	32.64

Table 4: Comparison of our model (BART AT + H-3D) with a video captioning model (i.e., HERO) and text-only summarizers for long dialogue summarization⁹. For HERO and all BART variants we perform content selection (FT: full fine-tuning, AT: adapter-tuning).

Models	Acc (NEs)		Acc (NNs)	
	text	+H-3D	text	+H-3D
LED FT	20.89	—	37.95	—
DialogLED FT	21.09	—	36.22	—
Summ ^N FT	18.03	—	34.91	—
Random	20.25	23.64	33.86	38.06
TP identification	21.65	24.07	40.42	40.68
Learned selection	20.65	24.71	38.58	39.37
Pseudo-oracle	28.53	29.64	41.73	42.00

Table 5: QA evaluation (test set) on named entities (NEs) and nouns (NNs). We denote our Hierarchical3D model with H-3D.

dialogues (DialogLED; Zhong et al. 2022), and Summ^N (Zhang et al., 2022), a two-stage hierarchical approach for long dialogue summarization. We also present text-only BART variants, with full fine-tuning (FT) and adapter-tuning (AT). Finally, we include a SOTA video-to-text model (HERO; Li et al. 2020) with a massively pre-trained encoder, which is tuned on another TV dataset for video captioning of short clips (i.e., TVC; Lei et al. 2020b).

As can be seen in the second block of Table 4, tuning only the adapter layers (BART AT) does not hurt performance compared to full fine-tuning (BART FT), presumably due to the small dataset size. Addition of multimodal information with hierarchical adapters (BART AT + Hierarchical3D) yields substantial ROUGE improvements. Interestingly, our performance is superior to fully fine-tuned, memory-heavy models like LED or DialogLED that process the entire transcript as input. This suggests that representations from multiple modalities are more informative and lead to higher performance compared to efficient self-attention mechanisms. Summ^N performs demonstrably worse than one-stage methods and HERO fails to produce long fluent outputs due to the shallow under-trained decoder and small dataset size.

Modality	R-1	R-2	R-L
Text	34.74	7.11	32.46
Audio	33.95	6.92	31.90
Video	34.86	7.24	32.73
Multimodal	34.95	7.51	33.01
w/ vanilla adapters	34.25	7.45	32.41
w/o eMLM loss	33.80	6.84	31.88
w/o random augmentation	33.45	6.48	31.81

Table 6: The role of multimodal information and hierarchical adapters (validation set).

QA Evaluation The results of our automatic QA evaluation are summarized in Table 5. The second block focuses on model performance with different content selection variants. We only compare text-only and multimodal (+H-3D) BART. Again, we find that augmenting BART with multimodal information regardless of the selection method improves accuracy, especially for named entities. This is true even when content is selected by a pseudo-oracle suggesting that multimodal information provides better associations between events and entities, even when the input contains all salient information. We further observe that supervised content selection and TP identification offer the best performance. The first block reports the performance of state-of-the-art models on dialogue summarization; we find these models perform on par or slightly worse than textual BART (depending on the content selection method) which casts doubts on their ability to efficiently consume longer inputs. Examples of output summaries (and QA pairs) are given in Table 7 and Appendix C.3. We also report additional (entity-specific) results in Appendix C.2.

Ablation Studies In Table 6 we summarize our ablation studies which isolate the contribution of individual modeling components. We observe that individual modalities (Text, Audio, Video) are worse on their own than in combination (Multimodal). The least informative modality is audio, while the most informative is video. In the multimodal case, we substitute hierarchical adapters in the encoder with vanilla adapters and observe a small drop in performance. Removing the auxiliary eMLM loss during training further decreases performance. The auxiliary loss is crucial for rendering the textual encoder multimodal and forcing an already tuned summarizer to consider a different type of input. Finally, data augmentation (via random content selection) during training is also important given the small size of our dataset and BART encoder length restrictions. We report additional ablation

Model	Summary
Gold	<p>Joshua tells Elizabeth he wants to turn Allison and demands she help ease Allison into her new life as his wife. Elizabeth tells Joshua she will kill him before she allows him to hurt Allison. Livvie is able to fend off her need to feed while she and Caleb make love. Frank searches for Allison. When Frank attempts to kidnap Allison from Rafe, he discovers that it really is Lucy and Ian in disguise. Allison and Rafe reappear in Caleb's cave.</p> <ul style="list-style-type: none"> • Who does Frank try to kidnap Allison from? Rafe • Who does Frank try to kidnap? Allison • Who tries to kidnap Allison? Frank
QA pairs	<ul style="list-style-type: none"> • Who can fend off her need to feed while she and Caleb make love? Livvie • Who tells Joshua she will kill him before she allows him to hurt Allison? Elizabeth • Who tells Elizabeth he wants to turn Allison into his wife? Joshua • What is Allison's new life? wife
Text-only	<p>Rafe tells Allison that he will never let Joshua take her for his bride, but she tells him that she has no choice in the matter. Elizabeth tells Joshua that she will not stand by and allow him to take her daughter. Joshua tells Elizabeth that he is going to ease Allison into her new lifestyle as his wife. Elizabeth says that she is not going to let her daughter suffer the kind of nightmare that she lived. She will kill Joshua before he is even that close to turning her. Allison tells Rafe that she thinks this is a little extreme, that is all. Rafe says he will not let Joshua get to her. He promises to keep her away from Joshua and all his goons. Caleb tells Livvie that she doesn't need to feed. He tells her that he can't make love to her the way she wants to. She tells him she can't turn him back, but he tells her he can. He says that he loves her and that he wants to make her his bride.</p> <ul style="list-style-type: none"> • Who does Frank try to kidnap Allison from? Joshua • Who does Frank try to kidnap? Joshua • Who tries to kidnap Allison? Rafe
QA pairs	<ul style="list-style-type: none"> • Who can fend off her need to feed while she and Caleb make love? Livvie • Who tells Joshua she will kill him before she allows him to hurt Allison? Elizabeth • Who tells Elizabeth he wants to turn Allison into his wife? Joshua • What is Allison's new life? wife
H-3D	<p>Livvie tries to convince Caleb to let her take the risk of biting him, but she is afraid that she won't be able to do it. Joshua tells Elizabeth that he wants Allison to be his bride. Elizabeth is shocked when she finds out that Joshua wants to take Allison away from Rafe. Elizabeth tells Joshua that she will find a way to stop him from taking Allison. Rafe tells Allison that he has a vision of a city of tortured souls, a master vampire race, and his vampire bride. He tells her that he can make a perfect bride for her. Allison tells Rafe that she doesn't want to leave her family, but Rafe assures her that she is not going to leave them. Frank tells Ian that he is going to have to tell his boss that his mission didn't work.</p> <ul style="list-style-type: none"> • Who does Frank try to kidnap Allison from? Rafe • Who does Frank try to kidnap? Allison • Who tries to kidnap Allison? Rafe
QA pairs	<ul style="list-style-type: none"> • Who can fend off her need to feed while she and Caleb make love? Livvie • Who tells Joshua she will kill him before she allows him to hurt Allison? Elizabeth • Who tells Elizabeth he wants to turn Allison into his wife? Joshua • What is Allison's new life? vampire bride

Table 7: Examples of gold and model generated summaries together with automatically generated questions and their answers based on gold and automatic summaries. Correct/wrong answers are in green/red color. We show output for a text-only BART model and a multimodal variant with hierarchical adapters (H-3D); in both cases content selection is performed with a model trained on pseudo-labels.

experiments on content selection in Appendix C.1.

8 Conclusions

In this work, we addressed the task of multimodal abstractive summarization and created SummScreen^{3D}, a new dataset which we hope will facilitate future research in this direction. We incorporated multimodal information into a pre-trained textual summarizer in a parameter-efficient manner and have experimentally shown performance gains over text-only models. Our experimental results further underscore the importance of (multimodal) content selection compared to approaches focusing

on self-attention variants for long dialogue summarization. In the future, we plan to explore more *structure-aware* representations for *all* input modalities in order to improve factuality (e.g., entity-event associations) in the generated summaries.

Acknowledgements

We thank the anonymous reviewers for their feedback. The authors also thank Marcus Rohrbach and Frank Keller for insightful comments on earlier versions of this paper. We gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (grant EP/W002876/1).

9 Limitations

Our approach considers only coarse-grained (i.e., utterance-level) multimodal information which we demonstrate is beneficial for summarization. More detailed frame-level visual information e.g., person identification and object recognition in frames, would be useful. However, considering frame-level representations for hour-long videos would bring a considerable increase in memory requirements and additional difficulties in aligning different modalities (e.g., frames vs. tokens vs. audio segments). We leave these challenges to future work and believe that structure-aware methods are necessary for addressing the current limitations.

Following previous work (Maynez et al., 2020; Kryscinski et al., 2020; Honovich et al., 2021), we advocate the use of automatic QA-based methods for evaluating the generated summaries. Although there is supportive analysis (e.g., Tang et al. 2022) that shows better correlation to human judgements for QA-based automatic evaluation in comparison with traditional summarization metrics such as ROUGE, more experimentation is necessary to determine the shortcomings of these metrics.

Finally, conducting human evaluation for SummScreen^{3D} is infeasible, since this would entail asking judges to watch 40-minute long episodes in order to evaluate the content and faithfulness of the summaries. We further cannot assume judges are familiar with the characters, specific details and (complex) storylines of different soap operas contained in our test set in order to be able to make reliable judgments. Therefore, using QA-based metrics for judging specific attributes of summarization quality, such as whether the correct entities are linked to the correct events in an episode (i.e., QA evaluation related to named entities), can provide us with useful insights.

References

- Christos Baziotis, Ivan Titov, Alexandra Birch, and Barry Haddow. 2021. Exploring unsupervised pre-training objectives for machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2956–2971, Online. Association for Computational Linguistics.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#). *CoRR*, abs/2004.05150.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33*. Curran Associates, Inc.
- Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. [SummScreen: A dataset for abstractive screenplay summarization](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.
- Sandra Eliza Fontes De Avila, Ana Paula Brandao Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. 2011. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68.
- Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. [Learning-based single-document summarization with compression and anaphoricity constraints](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1998–2008, Berlin, Germany. Association for Computational Linguistics.
- Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. [Slowfast networks for video recognition](#). In *2019 IEEE/CVF International Conference on Computer Vision*, pages 6201–6210, Seoul, Korea (South). IEEE Computer Society.
- Cheng-Yang Fu, Joon Lee, Mohit Bansal, and Alexander Berg. 2017. [Video highlight prediction using audience chat reactions](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 972–978, Copenhagen, Denmark. Association for Computational Linguistics.
- Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. [Bottom-up abstractive summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.
- Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 776–780. IEEE Computer Society.
- Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. [SAMSum corpus: A human-](#)

- annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.
- Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. **Creating summaries from user videos**. In *Proceedings of the 13th European Conference on Computer Vision*, volume 8695 of *Lecture Notes in Computer Science*, pages 505–520, Zurich, Switzerland. Springer.
- Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *Sentometrics Research*, 7(1):411–420.
- Or Honovich, Leshem Choshen, Roei Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. **q^2 : Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. **Parameter-efficient transfer learning for NLP**. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, California, USA. PMLR.
- Jeremy Howard and Sebastian Ruder. 2018. **Universal language model fine-tuning for text classification**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. **Adam: A method for stochastic optimization**. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA*.
- Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. **Reformer: The efficient transformer**. *CoRR*, abs/2001.04451.
- Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. **Evaluating the factual consistency of abstractive text summarization**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.
- Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara Berg, and Mohit Bansal. 2020a. **MART: Memory-augmented recurrent transformer for coherent video paragraph captioning**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2603–2614, Online. Association for Computational Linguistics.
- Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020b. **TVR: A large-scale dataset for video-subtitle moment retrieval**. In *Proceedings of the 16th European Conference on Computer Vision*, pages 447–463. Springer.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. **BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. 2020. **HERO: Hierarchical encoder for Video+Language omni-representation pre-training**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2046–2065, Online. Association for Computational Linguistics.
- Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luowei Zhou, Xin Wang, William Yang Wang, Tamara L. Berg, Mohit Bansal, Jingjing Liu, Lijuan Wang, and Zicheng Liu. 2021. **VALUE: A multi-task benchmark for video-and-language understanding evaluation**. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1*.
- Xiang Lisa Li and Percy Liang. 2021. **Prefix-tuning: Optimizing continuous prompts for generation**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. **ROUGE: A package for automatic evaluation of summaries**. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yang Liu and Mirella Lapata. 2019a. **Hierarchical transformers for multi-document summarization**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.
- Yang Liu and Mirella Lapata. 2019b. **Text summarization with pretrained encoders**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. 2020. [Univilm: A unified video and language pre-training model for multimodal understanding and generation](#). *CoRR*, abs/2002.06353.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Cory S Myers and Lawrence R Rabiner. 1981. A comparative study of several dynamic time-warping algorithms for connected-word recognition. *Bell System Technical Journal*, 60(7):1389–1409.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence RNNs and beyond](#). In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.
- Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. [Annotated Gigaword](#). In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada. Association for Computational Linguistics.
- Shashi Narayan, Shay Cohen, and Mirella Lapata. 2018. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807. Association for Computational Linguistics.
- Shruti Palaskar, Jindřich Libovický, Spandana Gella, and Florian Metze. 2019. [Multimodal abstractive summarization for how2 videos](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6587–6596, Florence, Italy. Association for Computational Linguistics.
- Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2019. [Movie plot analysis via turning point identification](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1707–1717, Hong Kong, China. Association for Computational Linguistics.
- Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2021a. [Film trailer generation via task decomposition](#). *CoRR*, abs/2111.08774.
- Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2021b. [Movie summarization via sparse graph construction](#). In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 13631–13639. AAAI Press.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2018. [A deep reinforced model for abstractive summarization](#). In *Proceedings of the 6th International Conference on Learning Representations, ICLR*, Vancouver, BC, Canada. OpenReview.net.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. [Learning multiple visual domains with residual adapters](#). volume abs/1705.08045.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.
- Stephen E. Robertson and Hugo Zaragoza. 2009. [The probabilistic relevance framework: BM25 and beyond](#). *Foundations and Trends in Information Retrieval*, 3(4):333–389.
- Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annetarie Friedrich, Manfred Pinkal, and Bernt Schiele. 2014. [Coherent multi-sentence video description with variable level of detail](#). In *Proceedings of the 36th German Conference on Pattern Recognition*, volume 8753 of *Lecture Notes in Computer Science*, pages 184–195, Münster, Germany. Springer.
- Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. 2017. Movie description. *International Journal of Computer Vision*, 123(1):94–120.

- Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: A large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL), NIPS*.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.
- Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: standardized comparison over long language sequences. *CoRR*, abs/2201.03533.
- Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. TVSum: Summarizing web videos using titles. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 5179–5187, Boston, MA, USA. IEEE Computer Society.
- Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. VL-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237, New Orleans, LA, USA.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, Las Vegas, NV, USA.
- Liyan Tang, Tanya Goyal, Alexander R Fabbri, Philippe Laban, Jiacheng Xu, Semih Yahvuz, Wojciech Kryściński, Justin F Rousseau, and Greg Durrett. 2022. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. *arXiv preprint arXiv:2205.12854*.
- Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. 2020. Sparse sinkhorn attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9438–9447, Virtual Event. PMLR.
- Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems 34*, pages 200–212. Curran Associates, Inc.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.
- Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Mousmeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. 2021. VLM: task-agnostic video-language model pre-training for video understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4227–4239, Online Event. Association for Computational Linguistics.
- Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5288–5296, Las Vegas, NV, USA.
- Tiezheng Yu, Wenliang Dai, Zihan Liu, and Pascale Fung. 2021. Vision guided generative pre-trained language models for multimodal abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3995–4007, Online and Punta Cana, Dominican Republic.
- Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big Bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia. OpenReview.net.
- Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed H Awadallah, Dragomir Radev, and Rui Zhang. 2022. Summⁿ: A multi-stage summarization framework for long input dialogues and documents. In *Proceedings of the 60th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 1592–1604, Dublin, Ireland. Association for Computational Linguistics.

Yusen Zhang, Ansong Ni, Tao Yu, Rui Zhang, Chenguang Zhu, Budhaditya Deb, Asli Celikyilmaz, Ahmed Hassan Awadallah, and Dragomir Radev. 2021. *An exploratory study on long dialogue summarization: What works and what’s next*. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4426–4433, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. DialogLM: Pre-trained model for long dialogue understanding and summarization. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 11765–11773, Virtual Event. AAAI Press.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online.

Luowei Zhou, Chenliang Xu, and Jason J. Corso. 2018. *Towards automatic learning of procedures from web instructional videos*. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 7590–7598, New Orleans, Louisiana, USA. AAAI Press.

As The World Turns (atwt)	1356
Bold and the Beautiful (bb)	1113
Guiding Light (gl)	836
One Life to Live (oltl)	1118
Port Charles (pc)	501

Table 8: Distribution of different TV shows in the augmented dataset.

	TMS	SummScreen ^{3D}
TV shows	10	5
Episodes	22,503	4,575
min #episodes per show	168	501
max #episodes per show	3,784	1,356
median #episodes per show	1,973.5	1,113
avg #episodes per show	2,250.0	984.8
Utterances/episode	360.8	322.8
Tokens/episode	6,420.7	5,720.6
Summaries/episode	1	1.53
#tokens/summary	327.0	395.7

Table 9: Comparison between the original SummScreen-TMS (Chen et al., 2022) and SummScreen^{3D} which is a subset of the original dataset.

A Dataset Analysis

As mentioned in Section 3, we create a multimodal version of the SummScreen dataset (Chen et al., 2022) by collecting full-length videos of the episodes contained in the original dataset. Overall, we retrieved videos from YouTube for five different TV shows (i.e., soap operas). We present in Table 8 the names of the TV shows and the number of episodes per show. We made sure to have enough episodes from each TV show and maintain the same distribution when splitting the dataset into train, validation, and test. Moreover, we present an example of the input transcript and output summary from SummScreen (Chen et al., 2022) and how we augment the dataset with additional information from the full-length video in Figures 2 and 3.

Next, we also compare the statistics of SummScreen^{3D}, which is a subset of SummScreen-TMS (Chen et al., 2022), with the original dataset in Table 9. Overall, we include episodes from half the TV shows contained in TMS. The number of episodes per TV show in our dataset is more balanced in comparison with the original (see rows 3–6 in Table 9). SummScreen^{3D} has similar input and output statistics per episode to the original dataset (e.g., number of utterances and tokens per transcript and number of tokens per summary). However, we also collect more summaries per episode when available (see Table 8) for creating an augmented training set and a more robust evaluation set.

Finally, we also compare our dataset against other video-to-text summarization datasets in Table 10. TACoS (Rohrbach et al., 2014) and How2 (Sanabria et al., 2018) are the only available multimodal summarization datasets we are aware of. In comparison, our dataset contains much longer videos (on average 40 minutes long), and fairly elaborate textual summaries (instead of short one-sentence descriptions with simple vocabulary).

B Implementation Details

B.1 Dataset Pre-processing

Given full-length video, we extract features for all modalities at the utterance-level as mentioned in Section 4.2. For text, we extract sentence-level features using Sentence-BERT (Reimers and Gurevych, 2019). Each utterance in the transcript is thus represented by a fixed-size vector. For the frames, we extract two types of features: frame-level features using the CLIP visual encoder (Radford et al., 2021) and motion-level features from video clips using Slowfast (Feichtenhofer et al., 2019). We then aggregate frame- and motion-level features to utterance-level given the automatic alignment by mean pooling. Finally, for audio, we use YAMNet pre-trained on the AudioSet-YouTube corpus (Gemmeke et al., 2017) for classifying audio segments into 521 audio classes (e.g., tools, music, explosion); for each audio segment contained in a shot, we extract features from the penultimate layer, and then aggregate representations again to utterance-level via mean pooling.

B.2 Training Details

We used the Adam algorithm (Kingma and Ba, 2015) for optimizing our networks. We trained all models with a learning rate of $3e-5$ for 12k steps using a linear warm-up of 500 steps, followed by inverted squared decay. All BART-based models were trained with batch size of 1 episode on 4 P100 GPUs with 16GB memory and label smoothing (Szegedy et al., 2016) of 0.1. To fine-tune the LED-based models, we used 4 A100 GPUs with 80GB memory. It took approximately 12 hours to fully train each of these models. Fully fine-tuned models have 406M parameters, which are all fine-tuned on the target dataset, whereas our multimodal adapter-augmented model has 421.6M parameters, from which we only train 15.6M parameters (i.e., multimodal projection layer and adapter layers) on the target dataset. This means that we only tune $\sim 3.8\%$

of the model parameters of the fully fine-tuned models. We report the results of a single run for all models following previous work (Chen et al., 2022; Zhong et al., 2022) due to the computational overhead of running some large comparison models. However, we report in Table 11 the average and standard deviation over three runs for BART AT and BART AT + H-3D in order to demonstrate the performance variation of these models.

C Additional Experimental Results

C.1 Ablation Study on Content Selection

In Table 12, we examine the performance of different content selectors. We report precision (Pre), recall (Re), and F1 score of model variants based on pseudo-oracle labels. We first consider selectors which have not been trained with pseudo-oracle labels, such as Random, Retrieval (i.e., BM25) and TP identification (we refer to these approaches as unsupervised). We observe that unsupervised baselines have significantly lower F1 score in comparison with a supervised approach. Interestingly, although TP identification “agrees less” with the pseudo-oracle labels in comparison with BM25, TPs still present competitive performance against the supervised content selector on abstractive textual summarization (e.g., Table 5). Finally, comparing the multimodal supervised content selector with equivalent unimodal models, we observe that the highest performance is achieved by combining all modalities. With respect to unimodal variants, we find that the textual modality is most informative, while using visual or audio cues alone is not enough to predict salient content.

C.2 Entity-specific Evaluation

Chen et al. (2022) propose a set of entity-specific metrics in order to investigate the role of characters, which are fundamental in TV shows, in the generated summaries. Specifically, they measure several bag of character (BoC) metrics based on character overlap between generated and gold standard summaries. They define precision as the fraction of correctly mentioned characters with respect to all characters that appear in the generated summary (BoC-p) and recall as the fraction of correctly mentioned characters with respect to all characters that appear in the gold summary (BoC-r). Given precision and recall, we also measure F1-score (BoC-f1).

Apart from correctly mentioned characters, Chen

	dataset size	video input	text input	video duration	output tokens
TACoS	147	✓	✗	4.5 minutes	9
How2	79k	✓	✓	90 seconds	20
SummScreen ^{3D}	4.5k	✓	✓	40 minutes	290

Table 10: Comparison between SummScreen^{3D} and other video-to-text summarization datasets (see Table 1).

	R-2	R-L
BART AT	6.71 (0.02)	30.96 (0.23)
BART AT + H-3D	7.58 (0.03)	7.58 (0.03)

Table 11: Results of two models from Table 3 across three different runs. We report the average and standard deviation in parentheses for R-2 and R-L.

Unsupervised	Precision (%)	Recall (%)	F1 (%)
Random	19.55	20.90	20.06
Retrieval	24.63	26.62	25.40
TP identification	20.35	22.10	21.04

Supervised	Precision (%)	Recall (%)	F1 (%)
Multimodal	47.57	50.68	48.57
Text	45.26	48.54	46.52
Vision	22.97	24.91	23.73
Audio	21.54	23.29	22.23

Table 12: The role of multimodal information in content selection. We report the Precision, Recall, and F1 for selecting important utterances from an episode. Supervised models are trained on pseudo-oracle labels.

et al. (2022) also compute similar bag of words metrics for relations between characters in the summaries. Specifically, they consider a pair of characters related if they appear in the same sentence in the summary. They do not account for the direction of relations and focus only on co-occurrence. They again consider precision (BoR-p) and recall (BoR-r) of the intersection of pairs of characters similarly to computing the BoC metrics. We also report F1-score (BoR-f1), given the precision and recall for character relations.

We summarize our entity-specific results in Table 13. Overall, especially when considering the F1 scores for characters and relations, we arrive to similar conclusions as with our automatic QA evaluation (Table 5). The multimodal information that is incorporated in our Hierarchical3D approach increases most entity-specific metrics in comparison with text-only variants. Regarding different content selection methods, TP identification and supervised content selection again perform best in comparison with random selection, although differences are not large. Finally, we achieve the best F1 scores in both entity- and relation-specific metrics by using oracle selection, indicating that there is still room

for improvement. Interestingly, we again observe a further increase in performance by adding multimodal information in the pseudo-oracle variant, suggesting that video-based information is important even when we consider the most salient parts of an episode.

We also compare our approach with state-of-the-art, fully fine-tuned textual summarizers for long dialogues. We again notice that Summ^N is weakest according to entity-specific metrics. Next, efficient architectures for modeling the entire input (i.e., LED, DialogLED) have competitive performance against our text-only variants with content selection. However, Hierarchical3D that considers multimodal information outperforms these memory-heavy models while training only a small fraction of model parameters. This further validates our hypothesis that the video can provide additional information which more important for high-quality summaries than processing the entire textual input.

C.3 Examples of Generated Summaries

In this section we provide examples of generated summaries based on different automatic systems. Moreover, we provide examples of questions and answers used for the automatic QA evaluation described in Section 6.

Table 14 shows examples of automatically generated question-answer pairs given gold standard summaries. We provide examples of QA pairs for named entities (first 4 rows of the table) and nouns (remaining 6 rows of the table). We observe that most QA pairs are reasonable and correspond to information given in human-written summaries (first column of the table). However, there are cases where the QA pairs do not provide reasonable questions. Such an example is illustrated in the last row of Table 14, where the question is generated given the summary segment “Jonathan and Lizzie find out their baby has a medical condition, and make a run for it”:

Q: “What do Lizzie and Jonathan do when they learn their baby has a medical condition?”

A: “run”

	BoC-p	BoC-r	BoC-f1	BoR-p	BoR-r	BoR-f1
Random selection	82.55	38.71	52.71	29.82	9.39	14.28
+ Hierarchical3D	81.80	47.37	60.00	31.75	13.77	19.21
TP identification	84.31	38.93	53.26	36.79	10.33	16.13
+ Hierarchical3D	82.20	47.10	59.89	34.82	14.10	20.07
Content Selection	81.60	36.59	50.52	30.54	8.58	13.40
+ Hierarchical3D	81.90	48.48	60.91	33.04	14.37	20.03
Pseudo-oracle	<i>87.42</i>	<i>46.95</i>	<i>61.09</i>	<i>37.92</i>	<i>14.40</i>	<i>20.87</i>
+ Hierarchical3D	85.53	52.37	64.96	36.67	17.51	23.70
LED FT	82.28	33.54	47.65	34.35	10.64	16.25
DialogLED FT	82.93	38.19	52.27	31.71	10.32	15.57
Summ ^N FT	82.74	29.14	43.10	34.73	9.39	14.78

Table 13: Entity-specific metrics (test set). We report bag of character precision (BoC-p), recall (BoC-r), and F1 (BoC-f1). Analogously, we compute bag of relations precision (BoR-p), recall (BoR-r), and F1 (BoR-f1).

Summary	Question	Answer
Sage goes to live with Jack after she learns Carly is planning to marry Craig. Meg agrees to marry Dusty.	Who does Meg agree to marry?	Dusty
	Who does Sage go to live with?	Jack
Joshua is busy preparing for Allison’s arrival, as he unveils Kevin’s latest creation; a portrait of Allison and Joshua in their wedding attire. Lucy goes to church to plead for answers. Ian overhears her plea and swears that he will not let her die. Livvie shows Joshua a picture of Allison appearing to be dead and tells him that he was right her fangs are poisoned.	Who goes to church to plead for answers?	Lucy
	Who swears he will not let Lucy die?	Ian
	What does Lucy do at church? What part of Allison’s body is poisoned?	plea fangs
Lizzie and Jonathan spend some time with their baby. Jonathan gives in to one of Alan s demands. Gus and Harley find a disk with some interesting information on it. Gus still can t figure out what it is that Blake has on him. Dinah and Mallet argue over who will be the next WSPR star. Tammy is heartbroken after a visit to the hospital. Jonathan and Lizzie find out their baby has a medical condition, and make a run for it. Alan realizes that he may have been outwitted by Jonathan. Gus vows to get to the bottom of his supposed secret.	What does Gus vow to find out about Blake?	secret
	What is Lizzie and Jonathan spending time with?	baby
	What do Gus and Harley find?	disk
	What do Lizzie and Jonathan do when they learn their baby has a medical condition?	run

Table 14: Examples of automatically generated QA pairs for the evaluation of generated summaries.

This QA pair does not correspond to a reasonable fact of the episode. This shows that although it is useful to filter the questions, there are still imperfections with the automatic generation of QA pairs, especially when considering nouns.

Next, we give examples of the generated summaries for the TV show "Port Charles" in Tables 15–18. We present the gold or generated summary alongside the QA pairs used for evaluation. First, we compare different content selection methods (i.e., supervised content selection (CS), TP identification (TPs), and pseudo-oracle) for a text-only summarizer based on BART with adapter tuning. We present two examples in Tables 15 and 17 (we also show gold summaries for each episode). In both cases, we observe that the pseudo-oracle selection provides summaries of better quality, with fewer errors in the questions answered (i.e., errors are illustrated with red). Moreover, when compar-

ing content selection (CS) with TP identification (TPs), we find that these two approaches provide similar results, as suggested by our main experimental results (Table 5). Specifically, in Table 15, TP identification seems to provide the most informative summary, whereas in Table 17 supervised content selection is the best option.

Secondly, we compare our approach that considers multimodal information (Hierarchical3D) against text-only BART with equivalent content selection, and LED which considers only text and uses an efficient self-attention mechanism for processing the entire input. We present two examples for the same episodes as above in Tables 16 and 18. We empirically validate that the quality of the generated summaries is improved by adding the multimodal information (both when using supervised content selection and TP identification). Our approach leads to summaries that answer a

larger percentage of automatic questions correctly (i.e., correct answers are illustrated with **green**) outperforming LED, which is fully fine-tuned and memory-heavy. Interestingly, LED summaries cannot answer a large proportion of the given questions, suggesting that such methods may not be suitable for our task and small size dataset.

Victor: To new beginnings and a new way of doing things.

Mary: Aw

Victor: Ladies and gentlemen, I would also like to raise my glass to Joshua. Mr. Joshua Temple. Some of you already know that Mr. Temple is going to be the new owner of our beloved Recovery Room. And he is certainly Port Charles' newest, most distinguished citizen. I haven't known him very long, but I can vouch for the fact that he's a man of drive and vision. Ladies and gentlemen, Joshua Temple.

All: Hear, hear!

Lucy: This is unbelievable.

Joshua: I have many ambitious plans, not just for this place but for all over my new adopted home, the lovely town of Port Charles.

Mary: Aw.

Joshua: I hope you all approve.

[Cheers and applause]

Jamal: Make room, make room, make room. Watch this.

Mary: Ah.

Alison: My God. It's like a vision of hell.

Caleb: It's your city -- the way Joshua intends it to be.

Rafe: We got to find a way to stop him.

Caleb: It looks like the destruction's already begun.

Rafe: This guy worked for you, Caleb. What are his weaknesses?

Caleb: Well, you might want to sit this one out.

Livvie: Or move.

Caleb: Don't worry. With Olivia's help, I won't be mortal for long. And then I'll crush that little worm.

Livvie: It might not be that easy.

Caleb: As long as we have the ring, we -- what happened to the ring?

Livvie: It's gone. I'm sorry, Caleb, but our protection against Joshua is gone.

Caleb is upset when Livvie tells him that Joshua has the ring.

Joshua attempts to sway Ian to the dark side, but Ian vows he will continue to fight Joshua and the other vampires. Rafe tells Caleb the only way he can defeat Joshua now is to remain human and Livvie reluctantly agrees. Lucy pleads with Victor to fight Joshua, however, it's too late, as Victor tells her he enjoys the power Joshua has given him. Karen realizes Frank is a vampire.

gold summary

part of the input transcript

Figure 2: Example of input and output for SummScreen dataset. A long transcript is considered as input for summarization, containing the dialogue parts of a full-length TV episode. Character names are given as part of the dialogue. The goal is to produce a textual summary of most important events in the episode.

Victor: To new beginnings and a new way of doing things.

Mary: Aw

Victor: Ladies and gentlemen, I would also like to raise my glass to Joshua. Mr. Joshua Temple. Some of you already know that Mr. Temple is going to be the new owner of our beloved Recovery Room. And he is certainly Port Charles' newest, most distinguished citizen. I haven't known him very long, but I can vouch for the fact that he's a man of drive and vision. Ladies and gentlemen, Joshua Temple.

All: Hear, hear!

Lucy: This is unbelievable.

Joshua: I have many ambitious plans, not just for this place but for all over my new adopted home, the lovely town of Port Charles.

Mary: Aw.

Joshua: I hope you all approve.

[Cheers and applause]

Jamal: Make room, make room, make room. Watch this.

Mary: Ah.

Alison: My God. It's like a vision of hell.

Caleb: It's your city -- the way Joshua intends it to be.

Rafe: We got to find a way to stop him.

Caleb: It looks like the destruction's already begun.

Rafe: This guy worked for you, Caleb. What are his weaknesses?

Caleb: Well, you might want to sit this one out.

Livvie: Or move.

Caleb: Don't worry. With Olivia's help, I won't be mortal for long. And then I'll crush that little worm.

Livvie: It might not be that easy.

Caleb: As long as we have the ring, we -- what happened to the ring?

Livvie: It's gone. I'm sorry, Caleb, but our protection against Joshua is gone.

part of the input transcript



video

Figure 3: We augment SummScreen (see example of Figure 2) with information from the full-length video, which is aligned to the input transcript. Additional information, such as Joshua touching the ring in a previous scene or Caleb looking concerned when talking to Livvie, can be acquired from the video frames.

Model	Summary	
Gold	Caleb is upset when Livvie tells him that Joshua has the ring. Joshua attempts to sway Ian to the dark side, but Ian vows he will continue to fight Joshua and the other vampires. Rafe tells Caleb the only way he can defeat Joshua now is to remain human and Livvie reluctantly agrees. Lucy pleads with Victor to fight Joshua, however, it's too late, as Victor tells her he enjoys the power Joshua has given him. Karen realizes Frank is a vampire.	
QA pairs	<ul style="list-style-type: none"> • Who tells Lucy that he enjoys the power Joshua has given him? • Who does Karen realize is a vampire? • Who pleads with Victor to fight Joshua? • Who tells Caleb that Joshua has the ring? • Who realizes Frank is a vampire? • What does Livvie tell Caleb Joshua has? • Who does Karen realize Frank is? 	Victor Frank Lucy Livvie Karen the ring vampire
CS (text-only)	Caleb and Rafe discuss how to get close to Joshua and Livvie. Lucy tries to convince Victor that Joshua is an evil vampire who should not be allowed to have his soul. Lucy tells Victor that she can't lose him and wants him to accept her offer to turn him back into a vampire. Joshua tells the people of Port Charles that he will do whatever it takes to breathe new life into this wonderful old place.	
QA pairs	<ul style="list-style-type: none"> • Who tells Lucy that he enjoys the power Joshua has given him? • Who does Karen realize is a vampire? • Who pleads with Victor to fight Joshua? • Who tells Caleb that Joshua has the ring? • Who realizes Frank is a vampire? • What does Livvie tell Caleb Joshua has? • Who does Karen realize Frank is? 	Victor Joshua Lucy Rafe Victor soul vampire
TPs (text-only)	Caleb and Livvie are shocked to find out that the ring has been taken away from them by Joshua. They are unable to get the ring back, but they are determined to find a way to get it back. Lucy tells Victor that Joshua is a liar and that he should not be allowed to have an important position in Port Charles. Victor tells Lucy that he will not give up on her, but she tells him that she will not go to the hospital because she has to be here for Victor. Lucy and Victor agree that they need to stay at the hospital for the sake of Victor, but Lucy is not willing to give up her job as Victor's guardian angel. Rafe tells Alison that if she turns back, she is going to be under Joshua's control, and he doesn't think she will be too happy about taking orders from someone else. He tells her that he would rather slit her own wrists than have Joshua touch her.	
QA pairs	<ul style="list-style-type: none"> • Who tells Lucy that he enjoys the power Joshua has given him? • Who does Karen realize is a vampire? • Who pleads with Victor to fight Joshua? • Who tells Caleb that Joshua has the ring? • Who realizes Frank is a vampire? • What does Livvie tell Caleb Joshua has? • Who does Karen realize Frank is? 	Victor Joshua Lucy Lucy the ring vampire
Pseudo-oracle	Caleb and Livvie find out that the ring that protects them from Joshua is gone, and they have no way of getting it back. Lucy and Ian try to convince Victor to turn Joshua into a vampire, but he tells them that he is not a monster and that he wants to be a good man. Lucy tells Victor that Joshua is evil, but Victor tells her that she is his friend and he will not let her get to him. Victor tells Lucy that he will do whatever it takes to breathe new life into Port Charles. Ricky tells Jamal that he does not want to be his guardian angel anymore, but Jamal tells him to go and hang with the bloodsuckers. Jack tells Jack that the only way he is going to win is to play both sides of the fence. Frank tells Karen that there are benefits to his new role as a vampire and he is willing to do it for the good of the town of Port Charles and his family.	
QA pairs	<ul style="list-style-type: none"> • Who tells Lucy that he enjoys the power Joshua has given him? • Who does Karen realize is a vampire? • Who pleads with Victor to fight Joshua? • Who tells Caleb that Joshua has the ring? • Who realizes Frank is a vampire? • What does Livvie tell Caleb Joshua has? • Who does Karen realize Frank is? 	Victor Frank Lucy Livvie Karen the ring vampire

Table 15: Examples of generated summaries and automatic QA evaluation for different content selection methods. In all model variants we only consider the textual modality (i.e., BART with adapter tuning).

Model	Summary
Gold	<p>Caleb is upset when Livvie tells him that Joshua has the ring. Joshua attempts to sway Ian to the dark side, but Ian vows he will continue to fight Joshua and the other vampires. Rafe tells Caleb the only way he can defeat Joshua now is to remain human and Livvie reluctantly agrees. Lucy pleads with Victor to fight Joshua, however, it's too late, as Victor tells her he enjoys the power Joshua has given him. Karen realizes Frank is a vampire.</p>
QA pairs	<ul style="list-style-type: none"> • Who tells Lucy that he enjoys the power Joshua has given him? Victor • Who does Karen realize is a vampire? Frank • Who pleads with Victor to fight Joshua? Lucy • Who tells Caleb that Joshua has the ring? Livvie • Who realizes Frank is a vampire? Karen • What does Livvie tell Caleb Joshua has? the ring • Who does Karen realize Frank is? vampire
Text-only (TPs)	<p>Caleb and Livvie are shocked to find out that the ring has been taken away from them by Joshua. They are unable to get the ring back, but they are determined to find a way to get it back. Lucy tells Victor that Joshua is a liar and that he should not be allowed to have an important position in Port Charles. Victor tells Lucy that he will not give up on her, but she tells him that she will not go to the hospital because she has to be here for Victor. Lucy and Victor agree that they need to stay at the hospital for the sake of Victor, but Lucy is not willing to give up her job as Victor's guardian angel. Rafe tells Alison that if she turns back, she is going to be under Joshua's control, and he doesn't think she will be too happy about taking orders from someone else. He tells her that he would rather slit her own wrists than have Joshua touch her.</p>
QA pairs	<ul style="list-style-type: none"> • Who tells Lucy that he enjoys the power Joshua has given him? Victor • Who does Karen realize is a vampire? Joshua • Who pleads with Victor to fight Joshua? Lucy • Who tells Caleb that Joshua has the ring? Lucy • Who realizes Frank is a vampire? Lucy • What does Livvie tell Caleb Joshua has? the ring • Who does Karen realize Frank is? vampire
H-3D (TPs)	<p>Caleb and Livvie are shocked when they find out that their protection against Joshua is gone. Victor and Lucy argue about Victor's role in Port Charles. Lucy tells Victor that Joshua is evil and that he should not be allowed to have an important position with the vampires. Victor tells Lucy that he still has so much to contribute and maybe this is his chance to have people listen to him again. Lucy is upset that Victor wants to give Joshua an important role in the town. Lucy and Victor are at the hospital and Lucy tells him that she is going to be there for Victor, but he tells her to stay away from him. Frank tells Karen that he has grown a pair of fangs. Karen is shocked to hear that Frank is a vampire.</p>
QA pairs	<ul style="list-style-type: none"> • Who tells Lucy that he enjoys the power Joshua has given him? Victor • Who does Karen realize is a vampire? Frank • Who pleads with Victor to fight Joshua? Lucy • Who tells Caleb that Joshua has the ring? Lucy • Who realizes Frank is a vampire? Karen • What does Livvie tell Caleb Joshua has? their protection against Joshua • Who does Karen realize Frank is? vampire
LED	<p>At the end of the night, Caleb and Livvie's wedding ring is revealed to Rafe and Caleb's ring. Caleb tells Rafe that the ring is a vampire slayer. Rafe tells Caleb that he's going to be a vampire again. Lucy tells Victor that Victor is going to take over the town of Port Charles. Victor tells Lucy that he wants to talk to Lucy. Lucy asks Victor to join her in the new venture, but Victor tells her that he is not going to do it. Caleb tells Olivia that he has been drugged by Kevin, and he's been turned into a vampire. Olivia tells him that she wants to be part of the new club, but Caleb tells her to stay away from him. Joshua tells Ian that he will not be able to get Victor away from Victor. Ian tells Joshua that Joshua is not one of the vampire slayers, but he is the one of them.</p>
QA pairs	<ul style="list-style-type: none"> • Who tells Lucy that he enjoys the power Joshua has given him? Victor • Who does Karen realize is a vampire? Caleb • Who pleads with Victor to fight Joshua? Ian • Who tells Caleb that Joshua has the ring? Ian • Who realizes Frank is a vampire? Rafe • What does Livvie tell Caleb Joshua has? wedding ring • Who does Karen realize Frank is? slayer

Table 16: Examples of generated summaries and automatic QA evaluation for different models. Here we compare our Hierarchical3D model (H-3D) with state-of-the-art textual summarizers (i.e., LED).

Model	Summary
Gold	<p>Joshua tells Elizabeth he wants to turn Allison and demands she help ease Allison into her new life as his wife. Elizabeth tells Joshua she will kill him before she allows him to hurt Allison. Livvie is able to fend off her need to feed while she and Caleb make love. Frank searches for Allison. When Frank attempts to kidnap Allison from Rafe, he discovers that it really is Lucy and Ian in disguise. Allison and Rafe reappear in Caleb s cave.</p> <ul style="list-style-type: none"> • Who does Frank try to kidnap Allison from? Rafe • Who does Frank try to kidnap? Allison • Who tries to kidnap Allison? Frank
QA pairs	<ul style="list-style-type: none"> • Who can fend off her need to feed while she and Caleb make love? Livvie • Who tells Joshua she will kill him before she allows him to hurt Allison? Elizabeth • Who tells Elizabeth he wants to turn Allison into his wife? Joshua • What is Allison s new life? wife
CS (text-only)	<p>Rafe tells Alison that he will never let Joshua take her for his bride, but she tells him that she has no choice in the matter. Elizabeth tells Joshua that she will not stand by and allow him to take her daughter. Joshua tells Elizabeth that he is going to ease Alison into her new lifestyle as his wife. Elizabeth says that she is not going to let her daughter suffer the kind of nightmare that she lived. She will kill Joshua before he is even that close to turning her. Alison tells Rafe that she thinks this is a little extreme, that is all. Rafe says he will not let Joshua get to her. He promises to keep her away from Joshua and all his goons. Caleb tells Livvie that she doesn t need to feed. He tells her that he can t make love to her the way she wants to. She tells him she can t turn him back, but he tells her he can. He says that he loves her and that he wants to make her his bride.</p> <ul style="list-style-type: none"> • Who does Frank try to kidnap Allison from? Joshua • Who does Frank try to kidnap? Joshua • Who tries to kidnap Allison? Rafe
QA pairs	<ul style="list-style-type: none"> • Who can fend off her need to feed while she and Caleb make love? Livvie • Who tells Joshua she will kill him before she allows him to hurt Allison? Elizabeth • Who tells Elizabeth he wants to turn Allison into his wife? Joshua • What is Allison s new life? wife
TPs (text-only)	<p>Livvie tells Caleb that she can t be with him, knowing what his bite might do to him. Joshua tells Elizabeth that he is going to steal Alison s slayer s intended and that will make him more respectable. He tells her that she is not going to do it, and that he knows that both roads lead Alison right back to Joshua. Elizabeth tells Joshua that she has no idea how much she would love to do that, but they both know that they are too busy figuring out which road to take. Rafe and Alison find themselves in Caleb s old hangout. Alison is shocked to see that Rafe is still there. She tells Rafe that she will feel better when all this is over with. She asks Rafe if he has his back and he says that he does, and he tells her he has her back. Lucy and Ian find out that they have succeeded in their mission to kidnap Alison. Ian and Frank tell Lucy that they will have to tell their boss that their mission didn t work. Frank tells Ian that he will pay for this, and they are both going to pay for it. Ian tells Frank that he has to tell his big, powerful boss that his mission failed.</p> <ul style="list-style-type: none"> • Who does Frank try to kidnap Allison from? Lucy • Who does Frank try to kidnap? Alison • Who tries to kidnap Allison? Lucy
QA pairs	<ul style="list-style-type: none"> • Who can fend off her need to feed while she and Caleb make love? Rafe • Who tells Joshua she will kill him before she allows him to hurt Allison? Elizabeth • Who tells Elizabeth he wants to turn Allison into his wife? Joshua • What is Allison s new life? Caleb
Pseudo-oracle	<p>Caleb tells Livvie that he loves her and wants to make love to her, but he can t because his fangs are poisoned and he can s not turn her back. He tells her to take the risk, but she is afraid to do it. She tells him that she loves him and that she wants to be with him, but that she can t do it because of the risk of his bite. He says that he will not let her go back to him. She says that she will never stop loving him. He promises her that he won t and they make love again. Joshua tells Elizabeth that he wants to take Alison as his bride. Elizabeth tells Joshua that she is not going to let him take her. She wants to stop him from taking Alison. Rafe tells Alison that he has a vision of a city of tortured souls, a master vampire race, and his vampire bride. He will never let that happen. He can t let Joshua get to her. He is going to keep her away from Joshua and all his goons. He wants her to be his bride and she will be his wife. She will never love him, Rafe says.</p> <ul style="list-style-type: none"> • Who does Frank try to kidnap Allison from? Joshua • Who does Frank try to kidnap? Allison • Who tries to kidnap Allison? Joshua
QA pairs	<ul style="list-style-type: none"> • Who can fend off her need to feed while she and Caleb make love? Livvie • Who tells Joshua she will kill him before she allows him to hurt Allison? Elizabeth • Who tells Elizabeth he wants to turn Allison into his wife? Joshua • What is Allison s new life? vampire bride

Table 17: Examples of generated summaries and automatic QA evaluation for different content selection methods. In all model variants, we only consider the textual modality (i.e., BART with adapter tuning).

Model	Summary	
Gold	Joshua tells Elizabeth he wants to turn Allison and demands she help ease Allison into her new life as his wife. Elizabeth tells Joshua she will kill him before she allows him to hurt Allison. Livvie is able to fend off her need to feed while she and Caleb make love. Frank searches for Allison. When Frank attempts to kidnap Allison from Rafe, he discovers that it really is Lucy and Ian in disguise. Allison and Rafe reappear in Caleb's cave.	
	<ul style="list-style-type: none"> Who does Frank try to kidnap Allison from? Who does Frank try to kidnap? Who tries to kidnap Allison? 	Rafe Allison Frank
QA pairs	<ul style="list-style-type: none"> Who can fend off her need to feed while she and Caleb make love? Who tells Joshua she will kill him before she allows him to hurt Allison? Who tells Elizabeth he wants to turn Allison into his wife? What is Allison's new life? 	Livvie Elizabeth Joshua wife

Text-only (CS)	Rafe tells Alison that he will never let Joshua take her for his bride, but she tells him that she has no choice in the matter. Elizabeth tells Joshua that she will not stand by and allow him to take her daughter. Joshua tells Elizabeth that he is going to ease Alison into her new lifestyle as his wife. Elizabeth says that she is not going to let her daughter suffer the kind of nightmare that she lived. She will kill Joshua before he is even that close to turning her. Alison tells Rafe that she thinks this is a little extreme, that is all. Rafe says he will not let Joshua get to her. He promises to keep her away from Joshua and all his goons. Caleb tells Livvie that she doesn't need to feed. He tells her that he can't make love to her the way she wants to. She tells him she can't turn him back, but he tells her he can. He says that he loves her and that he wants to make her his bride.	
	<ul style="list-style-type: none"> Who does Frank try to kidnap Allison from? Who does Frank try to kidnap? Who tries to kidnap Allison? 	Joshua Joshua Rafe
QA pairs	<ul style="list-style-type: none"> Who can fend off her need to feed while she and Caleb make love? Who tells Joshua she will kill him before she allows him to hurt Allison? Who tells Elizabeth he wants to turn Allison into his wife? What is Allison's new life? 	Livvie Elizabeth Joshua wife

H-3D (CS)	Livvie tries to convince Caleb to let her take the risk of biting him, but she is afraid that she won't be able to do it. Joshua tells Elizabeth that he wants Alison to be his bride. Elizabeth is shocked when she finds out that Joshua wants to take Alison away from Rafe. Elizabeth tells Joshua that she will find a way to stop him from taking Alison. Rafe tells Alison that he has a vision of a city of tortured souls, a master vampire race, and his vampire bride. He tells her that he can make a perfect bride for her. Alison tells Rafe that she doesn't want to leave her family, but Rafe assures her that she is not going to leave them. Frank tells Ian that he is going to have to tell his boss that his mission didn't work.	
	<ul style="list-style-type: none"> Who does Frank try to kidnap Allison from? Who does Frank try to kidnap? Who tries to kidnap Allison? 	Rafe Allison Rafe
QA pairs	<ul style="list-style-type: none"> Who can fend off her need to feed while she and Caleb make love? Who tells Joshua she will kill him before she allows him to hurt Allison? Who tells Elizabeth he wants to turn Allison into his wife? What is Allison's new life? 	Livvie Elizabeth Joshua vampire bride

LED	Caleb and Livvie try to convince Caleb that they can be together again, but Caleb refuses to believe that he can't make love to Livvie. Rafe and Alison agree that they will be together, but Rafe tells Alison that he will never be able to make love with her again. Elizabeth tells Joshua that she wants him to turn Alison into a vampire bride. Joshua tells Elizabeth that he wants her to be his wife, but Elizabeth tells him that she is not going to do that. Joshua says that he is going to make Alison a vampire, and he will be the one true love of his life. Elizabeth says that she will never let Alison suffer the kind of nightmare that she lived, and she will make her life as a vampire. Joshua and Elizabeth argue about how much she wants to be a vampire and how much he wants to help her. Elizabeth asks Joshua if he's going to help Alison, but he says he will not.	
	<ul style="list-style-type: none"> Who does Frank try to kidnap Allison from? Who does Frank try to kidnap? Who tries to kidnap Allison? 	Caleb Caleb Rafe
QA pairs	<ul style="list-style-type: none"> Who can fend off her need to feed while she and Caleb make love? Who tells Joshua she will kill him before she allows him to hurt Allison? Who tells Elizabeth he wants to turn Allison into his wife? What is Allison's new life? 	Livvie Elizabeth Joshua vampire

Table 18: Examples of generated summaries and automatic QA evaluation for different models. Here we compare our Hierarchical3D model (H-3D) with state-of-the-art textual summarizers (i.e., LED).

An Intra-Class Relation Guided Approach for Code Comment Generation

Zhenni Wang[†], Xiaohan Yu[†], Yansong Feng^{*}, Dongyan Zhao

Wangxuan Institute of Computer Technology, Peking University, China

The MOE Key Laboratory of Computational Linguistics, Peking University, China

{wangzhenni, yuxiaohan, fengyansong, zhaodongyan}@pku.edu.cn

Abstract

Code comments are critical for maintaining and comprehending software programs, but they are often missing, mismatched, or outdated in practice. Code comment generation task aims to automatically produce descriptive comments for code snippets. Recently, methods based on the neural encoder-decoder architecture have achieved impressive performance. These methods assume that all the information required to generate comments is encoded in the target function itself, yet in most realistic situations, it is hard to understand a function in isolation from the surrounding context. Furthermore, the global context may contain redundant information that should not be introduced. To address the above issues, we present a novel graph-based learning framework to capture various relations among functions in a class file. Our approach is based on a common real-world scenario in which only a few functions in the source file have human-written comments. Guided by intra-class function relations, our model incorporates contextual information extracted from both the source code and available comments to generate missing comments. We conduct experiments on a Java dataset collected from real-world projects. Experimental results show that the proposed method outperforms competitive baseline models on all automatic and human evaluation metrics.

1 Introduction

Code comment generation is the task of automatically producing natural language descriptions for given code snippets. Appropriate and sufficient comments are essential for software maintenance and understanding (Xia et al., 2018). They allow developers to grasp the purpose of source code quickly and accurately. However, in real-life software projects, comments are often missing, incomplete or outdated (Briand, 2003). Existing com-

```
1 private void firePropertyChange (String propName,  
2     Object oldValue, Object newValue) {  
3     PropertyChangeEvent evt = new PropertyChangeEvent();  
4     ...  
5 }  
6 /* Removes a time series from the map and  
   fires a TS_REMOVED PropertyChangeEvent.*/  
7 public removeTS (String name) {  
8     boolean fireChanged = false;  
9     ...  
10    if (fireChanged)  
11        firePropertyChange(TS_REMOVED, name, name);  
12 }  
13 /* Removes all time series from the map and  
   fires an ALL_TS_REMOVED PropertyChangeEvent.*/  
14 public removeAllTS() {  
15     ...  
16     firePropertyChange(ALL_TS_REMOVED, null, null);  
17 }
```

Table 1: Example illustrating the importance of utilizing class-level contextual information.

ments will also need to be adjusted as the associated programs are updated, which could cause large time and labor costs. Hence, there is a significant need for automatic generation technologies that can effectively produce high-quality comments.

Recent works in code comment generation take the neural encoder-decoder architecture as their cornerstone (Hu et al., 2018a; Alon et al., 2019; LeClair et al., 2020; Zhang et al., 2020; Wei et al., 2020). However, these works only utilize the information provided by the target function itself. In object-oriented programming, classes are the building blocks that express algorithmic intentions and they encapsulate the interaction between functions. Therefore, the class-level contextual information should not be ignored when we attempt to generate code comments. There are some existing studies that attempt to fill this gap. Haque et al. (2020) encode all functions in a source file using GRU (Cho et al., 2014) and apply an attention mechanism to learn associations between the encoding results to words in the generated comment. Yu et al. (2020) construct a class graph that connects the

[†]Equal contribution.

^{*}Corresponding author.

target function to all other functions in the same class to aggregate contextual information. Bansal et al. (2021) present a project-level encoder to augment existing models by introducing contextual information.

Although the above methods have shown promising performance, the way they introduce contextual information is somewhat crude. Since not all surrounding functions are closely related to the target function, indiscriminately utilizing the whole context may introduce noise, which would hurt the model performance. We propose that considering function relations is a better way to leverage the contextual information. For example, Table 1 presents three functions in a Java class. Within this class, we can observe two types of function relations. First, the function `removeTS` calls `firePropertyChange` in its function body. As we can see, the word "*PropertyChangeEvent*" in the human-written comment appears not in the target function, but in the callee function. Second, `removeTS` and `removeAllTS` perform very similar operations, and their comments are almost identical, with the exception of a few noun subjects. This example illustrates that the information required to generate a comment may be located outside the boundary of the target code snippet and within the related functions.

Motivated by the above observation, we define two types of relations between a function pair: extractive relation and inductive relation. The extractive relation captures connections between source code snippets at two levels: call dependencies and semantic similarity, allowing us to derive external knowledge directly from the relevant code snippets. The inductive relation captures common programming patterns within a class. We observe that developers usually create similar comments for functions that conform to a specific programming pattern. Therefore, comments of functions that have inductive relation to the target function can be used as a template to guide the target comment generation.

In this paper, we propose a graph-based encoder-decoder learning framework for code comment generation. Our approach is based on a common scenario where only a few functions in the class file are documented. We construct a heterogeneous graph to model both the extractive relation and inductive relation among functions within a class file. In the encoding stage, we encode all functions and available comments using bi-GRU. Then, we design

an intra-class relational GAT encoder to aggregate information and perform a fusion of both types of relations via a cross-graph mechanism. In the decoding stage, we employ a GRU decoder with a by-pointer mechanism to generate a comment utilizing the encoding results.

To evaluate the performance of our approach, we gather a Java dataset that preserves the class structure. We conduct experiments on this dataset and perform evaluation using automatic and human evaluation metrics. The experimental results show that our model outperforms prior methods by a significant margin, which demonstrates the effectiveness of our proposed framework.

2 Related Works

Early efforts on code comment generation are template-based or information retrieval (IR) based approaches (Sridhara et al., 2010; Haiduc et al., 2010a,b; Eddy et al., 2013; Rodeghero et al., 2014; McBurney and McMillan, 2014). In recent years, the neural encoder-decoder architecture is employed to the code comment generation field, which was designed for neural machine translation (NMT) task originally. CodeNN (Iyer et al., 2016) is an early work that attempts to adopt the encoder-decoder architecture for generating code comments. Followed works develop a variety of models by introducing the Abstract Syntax Tree (AST) to extract structural information of the source code (Hu et al., 2018a; Alon et al., 2019; Allamanis et al., 2018; Liang and Zhu, 2018; LeClair et al., 2019a). More recently, novel code representations are learned via well-designed encoders, such as GNN-based encoders (LeClair et al., 2020; Zhang et al., 2022) and pre-training encoders (Ahmad et al., 2020; Zügner et al., 2021; Guo et al., 2022).

Hybrid methods that integrate the IR-based and neural-based techniques proved to perform well on the code comment generation task. Zhang et al. (2020) retrieve two similar code snippets of the target function at syntax and semantics levels, then utilize their encoding information to generate comments in the decoding stage. Wei et al. (2020) retrieve the most similar code snippet and the corresponding comment to assist the generation process. Liu et al. (2021) retrieve the most similar code-comment pair and add it as auxiliary information to their proposed Hybrid GNN framework.

However, these works rarely utilize contextual information that is external to the target function.

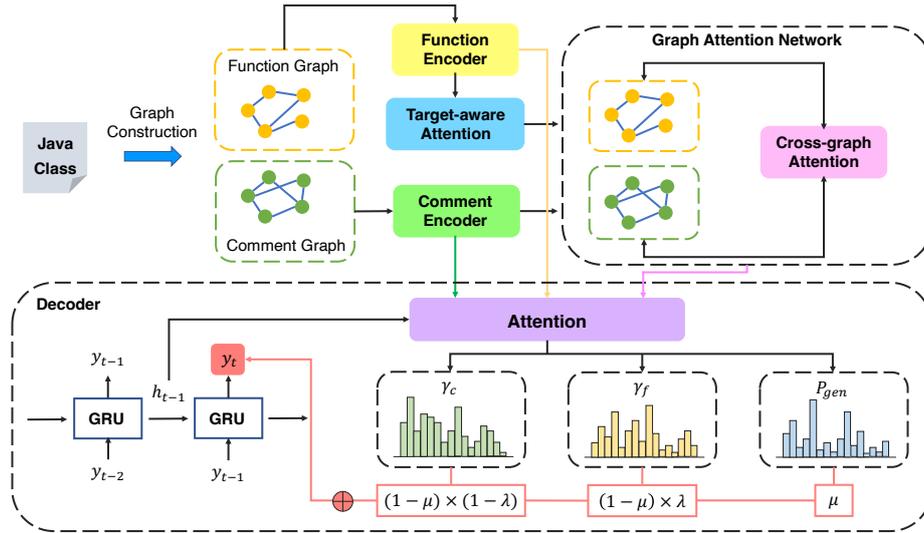


Figure 1: The overall architecture of our approach. Local encoders extract features from code snippets and known comments (Section 3.3). The GAT encoder aggregates class-level information and produces final representations. (Section 3.4). Lastly, these encoding results are fed into the decoder to create the target comment (Section 3.5).

Some of the most recent works make efforts to bridge this gap (Haque et al., 2020; Yu et al., 2020; Bansal et al., 2021). In contrast to these existing approaches, our method explores function relationships within a class and only incorporates related functions. This has the advantage of avoiding noise caused by irrelevant functions and focusing on the valuable contextual information.

3 Approach

This section introduces our proposed framework for code comment generation. Figure 1 illustrates an overview of our approach.

3.1 Relation Extraction

To focus on valuable class-level contextual information, we need to develop extraction rules for function relations. We define the extractive relation as (1) the call dependency; (2) the TF-IDF cosine similarity. Call dependencies can be extracted using the `java-callgraph*` toolkit, and the TF-IDF cosine similarity between functions X^i and X^j is calculated as:

$$s_{ij} = \frac{\overrightarrow{\text{tfidf}}(X^i)^T \overrightarrow{\text{tfidf}}(X^j)}{\|\overrightarrow{\text{tfidf}}(X^i)\| \|\overrightarrow{\text{tfidf}}(X^j)\|} \in [0, 1] \quad (1)$$

If the similarity score $s_{ij} > \alpha$, we consider an extractive relation between X^i and X^j , where α is a pre-defined threshold.

*<https://github.com/gousiosg/java-callgraph/>

Towards the inductive relation, we summarize some common programming patterns and organize them into five heuristic rules based on extensive observations of open-source software projects. Formally, we consider an inductive relation between two functions if:

- (R1) the verbs in function names are antonyms, with the same or no object entities;
- (R2) the verbs in function names are the same, with overlapping object entities;
- (R3) both functions have the same parameters as well as the same verbs in their names;
- (R4) both functions have the same parameters as well as the same return type;
- (R5) the return type of one function corresponds to the parameter type of another.

In (R1)-(R3), we conduct part-of-speech tagging on function names using the toolkit Stanford CoreNLP[†] to identify verbs and noun entities. And we use the NLTK[‡] interface of WordNet[§] to get antonyms of verbs in (R1). Appendix A provides several examples that correspond to the preceding rules.

3.2 Graph Construction

For each class, we build a graph structure that consists of two subgraphs, called function graph and

[†]<https://stanfordnlp.github.io/CoreNLP/>

[‡]<https://www.nltk.org/>

[§]<https://wordnet.princeton.edu/>

comment graph. The node of the function graph represents each function, while the node of the comment graph represents the corresponding comment. Since only a small fraction of comments in the class are known, we use function names to replace unknown comments. According to the previously defined extraction rules, we add edges between the corresponding function nodes if a pair of functions fulfill the extractive relation, and between comment nodes if they satisfy the inductive relation. Formally, we define a graph

$$G = \{ (v_i, r_{fun}, v_j) \cup (u_i, r_{com}, u_j) \},$$

where $v \in \mathcal{V}_f$ is the node of function, $u \in \mathcal{V}_c$ is the node of comment or function name, $\mathcal{V}_f, \mathcal{V}_c$ are the sets of function nodes and comment nodes, and r_{fun}, r_{com} denote edges that represent the extractive relation between functions, and the inductive relation between comments, respectively.

3.3 Local Encoder

Our model contains two local encoders, a function encoder and a comment encoder. They extract features from functions and comments separately. The function encoder employs a bi-GRU (Cho et al., 2014) to convert the source code sequence $\{x_1, \dots, x_n\}$ into numerical vectors $Z = \{z_1, \dots, z_n\}$, where $z_i = [\vec{z}_i || \overleftarrow{z}_i]$ is the concatenation of the hidden states from both directions. We take Z as the representation of the input function. The comment encoder also apply a bi-GRU to the comment sequence $\{w_1, \dots, w_m\}$, and produce hidden states $\{r_1, \dots, r_m\}$. The last hidden state r_m is considered as the comment representation.

3.4 Intra-class Relational GAT

We propose an intra-class relational graph attention network that performs on the previously constructed graph.

Node Initialization For function nodes, we first apply the function encoder to obtain their representations $\{Z_1, \dots, Z_K\}$, where $Z_i = \{z_{i1}, \dots, z_{in_i}\}$ represents the i -th function. Then, we use a target-aware attention mechanism to focus on information in other functions that is beneficial to the target function. We take the last hidden state as the representation of the target function Z_t , which can be denoted as z_t , and use it to compute the attention weight as:

$$\alpha_{t,ij} = \frac{\exp(z_t^T \mathbf{W}_t z_{ij})}{\sum_{k=1}^{n_i} \exp(z_t^T \mathbf{W}_t z_{ik})}, \quad (2)$$

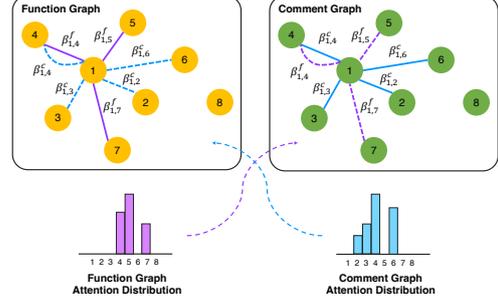


Figure 2: An example of updating node 1 via cross-graph attention mechanism.

where \mathbf{W}_t is a learnable parameter. The attention score $\alpha_{t,ij}$ measures the similarity between the target function and the j -th token in the code sequence of the i -th function. Then, we compute the weighted sum of all elements in Z_i :

$$\mathbf{f}_i^0 = \sum_{j=1}^{n_i} \alpha_{t,ij} z_{ij}, \quad (3)$$

and take it as the initial representation of the function node v_i . Finally, we obtain a set of target-aware initial vectors representing function nodes, which is denoted as $\{\mathbf{f}_i^0 | i : v_i \in \mathcal{V}_f\}$. For comment nodes, the last hidden states are used directly as the initial node representations, and we denote them as $\{\mathbf{c}_i^0 | i : u_i \in \mathcal{V}_c\}$.

Cross-Graph Attention We employ two separate GAT modules, one for the function graph and the other for the comment graph, which have the same structure. In order to interact information between these two graphs, we design a cross-graph attention mechanism that is applied to each layer of the GATs. Fig. 2 illustrates an example of this mechanism. Specifically, the $l+1$ -th layer of each GAT receives a set of messages $\{\mathbf{f}_i^l | i : v_i \in \mathcal{V}_f\}$ or $\{\mathbf{c}_i^l | i : u_i \in \mathcal{V}_c\}$ from the previous layer. Then, we obtain two output vectors $\mathbf{f}_{g,i}^{l+1}, \mathbf{c}_{g,i}^{l+1}$ and two attention distributions calculated as:

$$e_{ij}^f = \text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}_a \mathbf{f}_i^l || \mathbf{W}_a \mathbf{f}_j^l]),$$

$$\beta_{ij}^f = \text{Softmax}(e_{ij}^f) = \frac{\exp(e_{ij}^f)}{\sum_k \exp(e_{ik}^f)}, \quad (4)$$

$$e_{ij}^c = \text{LeakyReLU}(\mathbf{b}^T [\mathbf{W}_b \mathbf{c}_i^l || \mathbf{W}_b \mathbf{c}_j^l]),$$

$$\beta_{ij}^c = \text{Softmax}(e_{ij}^c) = \frac{\exp(e_{ij}^c)}{\sum_k \exp(e_{ik}^c)}, \quad (5)$$

where $\mathbf{a}, \mathbf{W}_a, \mathbf{b}, \mathbf{W}_b$ are trainable parameters. Next, we exchange the attention weights and com-

pute two cross-graph vectors as:

$$\mathbf{f}_{cross,i}^{l+1} = \sum_{j:u_j \in N_c(u_i)} \beta_{ij}^c \mathbf{W}_d \mathbf{f}_j^l, \quad (6)$$

$$\mathbf{c}_{cross,i}^{l+1} = \sum_{j:v_j \in N_f(v_i)} \beta_{ij}^f \mathbf{W}_e \mathbf{c}_j^l, \quad (7)$$

where u_i is the corresponding comment node of function node v_i , $N_f(v_i)$ is a set of the neighboring function nodes of v_i , $N_c(u_i)$ is a set of the neighboring comment nodes of u_i , and $\mathbf{W}_d, \mathbf{W}_e$ are trainable parameters. Finally, we aggregate the cross-graph vector and the original GAT output vector to obtain an integral context vector that takes into account both extractive and inductive relations:

$$\mathbf{f}_{aggr,i}^{l+1} = \tanh(\mathbf{W}_f[\mathbf{f}_{cross,i}^{l+1} \parallel \mathbf{f}_{g,i}^{l+1}]). \quad (8)$$

$$\mathbf{c}_{aggr,i}^{l+1} = \tanh(\mathbf{W}_c[\mathbf{c}_{cross,i}^{l+1} \parallel \mathbf{c}_{g,i}^{l+1}]). \quad (9)$$

where $\mathbf{W}_f, \mathbf{W}_c$ are trainable weights.

Update Gate Motivated by (Cho et al., 2014), we introduce an update gate to control how much information from the previous representation should be transferred to the current representation:

$$g = \text{Sigmoid}(\mathbf{W}_g[\mathbf{f}_{aggr,i}^{l+1} \parallel \mathbf{f}_i^l]), \quad (10)$$

$$\mathbf{f}_i^{l+1} = g * \mathbf{f}_{aggr,i}^{l+1} + (1 - g) * \mathbf{f}_i^l. \quad (11)$$

We also obtain \mathbf{c}_i^{l+1} by performing the same operation. These two representations are the outputs of the $l + 1$ -th layer. We repeat the above process L times and get the final node representations, denoted as \mathbf{f}_i^L and \mathbf{c}_i^L . Finally, we concatenate the representation of the function node and its corresponding comment node as the output of our GAT encoder, which is denoted as $\mathbf{g}_i = \mathbf{f}_i^L \parallel \mathbf{c}_i^L$.

3.5 Decoder

The decoder employs a GRU to generate comment for the target function Z_t . The initial hidden state is a concatenation of the last hidden state \mathbf{z}_t from the function encoder and the final output \mathbf{g}_t from the GAT encoder.

Attention We consider multiple context vectors: \mathbf{cz}_t toward the output from the function encoder, \mathbf{cr}_t toward the output from the comment encoder, and \mathbf{cg}_t toward the output from the GAT encoder, which can be calculated as follows:

$$\gamma_{tj} = \frac{\exp(\mathbf{h}_t^T \mathbf{W}_s \boldsymbol{\eta}_j)}{\sum_k \exp(\mathbf{h}_t^T \mathbf{W}_s \boldsymbol{\eta}_k)}, \quad (12)$$

$$\mathbf{cv}_t = \sum_j \gamma_{tj} \boldsymbol{\eta}_j, \quad (13)$$

where \mathbf{W}_s is a learnable parameter, \mathbf{h}_t is the current decoder hidden state, and $\boldsymbol{\eta}_j$ represents the function encoder output \mathbf{z}_j , the comment encoder output \mathbf{r}_j , and the GAT encoder output \mathbf{g}_j , respectively.

By-Pointer Since both code snippets and known comments may contain words that are not in the vocabulary, a portion of the predicted tokens could be copied directly from them. Motivated by (See et al., 2017) and (Sun et al., 2018), we design a by-pointer mechanism to solve this problem. In the t -th time step, the decoder takes embedding \mathbf{y}_t as input, and the copy distribution is formulated as:

$$\lambda = \text{Sigmoid}(\mathbf{W}_l[\mathbf{cr}_t \parallel \mathbf{h}_t \parallel \mathbf{y}_t]), \quad (14)$$

$$P_{copy} = \lambda * \gamma_c + (1 - \lambda) * \gamma_f, \quad (15)$$

where \mathbf{W}_l is the trainable parameter. The γ_f is the attention distribution between the current hidden state \mathbf{h}_t and source codes, γ_c is the attention distribution between \mathbf{h}_t and known comments, both calculated by Eq (12). Additionally, the generative distribution over all vocabulary tokens is calculated based on \mathbf{h}_t and three context vectors:

$$P_{gen} = \text{Softmax}(\mathbf{W}_v[\mathbf{h}_t \parallel \mathbf{cz}_t \parallel \mathbf{cr}_t \parallel \mathbf{cg}_t] + \mathbf{b}_v), \quad (16)$$

where $\mathbf{W}_v, \mathbf{b}_v$ are trainable parameters. Finally, we obtain the prediction distribution as follows:

$$\mu = \text{Sigmoid}(\mathbf{W}_m[\mathbf{cz}_t \parallel \mathbf{cr}_t \parallel \mathbf{cg}_t \parallel \mathbf{h}_t \parallel \mathbf{y}_t]), \quad (17)$$

$$P(w) = \mu * P_{gen} + (1 - \mu) * P_{copy} \quad (18)$$

where \mathbf{W}_m is the trainable parameter. This mechanism allows our model to both generate tokens from the vocabulary and copy tokens from two sources during inference.

4 Experimental Setup

4.1 Dataset

Due to most public datasets only consist of independent code snippets, we collect a dataset from Google Code Archive that preserves class-level information. With the help of Sourcerer (Bajracharya et al., 2014), we are able to trace and recover the entire architecture of 1,000 real-world JAVA projects. We assume that only 10% of functions or at least one function have comments in each class. To determine which functions will be treated as commented, we use two different sampling settings. (1)

Random Sampling: we randomly sample 10% of functions in each class as commented based on the assumption that commenting is a stochastic behavior for developers; (2) **Degree Sampling:** since functions that connect to others more frequently often play a key role in programming, we calculate function degrees in the class graph and rank them in descending order. Then we sample the top 10% of functions as commented. After sampling, we split our dataset by projects according to a ratio of 8:1:1. The more detail of our dataset is provided in the Appendix B.

4.2 Baselines

Retrieval-based Models **RandomCopy** randomly copies comments from a known comment set. **MaxCopy** computes the ROUGE-L score between the golden comment and known comments, then copies the comment with the highest score. **NNGen** (Liu et al., 2018) is a IR-based method for generating commit messages that can also be used in the code comment generation task.

Generation-based Models **Seq2Seq** (Sutskever et al., 2014) is a bi-GRU with an attention mechanism. **ASTGNN** (LeClair et al., 2020) applies a GRU encoder for the source code sequence, a GCN (Kipf and Welling, 2017) encoder for the AST and a GRU decoder for generation. **Rencos** (Zhang et al., 2020) retrieves two similar functions from a code retrieval base to enhance the neural generation. **ClassGAT** (Yu et al., 2020) employs a local bi-GRU encoder and a global GNN encoder to obtain two different levels of function representation. The encoder outputs are fed into a GRU decoder with an attention and copy mechanism. **CodeBERT** (Feng et al., 2020) is a pre-trained model that can be adapted to a variety of NL-PL applications. **GypSum** (Wang et al., 2022) learns representations from source codes and ASTs using a pre-trained encoder and a GAT encoder. The encoding results are fused in a Transformer decoder to generate comments.

with Known Comments The baseline models mentioned above only work on the source code itself, whereas our approach incorporates known comments additionally. To explore the influence of known comments, we perform a modification that introduces them into multiple baselines. For Seq2Seq, we produce a comment by combining the target function representation and the weighted

sum of known comment representations. Towards ClassGAT, we take the initial node representation as (i) a concatenation or (ii) a weighted sum of the function representations and their corresponding comment representations, then report the best performance. As for CodeBERT, we set its input as a concatenation of the target function and known comments within the class.

with CodeBERT We also incorporate CodeBERT into our model for verifying whether function relations still provide benefits when employing a strong pre-training model. Specifically, we use the CodeBERT and a transformer decoder to replace the bi-GRU encoder and the GRU decoder, respectively.

4.3 Implementation Details

The value of threshold α is set to 0.7. Word embeddings are randomly initialized, the size of embeddings and hidden states are set to 256. Both the encoder and decoder GRUs have a single layer and the GAT has 3 layers. We use Adam (Kingma and Ba, 2015) optimizer to train our model with the weight decay rate being $1e-6$. We set the learning rate to $1e-4$ and the dropout (Srivastava et al., 2014) rate is 0.3. There is also a scheduler that reduce learning rate when the BLEU on the validation set stops improving for 3 epochs, and the learning rate will not be less than $1e-6$. All our experiments were trained on Nvidia A40 GPUs.

4.4 Evaluation Metrics

We evaluate the quality of generated comments based on BLEU (Papineni et al., 2002) and ROUGE-1, -2, -L (Lin, 2004). We also report 1,2,3,4-gram precisions to determine how many n-grams in the generated text overlap with the reference text. For human evaluation, we invited three experienced raters to score fifty samples randomly selected from the test dataset. For each generated comment, raters assign scores in three aspects: (i) **Fluency**, which measures comment quality in terms of grammaticality and readability; (ii) **Relevance**, which examines whether the generated comment accurately summarizes the functionality of the code snippet; (iii) **Informativeness**, which evaluates whether the comment offers concrete information that is free of redundancy or repetition. These human evaluation metrics have a scale of 0 to 2 (where 2 indicates highly satisfied and 0 means highly unsatisfied).

Model	Degree-Sampling							
	BLEU	p_1	p_2	p_3	p_4	ROUGE-1	ROUGE-2	ROUGE-L
RandomCopy	13.08	30.0	14.4	9.2	7.4	30.72	14.04	29.41
MaxCopy	14.65	32.2	16.2	10.4	8.4	33.60	16.20	32.26
NNGen	16.11	29.1	16.4	13.1	12.1	30.29	17.68	29.48
Seq2Seq	15.10	37.1	18.1	11.9	9.9	37.49	18.47	36.05
ASTGCN	16.05	39.5	18.9	12.2	9.6	41.76	20.65	39.71
Rencos	16.08	39.1	20.9	14.4	12.3	37.83	20.06	36.63
ClassGAT	17.38	40.7	20.5	13.6	11.1	42.70	21.65	40.51
CodeBERT	18.29	46.9	25.2	16.8	13.5	45.00	24.01	43.25
GypSum	18.95	44.6	23.9	15.5	12.0	45.44	24.22	43.60
Seq2Seq+KC	16.51	39.4	20.4	12.9	10.6	39.82	20.76	38.37
ClassGAT+KC	18.52	42.8	22.1	14.9	12.6	43.34	22.67	41.05
CodeBERT+KC	19.64	51.9	29.0	18.3	13.6	49.95	27.54	47.87
Ours	21.39	47.3	26.6	18.6	15.9	46.78	25.72	44.87
+ CodeBERT	25.60	51.9	31.8	22.8	18.8	51.87	31.54	49.89

Table 2: Comparison between our model and baselines. "KC" refers to the known comments.

Model	Fluency	Relevance	Informativeness
Seq2Seq	1.19 (± 0.86)	0.72 (± 0.75)	0.93 (± 0.79)
ClassGAT	1.27 (± 0.82)	0.81 (± 0.75)	1.01 (± 0.76)
CodeBERT	1.39 (± 0.78)	1.13 (± 0.77)	1.31 (± 0.75)
Ours	1.54 (± 0.77)	1.21 (± 0.87)	1.36 (± 0.72)

Table 3: Results of human evaluation (standard deviation in parentheses).

5 Results and Analysis

5.1 Automatic Evaluation

The comparative results are summarized in Table 3. Overall, our model outperforms all baselines by a large margin. Retrieval-based models have relatively poor results since they do not adequately exploit the semantic information of the source code. In comparison, generation-based models perform better. ASTGCN surpasses Seq2Seq by incorporating structural information from the AST. Rencos and ClassGAT improve their performance with the assistance of external information. CodeBERT and GypSum exceed other baselines by utilizing their extensive pre-training knowledge. After aggregating related contextual information, our model outperforms all baseline models. This suggests that considering function relations is an effective way to enhance the comprehension of the target function.

We discover that introducing known comments can improve the performance of some baselines. As shown in Table 2, all of the methods achieve an improvement on BLEU and ROUGE. Due to the vast knowledge gained during the pre-training process, CodeBERT significantly improves ROUGE-L from 43.25 to 47.87 (+ 4.62%). This result suggests that known comments from the class context can help with code comment generation. We also ana-

lyze the effect of known comments on our model in the Appendix C.

Although these models present competitive performance with the incorporation of known comments, our model equipped with CodeBERT still manages to make a further improvement and achieves the best BLEU and ROUGE scores among all the involved models. This shows that combining our framework with the pre-trained model can effectively absorb both the contextual information and the pre-training knowledge, which allows our approach to collaborate with more advanced pre-trained models in the future.

5.2 Human Evaluation

We further conduct human evaluation to assess the quality of comments generated by different models, as shown in Table 3. Our model surpasses the baseline models on all metrics. The Seq2Seq model has a much lower score than others, because it only utilizes local information contained in the source code, whereas other models incorporate contextual information or pre-training knowledge as well. Since the by-pointer mechanism enables our model to copy tokens from both the source code and known comments, it significantly improves the fluency of generated comments. Besides, the highest relevance and informativeness score indicates that our model can effectively summarize the behavior of a given function.

5.3 Ablation Study

To examine the contribution of components in our framework, we evaluate the performance after removing each of them, as shown in Table 4. We discover that removing any of the modules has a neg-

Model	BLEU	p_1	p_2	p_3	p_4	ROUGE-1	ROUGE-2	ROUGE-L
Ours	21.39	47.3	26.6	18.6	15.9	46.78	25.72	44.87
w/o function encoder	16.39	42.2	21.4	13.1	10.0	43.45	22.44	41.40
w/o GAT encoder	16.75	44.3	22.0	13.4	10.1	45.09	22.20	42.84
w/o target-aware attention	19.35	46.3	24.7	16.2	12.9	46.41	24.72	44.24
w/o cross-graph attention	18.81	45.0	24.1	15.8	12.6	45.14	23.74	43.06
w/o by-pointer mechanism	20.27	46.9	25.7	17.4	14.7	46.58	25.20	44.53

Table 4: Ablation study results of our approach.

ative impact on the model performance. Without the function encoder or GAT encoder, the performance drops significantly, suggesting that both are critical components in our framework. Removing the target-aware attention or cross-graph attention mechanism also results in a noticeable performance degradation, indicating that both mechanisms contribute to overall performance. Besides, we observe a slight drop in performance without the by-pointer mechanism, confirming that this component can effectively copy tokens from the source code and known comments to improve comment generation.

5.4 Threshold α

The hyper-parameter α determines the lower limit of TF-IDF similarity scores. To explore how model performance varies with α , we run a series of experiments with different values of α , while keeping other hyper-parameters constant. Fig. 3 shows the corresponding results. It illustrates that $\alpha = 0.7$ achieves peak performance in both BLEU and ROUGE-L. When α is equal to 0.5 or 0.9, our model performs poorly in both metrics. This may be because when the α is too small or too large, there are too many or too few functions associated with the target function, and the model is unable to make effective use of contextual information.

5.5 Sampling Settings

To investigate the impact on our approach when programmers select functions to be commented in different ways, we conduct a series of experiments under random and degree sampling settings. The experimental results are reported in Fig. 4. Compared to the competitive baselines, our model presents better performance under both settings. Since our model is able to capture function relations within a class, even randomly commenting functions can help improve the quality of generated comments. Furthermore, it clearly shows that our model performs much better under degree sampling than random sampling, due to the reason that commenting functions with higher degrees can ben-

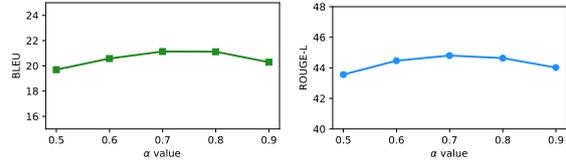


Figure 3: Performance of our model with different threshold α .

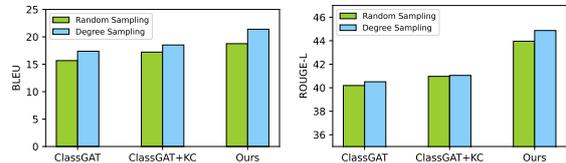


Figure 4: Performance of different models under random sampling and degree sampling.

Model	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L
Ours	21.39	46.78	25.72	44.87
w/o TF-IDFSim	19.21	46.11	24.64	43.95
w/o R1	19.70	45.86	24.09	43.81
w/o R2	20.52	46.48	25.67	44.37
w/o R3	20.28	46.22	24.98	44.03
w/o R4	20.48	46.39	25.09	44.30
w/o R5	21.09	46.68	25.72	44.54
fully connected	17.73	45.60	23.73	43.42

Table 5: Impact of different types of edges.

efit more functions in the same class. This finding implies that when engineering in the real world, it may be a good idea to start by writing comments for functions with higher degrees.

5.6 Relation Types

To examine the utility of edges constructed by TF-IDF similarity and five types of inductive relations, we remove them from the class graph to observe how it affects model performance, as presented in Table 5. It demonstrates that removing either type of edge leads to a drop in BLEU and ROUGE scores. Specifically, the performance degradation is greatest after dropping edges of type TF-IDFSim and R1, indicating that they are the most influential relation types. While removing R5 edges has the least impact on model performance, this can

	Case 1	Case 2
Target Function	<pre>public boolean isCurrent() { boolean result = false; if (this.getNumber().intValue() == EasyCalendar.getOne().getCurrentYear()) { result = true; } return result; }</pre>	<pre>public Vector parse(final String str, char separator) { if (str == null) { return new Vector(); } return parse(str.toCharArray(), separator); }</pre>
Class Context	<p>Related Function:</p> <pre>public Month getCurrentMonth() { Month month = null; if (isCurrent()) { int currentMonthNumber = EasyCalendar.getOne() .getCurrentMonth(); month = getMonth(currentMonthNumber); } return month; }</pre> <p>Known Comment: gets the current month, but only if this year is the current one</p>	<p>Related Function:</p> <pre>public Vector parse(final char[] chars, int offset, int length, char separator) { if (chars == null) { return new Vector(); } Vector params = new Vector(); return params; }</pre> <p>Known Comment: extracts a list of name value pairs from the given array of characters</p>
Graph		
Golden	checks if this year is the current year	extracts a list of name value pairs from the given string
Seq2Seq	gets the value of the attribute	parses a string from the given
ClassGAT	returns true if the current has the desired result	extracts a character value at the given character
CodeBERT	returns the current year	extracts a vector from the string buffer
Ours	returns true if this year is current one	extracts a list of name value pairs from the given string

Table 6: Examples of the source codes, graph structure and generated comments. "T" refers to the target function and "C" refers to the commented function in the class context.

be attributed to the low occurrence of this relation type in the dataset. Furthermore, we conduct an experiment to investigate the impact of exploiting contextual information in a crude manner. To be more specific, rather than modeling functional relations, we construct a fully connected graph to introduce the entire class context. Our model suffers greatly as a result of this operation, with the BLEU and ROUGE-L dropping 3.66% and 1.45%, respectively. This performance loss verifies the effectiveness of our design for utilizing class-level contextual information.

5.7 Case Study

Table 6 shows two examples of generated code comments. In the first case, there is a commented function in the class that is the caller of the target function. The second half of its comment provides an accurate description of the target function `isCurrent`. Although there is no direct connection between these two functions in the comment graph, our model is still able to extract information from the known comment through the cross-graph attention mechanism and generate a high-quality comment. In contrast, baseline models do not capture the true functionality of the source code and their generation results has a large deviation from the original intention.

In the second case, it is difficult to figure out the purpose of target function solely from the source code. Since there is a defined (R1) pattern between the target function and a commented function from the class context, the constructed edge in the comment graph allows our model to aggregate comment of this related function. Therefore, our model successfully generates "a list of name value pairs", whereas other models fail to capture this key information and produce meaningless comments. Moreover, it is worth noting that our model is unaffected by irrelevant information in the known comment, yielding the true object "string" rather than "array of characters". The final output of our model is exactly same as the human-written comment.

6 Conclusion

In this paper, we propose a graph-based learning framework for code comment generation. Our approach targets a practical scenario where only a few functions in the class file have human-written comments. To identify valuable information from the class context, we model function relations and develop a graph attention network to aggregate class-level contextual information. We conducted experiments on Java programs collected from real-world projects and the results demonstrate that our approach outperforms prior methods.

Limitations

There are four main limitations of our work. First, we only evaluate our model on Java code snippets. Although we expect that our approach could be generalized to other programming languages, further experiments is required to confirm this hypothesis. Second, our model does not utilize syntactic information (e.g. ASTs) of the source code. Thus, our next effort will incorporate this type of information into our framework to advance comment generation. Third, we do not employ Transformers in our approach due to limited resources. This will also be left to our future work. Fourth, in comparison with the widely used datasets TL-CodeSum (Hu et al., 2018b), CodeSearchNet (Husain et al., 2019) and Funcom (LeClair et al., 2019b), the size of our collected dataset is relatively small (Table 7). A large-scale code comment generation dataset that retains class structure information is needed in future studies.

Dataset	Ours	TL-CodeSum	CodeSearchNet	Funcom
Examples	40,328	87,136	496,688	2.1 M

Table 7: Number of dataset examples.

Acknowledgements

This work is supported in part by National Key R&D Program of China (No. 2020AAA0106600) and NSFC (62161160339). We would like to thank the anonymous reviewers for their insightful comments and helpful suggestions.

References

- Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2020. [A transformer-based approach for source code summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4998–5007. Association for Computational Linguistics.
- Miltiadis Allamanis, Marc Brockschmidt, and Mahmoud Khademi. 2018. [Learning to represent programs with graphs](#). In *Proceedings of the International Conference on Learning Representations*.
- Uri Alon, Omer Levy, and Eran Yahav. 2019. [code2seq: Generating sequences from structured representations of code](#). In *Proceedings of the 7th International Conference on Learning Representations*.
- Sushil Bajracharya, Joel Osher, and Cristina Lopes. 2014. [Sourcerer: An infrastructure for large-scale collection and analysis of open-source code](#). *Sci. Comput. Program.*, 79:241–259.
- Aakash Bansal, Sakib Haque, and Collin McMillan. 2021. [Project-level encoding for neural source code summarization of subroutines](#). In *Proceedings of the IEEE/ACM 29th International Conference on Program Comprehension (ICPC)*, pages 253–264.
- Lionel C. Briand. 2003. [Software documentation: how much is enough?](#) In *Proceedings of the 7th European Conference on Software Maintenance and Reengineering*, pages 13–15. IEEE.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. [Learning phrase representations using RNN encoder–decoder for statistical machine translation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics.
- Brian P. Eddy, Jeffrey A. Robinson, Nicholas A. Kraft, and Jeffrey C. Carver. 2013. [Evaluating source code summarization techniques: Replication and expansion](#). In *Proceedings of the 21st International Conference on Program Comprehension (ICPC)*, pages 13–22. IEEE.
- Zhangyin Feng et al. 2020. [CodeBERT: A pre-trained model for programming and natural languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1536–1547. Association for Computational Linguistics.
- Daya Guo, Shuai Lu, Nan Duan, Yanlin Wang, Ming Zhou, and Jian Yin. 2022. [Unixcoder: Unified cross-modal pre-training for code representation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 7212–7225. Association for Computational Linguistics.
- Sonia Haiduc, Jairo Aponte, and Andrian Marcus. 2010a. [Supporting program comprehension with source code summarization](#). In *Proceedings of the IEEE/ACM 32nd International Conference on Software Engineering, ICSE ’10*, pages 223–226. Association for Computing Machinery.
- Sonia Haiduc, Jairo Aponte, Laura Moreno, and Andrian Marcus. 2010b. [On the use of automated text summarization techniques for summarizing source code](#). In *Proceedings of the 17th Working Conference on Reverse Engineering, WCRE ’10*, pages 35–44. IEEE Computer Society.
- Sakib Haque, Alexander LeClair, Lingfei Wu, and Collin McMillan. 2020. [Improved automatic summarization of subroutines via attention to file context](#). In *Proceedings of the 17th International Conference on Mining Software Repositories, MSR ’20*, pages 300–310. Association for Computing Machinery.

- Xing Hu, Ge Li, Xin Xia, David Lo, and Zhi Jin. 2018a. [Deep code comment generation](#). In *Proceedings of the IEEE/ACM 26th International Conference on Program Comprehension, ICPC '18*, pages 200–210. Association for Computing Machinery.
- Xing Hu, Ge Li, Xin Xia, David Lo, Shuai Lu, and Zhi Jin. 2018b. [Summarizing source code with transferred api knowledge](#). In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI'18*, page 2269–2275. AAAI Press.
- Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. [Code-searchnet challenge: Evaluating the state of semantic code search](#).
- Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2016. [Summarizing source code using a neural attention model](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2073–2083. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *Proceedings of the 3th International Conference on Learning Representations*.
- Thomas N. Kipf and Max Welling. 2017. [Semi-supervised classification with graph convolutional networks](#). In *Proceedings of the 5th International Conference on Learning Representations*.
- Alexander LeClair, Sakib Haque, Lingfei Wu, and Collin McMillan. 2020. [Improved code summarization via a graph neural network](#). In *Proceedings of the IEEE/ACM 28th International Conference on Program Comprehension, ICPC '20*, pages 184–195. Association for Computing Machinery.
- Alexander LeClair, Siyuan Jiang, and Collin McMillan. 2019a. [A neural model for generating natural language summaries of program subroutines](#). In *Proceedings of the IEEE/ACM 41st International Conference on Software Engineering, ICSE '19*, pages 795–806. IEEE Press.
- Alexander LeClair, Siyuan Jiang, and Collin McMillan. 2019b. [A neural model for generating natural language summaries of program subroutines](#). In *Proceedings of the 41st International Conference on Software Engineering, ICSE '19*, page 795–806. IEEE Press.
- Yuding Liang and Kenny Q. Zhu. 2018. [Automatic generation of text descriptive comments for code blocks](#). In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. AAAI Press.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Proceedings of the Workshop on Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.
- Shangqing Liu, Yu Chen, Xiaofei Xie, Jingkai Siow, and Yang Liu. 2021. [Retrieval-augmented generation for code summarization via hybrid GNN](#). In *Proceedings of the 9th International Conference on Learning Representations*.
- Zhongxin Liu, Xin Xia, Ahmed E. Hassan, David Lo, Zhenchang Xing, and Xinyu Wang. 2018. [Neural-machine-translation-based commit message generation: How far are we?](#) In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE 2018*, page 373–384. Association for Computing Machinery.
- Paul W. McBurney and Collin McMillan. 2014. [Automatic documentation generation via source code summarization of method context](#). In *Proceedings of the 22nd International Conference on Program Comprehension, ICPC 2014*, pages 279–290. Association for Computing Machinery.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: A method for automatic evaluation of machine translation](#). In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.
- Paige Rodeghero, Collin McMillan, Paul W. McBurney, Nigel Bosch, and Sidney D’Mello. 2014. [Improving automated source code summarization via an eye-tracking study of programmers](#). In *Proceedings of the 36th International Conference on Software Engineering, ICSE 2014*, pages 390–401. Association for Computing Machinery.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083. Association for Computational Linguistics.
- Giriprasad Sridhara, Emily Hill, Divya Muppaneni, Lori Pollock, and K Vijay-Shanker. 2010. [Towards automatically generating summary comments for java methods](#). In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, ASE '10*, pages 43–52. Association for Computing Machinery.
- Nitish Srivastava et al. 2014. [Dropout: A simple way to prevent neural networks from overfitting](#). *The journal of machine learning research*, 15(1):1929–1958.
- Fei Sun, Peng Jiang, Hanxiao Sun, Changhua Pei, Wenwu Ou, and Xiaobo Wang. 2018. [Multi-source pointer network for product title summarization](#). In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18*, pages 7–16. Association for Computing Machinery.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. [Sequence to sequence learning with neural networks](#).

In *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS'14*, pages 3104–3112. MIT Press.

Y. Wang, Y. Dong, X. Lu, and A. Zhou. 2022. [Gypsum: Learning hybrid representations for code summarization](#). In *2022 IEEE/ACM 30th International Conference on Program Comprehension*, pages 12–23.

Bolin Wei, Yongmin Li, Ge Li, Xin Xia, and Zhi Jin. 2020. [Retrieve and refine: Exemplar-based neural comment generation](#). In *Proceedings of the IEEE/ACM 35th International Conference on Automated Software Engineering, ASE '20*, pages 349–360. Association for Computing Machinery.

Xin Xia, Lingfeng Bao, David Lo, Zhenchang Xing, Ahmed E. Hassan, and Shanping Li. 2018. [Measuring program comprehension: A large-scale field study with professionals](#). In *Proceedings of the 40th International Conference on Software Engineering, ICSE '18*, page 584. Association for Computing Machinery.

Xiaohan Yu, Quzhe Huang, Zheng Wang, Yansong Feng, and Dongyan Zhao. 2020. [Towards context-aware code comment generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3938–3947. Association for Computational Linguistics.

Jian Zhang, Xu Wang, Hongyu Zhang, Hailong Sun, and Xudong Liu. 2020. [Retrieval-based neural source code summarization](#). In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering, ICSE '20*, pages 1385–1397. Association for Computing Machinery.

Kechi Zhang, Wenhan Wang, Huangzhao Zhang, Ge Li, and Zhi Jin. 2022. [Learning to represent programs with heterogeneous graphs](#). In *Proceedings of the IEEE/ACM 30th International Conference on Program Comprehension*, pages 378–389.

Daniel Zügner, Tobias Kirschstein, Michele Catasta, Jure Leskovec, and Stephan Günnemann. 2021. [Language-agnostic representation learning of source code from structure and context](#). In *Proceedings of the 9th International Conference on Learning Representations*.

A Examples of the inductive relation

Table 8 shows examples that correspond to the five inductive relation rules defined in Section 3.1.

B Dataset

In order to better suit the scenario of our task, only well-commented JAVA classes are retained, which means classes containing more than three functions and at least 70% of them have manually written comments. The detailed statistics of our dataset are

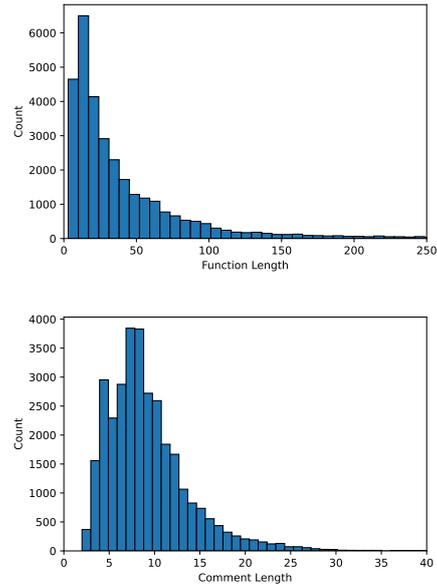


Figure 5: Length distribution of functions and comments in the dataset

shown in Table 9. Figure 5 shows the length distribution of target function and golden comment of our dataset. The length of functions is basically less than 100 and the length of comments are mainly between 3 and 25.

The data is preprocessed in following ways: for each function, (1) we extract the summative content in the Javadoc and take the first sentence as the comment; (2) we remove all the format controlling tokens and only retain comments having at least three words; (3) we serialize the function-comment pairs, remove non-alphabetical characters, and split tokens written in camelCase or underscore style; (4) we truncate the source code sequences to 200 tokens.

C Known Comments

As shown in the section 5.1, incorporating known comments into comment generation can significantly improve baseline model performance. To further demonstrate the effect of known comments, we remove them from our model and evaluate model performance. Specifically, we use function names instead of known comments for comment nodes during the graph construction step while leaving other settings unchanged. The experimental results are shown in Table 10. It illustrates that removing known comments reduces BLEU and ROUGE-L scores by 4% and 3%, respectively, in-

	Function 1	Function 2
Rule 1	<pre> /* Adds a service to the framework. */ public void add Service(Service service) { Settings settings = service.getSettings(); servicesMap.put(settings.getName(), service); settingsMap.put(settings.getName(), settings); } </pre>	<pre> /* Removes a service from the framework. public remove Service(Service service) { Settings settings = service.getSettings(); servicesMap.remove(settings.getName()); settingsMap.remove(settings.getName()); } </pre>
Rule 2	<pre> /* Get degrees of latitude in different formats. */ public String get LatDeg(int format) { switch (format) { case DD : return Double.toString(this.getLatDec()); ... case DMS : return getDMS(getLatDec(), 0, format); default : return ""; } } </pre>	<pre> /* Get degrees of longitude in different formats. */ public String get LonDeg(int format) { switch (format) { case DD : return Double.toString(this.getLonDec()); ... case DMS : return (((getLonDec() < 100.0) && (getLonDec() > -100.0)) ? "0" : "") + getDMS(getLonDec(), 0, format); default : return ""; } } </pre>
Rule 3	<pre> /* Method to calculate the bearing of a waypoint. */ public double get Bearing(CWPoint dest) { if (!this.isValid() dest == null !dest.isValid()) return 361; return GeodeticCalculator.calculateBearing(TransformCoordinates.WGS84, this, dest); } </pre>	<pre> /* Method to calculate the distance to a waypoint. */ public double get Distance(CWPoint dest) { ... return GeodeticCalculator.calculateDistance(TransformCoordinates.WGS84, this, dest) / 1000.0; } </pre>
Rule 4	<pre> /* Returns the Action at the specified index. */ public Action get(int i) { ... return (Action)m_actions.get(i); } </pre>	<pre> /* Removes the Action at the specified index. */ public Action remove(int i) { ... return (Action)m_actions.remove(i); } </pre>
Rule 5	<pre> /* Add a Log to the list. */ public int add(Log log) { resetRecommendations(); if (log != null && log.getLogType() != null) { return merge(log); } return -1; } </pre>	<pre> /* Get the Log at a certain position in the list. */ public Log getLog(int i) { ... return logList.get(i); } </pre>

Table 8: Examples of function pairs with the inductive relation

Item	Number
Classes	3,344
Functions	40,328
Training examples	25,247
Validation examples	3,900
Test examples	2,770
Avg functions per class	12.4
Avg tokens per function	62.8
Avg tokens per comment	8.14

Table 9: Statistics of Our Dataset

Model	BLEU	p_4	R-1	R-2	R-L
Ours	21.39	15.9	46.78	25.72	44.87
w/o KC	17.55	11.4	43.88	22.67	41.79

Table 10: Effect of known comments.

dicating that known comments are quite essential for our model to capture code features and generate accurate comments.

Spelling convention sensitivity in neural language models

Elizabeth Nielsen[†] Christo Kirov[°] Brian Roark[°]

[†]School of Informatics, University of Edinburgh, UK [°]Google
e.nielsen@ed.ac.uk {ckirov, roark}@google.com

Abstract

We examine whether large neural language models, trained on very large collections of varied English text, learn the potentially long-distance dependency of British versus American spelling conventions, i.e., whether spelling is consistently one or the other within model-generated strings. In contrast to long-distance dependencies in non-surface underlying structure (e.g., syntax), spelling consistency is easier to measure both in LMs and the text corpora used to train them, which can provide additional insight into certain observed model behaviors. Using a set of probe words unique to either British or American English, we first establish that training corpora exhibit substantial (though not total) consistency. A large T5 language model does appear to internalize this consistency, though only with respect to observed lexical items (not nonce words with British/American spelling patterns). We further experiment with correcting for biases in the training data by fine-tuning T5 on synthetic data that has been debiased, and find that finetuned T5 remains only somewhat sensitive to spelling consistency. Further experiments show GPT2 to be similarly limited.

1 Introduction

The probabilities that neural language models (LMs) assign to strings can be used to assess how effectively they capture linguistic dependencies found in their training data. Much as in psycholinguistic experiments on human language speakers, we can present LMs with strings both with and without agreement in key dependencies and measure the assigned probabilities to determine whether the model has learned these linguistic generalizations or not (see e.g., Futrell et al. 2018). For example, sentences both with and without subject/verb number agreement (but otherwise identical) can be used to assess whether the model accounts for that particular dependency, even over

long distances. Various long-distance dependencies have been investigated in this manner, from purely linguistic phenomena such as syntactic dependencies (e.g., Gulordava et al. 2018) to extralinguistic phenomena such as socio-cultural biases (e.g., Rudinger et al. 2018).

In this paper, we examine dependencies based on orthographic cues to language variety. Many LMs are trained on large corpora scraped from the web, and data from different language varieties are often combined. For example, LMs trained on web-scraped English (e.g., the WebText Corpus of Radford et al. 2019) encounter British English, North American English, and multiple World Englishes. Likewise, Spanish web corpora may include several distinct varieties of Latin American Spanish, as well as Iberian Spanish (e.g., Kilgarriff and Renau 2013). Here we use differences between British and American English spelling conventions to ask whether LMs trained on large and diverse collections of English learn to apply these conventions consistently within the same span of text. For example, if the British spelling of the word *labour* appears in a sentence prefix, will the LM assign higher probabilities to continuations that maintain British spelling conventions (e.g., *organisation*) over those that have American-spelled forms (*organization*)? To the extent that such models are used within response generation systems or for next word prediction in virtual keyboards, maintaining such consistency would be strongly desirable so users receive results appropriate for their locale.

Of course, as with any such dependencies, models can only learn generalizations that are present in the data, so we also look at the degree to which corpora used to train the large LMs (LLMs) that we investigate (as well as a few others) demonstrate spelling convention consistency. Assessing whether syntactic or semantic generalizations are learned by models trained on noisy, errorful and inconsistent data is complicated by the difficulty in

quantifying the actual degree of consistency of the dependency in the data itself. In contrast to structural linguistic generalizations or other implicit information, the explicitness of spelling conventions permits straightforward corpus analysis in addition to model probing, providing another avenue for explaining model performance.

The results of our data analysis are presented in §4. We find that relevant web-scraped English text used to train LLMs unsurprisingly does not provide perfect consistency — and further that it is heavily skewed towards American spelling conventions — but that it provides as much or more consistency than some curated corpora such as the British National Corpus (BNC Consortium, 2007). We then present methods, in §5, to measure the degree to which two neural LLMs – T5 (Raffel et al., 2020) (both with and without additional finetuning) and GPT2 – exhibit spelling variation consistency. We find that T5 without finetuning demonstrates a general preference for consistency, but that this preference is weaker for British than American English and does not extend robustly to nonce words. Finetuning T5 on a synthetically modified portion of the British National Corpus reduces the preference for American English. We then modify our conditional probability calculations to allow demonstration of similar patterns of model behavior for GPT2, a very differently architected and trained LLM (Radford et al., 2019). Lastly, in §6, we take a slightly deeper dive into the kinds of (and reasons for) spelling convention inconsistencies in some corpora analyzed in §4.

Overall, we demonstrate that, while T5 and GPT2 display some sensitivity to spelling convention differences, this cannot be relied on to produce consistent generated output. If reliable spelling consistency is an application requirement, additional post-processing may need to be applied to LLM output.

This paper makes several key contributions. First, we provide methods for straightforwardly assessing the ability of LLMs to capture certain well-attested long-distance dependencies in English, and demonstrate the strengths and shortcomings of two well-known models in doing so. This opens up the possibility of exploratory studies in languages where such conventions are less well documented. In contrast to the most heavily investigated types of long-distance dependencies (e.g., syntactic), the (previously unexplored) dependency of spelling

convention consistency is directly observable in the surface string and hence is relatively easy to assess in both models and data. As a result, it can be seen as a useful task for assessing LM learning in general. We also document the degree to which web-scraped corpora exhibit spelling consistency, making clear that the models have plenty of room for improvement. However, American English is shown to be far more heavily represented in the training corpora than British English, to the point that performance for British English is demonstrably far worse than for American English, something that language generation or word prediction systems must address for equitable performance.

2 Background

2.1 Dependencies and LMs

Much of the work investigating whether large language models capture long-distance linguistic generalizations has focused on non-surface dependencies, such as co-reference. In order to correctly identify that two expressions refer to the same entity, models often need to identify complex syntactic relationships (e.g., c-command), or build a model of entities over an entire discourse (e.g., Clark and Manning 2016). Despite this complexity, LLMs have shown some promise as general-purpose co-reference resolvers (Joshi et al., 2019). This suggests that they can learn to model complex long-distance dependencies.

Other research has shown more directly that LLMs model syntactic dependencies. A common methodology is to compare an LM’s surprisal directly to psycholinguistic data (Futrell et al., 2018). If the LM still performs like a human on examples that require modeling hierarchical relationships between tokens, this suggests that the LM has learned some part of the more complex syntactic structure of the language. Work such as Futrell et al. (2018) has shown that a recurrent neural network language model achieves surprisal rates that mimic human processing, including in these syntactically complex situations. This suggests that an RNN LM can be sensitive to complex syntactic relationships as well. Similar methods have been used to show LMs learning syntactic dependencies in Linzen et al. (2016), Frank et al. (2016), and Brennan et al. (2020).

Another class of methods for assessing whether LMs learn complex syntactic dependencies involves probing the models themselves to evaluate

whether syntax-like relationships between tokens can be discovered. Details of their methods vary widely, but Clark et al. (2019), Hewitt and Manning (2019), and Lin et al. (2019) all suggest that many LMs learn complex syntactic dependencies.

In contrast, the topic of the current paper – spelling convention dependencies – is a relatively surface-level dependency. A model does not need to capture the syntactic or semantic relationship between two words in order to evaluate spelling consistency, rather simply their co-occurrence. Given prior results showing that LMs can and do learn complex semantic and syntactic relationships between words, one might expect that a relatively simple dependency like spelling convention should be easy for an LM to learn.

2.2 Spelling variation

As discussed by Berg and Aronoff (2017), the orthography of English has never been regulated by an official body, but has rather emerged dynamically over time. Dictionaries played a key role in settling spelling conventions, with Samuel Johnson’s (1755) dictionary being the key source of contemporary British spelling conventions and Webster’s (1828) dictionary the key source of contemporary American spelling. The latter included spelling reforms such as using the suffix *-or* instead of *-our* for certain words, e.g., *labor* instead of *labour*. These reforms were adopted in American spelling but not in British spelling conventions.

This history makes English an interesting case study for spelling variation in particular. Languages that have historically had centralized regulatory institutions, such as the French or Royal Spanish Academies, have much less purely orthographic variation. For example, despite many lexical differences, there are few spelling differences between Iberian and Latin American Spanish. On the other hand, there are many language situations that have considerably more spelling variation. For example, speakers of South Asian languages that are traditionally written with Brahmic or Arabic scripts often write using the Latin alphabet in contexts like SMS messages and social media (Roark et al., 2020). This kind of informally romanized text presents many spelling variations due to these languages’ lack of orthography in the Latin script. The well-documented nature of English spelling variation and its close ties to standardized regional varieties make it a good initial case study

for whether LLMs learn systematic variation in the data. If so, such models may be useful in more exploratory studies, such as the above-mentioned scenario where no official orthography exists.

As far as we are aware, the issue of spelling convention consistency in language models has not been investigated. Nguyen and Grieve (2020) looked at whether word embeddings are *robust* to spelling variation, not whether generative language models capture spelling consistency. That paper focused mainly on the kinds of variation that arise in informal social media text, but they also examined British versus American spelling. Unsurprisingly, they found that cosine similarity between British and American spelled variants are high relative to other patterns of informal spelling variability.

2.3 Prompting LMs

In the present work, we construct prompts to measure the probability assigned to various tokens by LLMs. In constructing these prompts, we take into account the findings of recent work on prompting LMs. Our work is different from the sort of prompting described by these papers, which generally includes features such as task-specific prefixes containing instructions (e.g., Raffel et al. 2020), verbalized class labels (e.g., Schick and Schütze 2021), or in-context learning (e.g., Brown et al. 2020), none of which are present in our approach. However, work such as Webson and Pavlick (2022) has shown large effects due to small variations in the wording of prompts, even if the reasons for these effects are not apparent. Therefore, we choose to present the model with several different prompts and average the probabilities over all prompts, in order to account for possible variation.

3 Data and models

To assess the spelling convention consistency of data and models, we use a list of British and American English spelling differences that is part of the open-source American British English Translator.¹ We used the 1706 word pairs in the `data/american_spellings.json` file at that site. This list includes American and British spelling variants for words with common differences such as *-or/-our* (e.g., *vapor/vapour*), *-ize/-ise* (*realize/realise*), consonant doubling (*modeling/modelling*), *-er/-re* (*liter/litre*), along with

¹<https://github.com/hyperreality/American-British-English-Translator>

some number of term-specific spelling differences (aluminum/aluminium). We use this list to create prompts for probing the language models and to establish the consistency of usage within corpora, i.e., whether strings found in this list consistently follow one convention or the other when they co-occur.

For model probing, we examine T5 (Raffel et al., 2020), a general purpose encoder-decoder model. We use the t5-large architecture variant on the T5X codebase,² which has approximately 770M parameters. For English, T5 is (pre-)trained using a span corruption objective on the Colossal Clean Crawled Corpus (C4), an English language collection derived from Common Crawl (Raffel et al., 2020).³

We also examine GPT2, for which we use the open-source HuggingFace implementation (Radford et al., 2019). Unlike T5, GPT2 is a purely autoregressive language model rather than an encoder-decoder sequence-to-sequence model. It is trained to perform next-word prediction rather than fill in corrupted spans of text. GPT2 is built on OpenAI’s WebText corpus (Radford et al., 2019), of which there is an open-source variant available.⁴

We examine C4 and OpenAI’s WebText corpus for spelling convention consistency, along with several other corpora: English Wikipedia (downloaded 06-21-2020); the Billion Word Benchmark (Chelba et al., 2013), which is a collection of newswire text; and the British National Corpus (BNC Consortium, 2007),⁵ which is a balanced corpus of both written and spoken material.⁶

4 Training corpora consistency

To examine spelling consistency in training data, we made use of the list of spelling variants and the five corpora mentioned in Section 3: C4, the OpenWebText Corpus (OWT), English Wikipedia (EngWiki), the Billion Word Benchmark (BWB), and the British National Corpus (BNC). We convert all strings in each corpus to lowercase, and treat all characters outside of the a–z range as whitespace for tokenization. We look for exact matches of list items in the resulting whitespace-delimited tokens.

²<https://github.com/google-research/t5x/blob/main/docs/models.md#t5-checkpoints>

³<http://commoncrawl.org/>

⁴<https://skylion007.github.io/OpenWebTextCorpus/>

⁵<http://www.natcorp.ox.ac.uk/>

⁶Code for querying corpora and generating prompts, as well as other relevant data and code, can be found at https://github.com/google-research/google-research/tree/master/spelling_convention_nlm.

Corpus	total # of word pairs	X-matched %		
		US	UK	mis
C4	542,755,756	74.6	14.7	10.8
OWT	42,255,261	79.7	11.5	8.8
EngWiki	1,527,529	58.0	26.5	15.4
BWB	442,733	67.5	23.6	8.9
BNC	74,072	14.5	64.8	20.8

Table 1: Study of word pairs found in the same string from either UK or US spelling list over corpora of different sizes and characteristics, with percent of US-matched, UK-matched and mismatched US/UK pairs.

Let V_{US} be the US spelling variants⁷ of the words in the list and V_{UK} the UK spelling variants. For each corpus C , let $s^k = w_1 \dots w_{|s^k|}$ represent the k th string in the corpus, consisting of $|s^k|$ words. We extract all pairs of words (w_i, w_j) from s^k such that $i < j$ and $w_i, w_j \in V_{US} \cup V_{UK}$. Each extracted pair (w_i, w_j) is placed into one of three classes: the pair is (1) *US-matched* if $w_i, w_j \in V_{US}$; (2) *UK-matched* if $w_i, w_j \in V_{UK}$; and (3) *mismatched* otherwise. We then aggregate the counts for pairs in these three bins across all strings in the corpus.

Table 1 presents the number of pairs extracted from each corpus and the percentage of those within each class. Several things jump out from these results. First, all of the corpora, other than the British National Corpus, have significantly more US-matched pairs than UK-matched pairs, with OWT and C4 being the most skewed towards US-matched pairs. This likely indicates a heavy overall skew towards US spelling variants, leading to a high prior probability of US spelling variants in LLMs. Second, the percentage of extracted pairs that are mismatched are non-negligible, however there is a lot of consistency. For example, in the C4 corpus, if a word from V_{UK} is the first word of a pair, the probability that the next word will also be from V_{UK} is nearly three times the probability that it is from V_{US} .⁸ Finally, both English Wikipedia and the British National Corpus have somewhat elevated levels of mismatch compared to the other corpora, something we look at more closely in Section 6.

Having established that the level of mismatch in the C4 corpus used to train T5 is at the lower end

⁷For convenience, we use US as shorthand for American and UK as shorthand for British.

⁸Mismatched pairs in all corpora are roughly equally split between having V_{US} or V_{UK} words first. Hence, for C4, 5.4% of pairs are V_{UK} followed by V_{US} words (half of the mismatched probability), while 14.7% are V_{UK} followed by V_{UK} .

observed in the data we examined,⁹ we now move on to examine whether the trained models pick up on these dependencies.

5 Language model consistency

From the dictionary presented in Section 3, we kept only the words that can be described by a small number of rules, e.g., the variation between *-ize* and *-ise*, etc, leaving us with 1266 options. For efficiency, we sample $\approx 16k$ prompt-target pairs (16028) from all possible 1266^2 combinations.

To eliminate all sources of variation besides the pair of words being tested, we created several template sentences into which we can insert pairs of words. The full set of templates is presented in Table 9 in Appendix A. Several considerations informed how we formulated the templates so that they work for all the tokens we wanted to test.

First and most obviously, we need to ensure that all tokens in a template are variety-neutral. This ensures that the probability of any of our test words being British or American will not be swayed by any regional bias in the template. While neutrality is difficult to enforce perfectly within a single frame, we hope that by using multiple different templates, we can mitigate unknown sources of bias via averaging.

Second, we need templates that will be syntactically and semantically acceptable, regardless of the inserted tokens. LLMs may assign low probability to tokens that result in grammatically unacceptable or semantically unlikely sentences, and we want to avoid introducing this source of variation. This is challenging, since the tokens we are testing include different parts of speech and come from very different semantic domains, hence there are few contexts where all tokens would be acceptable.

Fortunately, this problem has an analogue in linguistics: linguists interested in detailed phonetic description often elicit tokens in set contexts to eliminate extraneous sources of acoustic variation (Bower, 2015). The approach these linguists often take is to use a template that *mentions* the tokens in question rather than *using* them. We follow this approach, and use templates similar to (1), which contain a list of word mentions.

(1) *My preferred words are ..., ..., and tree.*

⁹We note again the benefit of these explicit surface-level dependencies – we can easily assess the prevalence/consistency of the training data, in contrast to structural dependencies.

We then substitute pairs of words from our dictionary into the spaces marked with ellipses, both with consistent and inconsistent spelling conventions. In other words, given the pair of dictionary entries *realize/realise* and *center/centre*, we use the template above to generate the four test sentences:

- (2) a. US/US: *My preferred words are **realize**, **center**, and tree.*
- b. US/UK: *My preferred words are **realize**, **centre**, and tree.*
- c. UK/US: *My preferred words are **realise**, **center**, and tree.*
- d. UK/UK: *My preferred words are **realise**, **centre**, and tree.*

We use T5 to score the probability of generating the second bolded word, as shown in Example (2), given the first.

In the above template, the two words are adjacent in the string. We also include a non-adjacent condition, which augments the templates by adding ten variety-neutral tokens between the bold-face words. For the above sample, the non-adjacent variant would be:

- (3) *My preferred words are ..., flower, interesting, jump, ponderous, sky, skipping, desk, small, ladder, lovely, ..., and tree.*

Since T5 is a seq2seq model trained on a span-corruption objective, we present a prompt that includes a priming word and a blank span token representing the second word:

- (4) My preferred words are **flavour**, <BLANK-SPAN-1>, and tree

The decoder then scores an output string that replaces the blank, but represents the known inputs with span markers instead:

- (5) <INPUT-SPAN-1> **harbour** <INPUT=SPAN-2>

Thus we are effectively computing the probability that the blank span will be filled with a particular word (with a US or UK spelling), given the visible input sentence (which contains a US or UK primer) — $P(\text{“harbour”} \mid \text{“My preferred words are flavour, ..., and tree.”})$.

We report a few different measures to give a picture of how strongly each model prefers spelling consistency: mean conditional probabilities, prediction accuracy and mutual information. We then

Condition	Word 1	T5		T5+FT		C4	
		Word 2		Word 2		Word 2	
		US	UK	US	UK	US	UK
Adjacent	US	0.86	0.14	0.66	0.34	0.91	0.09
	UK	0.39	0.61	0.44	0.56	0.38	0.62
Non-adjacent	US	0.83	0.17	0.69	0.31	0.93	0.07
	UK	0.48	0.52	0.43	0.57	0.27	0.73

Table 2: Conditional probability of Word 2, given a template with Word 1, given by T5 (no finetuning) and T5+FT (finetuned on synthetic balanced BNC data). For each instance, the probability has been normalized over each condition (corresponding to each row for the model). We also present the conditional probabilities from pairs found in the training corpus C4.

examine behavior with nonce words.

5.1 Measure 1: conditional probability tables

The first measure we use to show the preferences of each model is a 2x2 table of the conditional probability of the second probe word, given the first. For ease of interpretation, we normalize the conditional probabilities for each conditioning word as though the two alternative second words (US and UK) are the only possibilities, i.e., the two conditional probabilities are made to sum to 1. That is, $P(UK|US) + P(US|US) = 1$ and $P(US|UK) + P(UK|UK) = 1$ for each example. These conditional probabilities are then averaged over the whole test corpus (16028 word pairs replicated across 29 template sentences¹⁰ for a total of 464812 samples) for both the adjacent and non-adjacent conditions. Table 2 presents these mean conditional probabilities for base T5 and T5 finetuned (TF+FT) on a synthetic balanced corpus derived from the BNC (see §5.2), alongside conditional probabilities calculated from the pairs extracted for the analysis in Table 1 from their training corpus (C4), under the same adjacent and non-adjacent conditions.¹¹

As can be seen from these results, T5 shows a preference for spelling consistency in both the adjacent and non-adjacent conditions — probabilities for both the consistent US and consistent UK conditions are higher than the probabilities for the respective inconsistent conditions. The differences are notably larger in the adjacent conditions than the non-adjacent conditions, indicating that the preference for spelling consistency attenuates somewhat

¹⁰For information on the variance across prompts, see Appendix A.

¹¹The conditional probabilities from C4 are simply the probability that Word 2 is from the UK or US class given the class of Word 1, with extracted pairs split by whether the words were adjacent or not in the string. Adjacent pairs account for roughly 1% of all pairs in the corpus.

over longer strings. The model also shows a preference for US forms overall, assigning a higher probability to a US form after a UK form than to a UK form after a US form. This is likely due to US forms being over-represented in the training data, leading to high prior probability.

Comparing the model and corpus columns in Table 2, the degree of consistency preference displayed by T5 in the adjacent condition is actually very similar to the consistency levels in the C4 training corpus (similarly replicating the bias for US forms). However, C4 is much more consistent in the non-adjacent condition than T5, indicating that the model is failing to capture some long-distance dependencies.

5.2 Finetuning on synthetic data

Finding naturally occurring English text using perfectly consistent spelling conventions of sufficient size to help improve a model’s consistency may be difficult, given the results presented in Table 1. It would be useful, however, to determine if T5 could be finetuned with some resource to exhibit better spelling consistency. To that end, we created a synthetic version of the BNC, which was changed to exhibit perfect consistency of British and American spelling conventions for the words in our lexicon.

This synthetic BNC corpus was produced as follows. Using our list of spelling variants, we identified strings in the corpus that contained an instance of either the American or British spelling. We then produced a synthetic consistent American spelling version of these strings by using the American spelling of all of the words, along with a synthetic consistent British spelling version of these strings by using the British spelling of all of the words. The resulting corpus is thus balanced between American and British spelling for these 1706 words, and every sentence is consistent in using one convention or the other. In total, the syn-

Condition	Word 1 = US		Word 1 = UK	
	T5	T5+FT	T5	T5+FT
Adjacent	92.2	71.1	65.1	63.4
Non-adjacent	88.7	77.7	54.3	63.2

Table 3: Percent of test set examples for which each model prefers consistent over inconsistent spelling.

thetic corpus contains 954238 sentences,¹² equally split between US and UK spelling conventions. A small random subset of 2560 sentences was reserved for validation, and T5 was finetuned on the rest. Finetuning used the same span-filling masked LM task used for pretraining, with dropout set to 0.1, and the loss normalizing factor set to 233472 as suggested in the T5 documentation. Fine-tuning started at the default T5-large checkpoint, which represents 1000700 steps, and proceeded another 99300 steps at a batch size of 128.

As seen in Table 2, finetuning on this synthetic corpus does not appear to improve overall spelling consistency – quite the opposite. However, it does have at least two interesting effects. First, as might be expected, the overwhelming preference for US English shown by base T5 is reduced. Furthermore, the finetuned model is better able to retain long-distance information — there is no dropoff in consistency between the adjacent and non-adjacent conditions as seen for T5 without finetuning.

5.3 Measure 2: prediction accuracy

While the conditional probabilities in Table 2 show the overall preferences of the models over the test set, we also want a measure that captures how often the LLMs assign consistent pairs a higher probability than inconsistent pairs. In Table 3 we show the percentage of the test set examples for which each model predicted consistency over inconsistency. The results show a similar pattern as the conditional probability measures in Table 2. Again, finetuning lowers overall consistency, but results in less drop-off in non-adjacent vs. adjacent conditions.

5.4 Measure 3: mutual information

We also calculated the average mutual information (MI) across all prompt/target pairs in order to measure the strength of association between spelling conventions in both words. For each pair,

¹²In our testing, this was not enough data to reliably train a T5-large LLM from scratch.

Condition	T5	T5+FT
Adjacent	0.0048	0.0017
Non-adjacent	0.0044	0.0015

Table 4: Average mutual information in the adjacent and non-adjacent conditions.

we calculate four joint probabilities — $P(\text{US}, \text{US})$, $P(\text{US}, \text{UK})$, $P(\text{UK}, \text{US})$, $p(\text{UK}, \text{UK})$. We assume these four probabilities make up the entire universe with respect to a particular prompt/target pair, and normalize them so they sum to 1. This also allows us to easily calculate marginal probabilities simply by adding the appropriate joint probabilities – e.g., $P(\text{US prompt}) = P(\text{US}, \text{US}) + P(\text{US}, \text{UK})$. To calculate MI, we use a formula based on the log-likelihood ratio calculation in Moore (2004), but equivalent to the standard formulation for mutual information, where x, y are the two probe words:

$$\sum_{x \in \{\text{UK}, \text{US}\}, y \in \{\text{UK}, \text{US}\}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

Since T5 is trained on masked token prediction, to measure the joint probability $p(x, y)$ of each pair of probe words x, y we can simply mask both probing tokens and measure the probability of generating both of them. That is, we present T5 with (6-a) and measure the probability of (6-b):

- (6) a. My preferred words are <BLANK-SPAN-1>, <BLANK-SPAN-2>, and tree.
 b. <INPUT-SPAN-1> flavour <INPUT-SPAN-2> harbour <INPUT-SPAN-3>

Table 4 presents these mutual information values. There doesn’t seem to be a significant difference between adjacent and non-adjacent conditions for either T5 variant, though finetuning does seem to cause an overall drop in MI, in line with the overall drop in consistency seen in the measures above.

5.5 Nonce forms

We want to determine if T5 assigns the probabilities reported above on the basis of dependencies between specific lexical items, or if it is learning sub-word generalizations. In other words, does the model learn that specific words like *flavour* and *realise* are more likely to co-occur than *flavour* and *realize*? Or does it learn that words containing *-our* are more likely to co-occur with words containing *-ise*? Since the model is trained using Sentence-

British	American	British	American
glavour	glavor	reptalise	reptalize
mentre	menter	amolirise	amolirize
unulise	unulize	sphectre	sphecter
malvour	malvor	imminise	imminize
larbour	larbor	voitre	voiter

Table 5: Nonce forms created by making one to three changes to words in the American-British dictionary.

		Word 2	
		US	UK
Word 1	US	0.68	0.32
	UK	0.56	0.44

Table 6: Conditional probability table for nonce forms given by T5. The table shows the conditional probability of Word 2 (which is a nonce form), given Word 1. For each instance, the probability has been normalized over each condition (i.e., each row in the table).

Piece tokenization (Kudo and Richardson, 2018), it is possible that it exploits sub-word features.

One way of testing if a model can use sub-word features is to create nonce words that contain British- or American-specific sub-words. If the model treats these as being British or American, this is an indication that the model is able to pick up on sub-word features.

We created a list of ten nonce forms by changing, adding, or removing one to three letters in existing words in our dictionary of American and British forms. These forms are shown in Table 5.

We use the same probing template and method as described above. For each probe, we use a real American or British word for the first probe word, and one of the nonce forms shown in Table 5 for the second. For this experiment we queried the base T5 model in the adjacent condition. The resulting conditional probability table is shown in Table 6.

Table 6 shows that the patterns shown in Section 5.1 above do not generalize very strongly to nonce forms. The probabilities assigned to US forms following UK forms are on average higher than UK forms following UK forms. However, the difference between these alternatives is attenuated compared to when Word 1 is a US form, indicating that (a) there is a heavy skew towards US spelling conditions in the training data; but (b) some sensitivity to the UK context, if not enough to counteract the high US form priors. This suggests that the results in Table 2 are to a large extent driven by lexical

dependencies rather than any lower-level spelling patterns encoded by wordpieces.

5.6 Autoregressive LLMs

Many commonly-used LLMs (including T5) are trained to predict words in the input that have been masked out. Another common class of LLMs, however, are trained to perform next-word prediction instead. To examine how such autoregressive architectures handle spelling consistency, we experiment with OpenAI’s GPT2 (Radford et al., 2019), which has a readily available open-source implementation through HuggingFace.¹³

As GPT2 is purely autoregressive, we cannot compute the probability that a particular probe word will fill a masked sentence span as easily as we could with T5. We can only efficiently compute the probability of a suffix given a prefix. Given this caveat, we have at least two options for assigning conditional probability scores, neither of which should be treated as exactly comparable to the T5 scores above. First, we can count only the logits corresponding to the target word:

$P(\text{“harbour”} \mid \text{“My preferred words are flavour,”})$. This local score ignores any words in the template occurring after the target word. Second, we can compute from the start of the target to the end of the sentence: $P(\text{“harbour, and tree”} \mid \text{“My preferred words are flavour,”})$, which accounts for the post-target suffix of the sentence.

Tables 7 and 8 show results for both of these methods for calculating the conditional probability, compiled in the same way as the T5 results in Tables 2 and 3. Table 7 also includes the conditional probabilities from GPT2’s training corpus, OWT. We see that GPT2 shows a similar preference for consistency as T5, but only very locally. There is a large drop-off in preference for consistency when moving from adjacent to non-adjacent conditions, or when including the completion of the sentence in the calculation. For UK English in particular, any preference for consistency completely disappears beyond the immediate vicinity of the priming word, and the model returns to chance performance on the task.

6 Further analysis of corpora

We now return to a slightly more detailed examination of two of the corpora presented in Table 1,

¹³<https://huggingface.co/gpt2>

Condition	Word 1	GPT2 (tgt only)		GPT2 (to EOS)		OWT	
		Word 2		Word 2		Word 2	
		US	UK	US	UK	US	UK
Adjacent	US	0.87	0.13	0.69	0.31	0.95	0.05
	UK	0.36	0.64	0.51	0.49	0.34	0.66
Non-adjacent	US	0.83	0.17	0.66	0.33	0.95	0.05
	UK	0.49	0.51	0.54	0.46	0.28	0.72

Table 7: Conditional probability of Word 2, given a template with Word 1, given by GPT2 scored until the end of the target word only (tgt only) and scored until the end of the sentence (to EOS). We also present the conditional probabilities from pairs found in the training corpus, OWT.

Condition	Word 1 = US		Word 1 = UK	
	GPT2 target	GPT2 EOS	GPT2 target	GPT2 EOS
Adjacent	94.2	70.1	70.8	49.4
Non-adjacent	92.5	67.5	54.6	45.1

Table 8: Percent of test set examples for which each GPT2 scoring variant prefers consistent over inconsistent spelling.

English Wikipedia and the British National Corpus, both of which had relatively high levels of mismatch compared to the other corpora.

Wikipedia is an interesting case, since the documents are collectively edited by potentially a large number of contributors, which may lead to higher expected mismatch than in other corpora. For example, one version of the article on *air lock* used both US-spelling of the word *vapor* and the UK-spelling (*vapour*). This is explained via three versions of the introductory sentence to the page, shown in Table 11 in Appendix B, where the two spellings are added to the sentence at different times, years apart.

The amount of mismatch in the British National Corpus is perhaps more surprising, given the provenance of the materials and intent of the collection. However the diversity of sources, which include things such as journal articles and edited volumes, likely leads to similar issues to those found in Wikipedia, along with simple human error and/or inconsistency. Table 12 in Appendix B presents a few examples of sentences with words from both spelling conventions, with American *-ize* spellings mixed with British *-ise* or *-our* versions.

7 Conclusion and Future Work

We have presented results showing that T5 does tend towards consistency in spelling, but not to the degree that could be relied upon should such

consistency be desired in generated text. We show that this general preference for consistency reflects the data that the model is trained on, which also is mostly consistent, but with a significant proportion of exceptions. The model’s behavior is also shown to be affected by the relative frequency of language varieties in the training data. We took advantage of the explicit and surface-accessible nature of these dependencies to attribute some model performance to the training data, while also demonstrating that modeling improvements should be possible, since the training data itself is substantially more consistent than the models.

These results suggest several possible avenues for future work. First, methods for addressing bias in training data should yield improvements for British spelling consistency in these models. We also intend to extend these results to languages other than English and investigate how spelling variation in other language situations is learned by LLMs. Some of the methods we used here rely on the fact that English spelling variation is quite thoroughly catalogued. Extending this work to less-documented cases of language variation will require us to either (1) collect data about spelling variation from language informants or data, or (2) develop methods that require less prior knowledge. In the interest of finding methods that are extensible to the greatest number of cases, we intend to pursue path (2), working on methods to mine information about language variation from large corpora and LLMs that have been trained on them.

Acknowledgements

Thanks to Alexander Gutkin, Shankar Kumar, Arya McCarthy and Richard Sproat for useful discussion and comments, and to the anonymous reviewers for helpful suggestions.

Limitations

Our work is focused on just a single case study of spelling variation. As detailed in Section 2, English is a good candidate for a case study for several reasons, but it would be beneficial to extend this work to other language situations.

Another limitation was our choice to focus on already existing pre-trained models, rather than directly controlling the training data that is input to each model. This means some of the conclusions about the connection between training data and outcome are tentative, pending experimental confirmation.

Ethics Statement

This work does not propose a new model or dataset, but rather probes the behavior of existing models. Thus novel ethical considerations about model behavior and dataset contents are not directly raised by this work. While not explicitly focused on ethical considerations, this paper’s methods hopefully contribute to better understanding model behavior, and could be used to understand the ways in which large language models treat underrepresented and marginalized language varieties.

References

- Kristian Berg and Mark Aronoff. 2017. Self-organization in the spelling of English suffixes: The emergence of culture out of anarchy. *Language*, 93(1):37–64.
- BNC Consortium. 2007. The British National Corpus, XML edition. Oxford Text Archive, <http://www.natcorp.ox.ac.uk>.
- Claire Bowern. 2015. *Linguistic fieldwork : a practical guide*, 2nd edition. Palgrave Macmillan, Basingstoke, Hampshire.
- Jonathan R. Brennan, Chris Dyer, Adhiguna Kuncoro, and John T. Hale. 2020. Localizing syntactic predictions using recurrent neural network grammars. *Neuropsychologia*, 146:107479.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT’s attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.
- Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Empirical Methods on Natural Language Processing*.
- Stefan L. Frank, Thijs Trompenaars, and Shrvan Vasishth. 2016. Cross-linguistic differences in processing double-embedded relative clauses: Working-memory constraints or language statistics? *Cognitive Science*, 40(3):554–578.
- Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv preprint arXiv:1809.01329*.
- Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.
- John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.
- Samuel Johnson. 1755. *A Dictionary of the English Language*. J. & P. Knapton, London.
- Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

- Adam Kilgarriff and Irene Renau. 2013. [esTenTen, a vast web corpus of Peninsular and American Spanish](#). *Procedia - Social and Behavioral Sciences*, 95:12–19. Corpus Resources for Descriptive and Applied Studies. Current Challenges and Future Directions: Selected Papers from the 5th International Conference on Corpus Linguistics (CILC2013).
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. [Open sesame: Getting inside BERT’s linguistic knowledge](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. [Assessing the ability of LSTMs to learn syntax-sensitive dependencies](#). *Transactions of the Association for Computational Linguistics*, 4:521–535.
- Robert C. Moore. 2004. [On log-likelihood-ratios and the significance of rare events](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 333–340, Barcelona, Spain. Association for Computational Linguistics.
- Dong Nguyen and Jack Grieve. 2020. [Do word embeddings capture spelling variation?](#) In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 870–881, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. [Processing South Asian languages written in the Latin script: the Dakshina dataset](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2413–2423, Marseille, France. European Language Resources Association.
- Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. [Gender bias in coreference resolution](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021. [Exploiting cloze-questions for few-shot text classification and natural language inference](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.
- Albert Webson and Ellie Pavlick. 2022. [Do prompt-based models really understand the meaning of their prompts?](#) In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.
- Noah Webster. 1828. *An American Dictionary of the English Language*. S. Converse, New York.

A Prompts

Table 9 presents the 29 prompt templates that were used in this study. Table 10 gives macro-averaged conditional probabilities for T5 runs across the different prompts, along with standard deviations to indicate how much performance varies due to the choice of prompt.

B Examples of corpus mismatch

Tables 11 and 12 present examples illustrating the mixture of American and British spelling in Wikipedia and the British National Corpus, respectively, as discussed in Section 6.

My preferred words are <CUE> and <FILLER>.

My preferred words are <CUE>, <FILLER>, and tree.

She wrote the words <CUE> and <FILLER>.

She wrote the words <CUE> and <FILLER> in her notebook.

She wrote the words <CUE>, <FILLER>, and cabbage.

I wrote the words <CUE> and <FILLER>.

I wrote the words <CUE> and <FILLER> in my notebook.

I wrote the words <CUE>, <FILLER>, and cabbage.

He wrote the words <CUE> and <FILLER>.

He wrote the words <CUE> and <FILLER> in his notebook.

He wrote the words <CUE>, <FILLER>, and cabbage.

We wrote the words <CUE> and <FILLER>.

We wrote the words <CUE> and <FILLER> in our notebook.

We wrote the words <CUE>, <FILLER>, and cabbage.

Mary wrote the words <CUE> and <FILLER>.

Mary wrote the words <CUE> and <FILLER> in her notebook.

Mary wrote the words <CUE>, <FILLER>, and cabbage.

Please spell <CUE> and <FILLER>.

Please spell <CUE>, <FILLER>, and panther.

Please spell <CUE> and <FILLER> correctly.

Say <CUE> and <FILLER>.

Say <CUE>, <FILLER>, and tapestry.

Say <CUE> and <FILLER> again.

The first words on the list were <CUE> and <FILLER>.

The first words on the list were <CUE>, <FILLER>, and oligarchy.

The easiest words on the list were <CUE> and <FILLER>.

The easiest words on the list were <CUE>, <FILLER>, and oligarchy.

The hardest words on the list were <CUE> and <FILLER>.

The hardest words on the list were <CUE>, <FILLER>, and oligarchy.

Table 9: Prompts used for model evaluation. Non-adjacent versions of each prompt were created by inserting the sequence “, flower, interesting, jump, ponderous, sky, skipping, desk, small, ladder, lovely,” between the <CUE> and <FILLER> word slots.

Condition	Word 1	T5 Word 2		T5+FT Word 2	
		US	UK	US	UK
Adjacent	US	0.86 (0.01)	0.14 (0.01)	0.66 (0.03)	0.34 (0.03)
	UK	0.39 (0.06)	0.61 (0.06)	0.44 (0.03)	0.56 (0.03)
Non-adjacent	US	0.83 (0.02)	0.17 (0.02)	0.69 (0.02)	0.31 (0.02)
	UK	0.48 (0.05)	0.52 (0.05)	0.43 (0.04)	0.57 (0.04)

Table 10: Conditional probability of Word 2, given a template with Word 1, given by base T5 and T5 with additional finetuning. Each cell includes a macro-average and standard deviation across 29 prompts.

version date	sentence version
4 Aug. 2017 (neither vapor nor vapour)	An air lock is a restriction of, or complete stoppage of liquid flow caused by gas trapped in a high point of a liquid-filled pipe system.
6 Sept. 2017 (vapour replaces gas)	An air lock is a restriction of, or complete stoppage of liquid flow caused by vapour trapped in a high point of a liquid-filled pipe system.
15 Feb. 2020 (vapor added)	An air lock (or vapor lock) is a restriction of, or complete stoppage of liquid flow caused by vapour trapped in a high point of a liquid-filled pipe system.

Table 11: Three versions of a Wikipedia page: (1) no use of *vapor* or *vapour* in the sentence; (2) the term *vapour* replaces *gas*; and (3) the alternative name for the phenomenon "*vapor lock*" is introduced.

Doc ID	sentence
CPD	‘What this guy will do is get a demoralized sales organisation revitalised ...’ said John Jones, analyst at Salomon Brothers.
CLW	They conceptualize these differences in terms of ‘separate local labour market cultures’ (ibid., p. 104).
CBH	It is a metaphor which attempts to create a reality of organization whereby cooperation is mobilised for fight with the outside world.

Table 12: Examples of spelling convention mismatches in the British National Corpus, sampled from varied books and periodicals.

Modelling Language Acquisition through Syntactico-Semantic Pattern Finding

Jonas Doumen^{1,2} and Katrien Beuls³ and Paul Van Eecke^{1,2,4}

¹KU Leuven, Faculty of Arts, Research Unit Linguistics, Blijde Inkomststraat 21, B-3000 Leuven

²KU Leuven, imec research group itec, Etienne Sabbelaan 51, B-8500 Kortrijk

³Faculté d'informatique, Université de Namur, rue Grandgagnage 21, B-5000 Namur

⁴Artificial Intelligence Laboratory, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels

jonas.doumen@kuleuven.be

katrien.beuls@unamur.be

paul@ai.vub.ac.be

Abstract

Usage-based theories of language acquisition have extensively documented the processes by which children acquire language through communicative interaction. Notably, Tomasello (2003) distinguishes two main cognitive capacities that underlie human language acquisition: intention reading and pattern finding. Intention reading is the process by which children try to continuously reconstruct the intended meaning of their interlocutors. Pattern finding refers to the process that allows them to distil linguistic schemata from multiple communicative interactions. Even though the fields of cognitive science and psycholinguistics have studied these processes in depth, no faithful computational operationalisations of these mechanisms through which children learn language exist to date. The research on which we report in this paper aims to fill part of this void by introducing a computational operationalisation of syntactico-semantic pattern finding. Concretely, we present a methodology for learning grammars based on similarities and differences in the form and meaning of linguistic observations alone. Our methodology is able to learn compositional lexical and item-based constructions of variable extent and degree of abstraction, along with a network of emergent syntactic categories. We evaluate our methodology on the CLEVR benchmark dataset and show that the methodology allows for fast, incremental and effective learning. The constructions and categorial network that result from the learning process are fully transparent and bidirectional, facilitating both language comprehension and production. Theoretically, our model provides computational evidence for the learnability of usage-based constructionist theories of language acquisition. Practically, the techniques that we present facilitate the learning of computationally tractable, usage-based construction grammars, which are applicable for natural language understanding and production tasks.

1 Introduction

Usage-based theories of language acquisition argue that the ability of children to learn language is based on two general cognitive capacities: *intention reading* and *pattern finding* (Tomasello, 2003, 2009). Intention reading refers to the capacity of children to understand the communicative intentions of their interlocutors. Pattern finding refers to the ability to recognise similarities and differences in sensory-motor experiences, and to use this ability for categorisation and schema formation (Tomasello, 2003, p. 3–4). Pattern finding thus provides mechanisms for generalising across different communicative interactions, thereby constructing abstract schemata that represent the linguistic knowledge of a language user. In the context of language acquisition, intention reading and pattern finding are two key cognitive capacities that are highly complementary. Intention reading allows a language learner to reconstruct the meaning of an utterance that they observe during a communicative interaction. Pattern finding then provides the mechanisms to learn a grammar based on the combination of these observed utterances and their reconstructed meanings.

There exists an impressive body of theoretical and empirical evidence for both intention reading (Bruner, 1983; Sperber and Wilson, 1986; Meltzoff, 1995; Nelson, 1998) and pattern finding (Goldberg, 1995; Croft, 2000; Diessel, 2004; Goldberg, 2006). However, no comprehensive mechanistic models that provide a faithful operationalisation of either of these cognitive processes exist to date. In this paper, we aim to fill part of this void by presenting a computational operationalisation of pattern finding mechanisms that can bootstrap a grammar based on a set of semantically annotated utterances alone. As such, we assume that the outcome of the intention reading process is given, hence the availability

of the utterances’ semantic representations, but that neither a segmentation of the utterances nor any pre-existing morpho-syntactic or other grammatical information can be used. For a computational model that operationalises the intention reading process, and that integrates it with the pattern finding mechanisms introduced in this paper, we refer the interested reader to [Nevens et al. \(2022\)](#).

A validation of our methodology on the CLEVR benchmark dataset for visual question answering ([Johnson et al., 2017](#)) shows that it allows for fast, incremental and effective grammar learning. The result of this learning process is a fully-operational, productive construction grammar that can be used for both language comprehension, i.e. mapping from utterances to their meaning representation, and language production, i.e. mapping from a meaning representation to an utterance.

The scientific contribution of this paper is twofold. On the one hand, it provides computational evidence for the cognitive plausibility of usage-based theories of language acquisition by introducing a mechanistic model of the acquisition of construction grammars from scratch. On the other hand, the techniques that we present pave the way for learning computationally tractable, large-scale, usage-based construction grammars that facilitate both language comprehension and production. Apart from their theoretical importance, such grammars are also highly valuable for a large range of application domains, including intelligent conversational agents ([Verheyen et al., 2022](#)) and the semantic analysis of discourse ([Willaert et al., 2020](#); [Beuls et al., 2021](#)).

The remainder of this paper is structured as follows. Section 2 presents the dataset, task and learning problem that we address. Section 3 introduces our novel methodology for learning construction grammars. Section 4 presents the evaluation results. Related work is discussed in Section 5. A concluding discussion is provided in Section 6.

2 Data

There are two main requirements for datasets to be compatible with the methodology that we present in this paper. First of all, they need to consist of utterances that are annotated with a representation of their meaning. Second, they need to be large enough so that they contain enough utterances that are similar to each other, but not equal, in terms of either form or meaning. The availability of ex-

emplars that are sufficiently close to each other is a necessary precondition for any generalisation process and is fully consistent with the prevailing hypotheses of how children acquire language ([Tomasello, 2003](#)). The exact required size of a dataset is as a consequence directly related to the variety and the degree of complexity of the utterances and meaning representations that it contains.

In this paper, we present and validate our methodology using the CLEVR dataset for visual question answering ([Johnson et al., 2017](#)). The utterances in the dataset are semantically annotated and the dataset contains ample examples of utterance-meaning pairs that are similar but not equal to each other. The utterances are English questions about images of scenes depicting different configurations of geometrical figures. Each question is annotated with a semantic representation that captures the logical meaning that underlies it. An example of such a scene, a question and its semantic representation is shown in Figure 1.

The semantic representation in Figure 1 takes the form of a set of predicates that share arguments with each other. In the figure, the predicates are drawn in the form of a network, based on the variables that they share. The meaning representation of a question can naturally be represented as a query, i.e. a series of steps that need to be taken in order to answer the question. Each predicate represents a step in this reasoning process, and intuitively corresponds to an atomic cognitive operation that a human or machine can perform. In the case of the example utterance ‘*How many rubber spheres are there?*’, the reasoning process consists of four main steps. The first predicate, GET-CONTEXT, binds the image to the variable ‘?source’. Then, the FILTER predicate filters the image for instantiations of the concept of SPHERE. The result of this filtering operation, i.e. the set of all spheres that are in the image, is bound to the variable ‘?spheres’. This set of spheres is subsequently filtered by another FILTER predicate for instantiations of the concept of RUBBER. The resulting set of rubber spheres is bound to the variable ‘?rubber-spheres’. Finally, the set of rubber spheres is counted by the COUNT predicate and the result is bound to the variable ‘?nr-of-rubber-spheres’. The meaning of the question ‘*How many rubber spheres are there?*’ corresponds thus informally to filtering an image for spheres, filtering the spheres for rubber objects and counting the result of this

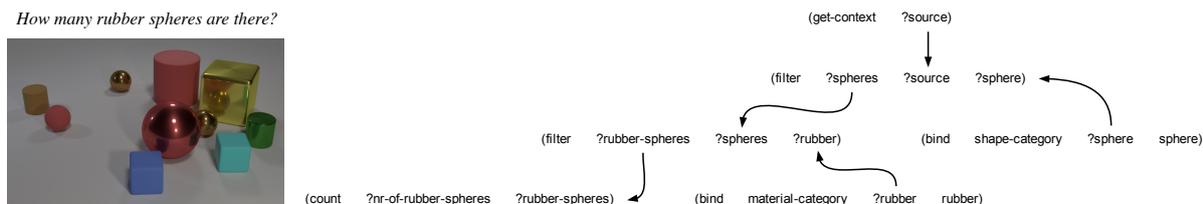


Figure 1: Example scene, question, and procedural semantic representation from the CLEVR dataset.

last operation. Such meaning representations are called *procedural semantic representations* as the representations themselves are at the same time executable procedures (Winograd, 1972; Johnson-Laird, 1977). Our methodology handles procedural semantic representations without problems, but is in no way restricted to it. It can handle any semantic representation, as long as it embraces some notion of compositionality and can be expressed as a set of predicates. Examples of other compatible semantic representations include abstract meaning representation (Banarescu et al., 2013), PropBank frames (Palmer et al., 2005) and the lambda calculus (Church, 1932; Montague, 1974).

The CLEVR dataset consists of three splits: a training split of 70,000 images and 699,989 questions, a validation split of 15,000 images and 149,991 questions, and a test split of 15,000 images and 149,988 questions. The questions in the training and validation splits come with semantic annotations, whereas the test set does not. As we require these annotations in order to evaluate our model, we use the training split of the CLEVR dataset as training set and the validation split as test set. The question-annotation pairs embrace various aspects of reasoning, including attribute identification (*‘There is a large cube; what is its color?’*), counting (*‘How many green spheres are there?’*), comparison (*‘Are there an equal number of large cubes and small things?’*), spatial relationships (*‘What size is the cylinder that is right of the yellow shiny thing that is left of the cube?’*) and logical operations (*‘How many objects are either red cubes or yellow cylinders?’*). For the purposes of this paper, we have selected the subset of CLEVR questions that do not involve comparison, spatial relationships or logical operations. The main reason for this is that these are complex cognitive operations that often correspond to long and complex utterances that are far removed from the linguistic expressions that children (or even other humans) are faced with. Our final training and test sets

consist of 47,134 questions and 10,044 questions respectively.

The learning task that we address consists in operationalising pattern finding mechanisms that facilitate the learning of a bidirectional construction grammar. The grammar should be able to map between the CLEVR utterances and their semantic representation, both in the comprehension (form to meaning representation) and the production (meaning representation to form) direction.

3 Methodology

The input to the learning process consists of utterances that are annotated with a representation of their meaning. The output of the learning process should consist in form-meaning mappings (constructions) that can be used for comprehending and producing utterances. The form-meaning mappings are represented in, and processed using, Fluid Construction Grammar (Steels, 2011; Van Eecke and Beuls, 2017; van Trijp et al., 2022).

3.1 Holophrase Constructions

Let us for a moment take the perspective of the learning algorithm. At the beginning of the learning process, the construction inventory is empty and the first utterance-meaning pair from the corpus comes in. At this point, the only thing that the learning algorithm can do is to store an exact mapping between the observed form and its meaning. Such a holistic mapping corresponds to a holophrase construction and is usable as such, albeit only for comprehending and producing the exact same utterance as the one that was observed. In order to use such a construction in the comprehension direction, it suffices to match the form side of the construction with an utterance and return the meaning side of the construction if the matching process succeeded. In order to use the same construction in the production direction, the meaning side of the construction must be matched with a semantic network and the form side must be returned.

When a next observation comes in, the learning algorithm first checks whether it is already covered by constructions that have been acquired previously. When this is the case, the constructions that are involved in the successful comprehension and production of the observation are reinforced by incrementing their entrenchment score. If the observation is not covered, the algorithm checks whether there are any generalisations that can be made based on the combination of the novel observation and any previously acquired constructions. It is these generalisation mechanisms that embody Tomasello (2003)'s pattern finding capacity and are thereby at the core of the construction learning process. We have identified three classes of mechanisms that facilitate the learning of general constructions by algorithmically reasoning over similarities and differences between existing constructions and novel observations.

3.2 Generalising over Holophrase Constructions

The first class of mechanisms facilitates the generalisation of holophrase constructions with respect to novel observations. These mechanisms can learn item-based constructions that capture the similarities between a novel observation and an existing holophrase construction that was learnt based on a similar, but not equal, observation. These item-based constructions abstract away from the differences between the observation and the holophrase construction.

For example, imagine that a holophrase construction has already been learnt based on the observation of the utterance *'How many rubber spheres are there?'* and the semantic network shown in Figure 1. Now, a novel utterance *'How many rubber cubes are there?'* is observed, along with a very similar meaning network in which the predicate '(bind shape-category ?cube cube)' appears at the place of '(bind shape-category ?sphere sphere)'. The generalisation mechanisms compute the similarities and differences between the construction and the observation in terms of both form and meaning, and make a new item-based construction that maps between the utterance *'How many rubber ?X are there?'* and the semantic network from Figure 1 in which the non-overlapping predicate has been replaced by a variable. At the same time, two new lexical constructions are created, which capture the differences between the observation and the

original holophrase construction. In our example, these will be a construction that maps between the utterance *'cubes'* and the meaning representation '(bind shape-category ?cube cube)' and a construction that maps between the utterance *'spheres'* and the meaning representation '(bind shape-category ?sphere sphere)'. Finally, categorial links are made between the ?X slot in the item-based construction and the new lexical constructions. These categorial links capture that *'cubes'* and *'spheres'* can both appear in the ?X slot of the construction for *'How many rubber ?X are there?'*. A schematic representation of this learning process is shown in Figure 2.

There are three different scenarios in which mechanisms of this class are active. The first scenario concerns utterances which extend holophrases that are already known. An example would be the generalisation of *'Are there any cylinders?'* to *'Are there any red cylinders?'*. In this case, an item-based construction *'Are there any ?X cylinders?'* is learnt, along with a lexical construction for *'red'* and a categorial link between the lexical construction and the open slot in the item-based construction. The second scenario concerns utterances which reduce known holophrases. An example would be the reduction of *'What is the size of the metal block?'* to *'What is the size of the block?'*. In this case, an item-based construction for *'What is the size of the ?X block?'* is learnt, along with a holophrase construction for *'What is the size of the block?'*, a lexical construction for *'metal'*, and a categorial link between the slot in the item-based construction and the lexical construction. The final scenario concerns utterances which are not a mere extension or reduction of each other, but contain different formal and/or semantic material. An example would be the utterances *'How many rubber spheres are there?'* and *'How many rubber cubes are there?'* discussed above, where a holophrase construction for *'How many rubber spheres are there?'* is already in place. An item-based construction for *'How many rubber ?X are there?'* is learnt along with a lexical construction for *'cubes'* and a categorial link between the open slot in the item-based construction and the new lexical construction. Additionally, a second lexical construction for *'spheres'* is learnt, along with a categorial link between the open slot in the item-based construction and the lexical construction for *'spheres'*.

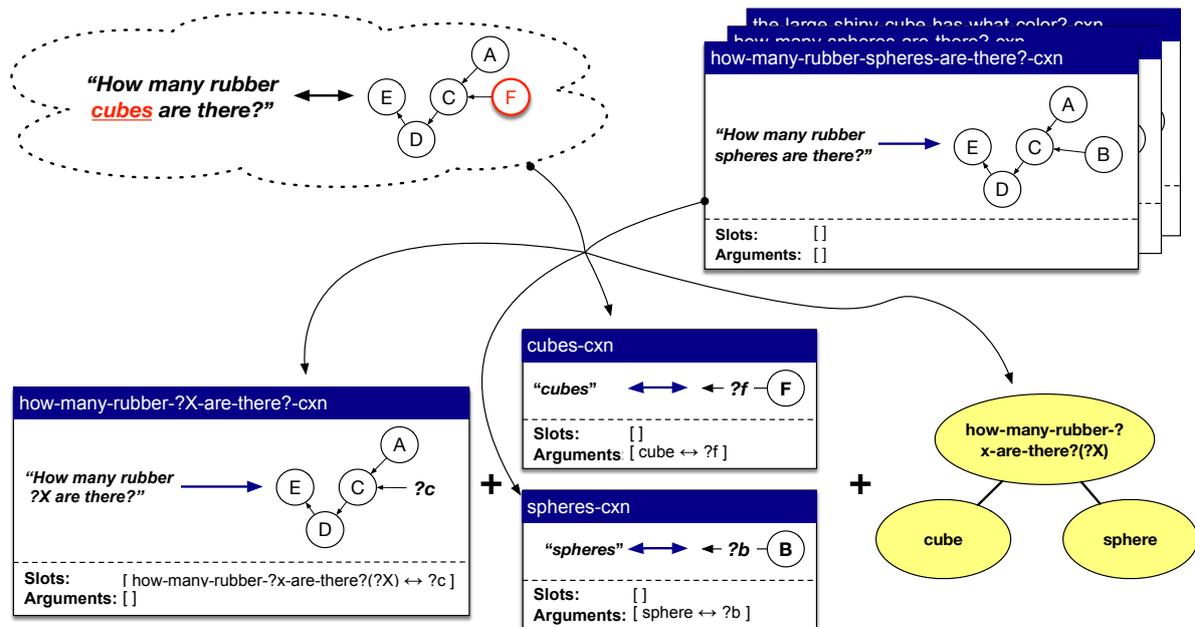


Figure 2: A schematic representation of a generalisation operator learning an item-based construction and two lexical constructions based on an existing holophrase construction and a novel observation.

3.3 Learning Constructions Based on a Partial Analysis

The second class of mechanisms is designed to handle cases where an observation could not completely be processed using the existing constructions of a grammar, but where a partial analysis could be provided. These mechanisms can then create novel constructions that can work together with existing constructions so that the entire observation can be processed successfully. They start thus from the combination of a novel observation on the one hand, and an item-based construction or one or more lexical constructions on the other. The second class of mechanisms is active in two different scenarios.

The first scenario concerns observations to which an item-based construction can apply, but where there remains material that is not covered by any of the existing constructions. An example would be an observation of ‘*What is the size of the green block?*’, where a construction for ‘*What is the size of the ?X block?*’ is already known, while no construction for ‘*green*’ has been learnt yet. The learning algorithm detects that some aspects of the form and the meaning of the observation are not covered by the existing item-based construction and it creates a novel lexical construction that maps between those parts of the form and meaning that were not covered. Additionally, a categorial link is made between the slot in the item-based construc-

tion and the lexical construction. In our example, this means that a lexical construction for ‘*green*’ is learnt, along with a categorial link between this construction and the ?X slot in the construction for ‘*What is the size of the ?X block?*’.

The second scenario concerns observations to which one or more lexical constructions can apply, but where these constructions do not fully cover the input. An example would be an observation of the utterance ‘*There is a big red cube; what is its material?*’, where lexical constructions for ‘*big*’, ‘*red*’, ‘*cube*’ and ‘*material*’ have already been learnt. The learning algorithm will then create a new item-based construction that incorporates all the form and meaning material that remains after the application of these lexical constructions, and that abstracts away from these constructions through the integration of four slots. The result is an item-based construction of the form ‘*There is a ?A ?B ?C; what is its ?D?*’ and four categorial links from the existing lexical constructions to the slots in the new item-based construction.

3.4 Extending the Categorial Network

The third class of mechanisms is designed to handle cases where all necessary constructions are already in place, but where they cannot combine due to the absence of certain links in the categorial network. An example would be the utterance ‘*How many things are there?*’ where an item-based construc-

tion covering ‘How many ?X are there?’ and a lexical construction covering ‘things’ already exist, but where there is no link in the categorial network between the lexical construction for ‘things’ and the ‘?X’ slot in the item-based construction. In such cases, the learning algorithm adds the missing link to the categorial network.

3.5 Entrenchment Scores

The constructions created by the learning operators have scores that reflect their entrenchment. During processing, higher scored constructions are preferred over lower scored ones. Upon creation, the score of a construction is set to 0.5. If used successfully, the score is increased by 0.1 and the score of other constructions of which the application would also have led to a solution is decreased by 0.3. The scores are bounded between 0 and 1. There is no built-in bias towards more general constructions. However, the fact that more general constructions are applicable in a broader range of situations and are therefore more frequently used, will, due to the dynamics of rewarding successful usage and punishing competitors, lead to higher entrenchment scores for more general constructions.

4 Experiments

This section presents a validation of our methodology for acquiring constructions on the CLEVR dataset discussed in Section 2. We first describe the experimental set-up (Section 4.1) and then present the evaluation results (Section 4.2).

4.1 Experimental Set-Up

The primary experiment consists in processing the 47,134 observations from our training set using the learning operators introduced above. For each experimental run, the observations are shuffled, so that any side-effects that might be caused by the order in which the observations are presented are levelled out. The learning operators are only active when an observation cannot be processed successfully by the constructions that have been learnt so far. Entrenchment scores are updated after each communicative interaction. The learning process is evaluated through four quantitative metrics: *communicative success*, *grammar size*, *number of constructions per type* and *active learning operators*. Communicative success is a binary measure computed by comparing the comprehended meaning with the gold standard annotation. In the graphs

below, communicative success and active learning operators are plotted using a sliding window of 50 observations.

For completeness, we also present a secondary experiment in which the grammar learnt on the training set is evaluated on the test set. Communicative success is here averaged over the whole test set, and grammar size and number of constructions per type do not change during evaluation.

The experimental results reported below are based on 10 independent experimental runs. The error bars that are plotted represent percentiles 5 and 95.

4.2 Results

The results obtained through the primary experiment are shown in Figures 3 to 5. Figure 3 displays the communicative success and grammar size metrics respectively on the left and right y-axis as a function of the number of observations (x-axis). We can see that the communicative success starts at 0, as the experiment starts with an empty inventory of constructions. The degree of communicative success rises rapidly, with more than 90% of the observations being successfully processed by the learned grammar after only 500 observations have been encountered. After 2000 observations, communicative success is already achieved in 99.6% of new observations.

The grammar size starts at 0 constructions and grows rapidly in the first phase of the experiment. After 500 observations, the grammar has reached its peak size of around 230 constructions that have some degree of entrenchment. This number then declines as a result of the rewarding and punishing of constructions. At the end of the learning process, the resulting grammar consists of 101.5 constructions on average.

An analysis of the types of constructions that are part of the learned construction inventory is provided in Figure 4. The results show that holophrase constructions flourish in the earliest phase of the experiment. In a second phase, item-based and lexical constructions take over the role of the holophrase constructions, with an abundance of item-based constructions being created. Over the course of the experiment, the linguistic inventory of the learner gradually reaches a stable state consisting of a limited number of entrenched lexical constructions and (more general) item-based constructions. At the end of the experiment, the grammar consists on av-

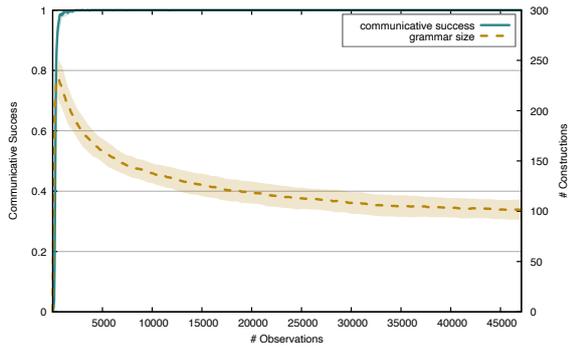


Figure 3: Evolution of communicative success (left y-axis) and grammar size (right y-axis) over time (full dataset).

erage of 10.2 holophrase constructions, 57.1 item-based constructions and 34.2 lexical constructions. These results show that the holophrase constructions have not yet completely disappeared after 47,134 observations and that the theoretical maximum of 35 lexical constructions was attained in 7 out of 10 experimental runs. Note that it is the dynamic evolution of the number of constructions per type over time that is important, rather than the absolute number of constructions at a given moment in time.

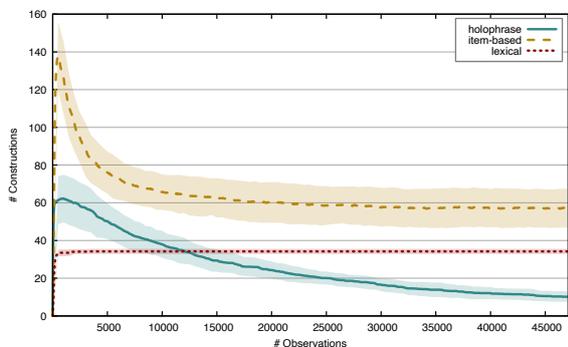


Figure 4: Evolution over time of the number of constructions per type with an entrenchment score > 0 (full dataset).

Figure 5 shows the active learning operators over time, zooming in on the first 1000 observations. In the beginning, only new holophrase constructions can be created. Then, operators of the first class can generalise over these holophrase constructions and create new item-based and lexical constructions. After that, operators of the second class take over and create constructions based on partial analyses. In the final phase of the experiment, mainly operators of the third class, which only create new categorial links, are active.

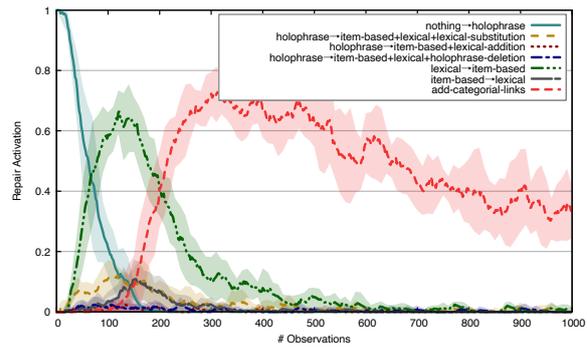


Figure 5: Active learning operators over time (first 1000 observations).

We finally conduct a secondary experiment, which consists in processing all observations from the test set using the grammars resulting from the different experimental runs of the primary experiment. The average communicative success amounts to a perfect 100% in both the comprehension and production direction. The average grammar size amounts to 101.5 constructions, of which 10.2 are holophrase constructions, 57.1 are item-based constructions and 34.2 are lexical constructions.

5 Related Work

Prior mechanistic models that operationalise the learning of constructions can be divided into two groups, based on the learning task that they address. A first class of models learns constructions paired with their meaning representation, either provided in the form of an annotated corpus (Dominey and Boucher, 2005; Chang, 2008; Abend et al., 2017) or obtained through task-oriented communicative interactions in a tutor-learner scenario (Gerasymova and Spranger, 2010; Beuls et al., 2010; Spranger and Steels, 2015). A second class of models, as introduced by Gaspers et al. (2011, 2016), is designed to learn form-meaning pairings under referential uncertainty. As such, the exact meaning representations of the input utterances are not provided to the learning algorithm, but grammars are learnt based on the combination of input utterances and situational context snippets. In these experiments, input utterances always correspond to a single term in the situational context. In general, both classes of models have explored interesting ideas on a rather small scale, either because they were limited to specific linguistic phenomena (Steels, 2004; Gerasymova and Spranger, 2010, 2012; Beuls et al., 2010; Spranger and Steels, 2015; Spranger, 2015, 2017;

Van Eecke and Beuls, 2017, 2018; Van Eecke, 2018), or because of the limited morpho-syntactic and semantic complexity of the input utterances (Dominey, 2005a,b, 2006; Chang, 2008; Gaspers et al., 2011; Gaspers and Cimiano, 2012, 2014; Gaspers et al., 2016; Abend et al., 2017). In all of the aforementioned work, either a segmentation of the input utterances, a lexicon or a set of predefined grammatical categories was provided. With the exception of the studies by Gaspers et al., the corpora that were used to learn and evaluate the models were not made available and were not described in sufficient detail to make reproduction and comparison feasible.

6 Discussion and Conclusion

The scientific contribution of the methodology and experiments presented in this paper is twofold. On the one hand, they provide computational evidence for the cognitive plausibility of constructivist theories of language acquisition. These theories, as most prominently put forward by Tomasello (2003), attribute the ability of children to acquire language to two main cognitive capacities: intention reading and pattern finding. Intention reading deals with reconstructing the intended meaning of observed utterances, while pattern finding implements generalisation processes that distil these reconstructed utterance-meaning pairs into abstract schemata embodying the linguistic knowledge of a language user. These schemata can then be used to fulfil the communicative function of language through the comprehension and production of natural language expressions. The methodology introduced in this paper presents a mechanistic model of the pattern finding capacity. Based on utterances paired with a representation of their meaning, the learning algorithm gradually builds up an inventory of concrete to abstract form-meaning mappings, called constructions, along with a network of emergent grammatical categories that captures how the constructions of the grammar can combine to collaboratively comprehend and produce utterances. The experiments show that a small number of general learning operators, which become active if an utterance cannot be successfully processed by the grammar learnt so far, effectively leads to learning dynamics that are similar to those described in the psycholinguistic literature (Pine and Lieven, 1997; Tomasello, 2003; Ambridge and Lieven, 2015). In the first phase of the learning

process, the learner acquires holistic mappings between utterances and their meaning representation. Soon after that, holophrase constructions are generalised to item-based constructions that integrate a variable slot. At the same time, this generalisation process leads to the emergence of slot-filling constructions, here called lexical constructions. Along with the item-based and lexical constructions, a network of grammatical categories emerges, capturing the distribution of construction slots and their observed fillers. In a third phase, more abstract item-based constructions emerge, with an increasingly large number of variable slots. In the final phase of the learning process, most constructions have already been acquired and most remaining impasses can be solved by adding new links to the categorial network. The learning dynamics are influenced by the degree of entrenchment of constructions. Constructions that are often successfully used become more entrenched, while their competitors are suppressed. As a result of this entrenchment process, the grammar reaches a stable state, while it remains adaptive to any changes in the discourse or environment. Similar dynamics have been observed in earlier experiments in the field of evolutionary linguistics, for instance in experiments on the emergence of compositionality in a population of autonomous agents (De Beule and Bergen, 2006; van Trijp, 2016).

On the other hand, the methodology and experiments presented in this paper pave the way for learning computationally tractable, large-scale, usage-based grammars that facilitate both language comprehension and production. The proposed learning algorithm supports online, interactive, incremental, transparent and data-efficient learning. The learner builds up its human-interpretable inventory of constructions and categories through the application of transparent syntactico-semantic generalisation processes. Already after a single observation, the fragment of linguistic knowledge acquired by the learner can be successfully used for language comprehension and production. As more and more utterance-meaning pairs are observed, the linguistic knowledge of the learner quickly expands and becomes better fit for achieving their communication goals. As a result of the dynamics of rewarding successful construction applications and punishing competing ones, the grammar of the learner remains ever-adaptive to any changes in the task or environment. Due to their online, interactive,

incremental, transparent and data-efficient nature, we strongly believe that the proposed mechanisms for learning computational construction grammars can serve as an excellent basis for implementing the language acquisition ability of truly intelligent agents.

Limitations

This paper has introduced a mechanistic model of the constructivist acquisition of language through syntactico-semantic pattern finding. Even though the results that we presented here proved to be promising and insightful, considerable challenges and limitations remain.

First of all, the learning operators that we present facilitate the learning of holistic, item-based and lexical constructions. At this point, the model does not include operators that give rise to constructions that capture more elaborate hierarchical patterns, including recursive patterns.

Second, the learning operators can adequately handle word order patterns, even non-contiguous ones. However, they provide no mechanisms to learn agreement patterns on an abstract level. As a consequence, different agreement patterns are captured in different constructions. This is a less-than-elegant solution, especially when applied to morphologically rich languages, as it can lead to a multiplication of the number of constructions.

Finally, the CLEVR dataset proved to be an excellent benchmark challenge for an initial validation of this novel methodology, as it consists of utterances with sufficient repetition, variation and overlap. It is however a synthetic dataset that does not reflect the richness of human language use. More research is needed before this methodology can be adequately applied to a broader range of linguistic resources, especially when it comes to finding generalisations over semantic structures.

Acknowledgements

The research reported on in this paper received funding from the imec's Smart Education research programme, with support from the Flemish government, the European Union's Horizon 2020 research and innovation programme under grant agreement no. 951846, and the Research Foundation Flanders (FWO) through a postdoctoral grant awarded to Paul Van Eecke (75929).

We would like to express our gratitude to Jens Nevens and Lara Verheyen for their valuable feed-

back on earlier versions of this manuscript, and to the three anonymous EACL reviewers for their constructive comments and support.

References

- Omri Abend, Tom Kwiatkowski, Nathaniel J. Smith, Sharon Goldwater, and Mark Steedman. 2017. [Bootstrapping language acquisition](#). *Cognition*, 164:116–143.
- Ben Ambridge and Elena Lieven. 2015. A constructivist account of child language acquisition. In Brian MacWhinney and William O'Grady, editors, *The Handbook of Language Emergence*, pages 478–510. John Wiley and Sons, Hoboken, NJ.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.
- Katrien Beuls, Kateryna Gerasymova, and Remi van Trijp. 2010. Situated learning through the use of language games. In *Proceedings of the 19th Annual Machine Learning Conference of Belgium and The Netherlands (BeNeLearn)*, pages 1–6.
- Katrien Beuls, Paul Van Eecke, and Vanja Sophie Cangalovic. 2021. A computational construction grammar approach to semantic frame extraction. *Linguistics Vanguard*, 7(1):20180015.
- Jerome Bruner. 1983. *Learning to use language*. Oxford University Press, Oxford.
- Nancy Chang. 2008. *Constructing grammar: A computational model of the emergence of early constructions*. Ph.D. thesis, University of California, Berkeley, Berkeley, CA.
- Alonzo Church. 1932. [A set of postulates for the foundation of logic](#). *Annals of Mathematics*, 33(2):346–366.
- William Croft. 2000. *Explaining language change: An evolutionary approach*. Pearson Education, Harlow.
- Joachim De Beule and Benjamin K. Bergen. 2006. [On the emergence of compositionality](#). In *The Evolution of Language. Proceedings of the 6th International Conference (EVOLANG6)*, pages 35–42, Singapore. World Scientific.
- Holger Diessel. 2004. *The acquisition of complex sentences*. Cambridge University Press, Cambridge.
- Peter Ford Dominey. 2005a. Emergence of grammatical constructions: evidence from simulation and grounded agent experiments. *Connection Science*, 17(3-4):289–306.

- Peter Ford Dominey. 2005b. From sensorimotor sequence to grammatical construction: Evidence from simulation and neurophysiology. *Adaptive Behavior*, 13(4):347–361.
- Peter Ford Dominey. 2006. From holophrases to abstract grammatical constructions: Insights from simulation studies. In Eve Clark and Barbara Kelly, editors, *Constructions in acquisition*, pages 137–162. CSLI Publications, Stanford.
- Peter Ford Dominey and Jean-David Boucher. 2005. [Learning to talk about events from narrated video in a construction grammar framework](#). *Artificial Intelligence*, 167(1):31–61.
- Judith Gaspers and Philipp Cimiano. 2012. A usage-based model for the online induction of constructions from phoneme sequences. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 1–6. IEEE.
- Judith Gaspers, Philipp Cimiano, Sascha S Griffiths, and Britta Wrede. 2011. An unsupervised algorithm for the induction of constructions. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6. IEEE.
- Judith Gaspers, Philipp Cimiano, Katharina Rohlfing, and Britta Wrede. 2016. Constructing a language from scratch: Combining bottom-up and top-down learning processes in a computational model of language acquisition. *IEEE Transactions on Cognitive and Developmental Systems*, 9(2):183–196.
- Judith Gaspers and Philipp Cimiano. 2014. A computational model for the item-based induction of construction networks. *Cognitive Science*, 38 3:439–88.
- Kateryna Gerasymova and Michael Spranger. 2010. Acquisition of grammar in autonomous artificial systems. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-2010)*, pages 923–928.
- Kateryna Gerasymova and Michael Spranger. 2012. An experiment in temporal language learning. In Luc Steels and Manfred Hild, editors, *Language Grounding in Robots*, pages 237–254. Springer, New York.
- Adele Goldberg. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press, Chicago.
- Adele Goldberg. 2006. *Constructions at work: The nature of generalization in language*. Oxford University Press, Oxford.
- Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2910.
- Philip N Johnson-Laird. 1977. Procedural semantics. *Cognition*, 5(3):189–214.
- Andrew Meltzoff. 1995. Understanding the intentions of others: re-enactment of intended acts by 18-month-old children. *Developmental Psychology*, 31(5):838–850.
- Richard Montague. 1974. English as a formal language. In R. H. Thomason, editor, *Formal Philosophy; Selected papers of Richard Montague*, pages 188–221. Yale University Press, New Haven, CT.
- Katherine Nelson. 1998. *Language in cognitive development: The emergence of the mediated mind*. Cambridge University Press, Cambridge.
- Jens Nevens, Jonas Doumen, Paul Van Eecke, and Katrien Beuls. 2022. [Language acquisition through intention reading and pattern finding](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 15–25, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.
- Julian M Pine and Elena VM Lieven. 1997. Slot and frame patterns and the development of the determiner category. *Applied Psycholinguistics*, 18(2):123–138.
- Dan Sperber and Deirdre Wilson. 1986. *Relevance: Communication and cognition*. Harvard University Press, Cambridge, MA.
- Michael Spranger. 2015. Incremental grounded language learning in robot-robot interactions: Examples from spatial language. In *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 196–201.
- Michael Spranger. 2017. Usage-based grounded construction learning: A computational model. In *The 2017 AAAI Spring Symposium Series*, pages 245–250, Palo Alto, Ca. AAAI Press.
- Michael Spranger and Luc Steels. 2015. Co-acquisition of syntax and semantics: an investigation in spatial language. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 1909–1915, Palo Alto, Ca. AAAI Press.
- Luc Steels. 2004. Constructivist development of grounded construction grammar. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 9–16.
- Luc Steels, editor. 2011. *Design patterns in Fluid Construction Grammar*. John Benjamins, Amsterdam.
- Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, Harvard.

- Michael Tomasello. 2009. The usage-based theory of language acquisition. In *The Cambridge handbook of child language*, pages 69–87. Cambridge University Press, Cambridge.
- Paul Van Eecke. 2018. *Generalisation and specialisation operators for computational construction grammar and their application in evolutionary linguistics Research*. Ph.D. thesis, Vrije Universiteit Brussel, Brussels: VUB Press.
- Paul Van Eecke and Katrien Beuls. 2017. Meta-layer problem solving for computational construction grammar. In *The 2017 AAI Spring Symposium Series*, pages 258–265, Palo Alto, Ca. AAAI Press.
- Paul Van Eecke and Katrien Beuls. 2018. Exploring the creative potential of computational construction grammar. *Zeitschrift für Anglistik und Amerikanistik*, 66(3):341–355.
- Remi van Trijp. 2016. *The evolution of case grammar*. Language Science Press, Berlin.
- Remi van Trijp, Katrien Beuls, and Paul Van Eecke. 2022. [The FCG editor: An innovative environment for engineering computational construction grammars](#). *PLOS ONE*, 17(6):1–27.
- Lara Verheyen, Jérôme Botoko Ekila, Jens Nevens, Paul Van Eecke, and Katrien Beuls. 2022. Hybrid procedural semantics for visual dialogue: An interactive web demonstration. In *Workshop on semantic techniques for narrative-based understanding: Workshop at IJCAI-ECAI 2022*, pages 48–52.
- Tom Willaert, Paul Van Eecke, Katrien Beuls, and Luc Steels. 2020. [Building social media observatories for monitoring online opinion dynamics](#). *Social Media + Society*, 6(2):2056305119898778.
- Terry Winograd. 1972. Understanding natural language. *Cognitive Psychology*, 3(1):1–191.

A Appendix

A.1 Computing Infrastructure and Runtime

The experiments were run on a single 2.5 GHz CPU, with 16 GB RAM. Running a single series of the training set takes less than one hour. The processing of the test set takes less than 15 minutes.

Benchmark Data and Evaluation Framework for Intent Discovery Around COVID-19 Vaccine Hesitancy

Shai Gretz^{1*}, Assaf Toledo^{1*}, Roni Friedman¹

Dan Lahav¹, Rose Weeks², Naor Bar-Zeev²

João Sedoc³, Pooja Sangha², Yoav Katz¹, Noam Slonim¹

¹IBM Research; ²Johns Hopkins Bloomberg School of Public Health; ³New York University
{avishaig,roni.friedman-melamed,katz,noams}@il.ibm.com
{assaf.toledo,dan.lahav}@ibm.com, {rweeks,nbarzee1,psangha1}@jhu.edu
{jsedoc}@stern.nyu.edu

Abstract

The COVID-19 pandemic has made a huge global impact and cost millions of lives. As COVID-19 vaccines were rolled out, they were quickly met with widespread hesitancy. To address the concerns of hesitant people, we launched VIRA, a public dialogue system aimed at addressing questions and concerns surrounding the COVID-19 vaccines. Here, we release VIRADialogs, a dataset of over 8k dialogues conducted by actual users with VIRA, providing a unique real-world conversational dataset. In light of rapid changes in users' intents, due to updates in guidelines or in response to new information, we highlight the important task of intent discovery in this use-case. We introduce a novel automatic evaluation framework for intent discovery, leveraging the existing intent classifier of VIRA. We use this framework to report baseline intent-discovery results over VIRADialogs, that highlight the difficulty of this task.

1 Introduction

As COVID-19 vaccines became available in late 2020, they were met with widespread vaccine hesitancy (Goldstein et al., 2015; Sallam, 2021), a phenomena recognized as a top global concern by the World Health Organization (WHO) in 2019. To address such hesitancy, one needs accurate, reliable, and up to date information, constantly available to the general public.

In recent years, task-oriented Dialogue Systems (DSs) have become an integral part of our daily lives, covering domains such as banking, tourism, and government agencies (Androustopoulos et al., 2019).

Correspondingly, we introduced VIRA,¹ the Vaccine Information Resource Assistant – an informational DS that aims to engage with COVID-19 vaccination concerns and questions. VIRA is able

to respond to 181 different concerns, accumulated over the course of the pandemic. VIRA responses were written and vetted by leading medical and public health experts, relying on up-to-date facts and guidelines. An example of a dialog conducted with VIRA is presented in Table 1.

We refer to this list of concerns and questions as *intents*. This is a slightly modified definition of intents, which are usually general tasks or goals which the user tries to accomplish (Jurafsky and Martin, 2009). In VIRA the intents are phrased as complete sentences, e.g., “Is the vaccine safe?”, as opposed to a synthetic class name like “vaccine_safe”, and the goal of the intent classifier is to predict the correct intent, to which VIRA responds to the user with a pre-defined message.

Advancing DSs depends on the availability of conversational datasets on which models can be trained. In domains where fact-based information is a necessity, e.g., healthcare, curating such datasets is often challenging: users in a crowdsourcing setting may not share their authentic concerns, may not represent public opinion, or may even attempt trolling. In addition, creating responses by highly-trained individuals is a very demanding process (Liu et al., 2021). Furthermore, even if one has collected data from a real-world DS, there could be limitations for making such data public.

The availability of VIRA enabled us to collect dialogs with real-world users, following word-of-mouth or social media advertising, presumably conveying genuine interest or concerns related to the vaccines. VIRA was launched in July 2021 and over the course of 10 months it accumulated over 8k conversations. We refer to this collection of conversations as VIRADialogs and release it as part of this work.²

After deploying a DS in a real-world setting, users may introduce new intents, which are not

*These authors equally contributed to this work.

¹<https://vaxchat.org>

²<https://vaxchat.org/research>

Side	Text	Intent
System	<i>Hey! Ask me your vaccine questions.</i>	
User	<i>Hi. Should I be concerned about side effects of the vaccine if I'm breastfeeding?</i>	Is breastfeeding safe with the vaccine
System	<i>Perhaps check this out: while trials did not include breastfeeding moms, COVID-19 vaccines are "non-replicating" vaccines. This type of vaccine poses no risk for breastfed infants, so COVID-19 vaccines are also safe when you're breastfeeding!</i>	

Table 1: An example of a dialog in VIRADialogs.

part of the system’s predefined intents (Grudin and Jacques, 2019). VIRA’s use case represents such an extreme example where users’ intents change rapidly due to updates in guidelines and protocols, or as a response to new information (e.g., the outbreak of novel variants). Hence, we needed to frequently update and expand the set of user intents. This makes VIRADialogs a unique resource for *Intent discovery* methods. These methods aim to reveal such new intents from conversational logs, trying to identify the most salient new intents, which can then be reviewed and added to the DS using a human-in-a-loop process.

Existing datasets for intent classification and discovery (e.g., Larson et al. (2019)) were collected, at least in part, by showing crowd annotators queries and asking them to provide rephrases. Thus, for each query, a similar number of rephrases is collected. VIRADialogs, on the other hand, comes from a real-world use-case, and thus presumably better reflects how people communicate; the real intent distribution; and how it evolves over time – an aspect which as far as we know, is not covered by any existing data.

To directly evaluate intent discovery methods, one would need to annotate each user utterance with its gold intent, and compare this intent with the prediction of each method.

While this annotation approach is typically more precise, it is far from trivial in our real-world use-case considering the size of VIRADialogs and the high number of intents involved. Moreover, as we are dealing with rapidly changing user intents in light of new information about the virus and new guidelines, the distribution of user intents over time is not uniform, which means that manual annotation – even for a test set – would require continuous annotation over the whole time period. This makes manual annotation quite challenging.

As a practical alternative, we propose a novel retrospective evaluation paradigm which leverages the existing intent classifier of VIRA. We assume that

this classifier, carefully developed over the entire relevant time period, covers most intents present in the data. Thus, we treat it as an Oracle to evaluate various intent discovery methods, independently in each month.

First, the Oracle is used to induce silver labels over the unlabeled user utterances. Next, to evaluate an intent discovery method, the same Oracle is used to classify intents predicted by this method to silver labels, enabling a fully automatic quantitative evaluation. We use this approach to evaluate various intent discovery methods on top of VIRADialogs and further share the code base to reproduce our experiments.³

To summarize, the contribution of this paper is three fold: i) We release VIRADialogs, a unique dataset of real-world human-machine conversations, reflecting COVID-19 vaccine hesitancy; ii) We propose and implement an automatic retrospective evaluation paradigm for intent discovery, relying on the availability of a high quality intent classifier; iii) We use our evaluation approach to report baseline performance of various intent discovery methods on top of VIRADialogs.

2 Related Work

Benchmark Datasets and COVID-19 DSs. Popular benchmark datasets for intent classification are also used to benchmark the task of intent discovery and were curated (at least in part) by asking crowd annotators to phrase intents suitable to a DS setting (e.g., Liu et al. (2019a); Larson et al. (2019)). Arora et al. (2020) introduce HINT3, a challenging benchmark whose test set comes from real chats in 3 domains. However, the test set contains less than 1,000 queries for each domain collected in a 15-day period, a relatively limited scope for intent discovery.

The pandemic outbreak led to the development of a few other DSs in this domain. Welch et al.

³<https://github.com/IBM/vira-intent-discovery>

(2020) introduce expressive interviewing – an interview style aiming to encourage users to express their thoughts and feelings by asking them questions about how COVID-19 has impacted their lives. Chalaguine and Hunter (2021) built and studied a DS specifically addressing COVID-19 vaccine hesitancy and showed that 20% of study participants changed their stance in favor of the vaccine after conversing with the system. While their motivation is similar to ours, the analyzed data is smaller and is coming from crowd annotators.

Intent Discovery Methods. Recent work by Rabinovich et al. (2022) introduced a fully unsupervised pipeline for detecting intents in unhandled DS logs. Utterances are encoded into vector representations, and a Radius-based Clustering (RBC) algorithm assigns each to an existing cluster, in case it surpasses a predefined similarity threshold; or use it to initiate a new cluster. The algorithm automatically selects the number of clusters, and does not enforce full partitioning of the underlying data, but rather enables outliers — instances that lay in isolation of discovered clusters. The paper suggests a method for selecting cluster representatives aimed at maintaining centrality and diversity.

Key Point Analysis (KPA) (Bar-Haim et al., 2020a,b, 2021a) proposes a framework that provides both textual and quantitative summary of the main points in a given data. KPA extracts the main points discussed in a collection of texts, and matches the input sentences to these key points. It has been shown to perform well on argumentative data, as well as in online surveys and on user reviews. To our knowledge, our work is the first to utilize KPA in the context of DSs.

3 The VIRA System

Users communicate with VIRA using either a web-based User Interface (UI)⁴ or a WhatsApp application. The general flow is that users enter free text expressing their questions and concerns about the vaccine, VIRA detects the intent within a pre-defined intent list, and in turn provides a suitable response, reviewed by medical experts. VIRA supports conversations in English.⁵ Below we describe VIRA’s main components.

⁴The UI is also embedded on the web pages of health departments, vaccine advocacy organizations, and health care facilities.

⁵A later version supported Spanish as well, however those conversations are left out of this work.

Profanity Classifier. We use a dictionary⁶ to identify utterances with suspected offensive language, to which VIRA presents a generic response.

Dialog-Act Classifier. We classify each user input to one of the supported dialog acts. For certain dialog acts, e.g., ‘Hi’, VIRA presents a generic response. Full details can be found in Appendix A.

Intent Classifier. Intents representing distinct vaccine concerns were carefully curated through various means: using a Twitter analysis, reviewing audience questions in Zoom-based public forums hosted by authors’ affiliated academic centers, and synthesizing web pages with frequently asked questions. The intents were defined also by taking into consideration the scientific knowledge towards the vaccine at that point. Over time, new concerns were identified by monitoring incoming queries to VIRA and eventually the list comprised of 181 intents, presented in Appendix G.

The requirement from VIRA was to provide specific answers to specific concerns, and general answers to general concerns – hence, “I am afraid the vaccine will change my DNA” and “I distrust this vaccine” required different answers, and thus were represented as separate intents, although the latter can be entailed from the former.

The intent classifier was trained on data collected from crowd annotators using the Appen platform.⁷ Annotators were presented with an intent and asked to express it in three different ways, as if conversing with a knowledgeable friend (see Section 6.1 for more details). The classifier’s top-ranked intent is selected for providing a response from the Response Database. If no intent passed a pre-defined threshold, a corresponding response is given.

Response Database. This database contains VIRA’s responses to intents. Each entry specifies multiple responses to a specific intent, to increase output diversity. The responses contain varying information and tone from which VIRA selects one randomly. The database was created and is maintained by experts in the field based on up-to-date facts and guidelines. All responses sought to minimize technical language and maintain brevity through a 280-character limit.

Feedback Mechanism. VIRA incorporates a feedback mechanism that enables users to correct the course of conversation. This feedback allows VIRA’s personnel to improve the system over time

⁶<https://github.com/LDNOOBW/>

⁷appen.com

# Dialogs	8,088
Total # Turns	28,202
Avg. turns per dialog	3.5
Total # Turns w/o feedback turns	20,304

Table 2: Stats of VIRADialogs. Row 2 includes turns that are both free text and a feedback selection (see Appendix B), whereas row 4 indicates free text turns only.

(see more details in Appendix B).

All VIRA’s chats, including feedback selections and classifiers output, are recorded for off-line analysis, without storing identifiable information.

4 The VIRADialogs Dataset

VIRADialogs contains the interactions conducted with VIRA by actual users from July 2021 to May 2022. The full dialogues, as well as user feedback, predicted intents, dialog acts, and offensive language predictions are released to the research community. The data has been anonymized by masking locations, names, e-mails, phone numbers, and birth-dates, along with suspected offensive terms, using a range of regular expressions, the Profanity Classifier, and the spaCy Named Entity recognizer.⁸ In addition, we have excluded dialogues between 29-30, July 2021, in which VIRA was confronted with multiple chats containing offensive language, presumably from individuals who attempted to break the system. Stats of VIRADialogs are presented in Table 2.

5 Retrospective Intent Discovery Evaluation

An important contribution of this work is to show how to leverage an existing DS intent *classifier* – like the one described in Section 3, referred to as an *Oracle* – to automatically evaluate intent *discovery* methods over a collection of dialogs. An overview of the proposed approach is depicted in Figure 1. The underlying components are described below, using the following terminology:

ORACLE INTENTS: The intents supported by the Oracle. **SILVER LABELS:** Subset of ORACLE INTENTS, induced over a given data. **PREDICTED INTENTS:** Intents predicted and phrased by an intent discovery method. **PREDICTED ORACLE INTENTS:** Subset of PREDICTED INTENTS mapped by the Oracle to ORACLE INTENTS.

⁸<https://spacy.io/>

5.1 Inducing SILVER LABELS

Given a set of unlabeled user utterances from conversational logs we randomly split it to train and test sets. The train set is used to induce SILVER LABELS, while the test set is used for evaluation. The motivation of the train-test split is three-fold: (i) enabling to evaluate how consistent is the Oracle itself to ensure the emerging SILVER LABELS are representative of the entire data; (ii) preserving an option to evaluate supervised intent discovery methods in future work; (iii) using the Oracle test set results to estimate upper bound test performance.

We apply the Oracle to predict (at most) one intent for each utterance in the train set. Utterances on which the Oracle confidence was below a pre-specified threshold are placed in a *none* cluster. Since each utterance is mapped to one intent, we obtain clusters of utterances around ORACLE INTENTS. Next, we sort all clusters by their size, and define the top K ranked ones and their intent representatives as the SILVER LABELS, where ranking criteria can vary (see Section 6.2 for a concrete example).

5.2 Evaluation Method

5.2.1 Matching PREDICTED INTENTS to SILVER LABELS

PREDICTED INTENTS often cannot be matched directly to SILVER LABELS. E.g., an intent discovery method might output “I don’t want to get a booster shot”, whereas the corresponding intent in the SILVER LABELS would be “Will I need a booster shot?”. Assuming manual mapping is not feasible, we use the Oracle to map each of the PREDICTED INTENTS to – at most – one of the ORACLE INTENTS, resulting in a set of PREDICTED ORACLE INTENTS. Utterances of PREDICTED INTENTS which are not mapped due to low confidence of the Oracle are placed in a *none* cluster. Note, that in principle this set may contain ORACLE INTENTS that were not selected as SILVER LABELS.

5.2.2 Evaluation Measures

We consider two types of measures to evaluate intent discovery methods: (a) the similarity of PREDICTED INTENTS to SILVER LABELS; and (b) the similarity of cluster partitions generated on the test data by the Oracle and the evaluated method.

Intent Discovery Measures

We estimate the quality of PREDICTED INTENTS (PIs) using the PREDICTED ORACLE INTENTS

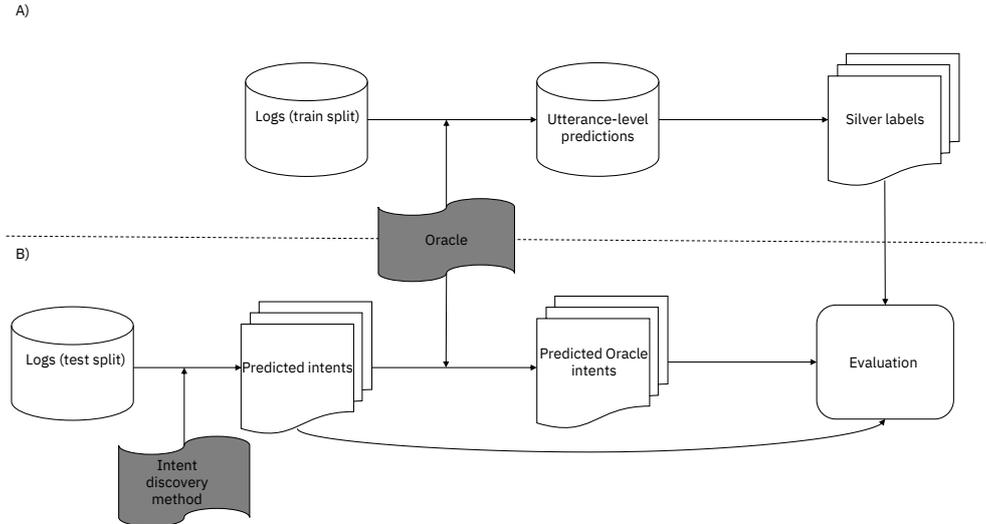


Figure 1: Overview of the evaluation pipeline.

(POIs) and SILVER LABELS (SLs) as follows:

Recall: Coverage of SILVER LABELS by method

$$\frac{|POIs| \cap |SLs|}{|SLs|}$$

Precision: Ratio of PREDICTED INTENTS mapped to SILVER LABELS

$$\frac{|POIs| \cap |SLs|}{|PIs|}$$

JS-distance: We place utterances of PREDICTED ORACLE INTENTS not in the SILVER LABELS in the *none* cluster. We normalize the sizes of the clusters induced by the SILVER LABELS and the PREDICTED ORACLE INTENTS— including the *none* cluster — into two probability distributions, and report their Jensen-Shannon divergence.

Intent Clusters’ Analysis

We compare the partitioning of the test data induced by the PREDICTED INTENTS and the Oracle using the following standard measures: **Adjusted Rand-Index (ARI):** The rand index corrected for chance (Vinh et al., 2010). **Adjusted Mutual-Information (AMI):** The mutual information corrected for chance (Meilă, 2007). **V-Measure:** The harmonic mean between homogeneity and completeness (Rosenberg and Hirschberg, 2007).

6 Experimental Setup

In this section we present a concrete implementation of the framework described in Section 5 using VIRA and VIRADialogs to automatically evaluate various unsupervised intent discovery methods.

6.1 The Oracle

For the Oracle we use VIRA’s intent classifier (Section 3), described below.

Data

For each intent amongst the final 181 intents covered by VIRA, we asked 18 Appen crowd annotators to contribute three different *intent expressions*, i.e., different phrasings of questions or comments by which they could have expressed the intent while chatting with a knowledgeable friend.⁹ Qualified annotators were paid on average 7.5-8\$ an hour.¹⁰ After manual cleaning we ended up with 7,990 expressions, between 20-100 for each intent.¹¹ We release this dataset as part of this work, contributing to the task of single-domain intent classification.¹²

Model and Training

We split the intent expressions associated with each intent to train (65%), dev (8%), and test (27%) sets, with 5,169, 664 and 2,139 examples, respectively, over which we fine-tuned RoBERTa-large (Liu et al., 2019b). Full model implementation de-

⁹Note that we collected data from crows annotators solely for training the intent model. VIRADialogs itself contains real interactions and is not crowd-sourced.

¹⁰For each annotator, we calculate the BLEU score of its expressions w.r.t the intent. Annotators with score < 0.07 are determined as qualified, aiming at promoting diversity.

¹¹The data also contains a small set of 324 intent expressions, extracted manually from VIRADialogs.

¹²https://research.ibm.com/haifa/dept/vst/debating_data.shtml

Fold	Train size	Test size	# SILVER LABELS
Jul-21	3,011	3,294	45
Aug-21	1,169	1,285	43
Sep-21	868	911	37
Oct-21	718	747	34
Nov-21	506	521	30
Dec-21	730	769	31
Jan-22	799	905	40
Feb-22	239	250	23
Mar-22	212	229	18
Apr-22	192	206	20

Table 3: # utterances in VIRADialogs splits for intent discovery evaluation.

tails and threshold tuning are in Appendix C. Note, when the confidence score of the top prediction was below a pre-specified threshold, the model does not predict any intent.

6.2 Inducing SILVER LABELS

We apply to VIRADialogs filters to reduce noise and irrelevant input.¹³ We split the remaining utterances into monthly intervals, resulting in 10 data folds, and subsequently evenly split the utterances in each fold to train and test (indifferent to which dialogue utterances came from).

To reduce noise in generating SILVER LABELS, we additionally filter from the train set utterances classified with a dialog act (e.g., ‘greeting’) or as offensive, as the ratio of intents related to COVID-19 vaccines in these utterances is much smaller.

We then apply the Oracle on each utterance in the train set, resulting in ORACLE INTENTS and corresponding clusters. We sort them based on their prevalence and define the top K as SILVER LABELS. In practice, we do this by accumulating the clusters until we reach a coverage of 80% (out of all texts on which the Oracle had a confident prediction) or that the number of utterances mapped to an intent is below 3 (removing a long tail of small clusters). The number of utterances and SILVER LABELS for each fold are reported in Table 3.

6.3 Intent Discovery Methods

6.3.1 Clustering Algorithms

We evaluate two clustering algorithms. Since one cannot assume that the number of SILVER LABELS is known *a priori*, we use \sqrt{N} as a simple

¹³We filter user feedback, utterances longer than 250 characters, contain at most one non-masked word, or less than 75% alpha-numeric characters.

heuristic to determine the number of clusters, including the *none* cluster, where N is the number of utterances being clustered. Short utterances, containing less than 5 recognized words, were placed in advance in the *none* cluster. Analysis takes a few minutes on CPU.

K-Means. We use the K-Means algorithm from the SciKit-Learn package (Pedregosa et al., 2011) with the default settings. Each utterance was represented using its Sentence-BERT representation (Reimers and Gurevych, 2019).

sequential Information Bottleneck (sIB). As a bag-of-words baseline, we use the sIB algorithm of Slonim et al. (2002).¹⁴ The algorithm uses as input the Term-Frequency vector representations and is executed with the default settings, after stop-word filtering and stemming.

Intent Extraction

We select a single user utterance per cluster to represent an intent, resulting with the list of PREDICTED INTENTS. The selection is based on a statistical analysis of n-grams in the data. For each cluster, we first find the n-grams that are significantly more common in this cluster compared to other clusters based on hyper-geometric test ($p = 0.05$). Then we select the user utterance in the cluster that includes the maximal number of significant n-grams found in that cluster.

6.3.2 End-to-End Methods

We evaluate two end-to-end methods with mostly default settings. These methods determine the number of clusters internally, and map utterances to a *none* cluster as they see fit. For comparison purposes, we take the top $\sqrt{N} - 1$ prevalent clusters for evaluation. The rest of the clusters are added to the *none* cluster.

Key Point Analysis (KPA). We use KPA as provided by the IBM Debater Academic Early Access Program (Bar-Haim et al., 2021b). The underlying model of KPA matches utterances with key point candidates, identified automatically. Adjustments for this task can be found in Appendix D. The service took about 3.5 hours to complete the analysis.

Radius-based Clustering (RBC). We approached the authors of Rabinovich et al. (2022) to produce the results for this evaluation. Adjustments for this task can be found in Appendix E. RBC took a few minutes to run on CPU.

¹⁴<https://github.com/IBM/sib>

Recall	Precision	f1	JS-distance
0.79(± 0.08)	0.8(± 0.08)	0.8(± 0.08)	0.16(± 0.04)

Table 4: Evaluation of the Oracle on VIRADialogs test sets (weighted-avg over the monthly intervals.)

7 Results and Discussion

7.1 The Oracle

We first establish the quality of VIRA’s intent classifier used as the Oracle in various ways.

Inference on Intent expressions test set. We evaluate the Oracle on the test set of the collected intent expressions, using the threshold tuned on the dev set (Section 6.1). The Oracle achieves a micro-averaged precision / recall / f1 of 0.85 / 0.74 / 0.79 on dev, and 0.88 / 0.77 / 0.82 on test.

Inducing SILVER LABELS and matching PREDICTED INTENTS. We manually evaluate the Oracle’s accuracy in (i) inducing SILVER LABELS (Section 5.1) and (ii) matching PREDICTED INTENTS to SILVER LABELS (Section 5.2.1).

For (i), we randomly sample 10 SILVER LABELS from the train set of each of the 10 folds. For each silver label we sample 2 utterances mapped to it ($200 < \text{utterance}, \text{SILVER LABELS} >$ pairs overall). For half of the pairs, we randomly replace the silver label with one of the other ORACLE INTENTS (thus, obtaining negative pairs). We asked 3 annotators to annotate whether a given pair of texts has a similar intent or meaning, and took the majority vote as the ground-truth (see more details in Appendix F). The accuracy of the Oracle on this data is 0.85.

For (ii), we randomly select from each fold and for each evaluated method 5 pairs of $< \text{PREDICTED INTENTS}, \text{PREDICTED ORACLE INTENTS} >$ where PREDICTED ORACLE INTENTS are part of the SILVER LABELS (200 pairs overall). We use the same annotation task as in (i). The accuracy of the Oracle on this data is 0.86.¹⁵

Consistency over VIRADialogs test. To recall, we evaluate methods on the *test* set w.r.t SILVER LABELS induced from the *train* set. Here, we would like to examine the consistency of the Oracle’s predictions between the sets which also implies the representativeness of the SILVER LABELS for the entire data. We do that by inferring the Ora-

¹⁵1. We note that on average for 24% of PREDICTED INTENTS the Oracle is not confident (covering 22% of the texts), and for an additional 18% the PREDICTED ORACLE INTENTS are not part of the SILVER LABELS. 2. For one of the methods there were less than 5 pairs, so the overall number of pairs is 199.

cle over the test set of each monthly fold to produce clusters around ORACLE INTENTS. We then rank them by prevalence and accumulate them to define the PREDICTED INTENTS (which are also trivially PREDICTED ORACLE INTENTS), as was done to induce SILVER LABELS on the train set. The results are presented in Table 4. The Oracle achieves a weighted-f1 of 0.795, demonstrating reasonable consistency between the train and test split in each fold. This also can be considered an upper limit of success for other methods.

Overall, the above evaluation has shown that the Oracle performs well in matching utterances and PREDICTED INTENTS to intents, and that SILVER LABELS are relatively representative.

7.2 Intent Discovery Methods

Results for the 4 methods we evaluate are presented in Table 5. RBC has the highest coverage uncovering 45% of the SILVER LABELS, and reaching an f1 of 0.51. These results also indicate the difficulty of this task, as the majority of SILVER LABELS remain undetected. Note that similar precision with worse recall, such as with K-Means compared to KPA, suggests more redundancy in the PREDICTED INTENTS of the former.

KPA is much better at the clustering measures, and is thus useful for finding good examples for each intent. This might be due to KPA’s matching engine, trained to match sentences with key points (similarly to intents in VIRA, key points are concise representations of main points in the data).

It should be noted that for simplicity we used “off-the-shelf” methods with minor adaptations, to resemble a real-world setting where a user would like to get a fast impression of how well such methods perform for a given use-case with minimal effort. In addition, we used a simple heuristic to determine the number of clusters. It is likely that with proper tuning of parameters, domain adaptation of underlying models, tuning of number of clusters, etc., the performance would have been higher.

7.3 Qualitative Analysis of Emerging Intents

The SILVER LABELS and PREDICTED ORACLE INTENTS cover varying issues, and so we sought to analyze some of the more high-profile ones in light of events that occurred in their context.

We selected two intents: i) *How effective is the vaccine against the Omicron variant*, coupled with the rise in Omicron-related cases in December

	Recall	Precision	f1	JS-distance	ARI	AMI	Clustering-f1	V-measure
sIB	0.39(± 0.09)	0.52(± 0.09)	0.44(± 0.09)	0.33(± 0.03)	0.05(± 0.03)	0.24(± 0.05)	0.07(± 0.04)	0.37(± 0.05)
K-Means	0.42(± 0.07)	0.57(± 0.08)	0.49(± 0.06)	0.34(± 0.03)	0.06(± 0.03)	0.32(± 0.06)	0.1(± 0.05)	0.43(± 0.05)
RBC	0.45 (± 0.11)	0.61 (± 0.11)	0.51 (± 0.11)	0.32 (± 0.04)	0.15(± 0.04)	0.28(± 0.05)	0.19(± 0.04)	0.39(± 0.04)
KPA	0.44(± 0.08)	0.57(± 0.09)	0.49(± 0.07)	0.32 (± 0.03)	0.24 (± 0.05)	0.38 (± 0.05)	0.3 (± 0.04)	0.48 (± 0.05)

Table 5: Evaluation of intent discovery methods on VIRADialogs. The numbers are a weighted-average over the monthly intervals. Best method for each metric is highlighted in bold. *Takeaway:* Methods are able to uncover up to 45% of the intents, demonstrating the difficulty of this task. RBC is able to uncover more intents and at better precision. KPA is much better at uncovering correct placements of utterances within clusters.

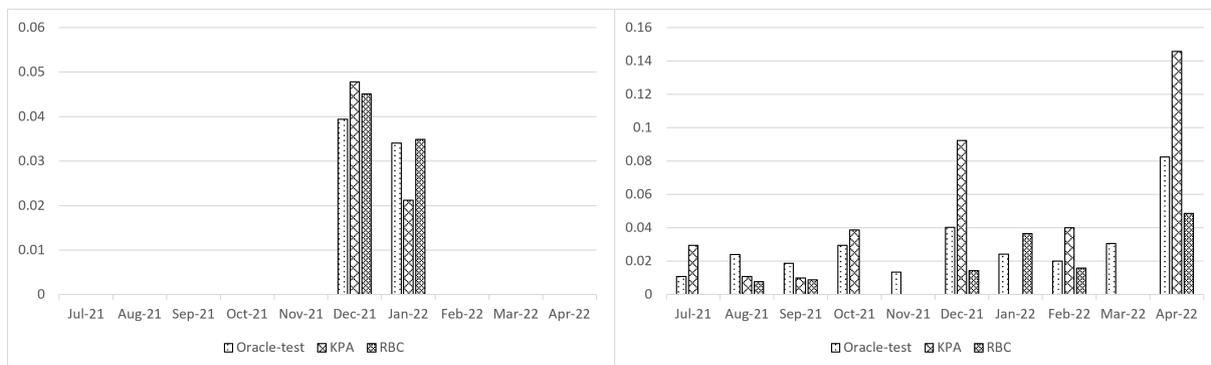


Figure 2: Cluster ratios of *How effective is the vaccine against the Omicron variant* (left); *Will I need a booster shot* (right). *Takeaway:* Predictions of methods on VIRADialogs correlate well with real-world developments.

2021;¹⁶ and ii) *Will I need a booster shot*, coupled with booster recommendations in late November 2021¹⁷ and March 2022.¹⁸ In Figure 2, we plotted the cluster ratio¹⁹ of each intent among all clusters in a given month, as predicted by the Oracle, KPA, and RBC on the test set. Presumably, high ratio indicates a peak of interest for this intent.

For Omicron, methods highlight emerging interest in December and January, correlated with its real-time occurrence. To the right, methods predict interest in boosters peaking in December and April. We also note that differences between systems are sometimes non-negligible (e.g., as evident by the different peaks in the right figure). Overall, this analysis demonstrates how outstanding events in the COVID-19 timeline can be captured by the evaluated intent discovery methods.

8 Conclusions

In this paper we first describe VIRA, an informational DS addressing hesitancy towards COVID-19

¹⁶<https://www.cdc.gov/coronavirus/2019-ncov/science/forecasting/mathematical-modeling-outbreak.html>

¹⁷<https://www.cdc.gov/media/releases/2021/s1129-booster-recommendations.html>

¹⁸<https://www.cdc.gov/media/releases/2022/s0328-covid-19-boosters.html>

¹⁹Cluster ratio is defined as the size of an intent cluster divided by the overall number of utterances for a given month.

vaccines. VIRA provides access to accurate, up-to-date information in English, written by experts. We believe that the associated VIRADialogs data, containing 8k dialogs of VIRA with real-world users, would be a valuable resource to the relevant research community. As an initial example of the potential of this data, we demonstrate how it can be utilized to evaluate intent discovery methods. We propose an automatic evaluation framework that relies on the availability of a corresponding intent classifier, and report the results of 4 diverse methods, concluding that this benchmark represents a significant challenge.

While automatic evaluation is clearly more practical than manual one, developing the required intent classifier involves a non-trivial effort. Still, we envision two potential outcomes of our work. First, additional intent-discovery methods can be easily evaluated over VIRADialogs data using our implementation, and compared to the baseline performance reported here. Second, the same framework can be implemented in other use cases as well for which a reliable intent classifier is available, opening the door for automatic evaluation of intent discovery methods over additional datasets.

Finally, VIRA is constantly maintained and updated, and is now being expanded to additional languages, along with a Whatsapp implementation, to expand its outreach. In future work we intend to

report the lessons learned from developing VIRA, and the implications for developing a DS in the public health domain.

Acknowledgements

We thank Ella Rabinovich, Roy Bar-Haim, Yoav Kantor, Lilach Eden and the anonymous reviewers for their insightful comments, and Edo Cohen-Karlik, Alex Michel and Elad Venezian for their contribution to VIRA,

9 Limitations

There are a few limitations to our approach, which stem from assumptions made to establish the evaluation pipeline.

- We implement an evaluation pipeline on a single dataset, which we were part of creating, and did not test its compliance with additional datasets.
- We assume a relatively accurate intent classifier, referred to as an Oracle, is available. Thus, our evaluation is not suited for cold-start scenarios.
- We assume the intents covered by the Oracle indeed cover most intents expressed in the data. It is quite possible that as VIRADialogs is a large dataset it included additional intents, beyond the 181 covered by the Oracle, which probably impacted the accuracy of the evaluation. We note, though, that automatic evaluation, as proposed in this work, is always prone to such issues.
- We evaluated only certain unsupervised methods for intent discovery. Other systems may perform better than the reported baselines.

10 Ethics Statement

This paper describes work around VIRA, a real-world DS addressing COVID-19 vaccine hesitancy. In an attempt to alleviate concerns that users would take action based on information given to them by VIRA which might harm them, the terms of use of the DS state that “This information ... is not intended as a substitute for medical advice”. We were guided with the principle of providing accurate information, thus when building VIRA we incorporated a direct mapping between intents and responses. Future endeavours based on this

dataset, e.g., for building a generative bot for addressing vaccine hesitancy, should be aware of the ramifications of showing to users such content.

In addition, the terms of use stated that queries are stored and may be used for research purposes.

The chats collected might have originally contained offensive language, often as a result of the sensitivity of the domain to some users. We made a dedicated effort to flag these cases and mask problematic terms. However, we did so with automatic measures, so the dataset might still contain such language. Finally, although the data was anonymized by masking various expressions, it is still possible that some sensitive medical concerns remain.

References

- Aggeliki Androutsopoulou, Nikos Karacapilidis, Euripidis Loukis, and Yannis Charalabidis. 2019. [Transforming the communication between citizens and government through ai-guided chatbots](#). *Government Information Quarterly*, 36(2):358–367.
- Gaurav Arora, Chirag Jain, Manas Chaturvedi, and Krupal Modi. 2020. [HINT3: Raising the bar for intent detection in the wild](#). In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 100–105, Online. Association for Computational Linguistics.
- Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020a. [From arguments to key points: Towards automatic argument summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4029–4039, Online. Association for Computational Linguistics.
- Roy Bar-Haim, Lilach Eden, Yoav Kantor, Roni Friedman, and Noam Slonim. 2021a. [Every bite is an experience: Key Point Analysis of business reviews](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3376–3386, Online. Association for Computational Linguistics.
- Roy Bar-Haim, Yoav Kantor, Lilach Eden, Roni Friedman, Dan Lahav, and Noam Slonim. 2020b. [Quantitative argument summarization and beyond: Cross-domain key point analysis](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 39–49, Online. Association for Computational Linguistics.
- Roy Bar-Haim, Yoav Kantor, Elad Venezian, Yoav Katz, and Noam Slonim. 2021b. [Project Debater APIs: Decomposing the AI grand challenge](#). In

- Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 267–274, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Lisa A. Chalaguine and Anthony Hunter. 2021. [Addressing popular concerns regarding COVID-19 vaccination with natural language argumentation dialogues](#). In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceedings*, volume 12897 of *Lecture Notes in Computer Science*, pages 59–73. Springer.
- Susan Goldstein, Noni E. MacDonald, and Sherine Guirguis. 2015. [Health communication and vaccine hesitancy](#). *Vaccine*, 33(34):4212–4214. WHO Recommendations Regarding Vaccine Hesitancy.
- Jonathan Grudin and Richard Jacques. 2019. [Chatbots, humbots, and the quest for artificial general intelligence](#). In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19*, page 1–11, New York, NY, USA. Association for Computing Machinery.
- D. Jurafsky and J.H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall.
- Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. [An evaluation dataset for intent classification and out-of-scope prediction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.
- Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021. [Towards emotional support dialog systems](#).
- Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019a. [Benchmarking natural language understanding services for building conversational agents](#).
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Marina Meilá. 2007. [Comparing clusterings—an information based distance](#). *Journal of Multivariate Analysis*, 98(5):873–895.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. [Scikit-learn: Machine learning in Python](#). *Journal of Machine Learning Research*, 12:2825–2830.
- Ella Rabinovich, Matan Vetzler, David Boaz, Vineet Kumar, Gaurav Pandey, and Ateret Anaby-Tavor. 2022. [Gaining insights into unrecognized user utterances in task-oriented dialog systems](#).
- Nils Reimers and Iryna Gurevych. 2019. [Sentencebert: Sentence embeddings using siamese bert-networks](#). *CoRR*, abs/1908.10084.
- Andrew Rosenberg and Julia Hirschberg. 2007. [V-measure: A conditional entropy-based external cluster evaluation measure](#). In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.
- Malik Sallam. 2021. [Covid-19 vaccine hesitancy worldwide: A concise systematic review of vaccine acceptance rates](#). *Vaccines*, 9(2): 160.
- Noam Slonim, Nir Friedman, and Naftali Tishby. 2002. [Unsupervised document classification using sequential information maximization](#). In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '02*, page 129–136, New York, NY, USA. Association for Computing Machinery.
- Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. [Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance](#). *Journal of Machine Learning Research*, 11(95):2837–2854.
- Charles Welch, Allison Lahkala, Veronica Perez-Rosas, Siqi Shen, Sarah Seraj, Larry An, Kenneth Resnicow, James Pennebaker, and Rada Mihalcea. 2020. [Expressive interviewing: A conversational system for coping with COVID-19](#). In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

A Dialog-Act Classifier

This classifier is used for categorizing the user input as one of the supported dialog acts: *greeting*, *farewell*, *negative reaction*, *positive reaction*, *concern* and *query*. The classifier was trained on utterances extracted from early chats labeled for their dialog act. VIRA responds to input texts that are classified with one of the first 4 dialog act types with corresponding generic texts. For example, a response to a greeting (e.g., ‘Hi’) is “Hello, what

are your thoughts about the COVID-19 vaccine?”. Utterances classified as either *concern* or *query* are passed to the Intent Classifier.

B Feedback Mechanism

VIRA incorporates a feedback mechanism that gives users the option to correct the course of conversation. When users give a thumbs down for a VIRA’s response, or when the intent classifier is not confident, VIRA shows to the user the top-3 predicted intents in a menu to select from with additional options for indicating that: (a) none of these intents address the concern, or (b) the input does not express a concern at all. This feedback allows VIRA’s developers and persons maintaining the Response Database to improve the system over time. For example, when (b) is selected, it indicates a false positive for the Dialog-Act Classifier.

C Intent Classification Model Details

As a base model for fine-tuning the intent classifier of VIRA, used as the Oracle, we use RoBERTa-large (354M parameters). We use AdamW optimizer with a learning rate of $5e-6$ and a batch size of 16. We fine-tune the model for 15 epochs and select the best performing checkpoint on the dev set according to overall accuracy. Training took 20 minutes on 1 v100 GPU. The confidence threshold of the model was tuned by taking the minimal threshold such that the precision on the dev set > 0.85 , resulting in a threshold of 0.296.

D Key Point Analysis Details

First, utterances for which no match was found above a threshold are placed in a *none* cluster.

Furthermore, preliminary experiments have shown KPA is producing too few intents, so as an adjustment for this task we: (i) set *limit_n_cands = false* to remove the limit on the number of key point candidates; (ii) set *n_top_kps = 1000* to remove the limit on number of clusters in the output, which also implies no minimal cluster size. The hypothesis is that (i)+(ii) will increase the amount and diversity of resulting key points at the expense of run-time.

E Radius-based Clustering Details

As an adjustment, chit-chat utterances which are filtered at the first phase of the algorithm are placed

in a *none* cluster. The minimal similarity threshold is set to 0.55. As with KPA we do not set a minimum size for clusters.

F Labeling User Utterances and PREDICTED INTENTS to SILVER LABELS

We presented annotators with pairs of texts, where one text can be either a user utterance or an intent from the PREDICTED INTENTS, and the other a silver label. We asked, “Do the above two texts convey the same meaning or intent?”. The annotators belong to a group with high success on previous tasks of our team, and the task included a few positive and negative examples to illustrate our objective. In addition, we included test questions of text pairs manually selected from the training data of the Oracle, and annotators with less than 70% accuracy on them were removed from the task.

G Intents Supported by VIRA

Intent
COVID-19 is not as dangerous as they say
Do I need to continue safety measures after getting the vaccine?
How long until I will be protected after taking the vaccine?
How many people already got the vaccine?
I am afraid the vaccine will change my DNA
I am concerned getting the vaccine because I have a pre-existing condition
I am concerned I will be a guinea pig
I’m concerned the vaccine will make me sick.
I am not sure if I can trust the government
I am young and healthy so I don’t think I should vaccinate
I distrust this vaccine
How much will I have to pay for the vaccine
I don’t think the vaccine is necessary
I don’t trust the companies producing the vaccines
I don’t want my children to get the vaccine
I think the vaccine was not tested on my community
I’m not sure the vaccine is effective enough
I’m waiting to see how it affects others
COVID vaccines can be worse than the disease itself
Long term side-effects were not researched enough
Are regular safety measures enough to stay healthy?
Should people that had COVID get the vaccine?
Side effects and adverse reactions worry me
The COVID vaccine is not safe
The vaccine should not be mandatory
Do vaccines work against the mutated strains of COVID-19?
They will put a chip/microchip to manipulate me
What can this chatbot do?
What is in the vaccine?

Intent
Which one of the vaccines should I take?
Will I test positive after getting the vaccine?
Can other vaccines protect me from COVID-19?
Do I qualify for the vaccine?
I don't trust vaccines if they're from China or Russia
Are the side effects worse for the second shot
Can I get a second dose even after a COVID exposure?
Can I get other vaccines at the same time?
Can I get the vaccine if I have allergies?
Can I get the vaccine if I have had allergic reactions to vaccines before?
Can I have the vaccine as a Catholic?
Can I have the vaccine if I'm allergic to penicillin?
Can I still get COVID even after being vaccinated?
Can you mix the vaccines?
COVID-19 vaccines cause brain inflammation
Do the COVID-19 vaccines cause Bell's palsy?
"Do the mRNA vaccines contain preservatives, like thimerosal?"
Do the vaccines work in obese people?
Do you have to be tested for COVID before you vaccinated?
Does the vaccine contain animal products?
Does the vaccine contain live COVID virus?
Does the vaccine impact pregnancy?
Does the vaccine work if I do not experience any side effects?
How can I stay safe until I'm vaccinated?
"How do I know I'm getting a legitimate, authorized vaccine?"
How do I report an adverse reaction or side-effect
How long do I have to wait between doses?
How many doses do I need?
How was the vaccine tested?
I am concerned about getting the vaccine because of my medications.
I don't want the v-safe app monitoring or tracking me
I don't want to share my personal information
Is breastfeeding safe with the vaccine
Is the Johnson & Johnson vaccine less effective than the others?
Is the vaccine halal?
Is the vaccine Kosher?
Is there vaccine safety monitoring?
Other vaccines have caused long-term health problems
Should I get the COVID-19 vaccine if I am immunocompromised
Should I get the vaccine if I've tested positive for antibodies?
The vaccine includes fetal tissue or abortion by-products
The vaccine was rushed
Vaccine side effects are not getting reported
What does vaccine efficacy mean?
What if I still get infected even after receiving the vaccine?
What if I've been treated with convalescent plasma?
What if I've been treated with monoclonal antibodies?
What is mRNA?
What is the difference between mRNA and viral vector vaccines?
When can I go back to normal life?
Why are there different vaccines?

Intent
Why do I need the COVID vaccine if I don't get immunized for flu
Why do we need the vaccine if we can wait for herd immunity?
Why get vaccinated if I can still transmit the virus?
Will 1 dose of vaccine protect me?
Can I take a pain reliever when I get vaccinated?
Will the vaccine benefit me?
Will the vaccine make me sterile or infertile?
Can we change the vaccine quickly if the virus mutates?
Can I get COVID-19 from the vaccine?
I'm still experiencing COVID symptoms even after testing negative - should I still take the vaccine?
Can children get the vaccine?
Can we choose which vaccine we want?
How long does the immunity from the vaccine last?
"The mortality rate of COVID-19 is low, why should I get the vaccine?"
There are many reports of severe side effects or deaths from the vaccine
How can I get the vaccine?
I am worried about blood clots as a result of the vaccine what is covid?
Who developed the vaccine?
Which vaccines are available?
What are the side effect of the vaccine?
Can I meet in groups after I'm vaccinated?
Is it safe to go to the gym indoors if I'm vaccinated?
How do I protect myself indoors?
What are the effects of long COVID?
Do you need a social security number to get a COVID-19 vaccine?
Do you need to be a U.S. citizen to get a COVID-19 vaccine?
Is it okay for me to travel internationally if I'm vaccinated?
Can my kids go back to school without a vaccine?
Will I need a booster shot?
"If I live with an immuno-compromised individual, do I still need to wear a mask outdoors if I'm vaccinated? "
Does the vaccine prevent transmission?
Why is AstraZeneca not approved in the USA?
Do I need to change my masking and social distancing practices depending on which COVID-19 vaccine I got?
Does the Pfizer vaccine cause myocarditis?
Does the Pfizer vaccine cause heart problems?
What can you tell me about COVID-19 vaccines?
Are there medical contraindications to the vaccines?
How many people died from COVID-19?
What about reports of abnormal periods due to the vaccine?
Do I need the vaccine?
Tell me about the vaccine
Is the Pfizer vaccine safe for young men?
Will vaccination lead to more dangerous variants?
Is it safe for my baby to get the vaccine?
Did a volunteer in the Oxford trial die?
Can I get COVID-19 twice?
Are some vaccines safer for younger children than others?
How long am I immune from COVID-19 if I had the virus?

Intent
Are women more likely to get worse side effects than men?
How do I convince my family and friends to get the COVID-19 vaccine?
Why are COVID-19 vaccination rates slowing in the U.S.?
I'm going to get vaccinated
Is getting vaccinated painful?
What do I do if I lose my COVID-19 vaccination card?
Can I get swollen lymph nodes from the vaccine?
Can my newborn become immune to COVID-19 if I'm vaccinated?
"COVID-19 is over, why should I get the vaccine?"
Did one woman die after getting the J&J vaccine?
Do people become magnetic after getting vaccinated?
Does the vaccine contain eggs?
How is the COVID-19 vaccine different than others?
How soon after I've had COVID-19 can I get the vaccination?
Is it safe for my teen to get the vaccine?
Is this Pfizer vaccine equally effective in kids as it is in adults?
Were the COVID-19 vaccines tested on animals?
What are the side effects of the vaccine in children?
What is the delta variant?
What is the J&J vaccine?
What is the Moderna vaccine?
What is the Pfizer vaccine?
Where are we required to wear masks now?
Who can get the Pfizer vaccine?
Who can I talk to about COVID-19 in person?
Why should I trust you?
Will my child need my permission to get vaccinated?
Will the US reach herd immunity?
Will my child miss school when they get vaccinated?
Is the vaccine FDA approved?
Why do vaccinated people need to wear a mask indoors?
Do vaccinated people need to quarantine if exposed to COVID-19?
What is Ivermectin?
Does the Johnson and Johnson vaccine cause Rare Nerve Syndrome?
What is the difference between quarantine and isolation?
Does the COVID-19 vaccine cause autism?
Does the vaccine cause impotence?
Who is required to get vaccinated under the federal vaccine mandate?
Is the Delta variant more dangerous for kids?
Will there be a booster shot for J&J and Moderna?
Is the booster the same as the original vaccine?
What are the side effects of booster shots?
What is the difference between the third shot and a booster shot?
How common are vaccine side effects?
Why do my kids need a vaccine if they're unlikely to get sick with COVID-19?
What happens if there is a COVID-19 case at my child's school?
Are booster shot side effects worse than those from the second shot?
Is the booster shot dangerous?
Can I get the vaccine if I have Multiple Sclerosis?

Intent
Do children receive the same dose of Pfizer as adults?
What is the Omicron variant?
How effective is the vaccine against the Omicron variant?

Learning Disentangled Representations for Natural Language Definitions

Danilo S. Carvalho¹ Giangiacomo Mercatali^{1†} Yingji Zhang^{1†} Andre Freitas^{1,2}

Department of Computer Science, University of Manchester, United Kingdom¹

Idiap Research Institute, Switzerland²

<firstname.lastname>@[postgrad.†]manchester.ac.uk

Abstract

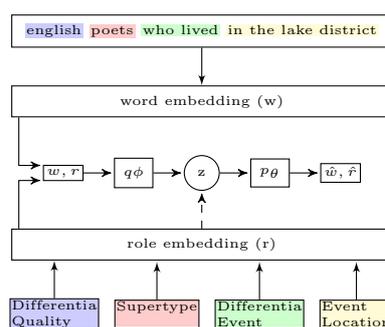
Disentangling the encodings of neural models is a fundamental aspect for improving interpretability, semantic control and downstream task performance in Natural Language Processing. Currently, most disentanglement methods are unsupervised or rely on synthetic datasets with known generative factors. We argue that recurrent syntactic and semantic regularities in textual data can be used to provide the models with both structural biases and generative factors. We leverage the semantic structures present in a representative and semantically dense category of sentence types, definitional sentences, for training a Variational Autoencoder to learn disentangled representations. Our experimental results show that the proposed model outperforms unsupervised baselines on several qualitative and quantitative benchmarks for disentanglement, and it also improves the results in the downstream task of definition modeling.

1 Introduction

Learning disentangled representations is a fundamental step towards enhancing the interpretability of the encodings in deep generative models, as well as improving their downstream performance and generalization ability. Disentangled representations aim to encode the fundamental structure of the data in a more explicit manner, where independent latent variables are embedded for each generative factor (Bengio et al., 2013).

Previous work in machine learning proposed to learn disentangled representations by modifying the ELBO objective of the Variational Autoencoders (VAE) (Kingma and Welling, 2014), within an unsupervised framework (Higgins et al., 2017; Kim and Mnih, 2018; Chen et al., 2018). On the other hand, a more recent line of work claims the benefits of supervision in disentanglement (Locatello et al., 2019) and it advocates the importance of designing frameworks able to exploit structures

Training



Evaluation

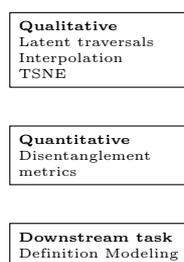


Figure 1: Left: Supervision mechanism with definition semantic roles (DSR) encoded in the latent space. The dotted arrow represent the conditional VAE version. Right: Evaluation framework.

in the data for introducing inductive biases. In parallel, disentanglement approaches for NLP have been tackling text style transfer, and evaluating the results with extrinsic metrics, such as style transfer accuracy (Hu et al., 2017; John et al., 2019; Cheng et al., 2020).

While style transfer approaches investigate the ability to disentangle and control syntactic factors such as tense and gender, the aspect of understanding and disentangling the semantic structure in language is under-explored, but with recent attempts of separating syntactic and semantic latent spaces showing promising results (Chen et al., 2019; Bao et al., 2019). Furthermore, evaluating disentanglement is challenging, because it requires knowledge of generative factors, leading most approaches to train on synthetic datasets (Higgins et al., 2017; Zhang et al., 2021).

In this work, we argue that recurrent semantic structures at sentence level can be leveraged both as inductive biases for enhancing disentanglement (RQ1) but also for providing meaningful generative factors that can be employed for evaluating the degree of disentanglement (RQ2). We also inves-

tigate whether organizing the generative factors in groups may facilitate learning and disentanglement (RQ3). As a result, this work focuses on natural language definitions, which are a textual resource characterised by a principled structure in terms of semantic roles, as demonstrated by previous work which proposed the extraction of structural and semantic patterns in this kind of data (Silva et al., 2016, 2018).

Seeking to address the highlighted issues and answer the research questions, we make the following contributions, also depicted in Figure 1.

1) We design a supervised framework for enhancing disentanglement in language representations by conditioning on the information provided by the semantic role labels (SRL) in natural language definitions. We present two mechanisms for injecting SRL biases into latent variables, firstly, reconstructing both words and corresponding SRL in a VAE, secondly, employing SRL information as input variables for a Conditional VAE (Zhao et al., 2017).

2) We propose a framework for evaluating the disentanglement properties of the encodings on non-synthetic textual datasets. Our evaluation framework employs semantic role label groupings as generative factors, enabling the measurement of several contemporary quantitative metrics. The results show that the proposed bias injection mechanisms are able to increase the degree of disentanglement (separability) of the representations.

3) We demonstrate that models trained with our disentanglement framework are able to outperform contemporary baselines in the downstream task of definition modeling (Noraset et al., 2017).

2 Disentangling framework

In this section we first describe the framework designed for improving disentanglement in natural language definitions with semantic role labels. Secondly, we present three models, shown in Figure 2 based on the Variational Autoencoder (VAE) (Bowman et al., 2016) architecture for achieving disentanglement.

2.1 Disentangling definitions

Definition semantic roles Our framework is based on natural language definitions, which are a particular type of linguistic expression, characterised by high abstraction, and specific phrasal properties. Previous work in NLP for dictionary

definitions (Silva et al., 2018) has shown that there are categories that can be consistently found in most definitions. In fact, Silva et al. (2018) define precise Semantic Role Labels (SRL) for phrases representing definitions, under the name of Definition Semantic Roles (DSR).

The example from (Silva et al., 2018) classifies the semantic roles within "english poets who lived in the lake district" as follows. "poets" as noun category (supertype), "english" as quality of the term (Differentia Quality), "who lived" as event that the subject is involved with (differentia event), and "in the lake district" as the location of the action (Event location). The full DSRs proposed by Silva et al. (2018) are reported in Table 9 in Appendix A.

Disentangling using SRL Our goal is to enhance disentanglement in natural language by injecting categorical structures into latent variables. We find that this goal is well aligned with the findings of Locatello et al. (2019), where it is claimed that a higher degree of disentanglement may benefit from supervision and inductive biases. Our hypothesis is that we may leverage such semantic information for learning representation with higher degree of disentanglement. While in the context of this work we use dictionary definitions as a target empirical setting, we conjecture that these conclusions can be extended to broader definitional sentence-types. The core intuition behind the approach is that the supervision signal should increase the likelihood of point clustering in regions corresponding to, or related to the discrete supervision labels, given the network architecture formulation.

2.2 Definition VAEs

Unsupervised VAE The first training framework that we consider is the traditional variational autoencoder (VAE) for sentences (Bowman et al., 2016), which operates in an unsupervised fashion, as in Figure 2a. The unsupervised VAE employs a multivariate gaussian prior distribution $p(z)$ and generates a sentence x with a decoder network $p_\theta(x|z)$. The joint distribution for the decoder is defined as $p(z)p_\theta(x|z)$, which, for a sequence of tokens x of length T result as $p_\theta(x|z) = \prod_{i=1}^T p_\theta(x_i|x_{<i}, z)$. The VAE objective consists into maximizing the expectation of the log-likelihood which is defined as $\mathbb{E}_{p(x)} \log p_\theta(x)$. Due to the computational intractability of the such expectation value, the variational distribution q_θ is employed to approximate $p_\theta(z|x)$.

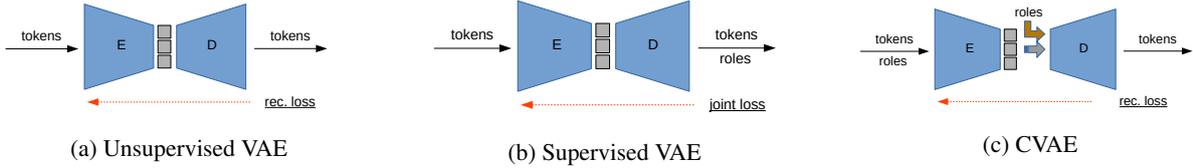


Figure 2: Proposed architectures for learning disentangled representations in definitions.

As a result, an evidence lower bound \mathcal{L}_{VAE} (ELBO) where $\mathbb{E}_{p(x)}[\log p_{\theta}(x)] \geq \mathcal{L}_{\text{VAE}}$, is derived as follows:

$$\mathcal{L}_{\text{Tokens}} = \mathbb{E}_{q_{\phi}(z|x)} \left[\log p_{\theta}(x|z) \right] - \text{KL}q_{\phi}(z|x)||p(z)$$

DSR supervised VAE The aim of this model is to inject the categorical structure of the definition semantic roles (DSR) into the latent variables, by factorizing them into the VAE auto-encoding objective function. In order to achieve this goal, we introduce the variable r for semantic roles, and train the "DSR VAE", where both sentence and semantic roles are auto-encoded. The variable r here operates just as x , with the corresponding label values. As a result, two separate losses are produced and added together for the final loss, as shown in Figure 2b. The ELBO for semantic roles is defined as follows:

$$\mathcal{L}_{\text{Roles}} = \mathbb{E}_{q_{\phi}(z|r)} \left[\log p_{\theta}(r|z) \right] - \text{KL}q_{\phi}(z|r)||p(z)$$

The final loss is given by $\mathcal{L}_{\text{Tokens}} + \mathcal{L}_{\text{Roles}}$.

Conditional VAE with SRL For explicitly leveraging the definition semantic roles, we propose a supervision mechanism based on the Conditional VAE (CVAE) (Zhang et al., 2017), shown in Figure 2c. Similar to the previously described model, we instantiate a VAE framework, where x is the variable for the tokens, and r for the roles. We perform auto-encoding for both roles and tokens, and additionally, we condition the decoder network on the roles. The CVAE is trained to maximize the conditional log likelihood of x given r , which involves an intractable marginalization over the latent variable z .

The ELBO is defined as:

$$\mathcal{L}_{\text{CVAE}} = \mathbb{E}_{q_{\phi}(z|r,x)} \left[\log p_{\theta}(x|z,r) \right] - \text{KL}q_{\phi}(z|x,r)||p(z|r)$$

Training We consider LSTM-based VAE and Transformer-based VAE (Optimus (Li et al., 2020))

as baselines. The training process follows the variational autoencoding methodology (Kingma and Welling, 2014). First, tokenization is performed in the sentences and the roles. The Encoder network involves feeding both first into embedding layers, then into LSTM / Transformer layers. Subsequently, two vectors μ and σ are sampled with two linear layers, and the vector z is computed with the re-parameterization trick. Finally, the decoder network is built with the LSTM / Transformer layers and another embedding layer, which return the same dimension that was given as input.

3 Evaluation framework

We first present the evaluation framework that for measuring disentanglement, then describe and justify the generative factor setup used in the experiments.

3.1 DSR as generative factors

While early approaches for disentanglement in NLP have been proposed in the context of in style transfer applications (John et al., 2019; Cheng et al., 2020) and are assessed purely in terms of style transfer accuracy, evaluating the intrinsic properties of the latent encodings is fundamental for disentanglement, as mentioned in several machine learning approaches (Higgins et al., 2017; Kim and Mnih, 2018). Recently, Zhang et al. (2021) proposed a framework for computing several popular quantitative disentanglement metrics such as (Higgins et al., 2017; Kim and Mnih, 2018) testing it on synthetic datasets. The limitation in (Zhang et al., 2021) is that it works only with synthetic datasets.

In this work, we propose a method where semantic role labels, such as the ones provided in (Silva et al., 2018), are used as generative factors for evaluating the degree of disentanglement in the encodings. The framework, illustrated in Figure 3, considers multiple generative factors, where each factor is composed by a number of semantic roles (for example the factor "location" includes, origin-location, and event-location). In this way, the dataset can be seen as the result of a sampling

of multiple generative factors, which is the same principle used when creating synthetic datasets for disentanglement. Once the generative factors are defined, the framework is enabled to compute a number of quantitative metrics for disentanglement, following the work from Zhang et al. (2021).

Supertype SUPERTYPE	Quality DIFFERENTIA-QUALITY	GF1: Semantics
Location EVENT-LOCATION ORIGIN-LOCATION	Modifier QUALITY-MODIFIER EVENT-TIME	
Statement PURPOSE ASSOCIATED-FACT	Accessory ACCESSORY-DETERMINER ACCESSORY-QUALITY	
Event DIFFERENTIA-EVENT		
Supertype SUPERTYPE	Main DIFFERENTIA-QUALITY DIFFERENTIA-EVENT	GF2: Syntax
Modifier Event EVENT-LOCATION EVENT-TIME	Modifier Quality QUALITY-MODIFIER PURPOSE ASSOCIATED-FACT	
Accessory ACCESSORY-DETERMINER ACCESSORY-QUALITY		
Quality DIFFERENTIA-QUALITY QUALITY-MODIFIER ACCESSORY-QUALITY	Event DIFFERENTIA-EVENT EVENT-TIME	GF3: Semantics
Location EVENT-LOCATION ORIGIN-LOCATION	Statement PURPOSE ASSOCIATED-FACT ACCESSORY-DETERMINER	
Main DIFFERENTIA-QUALITY DIFFERENTIA-EVENT	Modifier Event EVENT-LOCATION EVENT-TIME	GF4: Syntax
Modifier Quality QUALITY-MODIFIER PURPOSE ASSOCIATED-FACT	Accessory ACCESSORY-DETERMINER ACCESSORY-QUALITY	

Figure 3: Generative factors for definitions.

3.2 Semantics and Syntax groups of DSR

In order to categorize the definition semantic roles (DSR), we consider their structural and semantic dimensions in terms of their contribution to either the meaning (e.g., quality, location) or the structure (e.g., main terms, modifiers) of the definition sentence. We first create two DSR groups with semantic and two based on syntax, to evaluate which one would better facilitate disentanglement. For both syntax and semantic, we then create one group with "supertype" DSR and one without it, in order to understand the impact of the supertype DSR. The importance of "supertype" is due to its contribution to both abstraction groups and its predominant presence on the datasets analyzed ($\geq 97\%$).

Group 1: Semantics with Supertype Sets the factors in terms of their meaning, essentially abstracting categories of the DSRs, including the SUPERTYPE DSR as a single factor. Qualification,

location, modification, declaration (statement) and supplementation (accessory) are semantic roles of a given term to its definition, which are described by the DSRs.

Group 2: Syntax with Supertype Sets the factors in terms of their structural role in the definition sentence, including the SUPERTYPE DSR as a single factor. The ORIGIN-LOCATION DSR is omitted due to its syntactic overlap with EVENT-LOCATION and its low frequency in the datasets.

Group 3: Semantics without Supertype Similar to group 1, but excluding the SUPERTYPE DSR, and repositioning the factor from *modifier* and *accessory* for higher abstraction. Relations of modification and supplementation (present in group 1) are suppressed to focus on lexical semantics, moving label ACCESSORY-DETERMINER to the declaratory group, EVENT-TIME to the event group and all quality related labels to the qualification group.

Group 4: Syntax without Supertype Similar to group 2, but excluding the SUPERTYPE DSR. Further abstractions are not conducted, as the definition roles already offer a stable structure for sentence construction.

4 Related work

Disentangled VAEs in language Early approaches in text disentanglement use VAEs with multiple adversarial losses for style transfer (Hu et al., 2017; John et al., 2019). More recently, Cheng et al. (2020) propose a style transfer method which minimizing the mutual information between the latent and the observed variable, while Colombo et al. (2021) propose an upper bound of mutual information for fair text classification. Disentanglement of syntactic and semantic information on sentences is explored by Chen et al. (2019), using multiple losses for word ordering and paraphrasing, and by Bao et al. (2019) with linearized constituency tree losses. Finally, Dupont (2018) work on discrete factors for image models and the improvements in Mercatali and Freitas (2021) proposed method for NLP lead to this work, where we move from the latter's implicit language features and LSTM-based architecture to explicit automatic annotations and a state-of-the-art Transformer-based architecture. We focus our efforts into the representation of definitions, and propose to promote disentanglement by using biases provided as semantic roles, designing two VAE models to inject structural semantic information into the representation. As an alternative

architecture for generative modeling, Generative Adversarial Network (GAN) was not employed for this problem due to the non-contrastive nature of the input data (trying to leverage informed structural knowledge) and the emphasis on disentanglement as a mechanism to understand separability and control.

Disentanglement Evaluation Vishnubhotla et al. (2021) evaluate disentanglement in synthetic text on various NLP tasks such as classification, retrieval and style transfer. Zhang et al. (2021) evaluate disentanglement of various VAE models on synthetic datasets where generative factors are known. Differently from these methods, we propose a new framework to evaluate non-synthetic natural language, where semantic role labels are used as generative factors. We model linguistic features of natural language definitions, with the goal of exploring the semantic properties that are encapsulated in it.

Definition models Early approaches in definition encoding include (Hill et al., 2016), which propose the first neural embedding model for dictionaries, and (Bahdanau et al., 2017), which present an RNN-based encoder decoder architecture for textual entailment and reading comprehension. More recently, methods based on Autoencoders (Bosc and Vincent, 2018) and transformers (Tsukagoshi et al., 2021) have been proposed. Various approaches for the task of generating a definition from a word (Definition Modeling) have been proposed, including RNN-based methods (Noraset et al., 2017), soft attention mechanisms (Gadetsky et al., 2018), and span-based encoding schemes (Bevilacqua et al., 2020). The semantic aspect of natural language definitions are explored in (Silva et al., 2016, 2018), where the concept of definition semantic roles is proposed.

5 Empirical analysis

In this section, we firstly describe the empirical setup for experiments, secondly, we provide qualitative evaluation and thirdly, we measure various quantitative metrics. Finally, we demonstrate the capacity of the proposed models in the downstream task of definition modeling.

5.1 Experimental setup

Datasets Definition sentences and their respective semantic role structures are sourced from three different datasets by (Silva et al., 2016) with the characteristics described in Table 1. All datasets

Dataset	Num sents.	Avg. length	Version
Wordnet	93,699	9	WordNet 3.0
Wiktionary	464,243	8	Dec, 2016
Wikipedia	1,500,323	12	Dec, 2016

Table 1: Statistics from definition datasets.

are automatically annotated with DSR tags for each token, using the method proposed by (Silva et al., 2016). The datasets differ not only in sentence length and size, but also in textual style: while WordNet and Wiktionary sentences tend to be formatted as dictionary definitions, Wikipedia sentences are lengthier and less adherent to a typical definition structure. For brevity, hyperparameter choices and implementation details are covered in sections C and D of the supplementary material.

5.2 Qualitative Evaluation

We analyse the representations of the trained models in terms of their disentanglement and composition, by applying three different techniques 1) traversals of the latent space, 2) latent space arithmetic, 3) encoding interpolation.

Latent space traversals Traversal evaluation is a standard procedure with image disentanglement (Higgins et al., 2017; Kim and Mnih, 2018). The traversal of a latent factor is obtained as the decoding of the vectors corresponding to the latent variables, where the evaluated factor is changed within a fixed interval, while all others are kept fixed. If the representation is disentangled, when a latent factor is traversed, the decoded sentences should only change with respect to that factor. This means that after training the model we are able to probe the representation for each latent variable. In the experiment, the traversal is set up from a starting point given by a “seed” sentence. As illustrated in Table 2 we observed that the latent variables typically track a single abstract definition role (e.g., supertype, quality, purpose), and change the meaning of the original term according to an abstract interpretation axis (e.g, flying \rightarrow *movement*, art \rightarrow *doutrine/teachings*). This means a certain degree of control can be applied to the generation of both the sentence structure and semantics.

Latent space arithmetic In this experiment, the latent vectors for two sentences are added, subtracted or averaged, and then the resulting vectors are traversed. The sentence pairs are

a flying creature a flying animal a flying insect a robot a monster a creature a walking demon a flying creature a moving animal	a martial art developed in Israel an ancient Buddhist dagger used to stab others an ancient martial art practiced in Japan a Roman soldier's movement a military dress worn by monks a knight's ceremonial hat a religious rite in which communion is offered a literary rite in Bible study a medicine school
--	--

Table 2: Traversals showing **changed** and **held** semantic factors in Wiktionary definitions (Optimus-based model).

ADD	a flying machine a flying creature a flying dinosaur a flying robot a flying object	AVG	to make four copies of to make five copies of to make one copy of to make two copies of to make 3 copies of
SUB	a female monarch a monarch the subnormal condition in females originating from... the normal female pregnancy associated with some the female given name in the Japanese game...		

Table 3: Traversals showing **changed** and **held** semantic factors after latent vector arithmetic in Wiktionary definitions (Optimus-based model).

different by a single term, so that we can observe the latent variables affected by the change, and how they are affected. As illustrated in Table 3, these operations tend to produce vectors that, when traversed, generate sentences corresponding to the features manipulated by the operation (e.g., removing the *monarch* supertype, leaving the *female* quality).

Interpolation In this experiment, we analyse the capability of the models built with the proposed approach to provide a smooth transition between latent space representations of sentences (Bowman et al., 2016). In practice, the interpolation mechanism takes two sentences x_1 and x_2 , and uses their posterior mean as the latent features z_1 and z_2 , respectively. It interpolates a path $z_t = z_1 \cdot (1 - t) + z_2 \cdot t$ with t increased from 0 to 1 by a step size of 0.1. This is a deterministic process, and no search is performed. As a result, 9 sentences are generated on each interpolation step. In Table 4 we provide qualitative results with latent space interpolation on Wiktionary. We can observe the transition happening for each concept: *migratory* $\rightarrow \emptyset \rightarrow$ *microscopic*, *aquatic* \rightarrow *aquatic + terrestrial* \rightarrow *terrestrial*, *bird* \rightarrow *mammal* \rightarrow *organism* \rightarrow *invertebrate*. This type of localised semantic control provided by the operations of traversal and interpolation over intensional-level (definitional)

DSR Optimus-based	a migratory aquatic bird found in the temperate regions of the northern hemisphere 1 a migratory bird of the eastern Mediterranean 2 a marine gastropod of the subfamily 3 a terrestrial aquatic mammal of the family 4 a terrestrial aquatic mammal of the suborder 5 a terrestrial invertebrate 6 a microscopic organism or invertebrate a microscopic terrestrial animal or protozoan an automobile 1 a motorcycle a bicycle
-------------------	---

Table 4: Interpolation examples in Wiktionary (Optimus-based model). Only unique sentences are shown.

sentences can potentially support quasi-symbolic operations over the latent space. Such effects could not be observed within the baselines.

Based on those three experiments, the composition of such latent space could be conceptualised as in the projection illustrated in Figure 4.

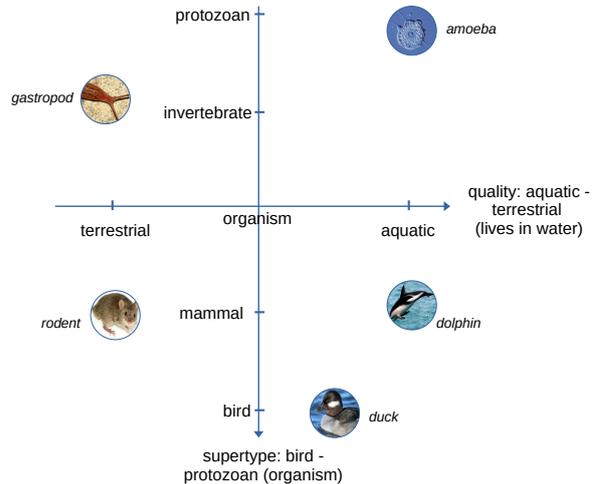


Figure 4: Conceptualisation of a two-dimension cut of the latent space, applied to the first example in Table 4.

UMAP plot UMAP (Uniform Manifold Approximation and Projection) (McInnes et al., 2018) is a popular method for non-linear dimensionality reduction, that allows the visualization of complex high-dimensional feature spaces, such as the representation space produced by a VAE. Figure 5 presents a 2D plot of UMAP transformations for both baselines under three training frameworks, from which the clustering of DSR patterns can be observed. While the supervision with DSR labels promotes clustering of the patterns around the center of the plot, cVAE compacts the cluster on the edges, allowing better separation. In the Optimus-based model, for example, the *SUPER* (green) cluster has a tendency to move

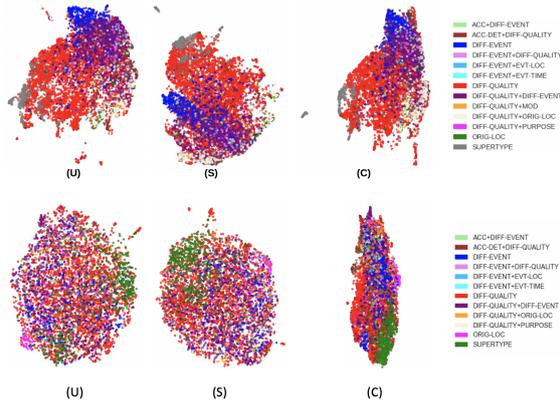


Figure 5: UMAP plot of latent representations from Un-supervised VAE (U), DSR supervision (S) and Conditional VAE (C) (Top: LSTM, Bottom: Optimus-based).

towards the edge of plot from left (U) to right (C). t-SNE transformations are also performed and the plots are presented in the supplemental material (Appendix E).

5.3 Quantitative Evaluation

In this experiment we probe the representation learned by the proposed VAE models using eight popular quantitative metrics for disentanglement, namely: z-diff (Higgins et al., 2017), z-min-var (Kim and Mnih, 2018), Mutual Information Gap (MIG) (Chen et al., 2018), Modularity & Explicitness (Ridgeway and Mozer, 2018), and from (Eastwood and Williams, 2018)(disentanglement, completeness, informativeness). Further details about the metrics are provided in Appendix B. It is relevant to mention that there are considerations regarding inconsistency on classification dependent probes (e.g., z-min-var, modularity), which are not discussed here due to space and scope considerations (we refer to Carbonneau et al. (2022)). Therefore, we decided to include all current metrics that could be applied in this scenario, and the results presented next should be interpreted considering these limitations.

Experimental Setup We evaluate VAE (U), DSR VAE (S) and CVAE (C) on Wordnet (WN), Wiktionary (WT) and Wikipedia (WP) datasets. Evaluation is performed under the framework explained in Section 3. Each combination of VAE architecture, generative factor grouping and representation size was trained and quantitatively tested, by calculating the previously mentioned disentanglement metrics. For computing the metrics we follow the

experiments of Zhang et al. (2021).

Analysis The results presented in Tables 2, 4, and 5 show that, specially when using the Optimus-based model:

LSTM												
D	z-diff			z-min-var ↓			MIG			Modularity		
	U	S	C	U	S	C	U	S	C	U	S	C
WN	.700	.691	.770	.482	.503	.532	.067	.057	.059	.793	.804	.765
WT	.597	.619	.635	.400	.385	.430	.112	.095	.065	.535	.424	.629
WP	.575	.630	.647	.398	.386	.420	.046	.041	.037	.771	.745	.757
D	Explicitness			Disentanglement			Completeness			Informativeness ↓		
U	S	C	U	S	C	U	S	C	U	S	C	
WN	.519	.532	.527	.022	.021	.031	.013	.013	.017	.364	.361	.399
WT	.584	.593	.616	.014	.011	.013	.013	.013	.011	.377	.373	.385
WP	.545	.557	.600	.007	.007	.005	.007	.007	.004	.375	.373	.374

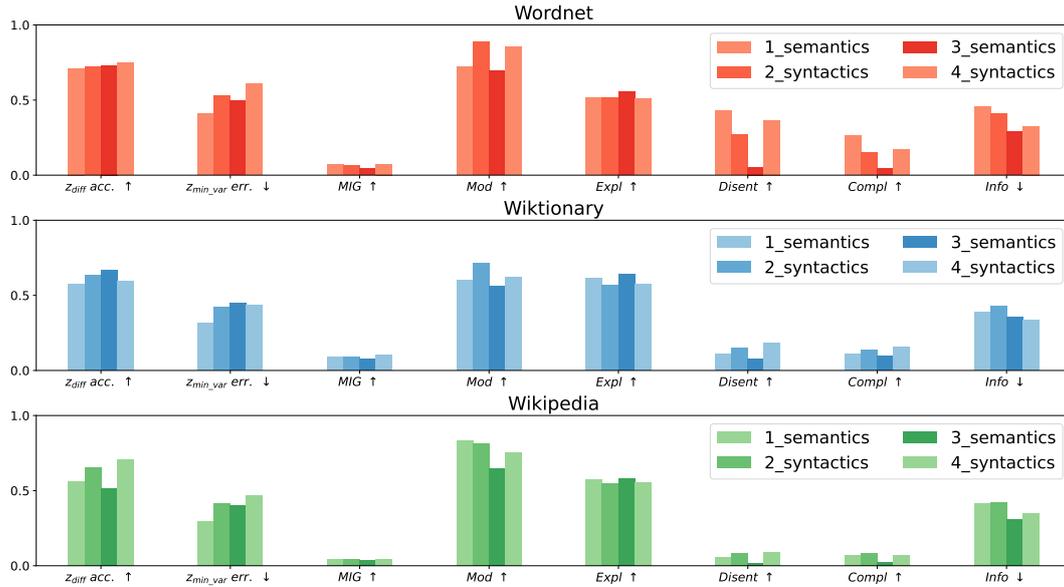
Optimus-based												
D	z-diff			z-min-var ↓			MIG			Modularity		
	U	S	C	U	S	C	U	S	C	U	S	C
WN	.645	.673	.669	.483	.509	.517	.023	.012	.006	.724	.766	.750
WT	.516	.532	.589	.458	.441	.480	.016	.013	.043	.827	.813	.809
WP	.513	.544	.641	.471	.486	.552	.010	.011	.033	.956	.942	.943
D	Explicitness			Disentanglement			Completeness			Informativeness ↓		
U	S	C	U	S	C	U	S	C	U	S	C	
WN	.501	.500	.501	.058	.040	.049	.039	.027	.032	.398	.377	.398
WT	.559	.547	.573	.013	.026	.028	.009	.018	.019	.333	.316	.305
WP	.548	.532	.594	.024	.054	.060	.016	.034	.038	.288	.282	.280

Table 5: Quantitative disentanglement metrics (Top: LSTM, Bottom: Optimus-based).

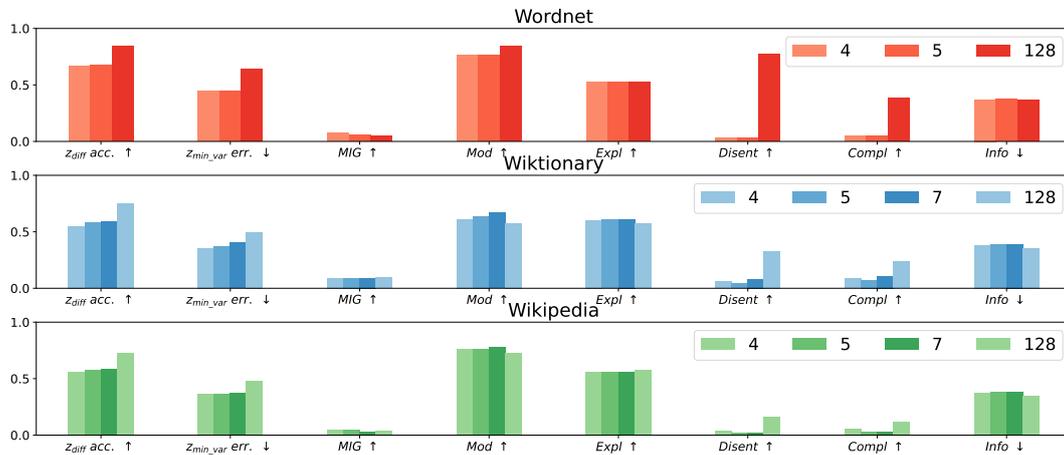
For the Wiktionary and Wikipedia datasets, the application of DSR categories as biases results in a measurable improvement in disentanglement (RQ1). This is evidenced by the proposed model outperforming the unsupervised baseline in six of the eight disentanglement metrics tested, by a margin of at least 2.5%, 81% in average.

The use of DSRs as generative factors produces meaningful disentangled representations (RQ2). The traversal results indicate the tendency of associating certain role abstractions to latent space dimensions, e.g., supertype, statement (purpose, among others). The interpolation results indicate the capture of semantic bridging across definitions, e.g., teaching → loading (process). The UMAP visualisation indicates slightly better factor separation and smoother transitions for the conditional model.

More specifically, in LSTM, z-diff presents the highest and most consistent improvement, specially with the CVAE, indicating higher interpretability when inferring single generative factors from the representations. Explicitness results are also consistent, indicating higher coverage of each factor. Improvements on Modularity, Disentanglement Score, Completeness and Informativeness are less consistent, indicating that the factors share substantial information between them. On the other hand, z-min-var, MIG counter the trend of improvement,



(a) By generative factor



(b) By number of latents

Figure 6: Metrics mean grouped.

Word	Definition Model	Unsupervised LSTM	Supervised LSTM
repuise	the act of making a gun	the act of moving forward	act in a hostile state
colonise	make a new or vital part	the state of being in a particular place	settle or cause to be easily removed
involve	make a specific purpose	make a specific effect	a specific act of making something
mitochondrion	a cell that is used to treat the blood	a substance that is used to treat a body reaction	a cell that is a source of an organic process
heat	a change in the surface of a liquid	a sudden increase in the flow of heat	a sudden increase in the temperature

Table 6: Definition generation examples for the Wordnet dataset.

due to the fact that they are designed to strongly penalize non-alignment of single pairs $\langle \text{factor} \leftrightarrow \text{latent dimension} \rangle$ (e.g., linear combinations). As a result, they penalize the existence of dependency and hierarchy relations which is present in most DSR categories, e.g., DIFFERENTIA-EVENT \rightarrow EVENT-TIME. As for the Optimus-based model, there are similar tendencies on WT and WP corpus. The conditional framework always performs better under 6 of 8 metrics, except z-min-var and

modularity. This result indicates that our conditional framework can improve the disentanglement performance of Optimus.

We also analyse how semantic groupings affect disentanglement in Figure 6b (RQ3). This is done only for the LSTM-based VAE, as the Transformer-based one was set to the optimal configuration in Li et al. (2020). Overall, we notice that syntax based groups have higher scores, indicating that it is easier to disentangle syntactic phrase components. For

Modularity the result is the opposite, indicating that semantic groupings promote higher independence between factors. Following (Zhang et al., 2021), the values in Table 5 for the metrics Completeness and Disentanglement score are multiplied by 10, in order to facilitate the visualization.

Finally, we find that a low number of latent dimensions leads to smaller degree of disentanglement. The experiments with 4,5,7 and 128 latents are reported in Figure 6a.

5.4 Definition Generation

In this experiment, we assess the proposed VAE models in the task of "Definition Modeling" (Noraset et al., 2017), where the goal is to generate a natural language definition given the word to be defined (definiendum).

Experimental setup During training, we adopt the "seed" setup (Noraset et al., 2017), which involves providing the definiendum concatenated with the definition tokens as input for the model. At generation time, the model takes as input only the word which needs to be defined, and leverages a trained model for computing the definition latent encoding. Such encoding is then fed into a softmax function and subsequently a multinomial probability distribution is sampled for decoding the latent variable into the final definition sentence.

To compare with the baseline of definition generation (Gadetsky et al., 2018), we only consider LSTM-based VAEs under the proposed unsupervised and DSR-supervised framework, both using the "seed" setup. The conditional LSTM and optimus-based models are not explored in this experiment in order to have a more fair comparison with the Definition model. We train the baseline and our models with similar setups, following (Gadetsky et al., 2018). We perform language model pretraining on the WikiText-103 dataset (Merity et al., 2016) for 1 epoch, then train on the downstream dataset for 10 epochs. Additionally, all models are initialised using Google Word2Vec pretrained vectors, following (Gadetsky et al., 2018).

Results We report the perplexity and Bleu (Papineni et al., 2002) results in Table 7. We observe that the proposed variational autoencoder models achieve an improvement on both perplexity and Bleu compared to the RNN baseline. The DSR

VAE achieves the best perplexity and Bleu on 2 out of 3 datasets while the unsupervised VAE is the best performing model in the other cases. Success of VAE models can be attributed to their disentangling properties, which promotes learning of latent spaces that are less sparse, a benefit deriving from sampling variable for re-parameterization. Improvements from the DSR VAE are marginal, but can be attributed to the additional information that is injected into its latent variables.

Data	Perplexity ↓			Bleu		
	DM	VAE	DSR	DM	VAE	DSR
WN	88.59	80.36	80.27	9.12	10.27	10.26
WT	42.51	39.09	38.64	6.70	7.53	7.59
WP	13.09	12.39	12.47	11.89	12.32	12.34

Table 7: Quantitative metrics for definition generation.

Some generation examples from the Wordnet dataset are provided in Table 6. Such examples show that the proposed VAE models are able to leverage the structural and semantic information of the learned definition roles to better approximate the defined concept. In particular, we notice some semantically strong linguistic elements in the definitions decoded with DSR supervision, for example DSR is the only model able to link the verb "repulse" with the hostile adjective, the verb colonise with the similar verb "settle", and the word "heat" with temperature. We include more generation examples of the Optimus-based model in Appendix E.

The strong performance in this definition generation task indicates that the disentangled representations have provided the VAE models with higher generalization capability, suggesting that disentangling is beneficial for diverse applications.

6 Conclusion

We propose a novel VAE-based framework for learning and evaluating disentangled representations in natural language definitions. We leverage the semantic structure present in dictionaries as inductive biases for improving disentanglement in VAEs, and as generative factors during evaluation. Our evaluation shows, both with qualitative investigations and with quantitative metrics, that the proposed framework is able to produce encodings with a higher degree of disentanglement. Finally, our models outperform existing baselines on a definition modeling application, demonstrating the generalization capabilities of disentangled representations.

Limitations

The type of structural supervision chosen for the approach here proposed is specifically fit to definition (dictionary style) sentences, in order to leverage semantic information from such structures. However, this limits the scope of comparison with other methods applied to general sentences. Additionally, the qualitative improvements we observed in terms of latent space traversals, arithmetic and interpolation do not clearly correlate with the disentanglement metrics, despite overall improvement. This raises some questions regarding the relation between explainability properties and general latent space separability.

References

- Dzmitry Bahdanau, Tom Bosc, Stanisław Jastrzebski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. 2017. Learning to compute word embeddings on the fly. *arXiv preprint arXiv:1706.00286*.
- Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019.
- Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828.
- Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or: “how we went beyond word sense inventories and learned to gloss”. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221.
- Tom Bosc and Pascal Vincent. 2018. Auto-encoding dictionary definitions into consistent word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1532.
- Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.
- Marc-André Carbonneau, Julian Zaidi, Jonathan Boilard, and Ghyslain Gagnon. 2022. Measuring disentanglement: A review of metrics. *IEEE Transactions on Neural Networks and Learning Systems*.
- Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A multi-task approach for disentangling syntax and semantics in sentence representations. In *NAACL*.
- Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. 2018. Isolating sources of disentanglement in vaes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2615–2625.
- Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving disentangled text representation learning with information-theoretic guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7530–7541.
- Pierre Colombo, Pablo Piantanida, and Chloé Clavel. 2021. A novel estimator of mutual information for learning to disentangle textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 6539–6550.
- Emilien Dupont. 2018. Learning disentangled joint continuous and discrete representations. *Advances in Neural Information Processing Systems*, 31.
- Cian Eastwood and Christopher KI Williams. 2018. A framework for the quantitative evaluation of disentangled representations. In *6th International Conference on Learning Representations*.
- Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271.
- Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.
- Felix Hill, KyungHyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596. PMLR.
- Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434.
- Hyunjik Kim and Andriy Mnih. 2018. **Disentangling by factorising**. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2649–2658. PMLR.

- Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes.
- Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699.
- Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR.
- Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.
- Giangiacomo Mercatali and André Freitas. 2021. Disentangling generative factors in natural language with discrete variational autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3547–3556.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.
- Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Karl Ridgeway and Michael C Mozer. 2018. Learning deep disentangled embeddings with the f-statistic loss. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 185–194.
- Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi Jaakkola. 2020. Educating text autoencoders: Latent representation guidance via denoising. In *International Conference on Machine Learning*, pages 8719–8729. PMLR.
- Vivian Silva, Siegfried Handschuh, and André Freitas. 2016. Categorization of semantic roles for dictionary definitions. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*, pages 176–184.
- Vivian S Silva, Siegfried Handschuh, and André Freitas. 2018. Recognizing and justifying text entailment through distributional navigation on definition graphs. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2021. Defsent: Sentence embeddings using definition sentences. In *ACL/IJCNLP*.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Krishnapriya Vishnubhotla, Graeme Hirst, and Frank Rudzicz. 2021. An evaluation of disentangled representation learning for texts. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1939–1951.
- Lan Zhang, Victor Prokhorov, and Ehsan Shareghi. 2021. Unsupervised representation disentanglement of text: An evaluation on synthetic datasets. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepLANLP-2021)*.
- Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664.

A Definition Semantic Roles

The datasets used in our experiments are introduced in (Silva et al., 2018). We report in Table 9 the annotated categories.

Role	Description
Supertype	the immediate or ancestral entity’s superclass
Differentia quality	a quality that distinguishes the entity from the others under the same supertype
Differentia event	an event (action, state or process) in which the entity participates and that is mandatory to distinguish it from the others under the same supertype
Event location	the location of a differentia event
Event time	the time in which a differentia event happens
Origin location	the entity’s location of origin
Quality modifier	degree, frequency or manner modifiers that constrain a differentia quality
Purpose	the main goal of the entity’s existence or occurrence
Associated fact	a fact whose occurrence is/was linked to the entity’s existence or occurrence
Accessory determiner	a determiner expression that doesn’t constrain the supertype / differentia scope
Accessory quality	a quality that is not essential to characterize the entity
Role particle	a particle, such as a phrasal verb complement, non-contiguous to the other role components

Table 8: Semantic Role Labels for dictionary definitions.

B Disentanglement Metrics

1. z_{diff} accuracy (Higgins et al., 2017): The accuracy of a predictor for $p(y|z_{diff}^b)$, where z_{diff}^b is the absolute linear difference between the inferred latent representations for a batch B of latent vectors, written as a percentage value. Higher values imply better disentanglement.
2. z_{min_var} error (Kim and Mnih, 2018): For a chosen factor k , data is generated with this factor fixed but all other factors varying randomly; their representations are obtained, with each dimension normalised by its empirical standard deviation over the full data (or a large enough random subset); the empirical variance is taken for each dimension of these normalised representations. Then the index of the dimension with the lowest variance and the target index k provide one training input/output example for the classifier. Thus, if the representation is perfectly disentangled,

the empirical variance in the dimension corresponding to the fixed factor will be 0. The representations are normalised so that the arg min is invariant to rescaling of the representations in each dimension. Since both inputs and outputs lie in a discrete space, the optimal classifier is the majority-vote classifier, and the metric is the error rate of the classifier. Lower values imply better disentanglement.

3. Mutual Information Gap (*MIG*) (Chen et al., 2018): The difference between the top two latent variables with the highest mutual information. Empirical mutual information between a latent representation z_j and a ground truth factor v_k , is estimated using the joint distribution defined by $q(z_j, v_k) = \sum_{n=1}^N p(v_k)p(n|v_k)q(z_j|n)$. A higher mutual information implies that z_j contains a more information about v_k , and the mutual information is maximal if there exists a deterministic, invertible relationship between z_j and v_k . *MIG* values are in the interval $[0, 1]$, with higher values implying better disentanglement.
4. *Modularity* (Ridgeway and Mozer, 2018): The deviation from an ideally modular case of latent representation. If latent vector dimension i is ideally modular, it will have high mutual information with a single factor and zero mutual information with all other factors. A deviation δ_i of 0 indicates perfect modularity and 1 indicates that this dimension has equal mutual information with every factor. Thus, $1 - \delta_i$ is used as a modularity score for vector dimension i and the mean of $1 - \delta_i$ over i as the modularity score for the overall representation. Higher values imply better disentanglement.
5. *Explicitness* (Ridgeway and Mozer, 2018): Mean of the ROC area-under-the-curve (AUC_{jk}) of a one-versus-rest logistic-regression classifier that takes the latent vectors as input and has factor values as targets, over a factor index j and an index k on values of factor j . Represents the coverage of the representation, in other words, how well each factor is represented. Higher values imply better disentanglement.
6. *Disentanglement Score* (Eastwood and

Williams, 2018): The degree to which a representation factorises or disentangles the underlying factors of variation, with each variable (or dimension) capturing at most one generative factor. It is computed as a weighted average of a disentanglement score $D_i = (1 - H_K(P_{i.}))$ for each latent dimension variable c_i , on the relevance of each c_i , where $H_K(P_{i.})$ denotes the entropy and P_{ij} denotes the 'probability' of c_i being important for predicting z_j . If c_i is important for predicting a single generative factor, the score will be 1. If c_i is equally important for predicting all generative factors, the score will be 0. Higher values imply better disentanglement.

7. *Completeness Score* (Eastwood and Williams, 2018): The degree to which each underlying factor is captured by a single latent dimension variable. For a given z_j it is given by $C_j = (1 - H_D(\tilde{P}.j))$, where $H_D(\tilde{P}.j) = -\sum_{d=0}^{D-1} \tilde{P}_{dj} \log_D \tilde{P}_{dj}$ denotes the entropy of the $\tilde{P}.j$ distribution. If a single latent dimension variable contributes to z_j 's prediction, the score will be 1 (complete). If all code variables contribute equally to z_j 's prediction, the score will be 0 (maximally over-complete). Higher values imply better disentanglement.
8. *Informativeness Score* (Eastwood and Williams, 2018): The amount of information that a representation captures about the underlying factors of variation. Given a latent representation c , It is quantified for each generative factor z_j by the prediction error $E(z_j, \hat{z}_j)$ (averaged over the dataset), where E is an appropriate error function and $\hat{z}_j = f_j(c)$. Lower values imply better disentanglement.

C Hyperparameter choices

Experiments are conducted to cover a set of 3 hyperparameters: First, the VAE architecture used: 1) Unsupervised VAE 2) Supervised with SRL 3) CVAE with SRL. Second, the generative factor grouping, which includes: 1) Semantic w/ supertype 2) Syntactic w/ supertype 3) Semantic w/o supertype 4) Syntactic w/o supertype. Third, the dimensionality of VAE latent representation (z): 4, 5, 7, 128.

The choice of architecture allows evaluation of the impact of DSR label conditioning in two distinct ways: as part of the autoencoding objective function, and as a conditional variable of the decoder, addressing our research questions **RQ1** and **RQ2**. The choice of generative factor grouping can indicate the best ways to organize the factors, addressing **RQ3**.

The dimensionality of the representation is set to match the number of generative factors, in an attempt to force disentanglement by alignment of each dimension to a single factor. The dimension sizes are then defined to be 4 (alignment with groupings 3 and 4), 5 (alignment with grouping 2) or 7 (alignment with grouping 1). However, different levels of disentanglement can be achieved with mismatching dimensions and factors. So all possible combinations of factors and representation sizes are tested and a size of 128 is included to evaluate the impact of a higher number of parameters in each grouping.

D Implementation Details

As for LSTM-based VAE, hyperparameters are chosen with the following values, based on a previous experiment from (Shen et al., 2020). (1) Number of hidden layers: 1, (2) Dimension of the hidden layer: 512, (3) VAE $\lambda_{KL} = 0.1$, (4) Epochs=20, (5) Batch size=32 for Wikipedia, 64 for the rest. Dropout (20%) is done for both encoder and decoder inputs. To provide the inputs and outputs for the VAEs, the definition sentences are tokenized into sub-words with a *Byte Pair Encoding* (BPE) scheme, and converted into token embeddings with the T5 transformer model (Raffel et al., 2020), with an embedding size of 512. With respect to Optimus, we use memory setup to inject latent representation into the decoder. The encoder and decoder are pretrained BERT with bert-base-cased version and GPT2, respectively. Some additional values of hyperparameters are: (1) Epochs=10, (2) Batch size=32. (3) latent size=32. In the supervised framework, a new embedding layer is considered to learn the representations of semantic roles. In the conditional framework, we add semantic roles into the vocabulary of pretrained BERT encoder.

E Further Experimental Results

t-SNE plot Alternative dimensionality reduction method (t-distributed Stochastic Neighbor Embedding) (Van der Maaten and Hinton, 2008), used to

visualise the clustering of DSR patterns, as seen in Figure 7.

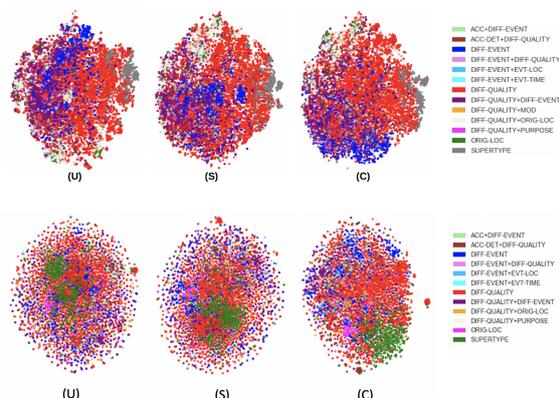


Figure 7: t-SNE plot of latent representation generated from LSTM and Optimus-based models under Unsupervised VAE (U), DSR supervision (S) and Conditional VAE (C) (Top: LSTM, Bottom: Optimus-based).

Optimus-based model definition generation

Table 9 lists the generated definitions from the Unsupervised Optimus-based model on Wordnet. The perplexity is 35.46 that is much lower than 80.27 from LSTM.

Word	Generated Definition
Fox	a member of the Mayflower
Untermeyer	United States writer of short stories
organise	make logical or comprehensible
dishrag	remove the fur from
altocumulus cloud	a clear blue sky
shuffle	move quickly on or move quickly forward
sharpen	make sharp or sharper
semantic error	discrimination that invalidates an earlier characteristic
railway station	station where planes take off and land or take off
Antonio Pignatelli	Italian cardinal and theologian
union	a cooperative level of play in league with other players
love knot	a knot of contrasting color or yarn used for tying a wedding band
commodity brokerage	a place where stockbrokers sell their stock

Table 9: Generation definitions from the Optimus-based model.

Distinguishability Calibration to In-Context Learning

Hongjing Li^{1*}, Hanqi Yan^{1*}, Yanran Li, Li Qian², Yulan He^{1,3,4}, and Lin Gui³

¹Department of Computer Science, University of Warwick, UK

²Xiaomi AI Lab, China

³Department of informatics, King’s College London, UK

⁴The Alan Turing Institute, UK

{Hongjing.Li,Hanqi.Yan}@warwick.ac.uk, yanranli.summer@gmail.com,
qianli@xiaomi.com, {yulan.he,lin.1.gui}@kcl.ac.uk

Abstract

Recent years have witnessed increasing interests in prompt-based learning in which models can be trained on only a few annotated instances, making them suitable in low-resource settings. When using prompt-based learning for text classification, the goal is to use a pre-trained language model (PLM) to predict a missing token in a pre-defined template given an input text, which can be mapped to a class label. However, PLMs built on the transformer architecture tend to generate similar output embeddings, making it difficult to discriminate between different class labels. The problem is further exacerbated when dealing with classification tasks involving many fine-grained class labels. In this work, we alleviate this *information diffusion* issue, i.e., different tokens share a large proportion of similar information after going through stacked multiple self-attention layers in a transformer, by proposing a calibration method built on feature transformations through rotation and scaling to map a PLM-encoded embedding into a new metric space to guarantee the distinguishability of the resulting embeddings. Furthermore, we take the advantage of hyperbolic embeddings to capture the hierarchical relations among fine-grained class-associated token embedding by a coarse-to-fine metric learning strategy to enhance the distinguishability of the learned output embeddings. Extensive experiments on the three datasets under various settings demonstrate the effectiveness of our approach. ¹

1 Introduction

Large pre-trained language models (PLMs) (Devlin et al., 2019; Lan et al., 2020; Liu et al., 2019) have been achieved state-of-the-art performance in many Natural Language Processing (NLP) downstream tasks. More recently, the PLMs with prompt learning demonstrate surprising capabilities in numerous

*Equal contribution.

¹Our code can be found at <https://github.com/donttal/TARA>

tasks both in NLP and computer vision, even outperforming their fine-tuned counterparts (Brown et al., 2020; Liu et al., 2021; Lester et al., 2021; Zhou et al., 2022b; Gao et al., 2021a).

Train#1:	Gotta protect'em! It was [MASK].
Train#2:	That's why it's only 20\$. It was [MASK].
Test:	On a boat trip to Denmark. It was [MASK].

Table 1: The prompt templates for emotion classification. The samples are from *GoEmotion* (Demszky et al., 2020) dataset.

In an emotion classification task shown in Table 1, an input sentence X , followed by a prompt, “*It was [MASK]*”, is fed to a PLM to predict the missing token at the position of $[MASK]$. The predicted word can be used to identify the emotion label of the input sentence. Such few-shot learning generates a probability distribution over the $[MASK]$ conditioning on the given prompt/context, which is considered as in-context learning of language models.

However, as in-context learning does not require updating PLM parameters, there arises the problem of distribution mismatch between the data used for LM pre-training and the test samples used in in-context learning, which hinders the full exploitation of the knowledge encoded in PLMs (Xie et al., 2022; Zhao et al., 2021; Ge et al., 2022; Shin et al., 2022). To alleviate the context shift, existing methods rely on prior knowledge to increase the overlapping between the two distributions. For example, *PTR* (Han et al., 2021) appends domain-agnostic tokens to prompts to discriminate the domains, such as “*sports*”, “*politics*”. Another line of studies designs sophisticated handcrafted verbalizers to map the test samples onto the label word space derived from PLMs (Schick and Schütze, 2021; Gao et al., 2021b). Although the gradient-optimized verbalizers (Hu et al., 2022) are proposed to ease the human effort and can be adapted to different downstream tasks via training, it is still consid-

ered inferior to the manual verbalizers, especially in both the few-shot and zero-shot settings where training data are scarce.

In this paper, we first show that PLMs have an inherent *information diffusion* issue in their generated output token embeddings, which share a large proportion of similar information after going through a stack of transformer layers (Gao et al., 2019; Yan et al., 2022). Such token embeddings occupy a narrow cone, leading to largely overlapped output distributions when applied to in-context learning. Next, we elaborate that the overlapped output distributions would violate the distinguishability condition (Xie et al., 2022) under in-context learning. To this end, we propose to flatten the singular value distributions of the output embeddings generated from PLMs to shape the space spanned by the singular values to a desirable manifold. On the one hand, we apply an orthogonal and a scaling constraints to the weight matrix applied to the output embeddings, which can avoid exploding and vanishing values in the feature matrix (Saxe et al., 2014), leading to better discriminative features when trained with limited labelled data. On the other hand, we leverage hyperbolic embeddings to capture the hierarchical relations among fine-grained class labels of training examples to further enhance the distinguishability of output embeddings.

Our proposed framework has been implemented on top of existing prompt-based few-shot learning methods and it demonstrates an average 5.86% performance improvement of F1-measure on three classification tasks under 100-shot learning. We also verify that the improvement stems from a more balanced singular value distribution for the output features and the learnt hierarchical feature space.

In summary, our contributions include:

- We propose a transformation-based constraint to output embeddings by rotation and ratio balancing which is able to guarantee the distinguishability of learned embeddings.
- The proposed hyperbolic embedding-based metric learning strategy not only improves the performance of prompt learning but also measures the relation between different categories.
- The experimental results outperform many strong baselines and the visualisation illustrates that the proposed method is able to project the embedding to a less overlapping

distribution and improve the interpretability and distinguishability of output. Specifically, across three evaluated datasets, our method surpasses the state-of-the-art by 9.60%, 5.11% and 2.87%, respectively, in the 100-shot setting.

2 Related works

Information diffusion in PLMs. In a typical L -layer transformer-based PLM, assuming the prompt is a concatenation of a few training examples and a test input X_{test} , consisting of m tokens in total, the goal of in-context learning is to predict the output distribution over the masked token at the t -th position, $[MASK]$. It is formally defined by the following equation:

$$p(\mathcal{O}_t | X_{\text{test}}) = \mathbb{E}_{h \sim p_{\text{prompt}}(h | X_{\text{test}})} [p(\mathcal{O}_t | X_{\text{test}}, h, \theta)],$$

where h denotes the last-layer hidden state corresponding to the token of X_{test} , θ is the parameters in prompt-based learning.

Although we have limited knowledge of the output distribution $p(\mathcal{O}_t | X_{\text{test}})$ over token $[MASK]$, many existing studies analyzed the geometry properties of the last layer feature h^L , and examined its effects in downstream tasks (Goyal et al., 2020; Zhou and Srikumar, 2022). Due to the softmax bottleneck (Yang et al., 2018) and the likelihood loss in language generation tasks (Gao et al., 2019), the output feature distribution in PLMs tends to be anisotropic and rank-deficient, which limits the expressiveness of the generated representations. Goyal et al. (2020) discussed the information diffusion issue among tokens within a sentence that feeding the tokens in different positions for classification only resulted in a 1.2% variance in classification accuracy. Gao et al. (2019) explored the information diffusion among different sentences via singular value decomposing and they found that the singular value distributions are skewed especially in deeper PLM layers, i.e., larger singular values become more predominant compared to the smaller ones.

Context shift in in-context learning. Many researchers studied the distribution shift (aka. domain shift) between the pretraining corpora and test samples and proposed solutions to decrease the performance variance in prompt-based few-shot learning (Xie et al., 2022; Zhao et al., 2021; Hu et al., 2022; Zhou et al., 2022b; Shin et al., 2022).

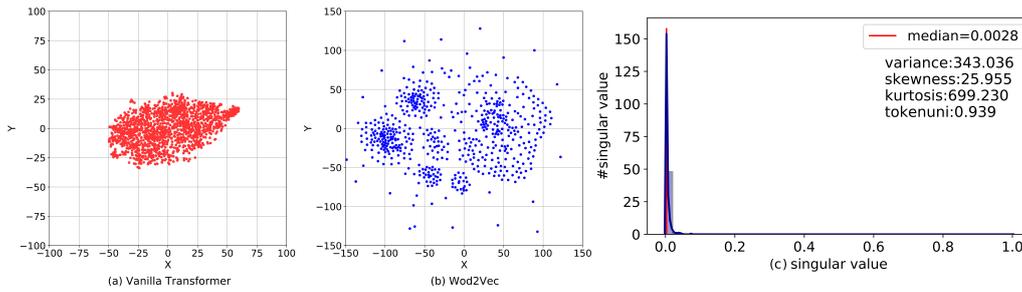


Figure 1: **(a)**: The mapping results of 1,500 *[MASK]* tokens randomly sampled from the *GoEmotions* dataset. Each red dot is the output representations derived from prompt-based learning for the *[MASK]* token of an input example, which will be used to predict the masked token in the corresponding position. **(b)**: Each blue dot is the static word representation of the corresponding predicted token with the largest probability on *[MASK]* for one of the 1,500 samples in (a) from the *GoEmotions* dataset. **(c)**: Singular value distribution (after normalisation) of the output representations of the randomly selected 1,500 *[MASK]*s. It is clear that the representations are dominated by very few singular values.

On the one hand, some in-context learning methods incorporated domain-specific words or learnable tokens in the prompt to discriminate different context. Ben-David et al. (2022) proposed to first generate the name of the domain and then generate domain-related features (DRFs) conditioned on the domain in a supervised manner. Both the generated domain name and DRFs were used as the prompt fed to the model. On the other hand, the sophisticated verbalizers contributed to minimising the distance between the two distributions (Schick et al., 2020; Schick and Schütze, 2021; Gao et al., 2021b; Hu et al., 2022). To broaden the coverage of single-choice verbalizer, *Knowledge Prompt Tuning (KPT)* (Hu et al., 2022) used the knowledge graph to extract more topic-related words as label words and then refine the label word candidates. To incorporate prior knowledge to calibrate the context shift, Xie et al. (2022) simplified a language model as the Hidden Markov Model, where the observed tokens are sampled from a family of concepts and proposed the *distinguishability condition* to measure context shift as the Kullback–Leibler (KL) divergence.

3 Contextual Calibration for Output Distribution

Many existing methods calibrate the probabilities of the generated tokens in a language model in order to improve the generation quality. In prompt-based learning, we want to find out if the output distribution $p(O_t|X_{\text{test}})$ or the output feature $h^{[\text{mask}]}$, which is a part of the hidden representation from the last layer of a PLM, h^ℓ , suffers from the *information diffusion* issue and occupies a narrow

cone. We take RoBERTa-based prompt learning as an example and derive the value of $h^{[\text{mask}]}$ from 1,500 randomly selected test samples from an emotion classification dataset, *GoEmotions* (Demszky et al., 2020), and visualise the results in a 2D plane in Figure 1(a). For comparison, we select the predicted token with the largest probability on each *[MASK]* and map their corresponding vectors from Word2Vec (Mikolov et al., 2013) to a 2D plane in 1(b). It is clear that the word embeddings learned from Word2Vec has a more uniform distribution around the origin. In contrast, the representations derived by RoBERTa degenerate into a narrow cone, which implies limited expressiveness. Inspired by the approach proposed in (Yan et al., 2022), we display the singular value distribution of $h^{[\text{mask}]}$ and calculate the distribution statistics, i.e., the matrix moment and the average cosine similarity between every *[MASK]* pair in Figure 1(c). From the empirical results, we can see that the value of the hidden representation for *[MASK]* in different samples share much similar information with the token uniformity value (Yan et al., 2022) (*tokenuni* in Figure 1(c)) of 0.939. This shows that most $h^{[\text{mask}]}$ concentrates at very few singular values, which implies a severe information diffusion issue.

3.1 Uniform Ratio-based Distinguishability

Although many calibration methods have been proposed, few of them focuses on explicitly addressing the information diffusion issue in the prompt-based learning framework. One main challenge in this task is that the unlabelled data used in language model pre-training is significantly larger than the labelled samples used for prompt tuning. Hence,

the optimised distribution in prompt-based few-shot learning can be very different from the true distribution. To avoid inheriting the *information issue* caused in the pre-training phase, we propose a calibration method to reduce the skewness of the output token distributions, such that the output representations are evenly distributed in the embedding space. The idea is to rotate the original embedding space to an isotropic metric space by an inner product-based operator on a learnable basis. For each dimension of the basis, we use the inner product to measure its relevance with a given input. The dimension-dependent relevance scores are sent to a Multi-layer Perceptron (MLP) decoder to generate the calibrated output embedding for final prediction.

The framework of the proposed calibration method is shown in Figure 2. In practice, due to the small size of training samples in prompt learning, the relevance scores might be dominated by very few dimensions. Therefore, inspired by Zhou et al. (2022a), who proposed a ratio estimator to balance the distribution from different label categories, we design a scaling matrix in our isotropic distribution scenario. That is, for both labelled and unlabelled data, the multi-class ratio between different dimensions should be similar.

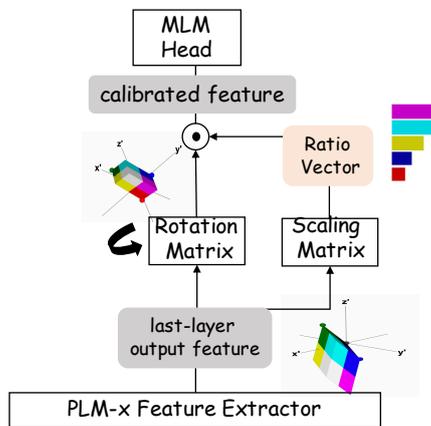


Figure 2: Our proposed calibration method is applied to the output embeddings from the last layer of a PLM. After being transformed with a rotation matrix through a Multi-layer Perceptron (MLP), the resulting output feature is assumed to have a more balanced singular value distribution in different basis directions. Moreover, as the vector norm on each projected direction would change in the new base, we derive a ratio vector to balance the distribution along the rotated directions.

Concretely, assuming we have N labelled data $\{y_j, x_j\}_{j=1}^N$ and M unlabelled data from pre-training $\{x_j\}_{j=N+1}^{N+M}$, where x_j is the input sample,

y_j is the true label, and $M \gg N$. To simplify the notation, in the rest of this paper, we use x_j to represent the feature of the last embedding layer and h_j to represent the output of our calibrated feature. Then, for the representation of a masked token, x_j , we assume there are K isotropic directions in the metric space and the corresponding inner product based relevance score is:

$$\mathcal{H}_k(x_j) = \sigma(\langle x_j, W_k \rangle), \quad (1 \leq k \leq K), \quad (1)$$

where $\sigma(\cdot)$ is the softmax activation function. Here, we can define a **rotation matrix** based on W_k since Eq. (1) projects an input embedding onto a new metric space by rotation. To guarantee the orthogonality of the basis in the new metric space, we use the following regulariser during training:

$$\mathcal{L}_{orth} = \left\| W^\top W - \mathbf{I} \right\|_2^2, \quad (2)$$

where W is the stacking of $\{W_k\}_{k=1}^K$. Correspondingly, for each dimension k , we can define a ratio score which aims to better separate them to avoid the skewed distribution by minimising the following loss:

$$\mathcal{L}_t = \frac{1}{N+M} \sum_{k=1}^K \sum_{j=1}^{N+M} \left\| \mathcal{R}_k(x_j) - \frac{1}{K} \right\|^2, \quad (3)$$

where $\mathcal{R}_k(x_j)$ is an MLP-based estimator with a softmax activation:

$$\mathcal{R}_k(x_j) = \sigma(S_k \cdot x_j + \beta). \quad (4)$$

By minimising \mathcal{L}_t , even if one input sample x_j is similar to a basis vector along a popular dimension k , there will still be a probability to assign it a low ratio score $\mathcal{R}_k(x_j)$ if there are other samples which are more closer to the basis vector in dimension k . In this way, we can balance the distribution after rotation. We define the stacking of S_k as a **scaling matrix** which aims to distribute x_j uniformly into K clusters in the metric space.²

However, it is difficult to optimise the loss defined in Eq. (3) since the size of the unlabelled data for pre-training is much larger than the labelled data and the unlabelled data is usually unseen to the downstream tasks. We instead define an alternative optimisation objective. First, according to Eq. (3), we need to ensure that for any two dimensions k and t , we have $\frac{1}{N+M} e^{S_k \cdot x_j} = \frac{1}{N+M} e^{S_t \cdot x_j}$.

²We measured the impact of different weight initialisations on S_k in Appendix A.2.

By the Jensen’s inequality, we have the following lower bound: $e^{\frac{1}{N+M}S_k \cdot x_j} \leq \frac{1}{N+M}e^{S_k \cdot x_j}$, in which we can achieve the lower bound for any two independent dimensions by taking $\frac{1}{N+M}S_k \cdot x_j = \frac{1}{N+M}S_t \cdot x_j$. It means that for any two dimensions, the sum of their ratio scores should be similar. As such, Eq. (3) can be approximated by:

$$\mathcal{L}_t \sim \sum_{k=1}^K (||S_k||^2 - 1)^2. \quad (5)$$

Accordingly, we can define the distinguishability loss in a more general form by both the relevance score and the ratio score without the need of sampling from unlabelled data:

$$\mathcal{L}_{dis} = \mathcal{L}_{orth} + \mathcal{L}_t. \quad (6)$$

From our findings in Section 3, much information encoded by the output representations generated by the last layer of a PLM only occupies a space spanned by very few singular value directions. This leads to the information diffusion issue. Therefore, our solution here is to re-project the output features into a new hyperplane, in which the information is more evenly distributed in different directions, and at the same time we can derive a ratio vector by aggregating the rotated components.

3.2 Supervised Prompt Learning

By our proposed distinguishability loss-based learning in Section 3.1, an input embedding has been separated into vectors along K independent dimensions. Then, for the labelled data $\{x_j\}_{j=1}^N$, we propose to use k independent decoders to produce the final prediction. The decoding result is based on the relevance score and ratio score on each independent dimension:

$$\mathbf{h}_j = \sum_{i=k}^K \text{Decoder}_k(\mathcal{H}_k(x_j) \cdot \mathcal{R}_k(x_j)), \quad (7)$$

where the Decoder_k is a decoder for the k -th dimension. Then the representation of \mathbf{h}_j can be used in the verbalizer $p_{\text{verbalizer}}(\hat{\mathcal{O}}|\mathbf{h}_j)$, where $\hat{\mathcal{O}}$ is the predicted masked token. Finally, the cross-entropy loss H is defined by the predicted $\hat{\mathcal{O}}$ and the true label y_j :

$$\mathcal{L}_{cls}(x_j) = H(y_j, p_{\text{verbalizer}}(\hat{\mathcal{O}}|\mathbf{h}_j)). \quad (8)$$

By combining the uniform ratio-based distinguishability loss of \mathcal{L}_{dis} and the prompt-based classification loss \mathcal{L}_{cls} , we propose our first model, named as **Transformation based Adaptation for**

Ratio bAlanced (TARA) prompt learning, which aims to minimise $\mathcal{L}_{\text{TARA}} = \mathcal{L}_{cls}(x_j) + \mathcal{L}_{dis}$. Note that $\mathcal{L}_{cls}(x_j)$ is the default loss term in all the baselines and our proposed methods.

3.3 Dimension Rotation by Hyperbolic Embeddings

In Section 3.1, we project the input mask embedding into a K dimensional metric space to avoid skewed distributions. However, we ignore the potential class relations between the dimensions. For example, in emotion classification, both the emotions of ‘gratitude’ and ‘approval’ belong to the *coarse* positive class, but they are associated with different *fine-grained* labels in the GoEmotions dataset (Demszky et al., 2020). Hence, in this section, we only consider those positive pairs under the same coarse category to achieve a better class disambiguation by a proxy based metric learning (Movshovitz-Attias et al., 2017; Yang et al., 2022), which uses an anchor vector to represent a category for metric loss optimisation and capture the hierarchical structure between coarse- and fine-grained labels in the hyperbolic space.

Strategies for Constructing Sample Pairs. Inspired by the hierarchical structure of coarse-to-fine emotion categories, we assume that a fine-grained emotion should be close to the coarse-grained emotion it belongs to. To implement this idea, we construct sample-anchor pairs (\mathbf{h}_j, z_i^+) for training, where \mathbf{h}_j is the representation for prompt prediction and $z_i^+ \in \mathbb{R}^d$ is a learnable anchor representation for each coarse class.

Metric Learning in a Hyperbolic Space. To maximise the similarity in sample-anchor positive pairs, where the sample and the anchor share the same coarse-grained label, while minimising the similarity in negative pairs, we adopt the following metric learning objective:

$$\mathcal{L}_{metric}(\mathbf{h}_j) = -\log \frac{e^{-d(\mathbf{h}_j, z_{p_j}^+)}}{\sum_{i=1}^C e^{-d(\mathbf{h}_j, z_i^+)}} \quad (9)$$

where $\{(\mathbf{h}_j, z_i^+)\}_{i=1}^C$ represents a set of sample-anchor pairs that we constructed for each sample i , C denotes the number of anchors, $z_{p_j}^+$ is the representation of positive pairing anchor of j -th sample, and $d(\cdot)$ is the hyperbolic distance metric defined by the Poincaré ball model of the hyperbolic space (Nickel and Kiela, 2017). In a n -dimensional hyperbolic space, all points will fall into a unit open

interval: $\mathcal{I}^n = \{x \in \mathbf{R}^n \mid \|x\| < 1\}$, where $\|\cdot\|$ denotes the Euclidean norm. The distance $d(\cdot)$ between two points $u, v \in \mathcal{I}^n$ can be formulated as:

$$d(u, v) = \operatorname{arcosh}\left(1 + 2 \frac{\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)}\right). \quad (10)$$

The motivation of using $\mathcal{L}_{metric}(\mathbf{h}_j)$ is to push similar categories together in the metric space. Hence, we can obtain our final learning objective by adding the loss of tree-structured metric learning $\mathcal{L}_{metric}(\mathbf{h}_j)$ to **TARA** as:

$$\mathcal{L}_{final} = \mathcal{L}_{cls}(x_j) + \mathcal{L}_{metric}(\mathbf{h}_j) + \mathcal{L}_{dis}. \quad (11)$$

For a comparison, we propose a variant called **TML** by keeping the learning architecture and simply adding $\mathcal{L}_{metric}(\mathbf{h}_j)$ to the classification loss of $\mathcal{L}_{cls}(x_j)$, but without the ratio balancing term of \mathcal{L}_{dis} , that is, $\mathcal{L}_{TML} = \mathcal{L}_{cls}(x_j) + \mathcal{L}_{metric}(\mathbf{h}_j)$.

4 Experiments

Datasets We evaluate our proposed approach on three multi-class text classification datasets, the *Emotion*³ (Saravia et al., 2018) dataset, an academic paper classification dataset, *WOS* (Kowsari et al., 2017), and a fine-grained emotion classification dataset, *GoEmotions*⁴ (Demszky et al., 2020). All of these datasets have hierarchical label structures. The datasets statistics are shown in Table 2.

Name	#Classes	#Train	#Dev	#Test
<i>Emotion</i>	6	16,000	2,000	2,000
<i>WOS</i>	11	5,736	1,147	1,147
<i>GoEmotions</i>	28	23,485	2,956	2,984

Table 2: Dataset statistics.

For all datasets, we remove punctuation, digits, and special characters that do not have specific semantic meanings. For the *Emotion* dataset which consists of tweet, we also remove user mentions.

Baselines We implement our proposed framework on top of the commonly used prompt-based learning methods and compare it with existing approaches including those which can be used for learning more discriminative representations:

- *Prompt-baselines*. Three commonly used prompt-based methods are selected including

Soft Prompts (Brown et al., 2020), *Prompt-Tuning* (Lester et al., 2021) and *PTR* (Han et al., 2021). The best-performing method is used as the default prompt-based training method for the following three comparison models, and denoted as *Prompt-baseline*.⁵

- *KPT* (Hu et al., 2022). It uses a knowledge graph to incorporate topic-related label words to increase the coverage of the verbaliser.
- *Context Calibration* (Zhao et al., 2021). This method calibrates the output representations by one-layer linear transformation, whose weight matrix is optimised to be diagonal.
- *Proxy-NCA* (Movshovitz-Attias et al., 2017). It creates a proxy for each class and uses the Neighbourhood Component Analysis (NCA) loss to pull samples closer to their assigned proxies while pushing negative samples away.

Prompt Settings As the performance of prompt-based methods heavily relies on prompt templates and verbalisers, we use the same template and verbaliser for all models for fair comparison. The prompt templates are shown in Table 3. The original class labels are used as label words in the verbaliser as in (Schick and Schütze, 2021).

Datasets	Prompt template
<i>Emotion</i>	<X>It’s [MASK].
<i>WOS</i>	<X>The domain of the text is [MASK].
<i>GoEmotions</i>	<X>The emotional aspect of this text is [MASK].

Table 3: Prompt templates used in three datasets.

4.1 Few-shot Learning on Three Datasets

We randomly select k different training samples for few-shot learning and show the results across the three datasets in Table 4.

For metric-learning, *Proxy-NCA* with contrastive loss leads to performance degradation compared to the *Prompt-baseline*, with more significant performance drops on the *GoEmotions* dataset, which has the largest label categories. By contrast, **TML** gives better results over the *Prompt-baseline* and *Proxy-NCA*, showing its efficiency in encoding hierarchical relations between the coarse- and fine-grained labels. It can be further demonstrated in Figure 3, which shows the similarity matrix

⁵The detailed performance of these three prompt-based training methods is shown in Table A3. We use *PTR* for *GoEmotion*, and use *P-tuning* for the other two datasets.

³<https://huggingface.co/datasets/emotion>

⁴https://huggingface.co/datasets/go_emotions

K -shot	Emotion				WOS				GoEmotions			
	5	10	50	100	5	10	50	100	5	10	50	100
Prompt-baseline	0.336	0.363	0.431	0.625	0.236	0.252	0.359	0.435	0.161	0.173	0.281	0.310
Proxy-NCA	0.333	0.384	0.412	0.637	0.214	0.246	0.295	0.383	0.149	0.166	0.208	0.233
Context Calibration	0.337	0.352	0.531	0.706	0.212	0.361	0.687	0.707	0.164	0.224	0.355	0.420
TML	0.339	0.387	0.466	0.699	0.229	0.277	0.372	0.529	0.158	0.227	0.309	0.355
TARA	0.348	0.401	0.697	0.783	0.245	0.418	0.705	0.728	0.172	0.249	0.364	0.442
Ours full model	0.355	0.441	0.713	0.802	0.278	0.439	0.719	0.757	0.206	0.255	0.384	0.448

Table 4: Weighted F1 scores on three Datasets. The proposed **TML** is better than *Proxy-NCA*. Our full method (**TML+TARA**) achieves the best performance among all the settings.

(28×28) of the 28 fine-grained emotion labels from 3 high-level categories, i.e., “*anger*”, “*joy*” and “*sad*”. The results of *Proxy-NCA* in (c) are similar to the *Prompt-baseline* as shown in (b). Our proposed **TML** in (d) can capture the hierarchical relations among the 28 labels, where the correlations among labels belonging to the same high-level emotion category are similar. By comparison, we replace the hyperbolic distance in **TML** with the Euclidean distance and show the results in (c). It can be observed that the resulting label embeddings fail to exhibit different patterns within and across different high-level emotion categories.

For the calibration methods, *Context Calibration* and **TARA** are overall better than the *Prompt-baseline*. This shows that the simple linear transformation of the output representations can greatly improve the performance of prompt-based learning. The superior performance of **TARA** over *Context Calibration* demonstrates the benefit of using our proposed rotation and scaling transformations. Combining **TML** with **TARA**, our full model achieves the best performance and the improvements are more predominant when K is larger. In the 100-shot setting, our method surpasses the state-of-the-art method, *Context Calibration*, by 9.6% on Emotion, 5.1% on WOS, and 2.9% on GoEmotions, respectively, verifying its superiority in the few-shot text classification task.

4.2 Information Diffusion Alleviation

In addition to the classification results, we also examine the characteristics of the generated output representations to check whether the information diffusion issue has been addressed. Figure 4 shows the PCA projection results of all the [MASK] representations, i.e., $h^{[MASK]}$ in the test samples, which are colour-coded according to their assigned class labels by the model. It is clear that our method can generate more widely distributed [MASK] representations, therefore better reducing the overlaps

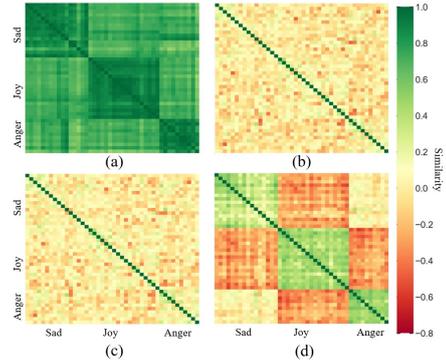


Figure 3: Heatmap for the pair-wise cosine similarity of fine-grained classes on *GoEmotion*. (a) Label representations from PLM without fine-tuning. (b) Fine-tuned label representations by classification module only. (c) Fine-tuned label representations with proposed constraint but based on Euclidean distance, i.e., *Proxy-NCA*. (d) Fine-tuned label representations by **TML**.

of the features from different class labels. For example, in the *Emotion* dataset, the output features from the baseline model mostly reside along the horizontal direction, while ours distribute more evenly across different directions.⁶

We also calculate the summary statistics of the singular value distribution of the output features, as well as the average similarity between every two [MASK] pairs. The results are shown in Table 5. The average cosine similarity (*CosSim*) between every token pair is used as a proxy measure of the degree of information diffusion. We can observe that the *CosSim* value calculated on the output representations generated by our model is significantly lower compared to the other baselines. We also observe an increase in the median and the decrease in variance of the singular value distribution from our model outputs in comparison to the prompt learning baseline. The results show that our model produces the output representations which have a more balanced singular value distribution. The

⁶The T-SNE results and singular value distribution of the output representations in *Emotion* and *GoEmotions* are shown in Figure A1 and Figure A2.

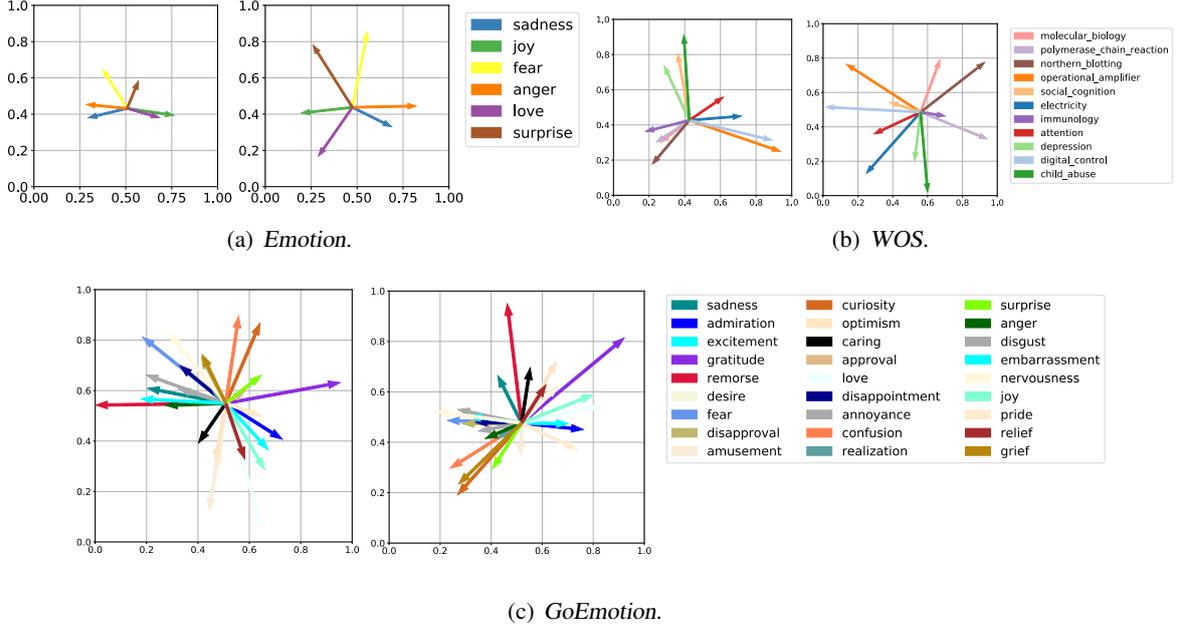


Figure 4: The PCA projection of the output representations belonging to different classes. In each sub-figure, the **left figure is the prompt-baseline**, while **the right figure is our method**. It is clear that our method distributes the output representations more evenly in the embedding space, while the output representations from the baseline appear to be more concentrated.

	Median	Variance	Skewness	CosSim
Emotion-prompt	0.0028	371.9	24.57	0.898
Emotion-Ours	0.0145	5.211	8.960	0.256
WOS-prompt	0.0036	235.8	22.06	0.817
WOS-Ours	0.0117	5.681	9.088	0.191
GoEmotions-prompt	0.0028	822.1	24.64	0.899
GoEmotions-Ours	0.0268	11.20	7.728	0.243

Table 5: The statistics of the singular value distribution of the output features, as well as the average cosine similarity of all $[MASK]$ token pairs.

	Ours	w/o \mathcal{L}_{orth}	w/o \mathcal{L}_t	w/o l_2	w/o all
Emotion	0.802	0.725	0.719	0.723	0.724
WOS	0.757	0.728	0.687	0.741	0.699
GoEmotions	0.448	0.422	0.415	0.427	0.412

Table 6: Ablation study of various loss terms in the learning objective for the distinguishability loss.

smaller skewness value further verifies that our proposed model can generate isotropic representations where the embedding dimensions are uncorrelated.

4.3 Ablation Study

To study the effect of different components of our proposed distinguishability loss, i.e., the constraints applied to the transformation operation for ratio balancing, we remove one of them and compare the performance changes in Table 6. Here, \mathcal{L}_{orth} is applied on W in Eq.2, \mathcal{L}_t is applied on

S_k (from Eq.4 and Eq.5), and l_2 is the weight for the L_2 regularisation term on all the other learnable parameters. The \mathcal{L}_{orth} and L_2 constraints have similar effects on the overall performance, as they both act as axis transformations, while the constraint L_t applied on S_k plays a more important role, whose removal leads to a larger performance drop among all the settings. It partly demonstrates the importance of the balancing ratio vector after the rotation transformation.

5 Conclusion

In this paper, to address the information diffusion issue in prompt-based few-shot learning, we propose a calibration method based on feature transformation which first rotates output embeddings into a new metric space, and then scales the ratio of each dimension to a uniform distribution to guarantee the distinguishability of the transformed embeddings. On the other hand, we utilise hyperbolic embeddings to capture the hierarchical relations between class labels to guide the metric learning strategy to enhance the interpretability of the learned output embeddings. Extensive experiments on the three multi-class classification tasks under various settings demonstrate the effectiveness of our approach with an average 5.9% performance improvement on the F1-measure.

Limitations

In this work, we only focus on the multi-class classification task with hierarchical class labels. Future work could explore extending our idea to other tasks, such as controllable text generation, which has the similar information diffusion issue. Another potential direction in future work is to learn a prior distribution rather than simply using the uniform distribution in ratio balancing. Since the uniform distribution-based ratio balancing is a strong assumption, it might not be suitable for some tasks in real-world applications. One could use VAE or VQ-VAE to learn a distribution which could be subsequently used to regularise the optimisation of feature transformation.

Acknowledgements

This work was supported in part by the UK Engineering and Physical Sciences Research Council (EP/T017112/1, EP/V048597/1, EP/X019063/1), and the National Science Foundation (NSF) grant 1750978. Yulan He is supported by a Turing AI Fellowship funded by the UK Research and Innovation (EP/V020579/1).

References

- Eyal Ben-David, Nadav Oved, and Roi Reichart. 2022. Pada: Example-based prompt learning for on-the-fly adaptation to unseen domains. *Transactions of the Association for Computational Linguistics*, 10:414–433.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan S. Cowen, Gaurav Nemade, and Sujith Ravi. 2020. [Goemotions: A dataset of fine-grained emotions](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4040–4054. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. [Representation degeneration problem in training natural language generation models](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. 2021a. [Clip-adapter: Better vision-language models with feature adapters](#). *CoRR*, abs/2110.04544.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021b. [Making pre-trained language models better few-shot learners](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.
- Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. 2022. [Domain adaptation via prompt learning](#). *CoRR*, abs/2202.06687.
- Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan T. Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. [Power-bert: Accelerating BERT inference via progressive word-vector elimination](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699. PMLR.
- Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. [WARP: word-level adversarial reprogramming](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4921–4933. Association for Computational Linguistics.
- Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. [Ptr: Prompt tuning with rules for text classification](#). *arXiv preprint arXiv:2105.11259*.
- Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong

- Sun. 2022. [Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2225–2240. Association for Computational Linguistics.
- Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, , Matthew S Gerber, and Laura E Barnes. 2017. [Hdltext: Hierarchical deep learning for text classification](#). In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. IEEE.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. [ALBERT: A lite BERT for self-supervised learning of language representations](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *CoRR*, abs/2107.13586.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.
- Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. 2017. [No fuss distance metric learning using proxies](#). In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 360–368. IEEE Computer Society.
- Maximilian Nickel and Douwe Kiela. 2017. [Poincaré embeddings for learning hierarchical representations](#). *CoRR*, abs/1705.08039.
- Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. [CARER: Contextualized affect representations for emotion recognition](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.
- Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2014. [Exact solutions to the nonlinear dynamics of learning in deep linear neural networks](#). In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. [Automatically identifying words that can serve as labels for few-shot text classification](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021. [Exploiting cloze-questions for few-shot text classification and natural language inference](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 255–269. Association for Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021. [It’s not just size that matters: Small language models are also few-shot learners](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.
- Seongjin Shin, Sang-Woo Lee, Hwijee Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woo-Myoung Park, Jung-Woo Ha, and Nako Sung. 2022. [On the effect of pre-training corpora on in-context learning by a large-scale language model](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5168–5186. Association for Computational Linguistics.
- Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. [An explanation of in-context learning as implicit bayesian inference](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Hanqi Yan, Lin Gui, Wenjie Li, and Yulan He. 2022. [Addressing token uniformity in transformers via singular value transformation](#). In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 2181–2191. PMLR.
- Zhibo Yang, Muhammet Bastan, Xinliang Zhu, Doug Gray, and Dimitris Samaras. 2022. [Hierarchical proxy-based loss for deep metric learning](#). In *IEEE/CVF Winter Conference on Applications of*

Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022, pages 449–458. IEEE.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. [Breaking the softmax bottleneck: A high-rank RNN language model](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. [Calibrate before use: Improving few-shot performance of language models](#). In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Doudou Zhou, Molei Liu, Mengyan Li, and Tianxi Cai. 2022a. [Doubly robust augmented model accuracy transfer inference with high dimensional features](#).

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. [Learning to prompt for vision-language models](#). *International Journal of Computer Vision*, 130(9):2337–2348.

Yichu Zhou and Vivek Srikumar. 2022. [A closer look at how fine-tuning changes BERT](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1046–1061. Association for Computational Linguistics.

A Appendix

A.1 Model Selection

Following previous research (Gao et al., 2021b; Hambardzumyan et al., 2021; Lester et al., 2021), BERT (Devlin et al., 2019), Roberta and ALBERT (Lan et al., 2020) were used when using the *cloze* prompts. The *cloze* is to fill in the blanks in the prompt template by the model itself.

Model	Zero-shot		
	Accuracy	Macro-f1	Weighted-f1
BERT (fine-tuning)	0.0342	0.0196	0.0329
BERT	0.0716	0.0165	0.0384
RoBERTa	0.1094	0.0465	0.0994
ALBERT	0.0538	0.0217	0.0459

Table A1: Classification results on GoEmotion dataset of different baseline models.

To select the baseline model used as the backbones of our proposed method, we evaluate the baseline models on *GoEmotions* dataset for zero-shot learning, the results are shown in Table A1. We compare the effects of the same model using prompt learning and fine-tuning, respectively (difference in effects between BERT (Devlin et al., 2019) and BERT (fine-tuning)). After comparison, we chose RoBERTa (Liu et al., 2019) as it shows the overall best performance.

Based on the large pretrained language model backbone, we compare different prompt-based training methods and select the best as our *Prompt-Baseline*. The details are shown in Table A3.

A.2 Weight Initialisation

The optimisation of W_k and S_k can be affected by different weight initialisations. As such, we experiment with different initialisation strategies and show the results of 100-shot learning in Table A2 (We use the same initialisation for W_k and S_k). The Gaussian distribution initialisation performs the best overall. Therefore we use the Gaussian distribution initialisation in all the experiments reported in the paper.

B Visualisation results

To better compare the results of Baseline methods and ours, we visualize the output of different labels by mapping them into 2D plane via T-SNE (Figure A1). It is clear that our model separates the data points of different labels ((b) and (d)) rather than

Initialisation	Emotion	WOS	GoEmotions
Gaussian	0.802	0.757	0.448
Xavier	0.817	0.749	0.420
Eye	0.798	0.747	0.387
Orthogonal	0.801	0.757	0.431

Table A2: Initialisation of different distributions on weight matrix.

mixing them up (shown in (a) and (c)). To explore the corresponding effects of singular values distribution, we visualise the normalized singular value distribution of the output embeddings in Figure A2. We observe a more balanced distribution after applying our transformation and metric learning.

K -shot	Emotion			WOS			GoEmotions		
	Soft	P-Tuning	PTR	Soft	P-Tuning	PTR	Soft	P-Tuning	PTR
5	0.295	0.336	0.330	0.165	0.236	0.213	0.072	0.135	0.161
10	0.312	0.363	0.351	0.180	0.252	0.230	0.151	0.151	0.173
50	0.363	0.431	0.409	0.328	0.359	0.319	0.230	0.245	0.281
100	0.423	0.625	0.631	0.412	0.435	0.391	0.331	0.336	0.310

Table A3: Weighted F1 of few-shot for different prompt-based training methods.

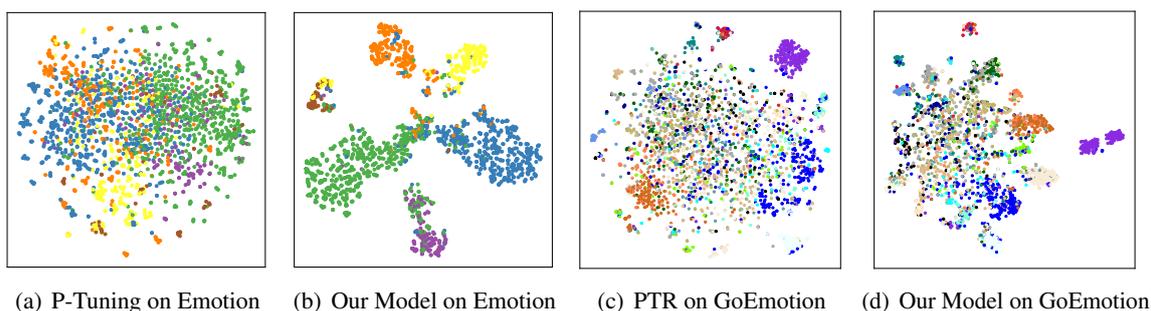


Figure A1: The T-SNE results of test samples in Emotion/GoEmotions Dataset under 100-shot.

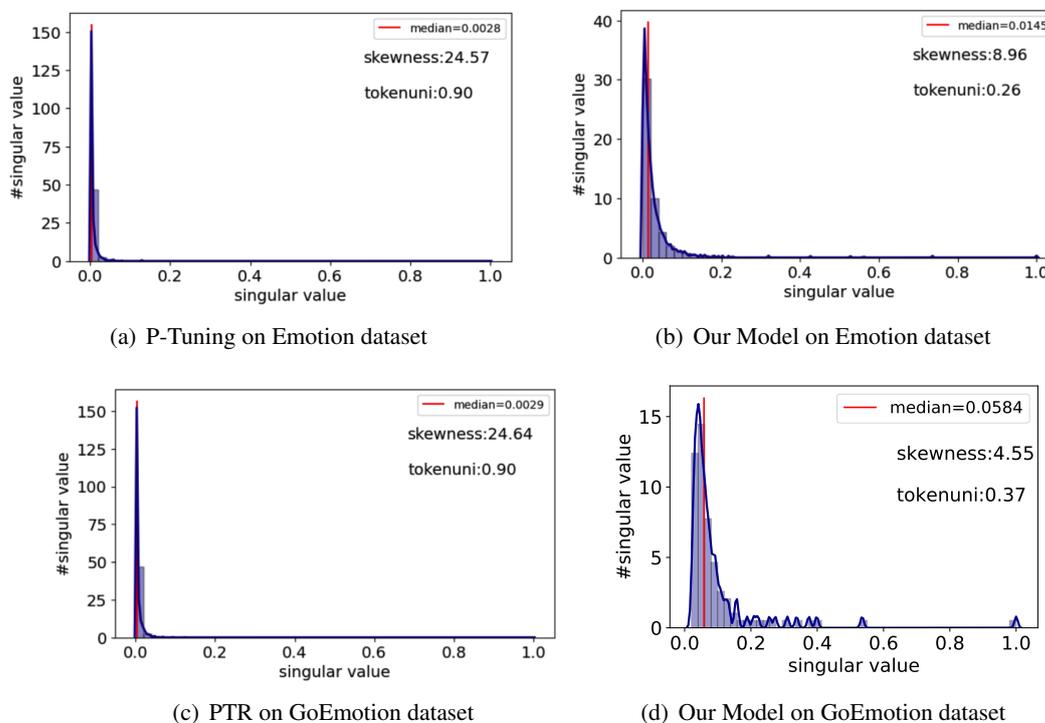


Figure A2: The singular value distribution of test samples under 100-shot. Our methods greatly balance the singular distribution, i.e., decrease the skewness, and alleviate the *information diffusion* issue, i.e., decrease the token similarity (*tokenuni*).

Investigating anatomical bias in clinical machine learning algorithms

Jannik Skyttegaard Pedersen*

The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
jasp@mimi.sdu.dk

Martin Sundahl Laursen*

The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
msla@mimi.sdu.dk

Pernille Just Vinholt

Department of Clinical Biochemistry
Odense University Hospital

Anne Bryde Alnor

Department of Clinical Biochemistry
Odense University Hospital

Thiusius Rajeeth Savarimuthu

The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

Abstract

Clinical machine learning algorithms have shown promising results and could potentially be implemented in clinical practice to provide diagnosis support and improve patient treatment. Barriers for realisation of the algorithms' full potential include bias which is systematic and unfair discrimination against certain individuals in favor of others.

The objective of this work is to measure *anatomical bias* in clinical text algorithms. We define anatomical bias as unfair algorithmic outcomes against patients with medical conditions in specific anatomical locations. We measure the degree of anatomical bias across two machine learning models and two Danish clinical text classification tasks, and find that clinical text algorithms are highly prone to anatomical bias. We argue that datasets for creating clinical text algorithms should be curated carefully to isolate the effect of anatomical location in order to avoid bias against patient subgroups.

1 Introduction

Research in clinical machine learning algorithms have shown promising results for automating clinical tasks. The algorithms could potentially be implemented in clinical practice to provide diagnosis support, improve patient treatment and provide time-savings for medical doctors (Topol, 2019; Matheny et al., 2020).

However, despite appealing research results, there are currently limited examples of algorithms being successfully deployed into clinical practice (Kelly et al., 2019). Barriers for realisation of the algorithms' full potential include bias and general-

sation issues (Char et al., 2018; Hovy and Prabhu-moye, 2021; Carrell et al., 2017).

Algorithmic bias can be defined as systematic and unfair discrimination against certain individuals or groups of individuals in favor of others (Friedman and Nissenbaum, 1996). Previous studies have raised serious concerns of algorithms that contain age, gender and racial bias (Sun et al., 2019; Davidson et al., 2019) — even for algorithms that have been taken into use (Obermeyer et al., 2019). Although machine learning algorithms are trained to be able to generalise to previously unseen data, they tend to overfit to the data they have been trained on. As a consequence of this, bias can unintendedly arise if some subgroups of the target population are not represented in the data used to train the algorithm. Moreover, if the training data itself include biases against some populations, e.g. data reflecting a negative attitude against people with disabilities (Hutchinson et al., 2020), these biases might be encoded and reinforced.

If biased algorithms are adopted, healthcare systems risk doing injustice to certain patient groups and harming patient safety (Obermeyer et al., 2019). Therefore, identifying and mitigating bias is important for successful implementation of novel clinical machine learning algorithms.

This paper investigates *anatomical bias* in clinical machine learning algorithms developed to classify and extract specific medical conditions from the narrative text of electronic health records (EHR). We define anatomical bias as unfair algorithmic outcomes against a subgroup of patients with the same medical condition, where the algorithm performs differently depending on the anatomical location of the condition. If the performance of clinical algorithms varies depending

*Equal contribution

on the anatomical location, it is reflected in some patient subgroups receiving worse treatment than others.

We hypothesised that careful dataset curation is needed to measure and mitigate anatomical bias because the text description of medical conditions in EHRs varies depending on the location, e.g. ‘epistaxis’ is a location-specific word describing nose bleedings.

Specifically, this paper investigates anatomical bias for classification of bleeding and venous thromboembolism (VTE) mentions in the narrative text of Danish EHRs. Automatic extraction of these conditions could be valuable for medical doctors in clinical practice, e.g. to guide diagnostic decision making and treatment options (Decousus et al., 2011). Previous papers (Hinz et al., 2013; Lee et al., 2017; Taggart et al., 2018; Li et al., 2019; Mitra et al., 2020, 2021; Elkin et al., 2021; Pedersen et al., 2021; Shi et al., 2021; Verma et al., 2022) have shown promising results for automatic extraction of these medical conditions but they did not investigate the performance of the algorithms across anatomical subgroups.

Our main contributions are:

- We find that clinical text algorithms are highly prone to anatomical bias.
- The performance of state-of-the-art algorithms developed to extract specific medical conditions varies significantly across anatomical locations with performance drops up to 89.1 percentage points (PP).
- We argue that datasets for creating clinical text algorithms should be curated carefully to isolate the effect of anatomical location in order to avoid bias against patient subgroups.

2 Methods

To investigate if machine learning algorithms are prone to anatomical bias, we performed two experiments. We (1) investigated the performance of a binary classifier on different anatomical subgroups of a medical condition when that subgroup was left out of the training set, and (2) measured how the performance on an anatomical subgroup varied depending on the amount of samples from that subgroup included in the training set.

Table 1: Distribution of the training, validation, and test samples for the balanced bleeding detection dataset.

Label	Location	Train	Validation	Test
Positive for bleeding	Gastrointestinal	750	250	250
	Urogenital	750	250	250
	Internal	750	250	250
	Otorhinolaryngeal	750	250	250
	Dermatological	750	250	250
	Gynecological	750	250	250
	Cerebral	750	250	250
	Ophthalmological	750	250	250
Negative for bleeding		6,000	2,000	2,000
Sum		12,000	4,000	4,000

Table 2: Distribution of the training, validation, and test samples for the balanced VTE detection dataset.

Label	Location	Train	Validation	Test
Positive for VTE	Lower extremity	1,600	200	200
	Lung	1,600	200	200
	Liver	0	0	239
	Cerebral	0	0	218
	Upper extremity	0	0	176
Negative for VTE		3,200	400	1,033
Sum		6,400	800	2,066

2.1 Datasets

We used the binary bleeding classification dataset from Pedersen et al. (2022) and present a new binary VTE classification dataset. The bleeding dataset consists of 20,000 sentences from Danish EHRs labeled as either positive or negative for bleeding mentions. The VTE classification dataset consists of 9,266 sentences from Danish EHRs labeled as either positive or negative for VTE mentions. Both datasets were constructed from Danish EHRs from Odense University Hospital and were labeled with a consensus label from three medical doctors.

In addition to the main labels of each dataset (positive and negative for bleeding or VTE), we created a subgroup label for the positive samples describing the anatomical location of either the bleeding or VTE mention. Samples that did not describe the anatomical location or described multiple locations were omitted.

For the bleeding dataset, we used the following eight anatomical locations: gastrointestinal, urogenital, internal, otorhinolaryngeal, dermatological, gynecological, cerebral, and ophthalmological.

For the VTE dataset, we used the following five anatomical locations: lower extremity, lung, liver, cerebral, and upper extremity.

The locations included for each medical condition were selected by two medical doctors.

For each dataset, we created a balanced training, validation, and test set containing an equal amount

of positive and negative samples. Moreover, for the bleeding dataset, the positive samples of the training, validation, and test sets were distributed equally between anatomical locations. For the VTE dataset, only samples from the lower extremity and lung locations were distributed equally between the train, validation, and test sets. The liver, cerebral, and upper extremity locations were only used for the test set because of a limited number of samples.

All samples were preprocessed by removing special characters, superfluous spaces, and duplicate samples. After preprocessing the samples, the bleeding and VTE datasets had an average token length of 13.3 and 13.6, respectively. The dataset distributions can be seen in [Table 1](#) and [Table 2](#).

2.2 Training set distributions

To measure performance differences for specific anatomical locations, we systematically removed all samples from a specific location, x , from the training set, creating the training set $\mathcal{T}_{\not{x}}$, trained a deep learning model on $\mathcal{T}_{\not{x}}$, and evaluated it on the test set. For example, for the bleeding dataset, we created 8 different training sets, one for each anatomical location being removed, containing 10,500 samples.

For comparison, we created a balanced training set, \mathcal{T} , which included the same amount of samples as $\mathcal{T}_{\not{x}}$, distributed equally between the positive and negative classes, and between anatomical locations.

2.3 Deep learning models

The deep learning models were a transformer-based ELECTRA model ([Clark et al., 2020](#)) and a Long Short-Term Memory (LSTM) model ([Hochreiter and Schmidhuber, 1997](#)).

The ELECTRA model was a Danish clinical ELECTRA (Clin-ELECTRA) ([Pedersen et al., 2022](#)) pretrained on the narrative text from 299,718 EHRs from Odense University Hospital. The model had ~ 13 M parameters and consisted of 12 transformer layers with 4 attention heads. We initialised Clin-ELECTRA from its pretrained checkpoint and followed the HuggingFace ([Wolf et al., 2019](#)) implementations for binary text classification.

The LSTM model had ~ 4 M parameters and consisted of a bidirectional LSTM layer with a hidden layer size of 512. The last hidden state of the LSTM was followed by a dropout layer with probability 0.2, a dense layer of size 256, a ReLU activation

function, a dropout layer of probability 0.2, and a dense classification layer. For word representation, the LSTM model used 300-dimensional FastText ([Bojanowski et al., 2017](#)) word embeddings pretrained on Danish EHRs consisting of 1.4B tokens.

2.4 Model evaluation

For each of the ELECTRA and LSTM deep learning models and training sets \mathcal{T} and $\mathcal{T}_{\not{x}}$, we:

1. Trained the deep learning model with five different learning rates and random initialisations.
2. Computed the test set accuracy of the best performing model based on the loss on the validation set.
3. Repeated step 1 and 2 five times.

We used the five accuracies to perform bootstrapping with 9,999 replicates and calculated mean accuracy, standard error (SE), and 95% confidence interval (CI) for \mathcal{T} and $\mathcal{T}_{\not{x}}$. Moreover, we computed the bootstrapped difference of means between \mathcal{T} and $\mathcal{T}_{\not{x}}$ to evaluate statistically significant differences in performance.

For both deep learning models, we used the Adam optimizer ([Kingma and Ba, 2015](#)) and searched for the best model using learning rates $7e-5$, $8e-5$, $9e-5$, and $1e-4$. Clin-ELECTRA was finetuned for a maximum of 10 epochs and the LSTM for a maximum of 30 epochs. One epoch was trained in <1 and ~ 5 seconds for the LSTM and ELECTRA model, respectively, using an NVIDIA v100 GPU.

We measured anatomical bias as the difference in sensitivity on a specific location, x , between two deep learning models trained on \mathcal{T} and $\mathcal{T}_{\not{x}}$.

3 Results

3.1 Bleeding classification

[Table 3](#) shows the binary accuracy of the bleeding classifiers on the test set for each of the training sets \mathcal{T} and $\mathcal{T}_{\not{x}}$. [Appendix A](#) shows additional metrics. With the exception of $\mathcal{T}_{\not{x}Otorhinolaryngeal}$ for the ELECTRA model, all training sets with an anatomical location removed resulted in models with significantly worse performance than when trained on \mathcal{T} .

The decreases in accuracy were caused by a significant drop in sensitivity for the anatomical locations which had been removed from the training

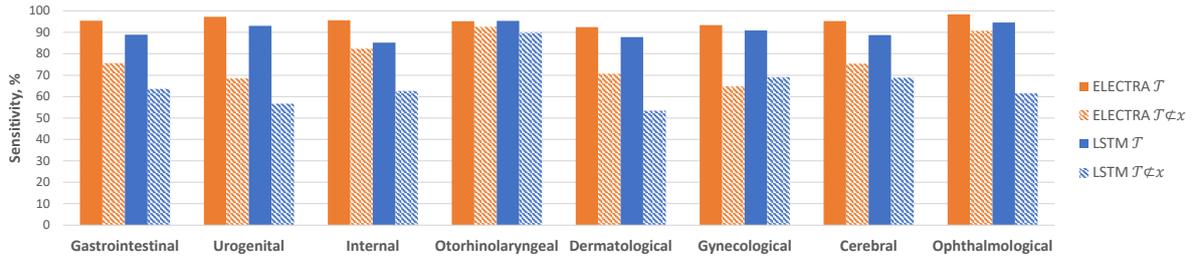


Figure 1: Sensitivity of models trained on $\mathcal{T}_{\setminus x}$ and \mathcal{T} for each anatomical location, x .

Table 3: Accuracy (%), standard error (SE), and 95% confidence interval (CI) for the bleeding classification dataset. $\mathcal{T}_{\setminus x}$ denotes the training set from which an anatomical location, x , has been removed. * denotes a significant difference at the 0.05 level between models trained on \mathcal{T} and $\mathcal{T}_{\setminus x}$.

	ELECTRA		LSTM	
	Accuracy±SE	CI	Accuracy±SE	CI
\mathcal{T}	95.6 ± 0.1	95.4 - 95.8	90.6 ± 0.1	90.4 - 90.7
$\mathcal{T}_{\setminus \text{Gastrointestinal}}$	94.4 ± 0.2*	94.0 - 94.7	88.8 ± 0.1*	88.6 - 88.9
$\mathcal{T}_{\setminus \text{Urogenital}}$	93.9 ± 0.1*	93.7 - 94.2	88.5 ± 0.1*	88.3 - 88.6
$\mathcal{T}_{\setminus \text{Internal}}$	95.0 ± 0.1*	94.8 - 95.2	88.8 ± 0.1*	88.5 - 89.0
$\mathcal{T}_{\setminus \text{Otorhinolaryngeal}}$	95.6 ± 0.1	95.4 - 95.8	90.0 ± 0.2*	89.8 - 90.4
$\mathcal{T}_{\setminus \text{Dermatological}}$	94.3 ± 0.1*	94.0 - 94.5	88.2 ± 0.1*	88.1 - 88.3
$\mathcal{T}_{\setminus \text{Gynecological}}$	93.8 ± 0.1*	93.6 - 94.0	89.1 ± 0.1*	88.9 - 89.4
$\mathcal{T}_{\setminus \text{Cerebral}}$	94.6 ± 0.2*	94.3 - 94.9	89.0 ± 0.1*	88.8 - 89.2
$\mathcal{T}_{\setminus \text{Ophthalmological}}$	95.3 ± 0.1*	95.1 - 95.4	88.2 ± 0.2*	87.8 - 88.5

data. Figure 1 shows the test set sensitivity for each anatomical location, x , for each training set \mathcal{T} and $\mathcal{T}_{\setminus x}$. The sensitivity for all anatomical locations was significantly worse when not present in the training set with performance drops up to 28.8 PP for ELECTRA and 36.3 PP for the LSTM model. On average, the sensitivity on the anatomies decreased with 17.8 PP (standard deviation ± 8.8 PP) for ELECTRA and 24.5 PP (standard deviation ± 9.4 PP) for the LSTM model.

Moreover, it is seen that even though models trained on $\mathcal{T}_{\setminus x}$ achieved high accuracies on the test set overall, the sensitivity on the anatomical location not present in the training set was low. E.g., for ELECTRA, $\mathcal{T}_{\setminus \text{Gynecological}}$ had a 93.8% accuracy on the test set, but the sensitivity for gynecological bleedings was only 64.8%. Appendix A shows the sensitivity, SE, and the differences of means for all anatomical locations and training sets.

Figure 2 shows the sensitivity on each anatomical location by the percentage of total subgroup samples in the training set. It is seen that the accuracy increased as more samples were present in the training set. For the LSTM model, the sensitivity on gastrointestinal, urogenital, cerebral, and ophthalmological bleedings was significantly worse - even when 80% of samples were present in the

Table 4: Accuracy (%), standard error (SE), and 95% confidence interval (CI) for the VTE classification dataset. $\mathcal{T}_{\setminus x}$ denotes the training set from which an anatomical location, x , has been removed. * denotes a significant difference at the 0.05 level between models trained on \mathcal{T} and $\mathcal{T}_{\setminus x}$.

	ELECTRA		LSTM	
	Accuracy±SE	CI	Accuracy±SE	CI
\mathcal{T}	84.8 ± 0.3	84.2 - 85.4	75.9 ± 0.3	75.4 - 76.4
$\mathcal{T}_{\setminus \text{Lower extremity}}$	67.6 ± 0.6*	66.5 - 68.7	71.6 ± 0.4*	70.8 - 72.6
$\mathcal{T}_{\setminus \text{Lung}}$	74.0 ± 1.1*	71.9 - 76.1	69.9 ± 0.1*	69.7 - 70.2

training set. For ELECTRA, the sensitivity on urogenital and internal bleedings was significantly worse when 80% of samples were present in the training set. Appendix A shows the accuracies and differences of means.

3.2 Venous thromboembolism classification

Table 4 shows the binary accuracy of the VTE classifiers on the test set for each of the training sets \mathcal{T} and $\mathcal{T}_{\setminus x}$. Appendix B shows additional metrics. Models trained on $\mathcal{T}_{\setminus \text{Lower extremity}}$ and $\mathcal{T}_{\setminus \text{Lung}}$ performed significantly worse than those trained on \mathcal{T} .

Similar to the bleeding classifier results, the decrease in the overall accuracy was caused by a significant drop in sensitivity on the anatomical locations which had been removed from the training data. Figure 3 shows the sensitivity for each anatomical location, x , for each training set \mathcal{T} and $\mathcal{T}_{\setminus x}$. The sensitivity on liver, cerebral, and lower extremity VTEs is only reported when not being present in the training set because of limited samples.

The sensitivity on lower extremity and lung VTEs was significantly worse when not present in the training set, e.g. the performance for the ELECTRA classifier decreased by 89.1 PP for lower extremity VTEs and 81.0 PP for lung VTEs. Appendix B shows the sensitivity, SE, and the differences of means for all anatomical locations and

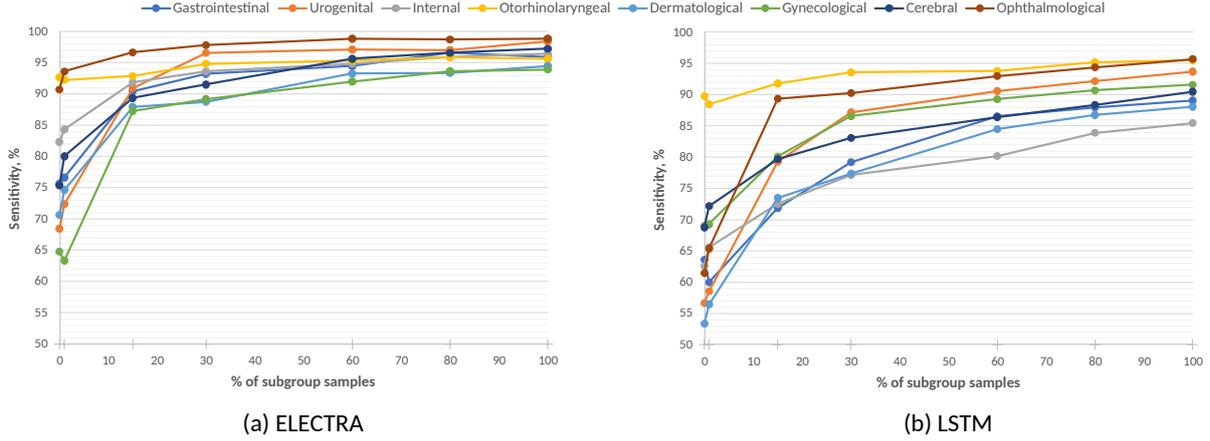


Figure 2: Test set sensitivity on the anatomical locations when removing a fraction of samples from that anatomy from the training set.

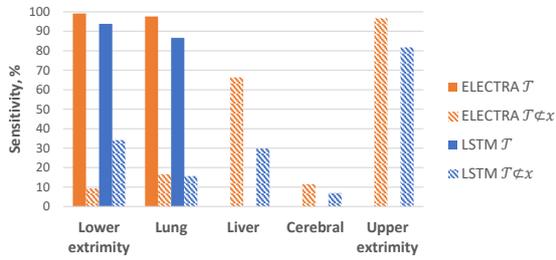


Figure 3: Sensitivity of models trained on $\mathcal{T}_{\mathcal{Q},x}$ and \mathcal{T} for each anatomical location, x . The sensitivity on liver, cerebral, and lower extremity VTEs is only reported when not being present in the training set because of limited samples.

training sets.

Figure 4 shows the sensitivity for lower extremity and lung VTEs by the percentage of total subgroup samples in the training set. Both locations performed significantly worse when 15% and 30% of samples from that location were present in the training set for the ELECTRA and LSTM classifier, respectively. Appendix B shows the accuracies and differences of means.

4 Analysis of word distributions

Medical conditions are often described using different words depending on the anatomical location for which the condition occurs. Table 5 shows the top-3 most frequent words used to describe VTE events for each anatomical location. The column *Location uniqueness* shows the fraction of times a word appears in samples from a specific anatomical

Table 5: Most frequent words used to describe VTE events for each anatomical location of the VTE classification dataset. Words are translated from Danish to English and, therefore, some cells include two words. PE = pulmonary embolism.

Word	Frequency	Location uniqueness
Location: Lower extremity		
dvt	1384	0.92
thrombus	135	0.70
blood clot	108	0.47
Location: Lung		
pulmonary embolism	1058	0.98
le (PE)	483	0.99
pulmonary embolisms	242	0.99
Location: Liver		
porta thrombosis	71	1.00
thrombosis	70	0.36
thrombus	22	0.11
Location: Cerebral		
infarct	93	0.97
sinus thrombosis	32	0.97
blood clot	26	0.11
Location: Upper extremity		
dvt	99	0.07
thrombus	28	0.14
thrombosis	14	0.07

location compared to the complete dataset:

$$Location\ uniqueness = \frac{f_x}{f_D} \quad (1)$$

where f_x is the frequency of the word in samples from anatomical location x and f_D is the total frequency of the word in the dataset, i.e. a value of 1 means that the word is unique for an anatomical location.

The top-3 words for upper extremity had a low uniqueness score (<0.15). This indicates that the vocabulary used to describe VTEs in the upper extremity was also used for other locations — e.g.

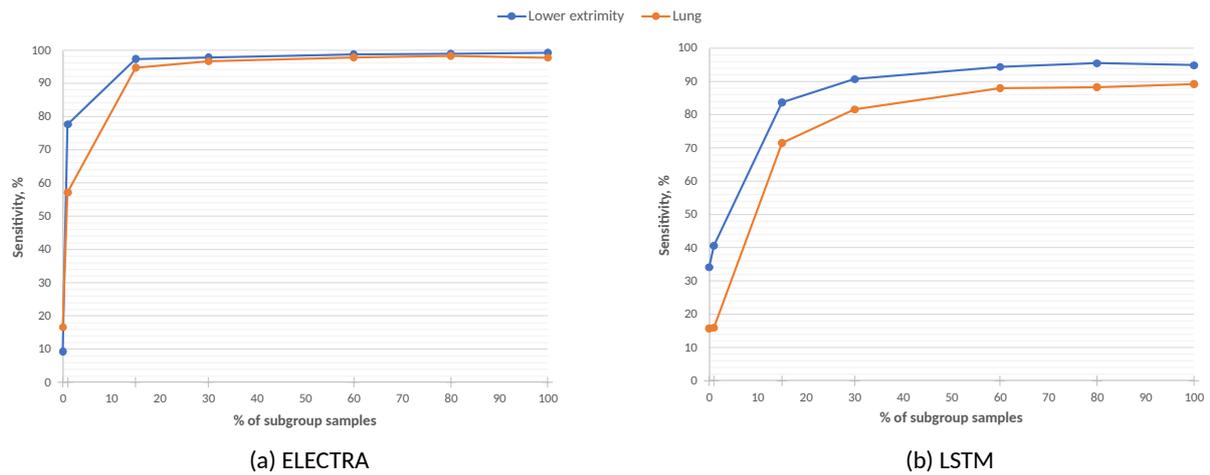


Figure 4: Test set sensitivity on the anatomical locations when removing a fraction of samples from that anatomy from the training set.

‘dvt’ (deep vein thrombosis) was the most frequent word but the uniqueness score was only 0.07. This might explain why the sensitivity for upper extremity was relatively high, as seen in Figure 3, even when samples from that location were not present in the training set. On the contrary, some of the frequent words from the lower extremity, lung, and cerebral locations were close to unique which could explain why the sensitivity of those locations were low. Appendix C shows word frequency and location uniqueness for the bleeding classification dataset which shows similar results.

5 Discussion

This paper has presented evidence that clinical natural language processing (NLP) classification algorithms are prone to anatomical bias, which is unfair algorithmic outcomes against patients with medical conditions in specific anatomical locations. We found that the performance of algorithms for both bleeding and VTE classification can vary significantly depending on the anatomical location with differences up to 36.3 PP and 89.1 PP, respectively.

Moreover, we found that small fluctuations in the training set distribution of anatomical locations can lead to significant performance drops for the under-represented anatomical locations. For the datasets presented in this study, we showed that the words used to describe medical conditions vary depending on the anatomical location. If classifiers do not learn to properly represent the full vocabulary for describing a medical condition, its performance will decrease for some anatomical locations.

We argue that datasets for clinical NLP algo-

gorithms should be created to be able to carefully measure anatomical bias, e.g. by subdividing each sample into an anatomical location. This is essential to avoid implementing clinical algorithms that might discriminate against specific subgroups of patients. For example, one of the developed VTE classifiers in this study performed with sensitivities of >96% for VTEs in the lungs and lower extremity while it performed with a sensitivity of only 11.5% for cerebral VTEs. Applying such a model in clinical practice or research would provide unfair algorithmic outcomes against patients with cerebral VTEs. We also showed that an algorithm not exposed to gynecological bleedings would perform worse on this anatomical location. This would lead to unfair algorithmic outcomes against woman with gynecological bleedings. Similarly, because alcoholics have an increased prevalence of gastrointestinal bleedings (Singal et al., 2018), this group of people would have a higher risk of unfair algorithmic outcomes if the algorithm has not been trained on such bleeding locations.

To the best of our knowledge, anatomical bias has not been investigated in previous research. However, some studies tried to automatically create datasets distributed between different patient groups by extracting data based on International Classification of Diseases 10 (ICD) codes — e.g. Pedersen et al. (2021) extract data based on different bleeding disorders. While this approach could, to some degree, mitigate the problem, studies (Valkhoff et al., 2014; Øie et al., 2018) found that ICD codes have low accuracy and, therefore, this does not ensure an evenly distributed dataset.

Moreover, in order to isolate and measure the performance on different anatomical locations, the dataset should be constructed with a known distribution of these anatomies.

Our work is closely related to the field of domain adaptation. For example, MacAvaney et al. (2017) find that an algorithm trained to extract temporal information from a specific patient population performs worse on another related patient population. Their results highlight that it is a challenging task to develop algorithms that can generalise well across domains. The main difference between our study and theirs is that the algorithms described in this paper are not developed to work on different domains. Rather, the algorithms are specifically developed to work on a specialised domain in the clinical field, e.g. bleeding detection. As our results have shown, the algorithms perform worse on some subpopulations of the population it is supposed to work on, and therefore, we describe this as a bias issue.

6 Conclusion

This paper presented evidence that clinical NLP algorithms are prone to anatomical bias. We found that the performance of clinical classification algorithms for both bleeding and VTE classification can vary significantly depending on the anatomical location of the medical condition. We argue that anatomical bias should be carefully examined when developing clinical text algorithms in order to avoid unfair algorithm performance against patient subgroups.

7 Limitations

Future work should investigate the degree of anatomical bias in other clinical areas and tasks, e.g. named entity recognition, to be able to compare the severity of the bias problem between algorithms and other clinical areas. Moreover, as the datasets used in this study are only from a single institution, the findings of the paper might not be widely representative.

The objective of this work was to stress the need for measuring anatomical bias. We leave it to future work to investigate algorithmic solutions other than dataset balancing for mitigating the problem, e.g. using techniques such as oversampling and data augmentation. Such techniques could also help mitigating anatomical bias in algorithms for which training set balancing is not sufficient.

The classification datasets and machine learning

models presented in this paper cannot be shared publicly due to privacy concerns but we advise interested researchers to contact us for sharing possibilities.

Ethics Statement

Machine learning researchers must be proactive in recognising and counteracting biases such as the one described in this paper. We hope that the findings and focus of this paper will lead other researchers to test and mitigate other kinds of algorithmic biases.

All datasets used in this research were obtained according to each dataset's respective data usage policy. The datasets were stored and processed on a secure platform¹ in compliance with GDPR regulations. According to section 14(2) of the Danish Act on Research Ethics Review of Health Research Projects², studies using retrospective data that do not involve human biological material do not require ethical approval.

References

- Piotr Bojanowski, Édouard Grave, Armand Joulin, and Tomáš Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- David S Carrell, Robert E Schoen, Daniel A Leffler, Michele Morris, Sherri Rose, Andrew Baer, Seth D Crockett, Rebecca A Gourevitch, Katie M Dean, and Ateev Mehrotra. 2017. Challenges in adapting existing clinical natural language processing systems to multiple, diverse health care settings. *Journal of the American Medical Informatics Association*, 24(5):986–991.
- Danton S Char, Nigam H Shah, and David Magnus. 2018. Implementing machine learning in health care—addressing ethical challenges. *The New England journal of medicine*, 378(11):981.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. *Electra: Pre-training text encoders as discriminators rather than generators*. In *International Conference on Learning Representations*.
- Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35.

¹<https://docs.cloud.sdu.dk/intro/security.html>

²<https://www.retsinformation.dk/eli/lta/2011/593> (English version is unfortunately not available)

- Hervé Decousus, Victor F Tapson, Jean-François Bergmann, Beng H Chong, James B Froehlich, Ajay K Kakkar, Geno J Merli, Manuel Monreal, Mashio Nakamura, Ricardo Pavanello, et al. 2011. Factors at admission associated with bleeding risk in medical patients: findings from the improve investigators. *Chest*, 139(1):69–79.
- Peter L Elkin, Sarah Mullin, Jack Mardekian, Christopher Crouner, Sylvester Sakilay, Shyamashree Sinha, Gary Brady, Marcia Wright, Kimberly Nolen, JoAnn Trainer, et al. 2021. Using artificial intelligence with natural language processing to combine electronic health record’s structured and free text data to identify nonvalvular atrial fibrillation to decrease strokes and death: Evaluation and case-control study. *Journal of medical Internet research*, 23(11):e28946.
- Batya Friedman and Helen Nissenbaum. 1996. Bias in computer systems. *ACM Transactions on information systems (TOIS)*, 14(3):330–347.
- Eugenia R McPeck Hinz, Lisa Bastarache, and Joshua C Denny. 2013. A natural language processing algorithm to define a venous thromboembolism phenotype. In *AMIA Annual Symposium Proceedings*, volume 2013, page 975. American Medical Informatics Association.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Dirk Hovy and Shrimai Prabhumoye. 2021. Five sources of bias in natural language processing. *Language and Linguistics Compass*, 15(8):e12432.
- Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, and Stephen Denuyl. 2020. Social biases in nlp models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5501.
- Christopher J Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. 2019. Key challenges for delivering clinical impact with artificial intelligence. *BMC medicine*, 17(1):1–9.
- Diederik P. Kingma and Jimmy Ba. 2015. **Adam: A method for stochastic optimization**. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Hee-Jin Lee, Min Jiang, Yonghui Wu, Christian M Shaffer, John H Cleator, Eitan A Friedman, Joshua P Lewis, Dan M Roden, Josh Denny, and Hua Xu. 2017. A comparative study of different methods for automatic identification of clopidogrel-induced bleedings in electronic health records. *AMIA Summits on Translational Science Proceedings*, 2017:185.
- Rumeng Li, Baotian Hu, Feifan Liu, Weisong Liu, Francesca Cunningham, David D McManus, Hong Yu, et al. 2019. Detection of bleeding events in electronic health record notes using convolutional neural network models enhanced with recurrent neural network autoencoders: deep learning approach. *JMIR medical informatics*, 7(1):e10788.
- Sean MacAvaney, Arman Cohan, and Nazli Goharian. 2017. Guir at semeval-2017 task 12: a framework for cross-domain clinical temporal information extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1024–1029.
- Michael E Matheny, Danielle Whicher, and Sonoo Thadaney Israni. 2020. Artificial intelligence in health care: a report from the national academy of medicine. *Jama*, 323(6):509–510.
- Avijit Mitra, Bhanu Pratap Singh Rawat, David McManus, Alok Kapoor, and Hong Yu. 2020. Bleeding entity recognition in electronic health records: A comprehensive analysis of end-to-end systems. In *AMIA Annual Symposium Proceedings*, volume 2020, page 860. American Medical Informatics Association.
- Avijit Mitra, Bhanu Pratap Singh Rawat, David D McManus, Hong Yu, et al. 2021. Relation classification for bleeding events from electronic health records using deep learning systems: an empirical study. *JMIR medical informatics*, 9(7):e27527.
- Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453.
- Lise R Øie, Mattis A Madsbu, Charalampis Gianidakis, Anders Vorhaug, Heidi Jensberg, Øyvind Salvesen, and Sasha Gulati. 2018. Validation of intracranial hemorrhage in the norwegian patient registry. *Brain and behavior*, 8(2):e00900.
- Jannik S Pedersen, Martin S Laursen, Thiusius Rajeeth Savarimuthu, Rasmus Sjøgaard Hansen, Anne Bryde Alnor, Kristian Voss Bjerre, Ina Mathilde Kjær, Charlotte Gils, Anne-Sofie Faarvang Thorsen, Eline Sandvig Andersen, et al. 2021. Deep learning detects and visualizes bleeding events in electronic health records. *Research and practice in thrombosis and haemostasis*, 5(4):e12505.
- Jannik S Pedersen, Martin S Laursen, Cristina Soguero-Ruiz, Thiusius R Savarimuthu, Rasmus Sjøgaard Hansen, and Pernille J Vinholt. 2022. Domain over size: Clinical electra surpasses general bert for bleeding site classification in the free text of electronic health records. In *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–4. IEEE.
- Jianlin Shi, John F Hurdle, Stacy A Johnson, Jeffrey P Ferraro, David E Skarda, Samuel RG Finlayson, Matthew H Samore, and Brian T Bucher. 2021. Natural language processing for the surveillance of

postoperative venous thromboembolism. *Surgery*, 170(4):1175–1182.

Ashwani K Singal, Ramon Bataller, Joseph Ahn, Patrick S Kamath, and Vijay H Shah. 2018. Acg clinical guideline: alcoholic liver disease. *The American journal of gastroenterology*, 113(2):175.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640.

Maxwell Taggart, Wendy W Chapman, Benjamin A Steinberg, Shane Ruckel, Arianna Pregoner-Wenzler, Yishuai Du, Jeffrey Ferraro, Brian T Bucher, Donald M Lloyd-Jones, Matthew T Rondina, et al. 2018. Comparison of 2 natural language processing methods for identification of bleeding among critically ill patients. *JAMA network open*, 1(6):e183451–e183451.

Eric Topol. 2019. *Deep medicine: how artificial intelligence can make healthcare human again*. Hachette UK.

Vera E Valkhoff, Preciosa M Coloma, Gwen MC Masclee, Rosa Gini, Francesco Innocenti, Francesco Lapi, Mariam Molokhia, Mees Mosseveld, Malene Schou Nielsson, Martijn Schuemie, et al. 2014. Validation study in four health-care databases: upper gastrointestinal bleeding misclassification affects precision but not magnitude of drug-related upper gastrointestinal bleeding risk. *Journal of clinical epidemiology*, 67(8):921–931.

Amol A Verma, Hassan Masoom, Chloe Pou-Prom, Saaha Shin, Michael Guerzhoy, Michael Fralick, Muhammad Mamdani, and Fahad Razak. 2022. Developing and validating natural language processing algorithms for radiology reports compared to icd-10 codes for identifying venous thromboembolism in hospitalized medical patients. *Thrombosis Research*, 209:51–58.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

A Bleeding classification results

Table 6: Precision, recall, and F1 performance for the bleeding classification dataset.

Table 7: Sensitivity and standard error for all anatomical locations of the bleeding classification dataset.

Table 8: Bootstrapped 95% confidence intervals for difference of means between models trained on $\mathcal{T}_{\zeta x}$ and \mathcal{T} of the bleeding classification dataset.

Table 9: Bleeding test set sensitivity and standard error on an anatomical location by percentage of subgroup samples in the modified training set.

Table 10: Bootstrapped 95% confidence intervals for difference of means between models trained on a modified training set, including a percentage of subgroup samples, and models trained on the full training set, \mathcal{T} , of the bleeding classification dataset.

B VTE classification results

Table 11: Precision, recall, and F1 performance for the VTE classification dataset.

Table 12: Sensitivity and standard error for all anatomical locations of the VTE classification dataset.

Table 13: Bootstrapped 95% confidence intervals for difference of means between models trained on $\mathcal{T}_{\zeta x}$ and \mathcal{T} of the VTE classification dataset.

Table 14: VTE test set sensitivity and standard error on an anatomical location by percentage of subgroup samples in the modified training set.

Table 15: Bootstrapped 95% confidence intervals for difference of means between models trained on a modified training set, including a percentage of subgroup samples, and models trained on the full training set, \mathcal{T} , of the VTE classification dataset.

C Bleeding word distribution

Table 16: Most frequent words used to describe bleeding mentions for each anatomical location of the bleeding classification dataset.

Table 6: Precision, recall, and F1 performance for the bleeding classification dataset. $\mathcal{T}_{\mathcal{L}x}$ denotes the training set from which an anatomical location, x , has been removed. SE = Standard error. CI = 95% confidence interval.

	ELECTRA			LSTM		
	Precision \pm SE (CI)	Recall \pm SE (CI)	F1 \pm SE (CI)	Precision \pm SE (CI)	Recall \pm SE (CI)	F1 \pm SE (CI)
$\mathcal{T}_{\mathcal{L}Gastrointestinal}$	94.7 \pm 0.2 (94.2 - 95.0)	94.2 \pm 0.4 (93.4 - 95.0)	94.4 \pm 0.2 (94.0 - 94.8)	89.1 \pm 0.7 (87.7 - 90.5)	88.5 \pm 0.7 (87.2 - 89.8)	88.8 \pm 0.1 (88.6 - 88.9)
$\mathcal{T}_{\mathcal{L}Urogenital}$	94.3 \pm 0.3 (93.6 - 94.8)	93.5 \pm 0.4 (92.6 - 94.3)	93.9 \pm 0.1 (93.6 - 94.2)	89.5 \pm 0.1 (89.3 - 89.8)	87.3 \pm 0.2 (87.0 - 87.7)	88.5 \pm 0.1 (88.3 - 88.6)
$\mathcal{T}_{\mathcal{L}Internal}$	95.0 \pm 0.4 (94.2 - 95.0)	95.1 \pm 0.6 (93.8 - 96.1)	95.0 \pm 0.1 (94.8 - 95.2)	89.9 \pm 0.4 (89.3 - 90.7)	87.6 \pm 0.5 (86.6 - 88.6)	88.8 \pm 0.1 (88.5 - 89.0)
$\mathcal{T}_{\mathcal{L}Otorhinolaryngeal}$	95.2 \pm 0.2 (94.8 - 95.6)	96.2 \pm 0.1 (96.0 - 96.3)	95.7 \pm 0.1 (95.5 - 95.8)	89.7 \pm 0.3 (89.1 - 90.3)	90.7 \pm 0.6 (89.6 - 91.7)	90.1 \pm 0.2 (89.8 - 90.5)
$\mathcal{T}_{\mathcal{L}Dermatological}$	94.7 \pm 0.5 (93.6 - 95.5)	93.7 \pm 0.4 (93.1 - 94.5)	94.2 \pm 0.1 (94.0 - 94.4)	89.5 \pm 0.3 (89.0 - 90.2)	87.0 \pm 0.3 (86.5 - 87.6)	88.2 \pm 0.1 (88.1 - 88.3)
$\mathcal{T}_{\mathcal{L}Gynecological}$	94.2 \pm 0.2 (93.7 - 94.6)	93.5 \pm 0.2 (93.1 - 93.8)	93.8 \pm 0.1 (93.6 - 94.0)	89.6 \pm 0.1 (89.4 - 89.8)	88.7 \pm 0.2 (88.3 - 89.0)	89.1 \pm 0.1 (88.9 - 89.4)
$\mathcal{T}_{\mathcal{L}Cerebral}$	95.2 \pm 0.2 (94.7 - 95.6)	94.0 \pm 0.2 (93.5 - 94.5)	94.6 \pm 0.1 (94.3 - 94.8)	89.1 \pm 0.2 (88.7 - 89.5)	88.9 \pm 0.2 (88.6 - 89.4)	89.0 \pm 0.1 (88.8 - 89.2)
$\mathcal{T}_{\mathcal{L}Ophthalmological}$	95.0 \pm 0.1 (94.7 - 95.3)	95.6 \pm 0.2 (95.2 - 96.0)	95.3 \pm 0.1 (95.1 - 95.5)	88.8 \pm 0.4 (88.0 - 89.6)	87.7 \pm 0.5 (87.0 - 88.8)	88.2 \pm 0.2 (87.8 - 88.5)
\mathcal{T}	95.9 \pm 0.2 (95.5 - 96.2)	95.4 \pm 0.3 (94.7 - 96.0)	95.6 \pm 0.1 (95.4 - 95.8)	90.5 \pm 0.5 (89.6 - 91.4)	90.4 \pm 0.6 (89.3 - 91.7)	90.5 \pm 0.1 (90.3 - 90.6)

Table 7: Sensitivity (%) and standard error for all anatomical locations of the bleeding classification dataset. * denotes a significant difference at the 0.05 level between models trained on $\mathcal{T}_{\mathcal{L}x}$ and \mathcal{T} .

	Gastrointestinal	Urogenital	Internal	Otorhinolaryngeal	Dermatological	Gynecological	Cerebral	Ophthalmological
ELECTRA								
$\mathcal{T}_{\mathcal{L}Gastrointestinal}$	75.7 \pm 2.4*	98.6 \pm 0.3*	96.2 \pm 0.3	96.2 \pm 0.4*	94.3 \pm 0.4*	95.9 \pm 0.4*	97.3 \pm 0.3*	99.1 \pm 0.1*
$\mathcal{T}_{\mathcal{L}Urogenital}$	97.1 \pm 0.3*	68.5 \pm 2.2*	97.4 \pm 0.1*	96.6 \pm 0.2*	95.8 \pm 0.6*	95.8 \pm 0.5*	98.1 \pm 0.1*	98.7 \pm 0.3
$\mathcal{T}_{\mathcal{L}Internal}$	96.6 \pm 0.2*	98.8 \pm 0.2*	82.3 \pm 2.0*	96.2 \pm 0.3*	94.6 \pm 0.6*	95.6 \pm 1.0*	97.3 \pm 0.3*	99.0 \pm 0.2*
$\mathcal{T}_{\mathcal{L}Otorhinolaryngeal}$	96.2 \pm 0.4	98.2 \pm 0.2*	96.2 \pm 0.3	92.6 \pm 0.4*	94.6 \pm 0.2*	95.1 \pm 0.5*	97.4 \pm 0.1*	99.1 \pm 0.2*
$\mathcal{T}_{\mathcal{L}Dermatological}$	96.6 \pm 0.4	98.2 \pm 0.3*	96.2 \pm 0.4*	95.8 \pm 0.5	70.6 \pm 1.4*	96.0 \pm 0.7*	97.6 \pm 0.2*	98.7 \pm 0.2
$\mathcal{T}_{\mathcal{L}Gynecological}$	97.0 \pm 0.2*	98.6 \pm 0.3*	97.2 \pm 0.3*	97.2 \pm 0.2*	95.5 \pm 0.4*	64.8 \pm 0.8*	97.9 \pm 0.4*	99.4 \pm 0.1*
$\mathcal{T}_{\mathcal{L}Cerebral}$	96.2 \pm 0.3	98.3 \pm 0.3*	96.6 \pm 0.4	99.1 \pm 0.1*	94.4 \pm 0.5*	95.8 \pm 0.4*	75.4 \pm 0.7*	99.1 \pm 0.1*
$\mathcal{T}_{\mathcal{L}Ophthalmological}$	96.6 \pm 0.2*	98.5 \pm 0.2*	96.5 \pm 0.2*	95.2 \pm 0.3	93.8 \pm 0.6	95.4 \pm 0.6*	97.4 \pm 0.3*	90.7 \pm 0.5*
\mathcal{T}	95.4 \pm 0.4	97.3 \pm 0.4	95.6 \pm 0.5	95.2 \pm 0.2	92.4 \pm 0.8	93.4 \pm 0.2	95.3 \pm 0.4	98.4 \pm 0.3
LSTM								
$\mathcal{T}_{\mathcal{L}Gastrointestinal}$	63.6 \pm 3.0*	94.6 \pm 0.7*	85.8 \pm 1.0*	96.1 \pm 0.5*	88.2 \pm 0.9	92.2 \pm 0.8	90.6 \pm 0.7*	96.6 \pm 0.6*
$\mathcal{T}_{\mathcal{L}Urogenital}$	90.2 \pm 0.4	56.7 \pm 1.6*	86.0 \pm 0.5	97.4 \pm 0.2*	89.8 \pm 0.3*	91.6 \pm 0.4	90.9 \pm 0.3*	96.2 \pm 0.1
$\mathcal{T}_{\mathcal{L}Internal}$	86.9 \pm 0.6	93.0 \pm 0.7	62.6 \pm 1.2*	95.1 \pm 0.5	86.9 \pm 0.5	89.7 \pm 0.4	90.6 \pm 0.5	96.1 \pm 0.4
$\mathcal{T}_{\mathcal{L}Otorhinolaryngeal}$	88.5 \pm 0.6	93.8 \pm 0.6	85.2 \pm 1.0	89.8 \pm 0.5*	89.4 \pm 0.8	92.4 \pm 0.7	90.2 \pm 0.5*	96.1 \pm 0.4*
$\mathcal{T}_{\mathcal{L}Dermatological}$	90.6 \pm 0.3	93.4 \pm 0.5	85.4 \pm 0.4*	96.0 \pm 0.1*	53.4 \pm 0.5*	91.8 \pm 0.5	90.4 \pm 0.6*	95.2 \pm 0.2
$\mathcal{T}_{\mathcal{L}Gynecological}$	89.3 \pm 0.5	92.4 \pm 0.3	86.4 \pm 0.2*	96.1 \pm 0.3	89.2 \pm 0.4	69.1 \pm 1.2*	90.8 \pm 0.4*	96.2 \pm 0.2
$\mathcal{T}_{\mathcal{L}Cerebral}$	87.8 \pm 0.6	93.7 \pm 0.2	85.6 \pm 0.6	96.8 \pm 0.3*	89.0 \pm 0.4	93.0 \pm 0.2*	68.8 \pm 0.5*	96.7 \pm 0.2*
$\mathcal{T}_{\mathcal{L}Ophthalmological}$	89.5 \pm 1.0	93.1 \pm 0.7	86.6 \pm 0.9*	96.9 \pm 0.4*	89.4 \pm 1.0	93.6 \pm 0.3*	90.9 \pm 0.3*	61.5 \pm 2.1*
\mathcal{T}	88.9 \pm 1.1	93.0 \pm 0.8	85.2 \pm 0.8	95.4 \pm 0.2	87.8 \pm 0.8	90.9 \pm 0.6	88.7 \pm 0.3	94.6 \pm 0.7

Table 8: Bootstrapped 95% confidence intervals for difference of means between models trained on $\mathcal{T}_{\mathcal{L}x}$ and \mathcal{T} of the bleeding classification dataset. Means are computed as performance of models trained on $\mathcal{T}_{\mathcal{L}x}$ minus \mathcal{T} . Total = difference of means on the full test set.

	Total	Gastrointestinal	Urogenital	Internal	Otorhinolaryngeal	Dermatological	Gynecological	Cerebral	Ophthalmological
ELECTRA									
$\mathcal{T}_{\mathcal{L}Gastrointestinal}$	-1.5, -0.9	-23.6, -15.0	0.2, 2.3	-0.6, 2.0	0.5, 1.5	0.6, 3.3	1.6, 3.3	1.4, 2.6	0.5, 1.1
$\mathcal{T}_{\mathcal{L}Urogenital}$	-2.0, -1.4	0.6, 2.8	-33.2, -24.7	1.0, 2.7	0.9, 2.1	1.8, 5.0	1.2, 3.6	1.8, 3.8	-0.6, 1.3
$\mathcal{T}_{\mathcal{L}Internal}$	-0.8, -0.4	0.2, 2.4	0.7, 2.4	-18.5, -9.1	0.1, 2.2	0.1, 4.6	0.4, 3.9	1.2, 3.0	0.1, 1.2
$\mathcal{T}_{\mathcal{L}Otorhinolaryngeal}$	-0.3, 0.4	-0.2, 1.6	0.2, 1.7	-0.5, 1.9	-2.9, -2.1	0.7, 4.0	0.7, 2.5	1.5, 2.6	0.2, 1.4
$\mathcal{T}_{\mathcal{L}Dermatological}$	-1.8, -1.1	0.0, 2.4	0.4, 1.7	0.1, 1.3	-0.2, 1.6	-23.5, -19.8	1.8, 3.6	1.9, 2.9	-0.1, 1.2
$\mathcal{T}_{\mathcal{L}Gynecological}$	-2.2, -1.5	1.1, 2.3	0.7, 2.1	0.8, 2.2	1.3, 2.7	1.6, 4.9	-29.5, -27.3	2.1, 3.4	0.6, 1.4
$\mathcal{T}_{\mathcal{L}Cerebral}$	-1.5, -0.5	-0.1, 1.6	0.2, 1.9	-0.2, 2.3	0.5, 1.6	0.4, 3.5	1.7, 3.3	-21.3, -18.7	0.3, 1.3
$\mathcal{T}_{\mathcal{L}Ophthalmological}$	-0.6, -0.1	0.5, 2.3	0.5, 2.1	0.1, 1.7	-0.7, 0.7	-0.1, 3.7	1.0, 3.4	1.5, 2.9	-8.6, -6.5
LSTM									
$\mathcal{T}_{\mathcal{L}Gastrointestinal}$	-1.9, -1.4	-30.3, -18.6	1.0, 2.1	1.0, 2.2	0.1, 1.4	-0.6, 1.4	-0.2, 2.6	0.8, 2.9	1.2, 3.0
$\mathcal{T}_{\mathcal{L}Urogenital}$	-2.1, -1.8	-1.7, 4.0	-38.6, -34.0	-0.5, 3.6	1.4, 2.8	0.1, 4.1	-1.3, 2.6	1.2, 3.1	-0.1, 3.0
$\mathcal{T}_{\mathcal{L}Internal}$	-1.8, -1.4	-4.6, 0.7	-1.4, 1.7	-23.0, -20.3	-1.4, 1.1	-2.1, 1.0	-2.6, 0.2	-2.6, 0.2	0.0, 3.0
$\mathcal{T}_{\mathcal{L}Otorhinolaryngeal}$	-0.7, -0.1	-2.7, 1.8	-0.8, 2.1	-1.2, 3.4	-6.9, -4.5	-1.3, 4.1	-0.6, 3.0	0.1, 2.6	0.2, 2.7
$\mathcal{T}_{\mathcal{L}Dermatological}$	-2.2, -2.0	-1.1, 4.2	-2.2, 2.5	-1.3, 3.0	0.2, 1.0	-36.3, -32.1	-0.2, 2.0	0.4, 3.4	-1.0, 2.2
$\mathcal{T}_{\mathcal{L}Gynecological}$	-1.5, -1.1	-1.9, 2.7	-2.0, 0.4	0.3, 3.8	-0.1, 1.6	-0.4, 3.7	-24.3, -19.0	1.4, 2.8	0.0, 2.8
$\mathcal{T}_{\mathcal{L}Cerebral}$	-1.7, -1.2	-3.4, 0.4	-1.0, 1.7	-0.3, 3.0	1.0, 1.8	-0.6, 3.4	1.4, 2.7	-21.0, -19.0	0.6, 3.4
$\mathcal{T}_{\mathcal{L}Ophthalmological}$	-2.6, -1.8	-2.9, 3.8	-1.1, 1.4	0.1, 4.8	0.7, 2.4	-0.2, 3.4	1.4, 3.9	1.3, 3.0	-38.5, -28.5

Table 9: Bleeding test set sensitivity (%) and standard error on an anatomical location by percentage of subgroup samples in the modified training set. * denotes a significant difference at the 0.05 level between models trained on the modified training set and the full training set, \mathcal{T} .

	Anatomical subgroup fraction						
	0.0	0.01	0.15	0.30	0.60	0.80	1.0
	ELECTRA						
Gastrointestinal	75.7±2.4*	76.6± 0.6*	90.4± 1.1*	93.2± 0.3*	94.5± 0.5*	96.6± 0.5	95.8± 0.4
Urogenital	68.5±2.2*	72.4± 1.1*	91.0± 1.3*	96.6± 0.3*	97.1± 0.1*	97.0± 0.1*	98.4± 0.4
Internal	82.3±2.0*	84.3± 1.8*	91.8± 0.6*	93.6± 0.2*	94.8± 0.5*	95.9± 0.4*	96.4± 0.4
Otorhinolaryngeal	92.6±0.4*	92.2± 0.7*	92.9± 0.3*	94.8± 0.2*	95.4± 0.6	95.8± 0.2	95.6± 0.3
Dermatological	70.6±1.4*	74.6± 1.3*	87.9± 0.7*	88.7± 0.9*	93.3± 0.7	93.4± 0.3	94.5± 0.6
Gynecological	64.8±0.8*	63.4± 1.3*	87.3± 0.6*	89.2±0.3*	92.0± 0.7*	93.7± 0.2	93.9± 0.4
Cerebral	75.4±0.7*	80.0± 1.5*	89.4± 0.4*	91.5± 0.7*	95.6± 0.5*	96.6± 0.6	97.2± 0.3
Ophthalmological	90.70±0.5*	93.6± 0.3*	96.7± 0.4*	97.8± 0.3*	98.8± 0.1	98.7± 0.1	98.9± 0.2
	LSTM						
Gastrointestinal	63.6 ± 3.0*	60.0 ± 1.0*	71.9 ± 1.7*	79.2 ± 0.7*	86.6 ± 0.5*	88.0 ± 0.9*	89.1 ± 0.6
Urogenital	56.7 ± 1.6*	58.6 ± 0.9*	79.3 ± 0.7*	87.2 ± 0.6*	90.6 ± 0.2*	92.2 ± 0.3*	93.7 ± 0.1
Internal	62.6 ± 1.2*	65.6 ± 0.8*	72.6 ± 0.8*	77.2 ± 0.5*	80.2 ± 0.8*	83.9 ± 0.5	85.5 ± 0.5
Otorhinolaryngeal	89.8 ± 0.5*	88.5 ± 0.3*	91.8 ± 0.3*	93.6 ± 0.3*	93.8 ± 0.1*	95.2 ± 0.5	95.5 ± 0.3
Dermatological	53.4 ± 0.5*	56.5 ± 1.5*	73.5 ± 2.4*	77.4 ± 0.6*	84.5 ± 0.3*	86.8 ± 0.4	88.1 ± 0.5
Gynecological	69.1 ± 1.2*	69.3 ± 0.5*	80.1 ± 0.7*	86.6 ± 0.7*	89.3 ± 0.5*	90.7 ± 0.4*	91.6 ± 0.3
Cerebral	68.8 ± 0.5*	72.2 ± 1.6*	79.7 ± 0.4*	83.1 ± 0.6*	86.4 ± 0.4*	88.4 ± 0.5*	90.5 ± 0.2
Ophthalmological	61.5 ± 2.1*	65.4 ± 1.4*	89.4 ± 0.4*	90.3 ± 0.5*	93.0 ± 0.7*	94.4 ± 0.4*	95.7 ± 0.4

Table 10: Bootstrapped 95% confidence intervals for difference of means between models trained on a modified training set, including a percentage of subgroup samples, and models trained on the full training set, \mathcal{T} , of the bleeding classification dataset. Means are computed as performance of models trained on the modified training set minus \mathcal{T} .

	Anatomical subgroup fraction						
	0.0	0.01	0.15	0.30	0.60	0.80	1.0
	ELECTRA						
Gastrointestinal	-23.6 , -15.0	-20.2 , -18.2	-8.3 , -2.6	-4.2 , -1.2	-2.6 , -0.2	-0.2 , 1.8	
Urogenital	-33.2 , -24.7	-27.7 , -24.0	-10.0 , -4.8	-2.6 , -1.4	-2.2 , -0.4	-2.2 , -0.6	
Internal	-18.5 , -9.1	-15.2 , -9.0	-5.8 , -3.4	-3.8 , -1.8	-2.4 , -1.0	-3.0 , -0.1	
Otorhinolaryngeal	-2.9 , -2.1	-4.8 , -2.1	-3.5 , -1.8	-1.5 , -0.2	-0.7 , 0.3	-0.6 , 1.0	
Dermatological	-23.5 , -19.8	-22.6 , -16.5	-7.8 , -5.0	-7.8 , -3.4	-3.0 , 1.0	-2.4 , 0.1	
Gynecological	-29.5 , -27.3	-33.6 , -27.5	-8.1 , -5.3	-5.5 , -3.8	-3.6 , -0.8	-0.8 , 1.5	
Cerebral	-21.3 , -18.7	-20.1 , -13.8	-8.4 , -7.3	-7.0 , -4.4	-2.6 , -0.7	-1.3 , 0.0	
Ophthalmological	-8.6 , -6.5	-6.0 , -4.5	-2.7 , -1.8	-1.8 , -0.5	-0.6 , 0.6	-0.2 , 0.6	
	LSTM						
Gastrointestinal	-31.7 , -19.8	-31.0 , -27.2	-21.4 , -14.3	-12.3 , -7.6	-4.1 , -0.6	-1.9 , -0.1	
Urogenital	-39.8 , -33.8	-36.7 , -32.9	-15.8 , -13.0	-7.8 , -4.8	-3.5 , -2.5	-2.1 , -0.9	
Internal	-25.7 , -19.6	-22.3 , -17.4	-15.0 , -10.6	-9.5 , -7.1	-7.4 , -3.1	-3.4 , 0.4	
Otorhinolaryngeal	-7.0 , -4.5	-8.2 , -6.2	-4.5 , -3.0	-2.5 , -1.4	-2.3 , -1.1	-1.8 , 1.0	
Dermatological	-35.6 , -33.8	-35.5 , -28.6	-19.0 , -10.1	-12.8 , -9.0	-4.9 , -2.5	-3.1 , 0.0	
Gynecological	-25.4 , -19.5	-23.3 , -21.2	-13.4 , -10.3	-6.5 , -3.4	-3.9 , -0.9	-1.2 , -0.6	
Cerebral	-22.3 , -21.2	-21.1 , -15.1	-11.8 , -9.8	-8.7 , -5.8	-5.0 , -3.2	-3.4 , -0.8	
Ophthalmological	-38.4 , -30.5	-33.4 , -27.2	-7.4 , -5.1	-6.5 , -4.1	-3.9 , -1.1	-2.1 , -0.6	

Table 11: Precision, recall, and F1 performance for the VTE classification dataset. $\mathcal{T}_{\mathcal{Q}x}$ denotes the training set from which an anatomical location, x , has been removed. SE = Standard error. CI = 95% confidence interval.

	ELECTRA			LSTM		
	Precision ± SE (CI)	Recall ± SE (CI)	F1 ± SE (CI)	Precision ± SE (CI)	Recall ± SE (CI)	F1 ± SE (CI)
$\mathcal{T}_{\mathcal{Q}Lower\ extremity}$	86.3 ± 0.7 (84.9 - 87.7)	41.8 ± 1.5 (39.1 - 44.7)	56.3 ± 1.3 (53.8 - 58.8)	77.2 ± 0.2 (76.9 - 77.6)	61.4 ± 1.5 (58.8 - 64.5)	68.3 ± 0.8 (66.8 - 70.1)
$\mathcal{T}_{\mathcal{Q}Lung}$	86.1 ± 1.0 (84.1 - 87.9)	57.4 ± 3.2 (51.8 - 63.8)	68.6 ± 2.0 (64.9 - 72.7)	78.6 ± 0.5 (77.6 - 79.6)	54.8 ± 0.7 (53.5 - 56.1)	64.6 ± 0.3 (63.9 - 65.3)
\mathcal{T}	96.4 ± 0.3 (95.9 - 97.0)	72.3 ± 0.7 (71.0 - 73.6)	82.7 ± 0.4 (81.9 - 83.5)	91.4 ± 0.1 (91.2 - 91.6)	57.2 ± 0.6 (56.0 - 58.4)	70.4 ± 0.4 (69.4 - 71.2)

Table 12: Sensitivity (%) and standard error for all anatomical locations of the VTE classification dataset. * denotes a significant difference at the 0.05 level between models trained on $\mathcal{T}_{\mathcal{Z}_x}$ and \mathcal{T} .

	Lower extremity	Lung	Liver	Cerebral	Upper extremity
ELECTRA					
$\mathcal{T}_{\mathcal{Z}_{Lower\ extremity}}$	9.3 ± 0.8*	98.6 ± 0.2*	50.1 ± 3.6*	23.8 ± 1.9*	25.3 ± 1.4*
$\mathcal{T}_{\mathcal{Z}_{Lung}}$	99.7 ± 0.2*	16.6 ± 3.5*	65.7 ± 7.7	13.6 ± 4.1	99.0 ± 0.3*
\mathcal{T}	99.1 ± 0.2	97.6 ± 0.3	66.3 ± 1.6	11.5 ± 1.6	96.7 ± 0.3
LSTM					
$\mathcal{T}_{\mathcal{Z}_{Lower\ extremity}}$	34.2 ± 1.5*	92.6 ± 0.5*	76.2 ± 2.2*	52.6 ± 2.2*	47.6 ± 2.8*
$\mathcal{T}_{\mathcal{Z}_{Lung}}$	95.3 ± 0.2*	15.7 ± 0.4*	54.2 ± 1.8*	24.7 ± 1.3*	91.4 ± 0.4*
\mathcal{T}	93.8 ± 0.6	86.6 ± 0.6	29.8 ± 1.6	7.0 ± 0.5	81.8 ± 1.3

Table 13: Bootstrapped 95% confidence intervals for difference of means between models trained on $\mathcal{T}_{\mathcal{Z}_x}$ and \mathcal{T} of the VTE classification dataset. Means are computed as performance of models trained on $\mathcal{T}_{\mathcal{Z}_x}$ minus \mathcal{T} . Total = difference of means on the full test set.

	Total	Lower extremity	Lung	Liver	Cerebral	Upper extremity
ELECTRA						
$\mathcal{T}_{\mathcal{Z}_{Lower\ extremity}}$	-18.0, -16.3	-91.3, -88.3	0.2, 1.6	-23.9, -7.3	8.0, 16.6	-74.0, -68.7
$\mathcal{T}_{\mathcal{Z}_{Lung}}$	-12.4, -9.2	0.3, 0.9	-87.2, -73.7	-12.6, 11.6	-2.0, 8.3	1.8, 2.7
LSTM						
$\mathcal{T}_{\mathcal{Z}_{Lower\ extremity}}$	-5.1, -3.5	-63.0, -56.5	4.3, 7.7	43.8, 50.0	40.3, 49.6	-41.9, -28.2
$\mathcal{T}_{\mathcal{Z}_{Lung}}$	-6.6, -5.4	0.6, 2.6	-72.5, -69.5	18.9, 28.8	14.9, 19.7	7.5, 11.9

Table 14: VTE test set sensitivity (%) and standard error on an anatomical location by percentage of subgroup samples in the modified training set. * denotes a significant difference at the 0.05 level between models trained on the modified training set and the full training set, \mathcal{T} .

	Anatomical subgroup fraction						
	0.0	0.01	0.15	0.30	0.60	0.80	1.0
ELECTRA							
Lower extremity	9.3 ± 0.8*	77.6 ± 3.7*	97.2 ± 0.3*	97.7 ± 0.4	98.6 ± 0.3	98.8 ± 0.2	98.4 ± 0.2
Lung	16.6 ± 3.5*	57.1 ± 8.1*	94.6 ± 0.3*	96.6 ± 0.3	97.7 ± 0.2	98.2 ± 0.1	97.6 ± 0.3
LSTM							
Lower extremity	34.2 ± 1.5	40.6 ± 0.7	83.7 ± 0.6	90.7 ± 0.5	94.4 ± 0.6	95.5 ± 0.6	94.9 ± 0.7
Lung	15.7 ± 0.4	16.0 ± 0.9	71.5 ± 1.1	81.6 ± 0.7	88.0 ± 0.9	88.3 ± 0.9	89.2 ± 0.8

Table 15: Bootstrapped 95% confidence intervals for difference of means between models trained on a modified training set, including a percentage of subgroup samples, and models trained on the full training set, \mathcal{T} , of the VTE classification dataset. Means are computed as performance of models trained on the modified training set minus \mathcal{T} .

	Anatomical subgroup fraction					
	0.0	0.01	0.15	0.30	0.60	0.80
ELECTRA						
Lower extremity	-91.3, -88.3	-28.8, -15.0	-2.2, -0.4	-1.9, 0.4	-0.2, 0.7	-0.4, 1.1
Lung	-87.2, -73.7	-58.8, -27.4	-3.6, -2.2	-2.0, 0.0	-0.7, 0.8	-0.1, 1.2
LSTM						
Lower extremity	-63.0, -56.5	-56.2, -52.2	-13.2, -9.4	-5.6, -2.9	-2.3, 1.2	-1.3, 2.2
Lung	-72.5, -69.5	-76.3, -70.1	-20.6, -14.4	-9.2, -6.0	-3.5, 0.8	-3.6, 1.9

Table 16: Most frequent words used to describe bleeding mentions for each anatomical location of the bleeding classification dataset. Words are translated from Danish to English and, therefore, some cells include two words.

Word	Frequency	Location uniqueness	Word	Frequency	Location uniqueness
Location: Otorhinolaryngeal			Location: Gynecological		
bleeding	324	0.13	bleeding	714	0.29
nose bleeding	273	1.0	uterus	108	0.97
epistaxis	254	0.99	allowable	78	0.94
nostril	148	1.0	vagina	69	1.0
Location: Dermatological			Location: Cerebral		
haematoma	354	0.56	sah	217	0.99
bleeding	170	0.07	bleeding	190	0.08
skin	122	0.73	ct	185	0.63
right	97	0.29	haematoma	161	0.25
Location: Urogenital			Location: Internal		
haematuria	536	0.99	bleeding	273	0.11
urine	311	0.98	haemothorax	249	1.0
blood	205	0.23	fluid	174	0.80
macroscopic	186	0.99	blood	140	0.16
Location: Gastrointestinal			Location: Ophthalmological		
bleeding	560	0.23	corpus hemorrhagicum	270	1.0
blood	247	0.28	corpus hem	205	1.0
fresh	180	0.48	bleeding	198	0.08
melaena	165	0.98	haemorrhage	180	0.70

Topic Ontologies for Arguments

Yamen Ajjour

Lebiniz University Hannover

Benno Stein

Bauhaus-Universität Weimar

Johannes Kiesel

Bauhaus-Universität Weimar

Martin Potthast

Leipzig University and ScaDS.AI

Abstract

Many computational argumentation tasks, such as stance classification, are topic-dependent: The effectiveness of approaches to these tasks depends largely on whether they are trained with arguments on the same topics as those on which they are tested. The key question is: What are these training topics? To answer this question, we take the first step of mapping the argumentation landscape with The Argument Ontology (TAO). TAO draws on three authoritative sources for argument topics: the World Economic Forum, Wikipedia’s list of controversial topics, and Debatepedia. By comparing the topics in our ontology with those in 59 argument corpora, we perform the first comprehensive assessment of their topic coverage. While TAO already covers most of the corpus topics, the corpus topics barely cover all the topics in TAO. This points to a new goal for corpus construction to achieve a broad topic coverage and thus better generalizability of computational argumentation approaches.

1 Introduction

The term “topic” refers to the subject matter of a text. A text may be about one or more topics and the relationship between topics and texts is called “aboutness” (Yablo, 2014). Topics play a central role in argumentation because they determine argumentation strategies and rhetorical devices by setting the appropriate and expected universe of discourse. This view is supported by pragmatics (van Eemeren, 2015): “The basic aspects of strategic maneuvering [...] are making an expedient selection from the ‘topical potential’ available at a certain discussion.” Although debaters often use commonplace arguments across topics (Bilu et al., 2019), they must be relevant: a black market argument, for example, can be equally well applied to topics such as banning drugs or banning firearms. As recently shown, for example, by Reuver et al. (2021), training computational models to extract, analyze, or generate arguments with a broad topic coverage improves their generalizability.

A set of topics can be organized as a graph, sometimes called a “topic space”. Information theorists and library scientists map hierarchical subject relationships into ontologies in this way (Hjørland, 2001). For this purpose, topics are labeled with a subject heading, a phrase from a controlled vocabulary that describes a topic in a concise and discriminating manner. While library ontologies are not focused on argumentation, others deal specifically with *argumentative* topic spaces. We have identified and tapped three authoritative sources of ontological knowledge covering global issues, controversies, and popular debates: the World Economic Forum’s “Strategic Intelligence” site, Wikipedia’s list of controversial topics, and Debatepedia’s debate classification system (Section 4). They form the basis for The Argument Ontology (TAO).¹

We compile a comprehensive survey of 59 argument corpora (Section 3) and investigate their topic coverage with respect to the three authoritative ontologies (Section 5). The coverage of corpora with topic labels is manually assessed by matching each label with the topics of the ontologies. From this, the ontology topics covered by a corpus and the distribution of corpus arguments in the ontologies are calculated. Our analyses show that the existing corpora focus on only a subset of the known topics. For corpora without topic labels, we categorize their argumentative texts by measuring their semantic relatedness to ontology topics. Given the large number of ontology topics (748 for Wikipedia), this is a challenging classification for which we achieve a remarkable F_1 of 0.59. (Section 6).²

Altogether, we lay the foundation for the study and systematic exploration of controversial topics within computational argumentation analysis. The authoritative sources identified already cover their respective areas quite comprehensively. Future work will need to extend our approach to other subject areas, such as business, domestic, historical, and scientific argument spaces.

¹Data: <https://zenodo.org/record/3928096>.

²Code: <https://github.com/webis-de/EACL-23>.

2 Related Work

Our review of related work focuses on the role of the variable “topic” in computational argumentation. Moreover, we briefly review topic ontologies and hierarchical topic classification.

2.1 Topics in Computational Argumentation

In computational argumentation, arguments are typically modeled as compositions of argument units, where an argument unit is represented as a span of text. Habernal and Gurevych (2016a) adopts Toulmin (1958)’s (1958) model, which defines six unit types, among which are “claim” and “data”. Wachsmuth et al. (2017) employ a more basic model of two units, which defines an argument as a claim or conclusion supported by one or more premises. These models capture arguments without explicitly identifying the topic they address. Levy et al. (2014) consider claims to be topic-dependent and study their detection in the context of a random selection of 32 topics from idebate.org. This work raises the question why topic-dependence has not been addressed more urgently until now.

Key tasks for computational argumentation include the mining of arguments from natural language (Moens et al., 2007; Al-Khatib et al., 2016), classifying their stances with regard to a thesis (Bar-Haim et al., 2017), and analyzing which arguments are more persuasive (Tan et al., 2016; Habernal and Gurevych, 2016a). Current approaches to these tasks rely on supervised classification. Daxenberger et al. (2017) show that supervised classifiers fail to generalize across domains (\sim topics). More recently, Stab et al. (2018) tweak BiLSTM (Graves and Schmidhuber, 2005) to integrate the topic while jointly detecting (1) whether a sentence is an argument and (2) its stance to the topic. The designed neural network outperforms BiLSTM without topic integration in both tasks; further evidence for the topic-dependence of argument mining and stance classification. Whether model transfer between more closely related topics works better is unknown. As a first step, Reuver et al. (2021) show that cross-topic stance-classification with BERT (Devlin et al., 2018) produces mixed results depending on the topics, but misses the relations between the topics. Gu et al. (2018) show that integrating the topic of an argument helps assessing its persuasiveness.

Topic plays a central role in argument retrieval and generation since it defines what arguments are

relevant. Argument retrieval aims at delivering pro and con arguments on a given topic query. A major challenge in argument retrieval is the grouping of arguments that address common aspects of a topic. As shown by Reimers et al. (2019) and Ajour et al. (2019a), integrating the topic is an important step while clustering arguments. For argument generation, Bilu et al. (2019) introduce an approach that matches an input topic against a list of topics that are paired with sets of topic-adjustable commonplace arguments (e.g., black-market arguments). In a similar vein, Bar-Haim et al. (2019) identify consistent and contrastive topics for a given topic with the goal of expanding the topic in a new direction (e.g., fast food versus obesity). Both approaches show the merit of utilizing argument topic ontologies in argument generation.

Only abstract argumentation may be truly topic-independent, where only the structure and relations among arguments, not their language, are studied.

2.2 Topic Ontologies

In information science, an ontology is defined as “an explicit specification of a conceptualization” (Gruber, 1993). Topic ontologies are a specific type of ontologies which specify topics as nodes of a directed acyclic graph. An edge in the graph then implies an “is part of”-relation between the topics (Xamena et al., 2017). The effort in creating topic ontologies ranges from ad-hoc decisions (e.g., tags for blog posts) to extensive classification schemes for libraries. The oldest classification scheme that is still used today in libraries is the Dewey Decimal Classification. It has been translated into over 30 languages, and it contains several tens of thousands of classes. Most topic ontologies focus on a specific domain, such as the ACM Computing Classification System for computer science, or DMOZ for web pages.³ The only topic ontology directly linked to arguments is that of Debatepedia.

2.3 Hierarchical Text Classification

Hierarchical text classification aims at classifying a document into a class hierarchy. Depending on how the hierarchical structure is exploited, classification can be done top-down (from higher classes downwards), bottom-up, or flat (ignoring hierarchical relations) (Silla and Freitas, 2011). Researchers usually train supervised classifiers for each class in the hierarchy (Sun and Lim, 2001).

³<https://dl.acm.org/ccs> and <https://dmoz-odp.org/>

Corpus	Authors	Source	Unit granularity	Units	Topics	Exp.
Manual selection						
Arguing Subjectivity	Conard et al. (2012)	Editorials	Editorial/blog	84	1	1
Arguments Moderation	Falk et al. (2021)	Discussion forum	Argument	112	2	2
Argumentative Sentences	Eyal et al. (2020)	Wikipedia	Arguments	700	20	1
Argument Facet Similarity	Misra et al. (2016)	Debate portals	Argument	6,188	3	12
AURC	Trautmann et al. (2020)	Web	Argument Unit	8,000	8	6
Basn	Kondo et al. (2021)	Debate portals	Argument pair	2,370	6	1
CCSA	Li et al. (2022)	Scientific papers	Argument unit	18,332	1	1
Claim and Evidence 1	Aharoni et al. (2014)	Wikipedia	Wikipedia article	315	33	22
Claim and Evidence 2	Rinott et al. (2015)	Wikipedia	Wikipedia article	547	58	16
Claim Generation	Gretz et al. (2020)	Generated text	Argument Unit	2,839	136	1
Claim Stance	Bar-Haim et al. (2017)	Wikipedia	Argument Unit	2,394	55	15
Claim Sentence Search	Levy et al. (2018)	Wikipedia	Argument unit	1,492,077	150	5
COMARG	Boltužić and Šnajder (2014)	Debate portals	Argument pair	2,298	2	3
Evidence Sentences	Schnarch et al. (2018)	Wikipedia	Argument unit	5,783	118	6
Evidence Sentences 2	Ein-Dor et al. (2020)	Wikipedia	Argument unit	29,429	221	4
Evidence Quality	Gleize et al. (2019)	Wikipedia	Argument pair	5,697	69	2
IAM	Cheng et al. (2022)	Wikipedia	Argument unit	69,666	100	1
ICLE Essay Scoring	Persing et al. (2010)	Essays	Essay	1,000	10	12
Ideological Debates Reasons	Hasan and Ng (2014)	Debate portals	Argument	4,903	4	12
Internet Argument Corpus v2	Abbott et al. (2016)	Web	Discussion	16,555	19	22
Key Point Analysis	Bar-Haim et al. (2020)	Wikipedia	Argument	24,093	28	15
M-Arg	Mestre et al. (2021)	Presidential debate	Argument pair	4,104	18	1
Micro Text v1	Peldszus and Stede (2015)	Essays	Essay	112	18	13
Micro Text v2	Skeppstedt et al. (2018)	Essays	Essay	171	35	2
Multilingual Argument Mining	Toledo-Ronen et al. (2020)	Wikipedia	Argument unit	65,708	347	4
Political Argumentation	Menini et al. (2018)	Presidential debate	Argument pair	1,462	5	3
Record Debating Dataset 2	Mirkin et al. (2018)	Debating	Speech	200	50	5
Record Debating Dataset 3	Lavee et al. (2019)	Debating	Speech	400	199	1
Record Debating Dataset 4	Orbach et al. (2019)	Debating	Speech	200	50	1
Record Debating Dataset 5	Orbach et al. (2020)	Debating	Speech	3,562	397	1
Sci-arg	Lauscher et al. (2018)	Scientific papers	Paper	40	1	7
SciARK	Fergadis et al. (2021)	Scientific papers	Abstract	1,000	6	1
UKP Sentential	Stab et al. (2018)	Web	Argument	25,492	8	20
UKP Aspect	Reimers et al. (2019)	Web	Argument pair	3,595	28	11
UKPConvArg1	Habernal and Gurevych (2016c)	Debate portals	Argument pair	11,650	16	16
UKPConvArg2	Habernal and Gurevych (2016b)	Debate portals	Argument pair	9,111	16	6
WebDiscourse	Habernal and Gurevych (2016a)	Web	Document	340	6	12
Webis-debate-16	Al-Khatib et al. (2016a)	Debate portals	Debate	445	14	5
VivesDebate	Ruiz-Dolz et al. (2021)	Debating	Debate	29	1	2
Source-driven: greedy within a time-span						
AIFdb	Bex et al. (2013)	Web	Argument unit	67,408	n/a	8
Args-me	Ajjour et al. (2019b)	Debate portals	Argument	387,692	n/a	31
ChangeMyView	Tan et al. (2016)	Discussion forum	Post/comment	14,066	n/a	37
CJEU	Grundler et al. (2022)	Law Case	Court Decision	40	n/a	1
DebateSum	Roush and Balaji (2020)	Debating	Debate	187,386	n/a	1
IMHO	Chakrabarty et al. (2019)	Discussion forum	Argument Unit	5,569,962	n/a	3
Intelligence Squared Debates	Zhang et al. (2016)	Debate portals	Debate	108	n/a	9
Kialo	Kialo (2020)	Debate portals	Argument unit	331,684	n/a	23
Political Speech	Lippi and Torroni (2016)	Ministerial debate	Argument unit	152	n/a	2
USElecDeb60To16	Haddadan et al. (2019)	Presidential debate	Debate	42	n/a	5
MultiOpEd	Liu et al. (2021)	Editorials	Editorial	2,794	n/a	2
QT30	Hautli-Janisz et al. (2022)	Debating	Argument unit	19,842	n/a	1
Review-Rebuttal	Cheng et al. (2020)	Scientific reviews	Argument pair	4,764	n/a	5

Table 1 (continued on next page).

Table 1 (continued).

Corpus	Authors	Source	Unit granularity	Units	Topics	Exp.
Source-driven: sampled						
Argument Annotated Essays	Stab and Gurevych (2017)	Essays	Essay	402	n/a	64
E-rulemaking	Park and Cardie (2018)	Discussion forum	Argument	731	n/a	9
ECHR	Poudyal et al. (2020)	Law Case	Argument	743	n/a	8
Editorials	Al-Khatib et al. (2016b)	Editorials	Editorial	300	n/a	15
GAQCorpus	Ng et al. (2020)	Web	Argument	6,424	n/a	4
IDebate Persuasiveness	Persing and Ng (2017)	Debate portals	Argument	1,205	n/a	1
Scinf-biomed	Gao et al. (2022)	Scientific papers	Paper	27,924	n/a	1

Table 1: Survey of argument corpora indicating data source, unit granularity, and size in terms of units and topics (if authors remarked on it). The unit granularity is the one in the corpus’ files, using premises and conclusions as one unit each and the best context-preserving unit for corpora featuring multiple granularities. We presume these topic selection directives from the corpus description: either *manual selection* by the authors, or *source-driven*—i.e., the topics in the selected source(s)—from the units of a specific *time-span* or by random *sampling*. Experiments (Exp.) denotes the count of papers that use the corpus in an experiment among those papers that cite the corpus’ paper.

3 Survey of Argument Corpora

To study arguments and computational argumentation tasks, researchers compile corpora with argumentative texts. To the best of our knowledge, Table 1 compiles all corpora dedicated to argumentation until 2022. We review these corpora and their associated publications with regard to what are the sources of arguments, what is the granularity of the corpus, what is the size of the corpora in terms of their units, and which and how many different topics are covered in them. Reviewing all papers citing a corpus, we also analyzed how many experiments were carried out using them.

The most elaborate discussion of topic selection is given in Habernal and Gurevych (2016a), who chose six topics (homeschooling, public versus private schools, redshirting, prayers in schools, single sex education, mainstreaming) to focus on different education-related aspects. The broadest selection of topics is reported by the researchers of IBM Debater,⁴ who obtain arguments from Wikipedia. However, samples of the topics have been used in their papers without mentioning which ones. The only other work mentioning their source of topics stems from Stab et al. (2018), who randomly select 8 topics from two lists of controversial topics that originate from an online library and the debate portal ProCon.org, respectively. Peldszus and Stede (2015) predefine a set of topics and give writers the freedom to choose which one to write about, but nothing is said about where the set of predefined topics originate from. Conard et al. (2012) and Hasan and Ng (2014) explicitly select one and

four topics, respectively. For all other corpora with topic labels, their authors do not argue on choosing topics, nor selection or sampling criteria. Neither do the authors of corpora without topic labels.

Altogether, it appears that the best practices in argumentation do not as of yet consider topic sampling as a prerequisite task to ensure coverage of a certain domain of interest, diversity, or reproducibility. Based on our review, we presume three basic topic selection directives are in use today: (1) *Manual selection*. Topics are manually defined or selected. Although the process may be random, when aiming for controversial topics, one may often end up with commonplace topics in Western culture (e.g., abortion, death penalty, gay marriage). Still, they are relevant and important today. (2) *Source-driven (greedy within a time-span)*. A source of argument ground truth is either exploited in its entirety, or a maximum subset fulfilling desired properties is used. Since argument-related ground truth is hard to come by, it is understandable that many readily available sources are being exploited. (3) *Source-driven (sampled)*. A source or argument ground truth is exploited and a subset is sampled. Here, it may be infeasible to exploit a source in its entirety. Al-Khatib et al. (2016b) randomly select 300 documents from three websites. Park and Cardie (2018) and Stab and Gurevych (2017) do not mention anything about their sampling process. In general, both source-driven corpus construction approaches inevitably incur the source’s idiosyncracies of topic selection in terms of skew towards certain topics. Scaling up may or may not be a remedy for this problem.

⁴https://www.research.ibm.com/haifa/dept/vst/debating_data.shtml

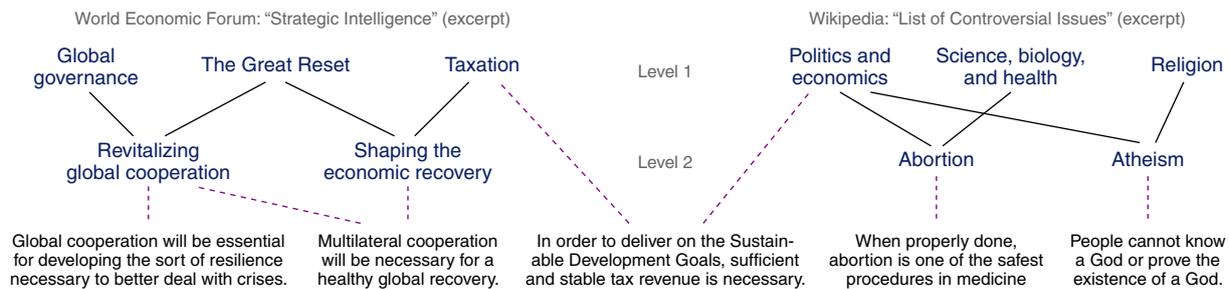


Figure 1: Example for an assignment of arguments (bottom) to topics of a two-leveled ontology. Level 2 topics are subtopics of their linked Level 1 topics. Arguments linked to a Level 2 topic also pertain to its Level 1 ancestors.

We assess how many experiments have been reported on each of the corpora by collecting the publications referring to a corpus as per Google Scholar, focusing on conference and journal papers, but excluding books and web pages. We then check whether the cited corpus is mentioned in its data, experiment, or results section. As can be seen in Table 1, corpora with fewer topics tend to be used more often in experiments than those with larger amounts. In total, 230 experiments were carried out on argument corpora with no clearly defined topic selection directive. The skew towards smaller-scale experiments may affect generalizability.

4 Bootstrapping The Argument Ontology

Topic ontologies provide for a knowledge organization principle, and, especially if widely accepted, also a standard. They are typically modeled as directed acyclic graphs, where nodes correspond to topics and edges indicate “is part of” relations. Topics that are part of other topics are called their subtopics. A topic ontology is often displayed in levels, starting with the topics that are not subtopics of others, continuing recursively with each lower level of subtopics. Figure 1 shows an excerpt of a two-level topic ontology for arguments.

The identification of the topics to be included in The Argument Ontology (TAO), as well as their relations, requires domain expertise. Building an all-encompassing ontology thus requires experts from every top-level domain where argumentation of scientific interest is expected. In the following, we suggest and outline three authoritative sources of expert topic ontologies, which comprise a wide selection of important argumentative topics. We use them to bootstrap a first version of TAO.

World Economic Forum (WEF) The World Economic Forum is a not-for-profit foundation that coordinates organizations from both the public and

the private sector to work on economical and societal issues. As part of their efforts, their “Strategic Intelligence” platform⁵ strives to inform decision makers on domestic and global topics, specifically global issues (e.g., artificial intelligence and climate change), industries (e.g., healthcare delivery and private investors), and economies (e.g., Africa and ASEAN). Domain experts for each topic curate a stream of relevant news articles which they each tag with 4-9 subtopics of their topic (e.g., the continuous monitoring of mental health).

Wikipedia Wikipedia strives for a neutral point of view, but many topics of public interest are discussed controversially. Some editors thus curate a list of controversial Wikipedia articles to highlight where special care is needed, grouped into 14 top-level topics (e.g., environment and philosophy) and 4-176 subtopics (e.g., creationism and pollution).⁶ We omit the “People” topic and articles on countries; their controversiality is not universal.

Debatepedia The Debatepedia portal’s goal is to create an encyclopedia of debates which are organized as “pro” and “con” arguments. A list of 89 topics helps visitors to browse the debates. The debates are contributed by anonymous web users. Topics in Debatepedia tend to address issues of Western culture. For example, the topic “United States” covers 306 debates while “Third World” covers only 12. The site is no longer maintained, but accessible through the Wayback Machine.⁷

The three ontologies are publicly accessible, and two of them are actively maintained and updated. Acquiring the ontologies is straightforward—not straightforward is to make use of them. A key task associated with every topic ontology is to categorize a given document. Having just a short string

⁵<https://intelligence.weforum.org>

⁶https://en.wikipedia.org/wiki/Wikipedia:List_of_controversial_issues

⁷https://web.archive.org/web/20180222051626/http://www.debatepedia.org/en/index.php/Welcome_to_Debatepedia%21

Type	Count	Example topic		
		Topic label	Normalized form	Corpus
Concept	1,394	Abortion	abortion	Claim Sentence Search
Conclusion	707	We should ban partial birth abortion	partial birth abortion	Evidence Quality
Question	110	Should abortion be prohibited?	abortion	IAM
Imperative	25	Ban abortions	abortion	Record Debating Dataset 5
Comparison	23	Pro Choice vs. Pro Life	pro choice vs pro life	UKPConvArg1

Table 2: Counts of the topic types in the 39 preprocessed corpora with examples and their normalized form.

label describing a (potentially multifaceted) topic, such as “The Great Reset”, renders this task exceedingly difficult. Fortunately, domain experts have been pre-categorizing documents into the aforementioned ontologies. In particular, regarding the WEF, invited domain experts categorize news articles for every topic, regarding Wikipedia, the text of the associated articles is available, as are the associated debates on Debatepedia.

Articles that are categorized into Level 2 topics are propagated up to their respective Level 1 topics. Table 3 shows the large differences between the ontologies. The WEF ontology contains the most topics, links the most documents, and has the most tokens overall. Wikipedia’s Level 2 topics link to a single article each, yielding less text overall.

5 Topic Coverage

To assess the topic coverage of an argument corpus given the three ontologies, we map their topic labels (if provided) to matching ontology topics.

5.1 Topic Label Normalization

Table 1 lists 39 argument corpora that provide topic labels. Altogether 2,259 different labels have been assigned. They are concise descriptions of the main issues of an argument provided by the corpus authors. The labels possess the text register of the respective corpus: In essays, for instance, topics are usually thesis statements, while Wikipedia-derived corpora use article titles, and the topics of debate corpora include clichés such as “This house should”. Often, topic labels express a stance towards a target issue, e.g., “ban guns”. Five types of topic labels can be distinguished: concept, comparison of concepts, conclusion (includes claim and thesis), question, and imperative. We normalize the topic labels by converting all concepts to singular form, removing clichés, and dropping stance-indicating words such as “legalize”. Our normalization aims at retaining only the central target issue of a topic label and leads to 798 unique topic labels.

5.2 Mapping Topic Labels to Ontology Topics

Using the preprocessed topic labels as queries, we retrieve for each topic label the 50 top-most relevant topics in each level of the three ontologies. To facilitate the retrieval of ontology topics, we employ a BM25-weighted (Robertson et al., 2004) index of the concatenated documents for each topic. This enables us to narrow down the mapping of a topic label to a manageable size. Except for a handful of cases, 50 ontology topics can be retrieved for each topic label. The topic labels were then manually mapped to an ontology topic, if they form synonyms, or if the former is a subtopic of the latter—which thus indicates that all arguments in the corpus with that topic label are about the ontology topic. A topic label can thus be mapped to multiple ontology topics. For example, the topic label “plastic bottles” is mapped to “pollution” and “recycling” in Wikipedia Level 2.

5.3 Analysis of Topic Coverage

Table 3 shows general statistics of this mapping of corpora topic labels to ontology topics. Most of the topic labels (2,141 out of 2,259) are mapped to at least one Debatepedia topic while only 395 labels are mapped to WEF Level 2 topics. For Wikipedia Level 2, only 298 out of the 748 topics are actually covered by argument corpora. This first analysis already suggests that existing argument corpora often only cover a small subset of possible argumentative topics that people are trained to debate. For those topic labels that can be mapped, they belong on average to 2.78 topics in Debatepedia, 1.24 topics in Wikipedia Level 1, and 1.53 topics in WEF Level 1. As discussed in Section 4, topics in Debatepedia focus on the Western culture and are easily accessible, whereas topics in WEF require in-depth domain knowledge and have more global relevance. The broad coverage of Debatepedia’s topics indicates that argument corpora focus on common, widely discussed topics rather than global issues or those that need domain knowledge.

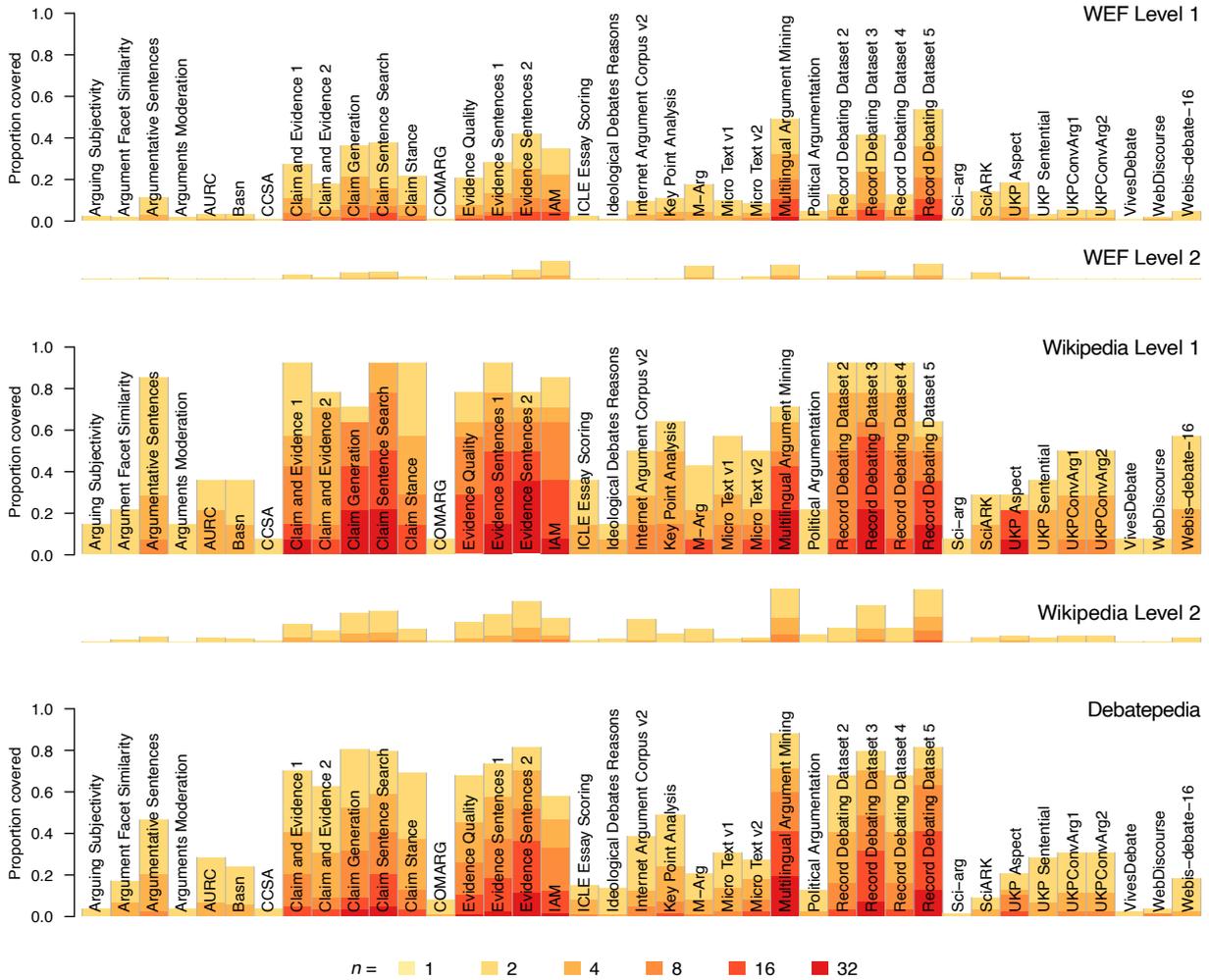


Figure 2: Proportion of ontology topics covered by at least n corpus topics (per ontology level and per corpus).

For a more fine-grained analysis, Figure 2 illustrates the differences regarding the number of ontology topics covered by a corpus: While topics in Wikipedia Level 1 are covered well by some argument corpora, topics in Wikipedia and WEF Level 2 are covered only marginally. Note that topic coverage varies significantly between the corpora: the Claim Sentence Search dataset’s topics cover 93% of the Wikipedia Level 1 topics, while the Ideological Debates Reasons dataset covers only 14%. The colors show the topic granularity of the corpus; especially the Record Debating Dataset 3 dataset is fine-grained: as the highest value, 36 of its topics are mapped to the Wikipedia Level 1 category “Politics and Economics”.

Figure 3 shows how the set of the units of the 39 labeled corpora distribute over the top-matching topics in Debatepedia, Wikipedia Level 1, and WEF Level 1. Distributions over Level 2 are omitted for brevity and can be found in Figure 4 in the Appendix. The distribution is significantly skewed:

while the top ten topics in Debatepedia are matched by 354,811 to 138,407 corpora units, the top ten topics in WEF Level 1 are matched by 344,345 to 28,725 corpora units. This supports our finding that the corpora cover easily accessible topics (e.g., “Media and Entertainment” and “Society”).

6 Unit Categorization

The previous analysis assesses argument corpora which contain topic labels. About a third of the argument corpora do not. As a heuristic step to assessing their topic coverage, we map the ontology topics for a unit (Table 1) in an argument corpus by treating the unit as a (long) query in a standard information retrieval setup, where ontology topics are the retrieval targets. The documents categorized into each topic have been concatenated and used as the topic’s representation. Though the documents associated with a topic are not necessarily argumentative, they cover the salient topic aspects.

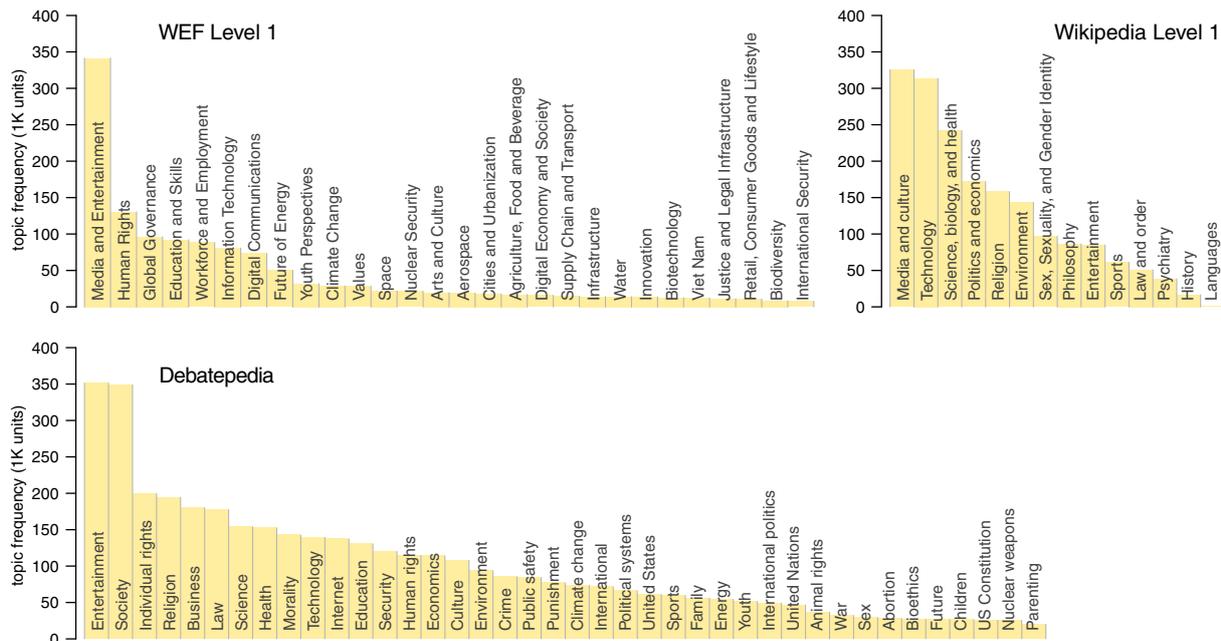


Figure 3: Distribution of corpora units over the top matching topics in an ontology (39 labeled corpora).

Ontology	Acquired ontologies (Section 4)			Topic coverage (Section 5)				Unit categorization (Section 6)												
	Topics	Topic statistics		Mapped topic labels	Covered ontology topics			Direct match			Semantic interpretation			Text2vec-SI						
		Authors	Docs		Tokens	All	Min	Mean	Max	P	R	F	Policy	P	R	F	Policy	P	R	F
WEF L1	137	334.1	940.7	490,576.6	1,339	92	1	1.53	13	0.38	0.23	0.29	$k = 12$	0.22	0.75	0.34	$k = 7$	0.19	0.53	0.28
WEF L2	822	216.8	550.3	310,229.7	395	154	1	1.56	22	0.59	0.11	0.19	$k = 30$	0.21	0.68	0.32	$\theta = 0.93$	0.15	0.49	0.23
WP L1	14	78,013.7	68.0	339,088.0	1,647	14	1	1.24	3	0.12	0.04	0.06	$k = 3$	0.32	0.65	0.43	$k = 2$	0.41	0.55	0.47
WP L2	748	1,929.5	1.0	6,149.1	1,560	298	1	1.80	16	0.47	0.34	0.40	$\theta = 0.05$	0.54	0.64	0.59	$\theta = 0.89$	0.22	0.52	0.31
DP	89	145.0	61.7	84,787.6	2,141	88	1	2.78	10	0.49	0.37	0.42	$\theta = 0.02$	0.52	0.61	0.56	$k = 23$	0.36	0.80	0.50

Table 3: Statistics for each topic ontology level: for topics and topic documents (Section 4), Count of mapped topic labels of the analyzed corpora for each ontology level, Count of all covered ontology topics by the topic labels and the min, max, and mean count of covered ontology topics per topic label (Section 5), and the effectiveness of the approaches and baseline in unit categorization (in terms of precision, recall, and F_1 -score) (Section 6).

To retrieve topics for a corpus unit, we implement and evaluate the following approaches: Semantic Interpretation (SI) and SI with Text Embeddings (Text2vec-SI). The Semantic interpretation approach computes the semantic similarity of a unit and a topic as follows: it uses the cosine similarity of the TF-IDF vectors for the unit and the concatenated topic’s documents. This corresponds to the semantic interpretation step that is at the core of the well-known ESA model (Gabrilovich and Markovitch, 2007). Text2vec-SI calculates the similarity of topics and corpus units using BERT embeddings (Devlin et al., 2018). Following common practice, we take the dimension-wise average of the word embeddings for all tokens in the text.⁸ We tried other embeddings and approaches that performed similarly. The results of these approaches

⁸For efficiency, we limited the embeddings to 10,000 randomly sampled sentences for the topics that had more sentences associated with them.

can be found in the appendix. As a baseline, we implement a direct match approach, which assigns a unit an ontology topic if the topic’s text appears in the unit text (ignoring case).

For evaluation, we collect 34,638 pooled query relevance judgments (0.53 inter-annotator agreement as per Krippendorff’s α) on 104 randomly selected argument units as queries from 26 corpora. The annotation process is detailed in the Appendix.

Based on the similarity scores of the approaches, we derive Boolean labels that indicate whether a unit is or is not about one of the ontologies’ topics using two policies. The *threshold* policy labels a unit as about a topic if their similarity is above a threshold θ . The *top-k* policy labels a unit as about a topic if the topic is among the top- k topics with the highest similarity to the unit. We report the parameter of the policy that achieved the highest F_1 -score on the pooled judgments for each approach.

Table 3 shows the results of this evaluation. The baseline produces different results across ontologies—it performs poorly for both the abstract topics in Wikipedia Level 1 and the specific topics in WEF Level 2. The semantic interpretation approach clearly outperforms the baseline for all ontologies in terms of the F_1 -score. The Text2vec-SI approach outperforms the baseline and the semantic interpretation on abstract topics (Wikipedia Level 1), but its effectiveness is below that of the semantic interpretation approach on the other ontology levels.

7 Conclusion

The computational argumentation community risks topic bias in its approaches if the representativeness of topics in future corpora is not ensured. Achieving topic coverage is complicated by the fact that the landscape of controversial topics has not yet been well explored, and that there are no widely accepted ontologies for argument topics. In this paper, we venture into this future by mapping the landscape of argument topics and making it accessible for corpus construction and experimental design. We have identified three authoritative sources of ontological knowledge related to argument topics that provide an initial foundation for The Argument Ontology (TAO). For each source ontology, we evaluate the topic coverage of 39 argument corpora labeled with topics by matching the labels with the topics of the ontologies. To evaluate the topic coverage of corpora without topic labels, we develop an approach to identify the ontology topics of an argumentative text and achieve an F_1 of 0.59.

Our analyses show that the topic coverage of existing argument corpora is both limited to a subset of the topics of the ontologies and skewed. Most topics that require expertise, such as mental health, philosophy, or international security, are treated only peripherally in argumentation corpora. Therefore, existing argumentation technologies are more suited to teaching people how to construct arguments in general than to helping them make decisions about such and similarly complex topics. For the development of robust argumentation technologies, corpora need to be carefully drawn from a specific domain to allow for reliable experiments and the development of generalizable classifiers.

Future work for further development of TAO consists mainly of further surveying the argument topic landscape and unifying the various available

ontologies. In addition to “is part of” relationships between topics, other relationship types can also be considered to build an argument topic knowledge base. However, our first version of TAO and our analyses can already help in selecting arguments for future corpus construction and model training.

Limitations

The three topic ontologies we used to evaluate topic coverage of argument corpora are from authoritative sources. Nevertheless, they probably do not cover all possible controversial topics relevant to argumentation (e.g., topics concerning private life). A comprehensive coverage of controversial topics in breadth and depth will likely remain an unattainable goal. Moreover, unifying the three thematic ontologies into a standard ontology is still an open problem given the many possible interpretations and relationships between the topics.

Another limitation is the moderate effectiveness achieved by our approaches for categorizing argument units. This is the case due to the large collection of controversial topics (about 748 for Wikipedia). Future research can be improved by using the structure of the topic ontology and hierarchical classifiers. Furthermore, it is also unclear whether the topic dependence of argumentation approaches decreases with increasing corpus size.

Ethics Statement

Our goal is to investigate whether and to what extent existing argumentation corpora are topic biased. This serves to critically examine the state of the art. However, we by no means want to give the impression that previous corpus authors lack ambition or diligence. Rather, the opposite is the case. The number of corpora that have been created in the last decade shows that the community is aware of the fact that not all areas of the argumentation landscape have been covered yet, and is therefore doing its utmost to explore it further. In a dynamic and rapidly growing research field, standards are usually developed in parallel with contributions, not in advance. Our research may therefore contribute to the further standardization of the corpus linguistics of argumentation.

The manual annotation of arguments and topics was done by expert annotators of our research groups. They were compensated fairly under German law. No personal data was collected.

References

- Rob Abbott, Brian Ecker, Pranav Anand, and Marilyn Walker. 2016. [Internet Argument Corpus 2.0: An SQL Schema for Dialogic Social Media and the Corpora to go with it](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4445–4452. European Language Resources Association (ELRA).
- Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. 2014. A Benchmark Dataset for Automatic Detection of Claims and Evidence in the Context of Controversial Topics. In *Proceedings of the 2014 Workshop on Argumentation Mining (ArgMining 2014)*, pages 64–68. Association for Computational Linguistics.
- Yamen Ajjour, Milad Alshomary, Henning Wachsmuth, and Benno Stein. 2019a. Modeling Frames in Argumentation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP 2019)*. ACL.
- Yamen Ajjour, Henning Wachsmuth, Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. 2019b. Data Acquisition for Argument Search: The args.me corpus. In *42nd German Conference on Artificial Intelligence (KI 2019)*. Springer.
- Khalid Al-Khatib, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein. 2016. [Cross-Domain Mining of Argumentative Text through Distant Supervision](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*, pages 1395–1404. Association for Computational Linguistics.
- Khalid Al-Khatib, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein. 2016a. [Cross-Domain Mining of Argumentative Text through Distant Supervision](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*, pages 1395–1404. Association for Computational Linguistics.
- Khalid Al-Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. 2016b. [A News Editorial Corpus for Mining Argumentation Strategies](#). In *26th International Conference on Computational Linguistics (COLING 2016)*, pages 3433–3443. Association for Computational Linguistics.
- Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. [Stance Classification of Context-Dependent Claims](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 251–261. Association for Computational Linguistics.
- Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020. [From arguments to key points: Towards automatic argument summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2021)*, pages 4029–4039. Association for Computational Linguistics.
- Roy Bar-Haim, Dalia Krieger, Orith Toledo-Ronen, Lilach Edelstein, Yonatan Bilu, Alon Halfon, Yoav Katz, Amir Menczel, Ranit Aharonov, and Noam Slonim. 2019. [From Surrogacy to Adoption; From Bitcoin to Cryptocurrency: Debate Topic Expansion](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 977–990. Association for Computational Linguistics.
- Floris Bex, John Lawrence, Mark Snaith, and Chris Reed. 2013. Implementing the Argument Web. *Communications of the ACM*, 56:66–73. Crawled in Jan, 2020.
- Yonatan Bilu, Ariel Gera, Danel Hershcovich, Benjamin Sznajder, Dan Lahav, Guy Moshkovich, Anael Malet, Assaf Gavron, and Noam Slonim. 2019. [Argument Invention from First Principles](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 1013–1026. Association for Computational Linguistics.
- Filip Boltužić and Jan Šnajder. 2014. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58. The Association for Computational Linguistics.
- Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. 2019. [IMHO fine-tuning improves claim detection](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 558–563.
- Liyong Cheng, Lidong Bing, Ruidan He, Qian Yu, Yan Zhang, and Luo Si. 2022. [Iam: A comprehensive and large-scale dataset for integrated argument mining tasks](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 2277–2287.
- Liyong Cheng, Lidong Bing, Qian Yu, Wei Lu, and Luo Si. 2020. [Ape: Argument pair extraction from peer review and rebuttal via multi-task learning](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7000–7011.
- Alexander Conard, Janyce Wiebe, and Rebecca Hwa. 2012. Recognizing Arguing Subjectivity and Argument Tags. In *Proceedings of the Workshop on Extrapositional Aspects of Meaning in Computational Linguistics (ExProM 2012)*, pages 80–88.

- Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. 2017. [What is the Essence of a Claim? Cross-Domain Claim Identification](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 2045–2056. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Liat Ein-Dor, Eyal Shnarch, Lena Dankin, Alon Halfon, Benjamin Sznajder, Ariel Gera, Carlos Alzate, Martin Gleize, Leshem Choshen, Yufang Hou, Yonatan Bilu, Ranit Aharonov, and Noam Slonim. 2020. [Corpus wide argument mining - A working solution](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7683–7691.
- Shnarch Eyal, Leshem Choshen, Guy Moshkovich, Ranit Aharonov, and Noam Slonim. 2020. [Unsupervised expressive rules provide explainability and assist human experts grasping new domains](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2678–2697. Association for Computational Linguistics.
- Neele Falk, Iman Jundi, Eva Maria Vecchi, and Gabriella Lapesa. 2021. [Predicting moderation of deliberative arguments: Is argument quality the key?](#) In *Proceedings of the 8th Workshop on Argument Mining*, pages 133–141.
- Aris Fergadis, Dimitris Pappas, Antonia Karamolegkou, and Harris Papageorgiou. 2021. [Argumentation mining in scientific literature for sustainable development](#). In *Proceedings of the 8th Workshop on Argument Mining*, pages 100–111.
- Evgeniy Gabrilovich and Shaul Markovitch. 2007. [Computing semantic relatedness using wikipedia-based explicit semantic analysis](#). In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 1606–1611.
- Yingqiang Gao, Nianlong Gu, Jessica Lam, and Richard H.R. Hahnloser. 2022. [Do discourse indicators reflect the main arguments in scientific papers?](#) In *Proceedings of the 9th Workshop on Argument Mining*, pages 34–50.
- Martin Gleize, Eyal Shnarch, Leshem Choshen, Lena Dankin, Guy Moshkovich, Ranit Aharonov, and Noam Slonim. 2019. [Are You Convinced? Choosing the More Convincing Evidence with a Siamese Network](#). In *Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 967–976.
- Alex Graves and Jürgen Schmidhuber. 2005. [Frame-wise phoneme classification with bidirectional lstm and other neural network architectures](#). *Neural networks: the official journal of the International Neural Network Society*, 18:602–10.
- Shai Gretz, Yonatan Bilu, Edo Cohen-Karlik, and Noam Slonim. 2020. [The workweek is the best time to start a family – a study of GPT-2 based claim generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 528–544, Online. Association for Computational Linguistics.
- Thomas R. Gruber. 1993. [A translation approach to portable ontology specifications](#). *Knowledge Acquisition*, 5:199–220.
- Giulia Grundler, Piera Santin, Andrea Galassi, Federico Galli, Francesco Godano, Francesca Lagioia, Elena Palmieri, Federico Ruggeri, Giovanni Sartor, and Paolo Torroni. 2022. [Detecting arguments in cjeu decisions on fiscal state aid](#). In *Proceedings of the 9th Workshop on Argument Mining*, pages 143–157.
- Yunfan Gu, Yhongyu Wei, Maoran Xu, Hao Fu, Yang Liu, and Xuanjing Huang. 2018. [Incorporating Topic Aspects for Online Comment Convincingness Evaluation](#). In *Proceedings of the 5th Workshop on Argument Mining (ArgMining 2018)*, pages 97–104. Association for Computational Linguistics.
- Ivan Habernal and Iryna Gurevych. 2016a. [Argumentation Mining in User-Generated Web Discourse](#). *Computational Linguistics*.
- Ivan Habernal and Iryna Gurevych. 2016b. [What makes a convincing argument? Empirical analysis and detecting attributes of convincingness in web argumentation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223. Association for Computational Linguistics.
- Ivan Habernal and Iryna Gurevych. 2016c. [Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1589–1599. Association for Computational Linguistics.
- Shohreh Haddadan, Elena Cabrio, and Serena Villata. 2019. [Yes, we can! Mining Arguments in 50 Years of US Presidential Campaign Debates](#). In *Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 4684–4690.
- Kazi Saidul Hasan and Vincent Ng. 2014. [Why are You Taking this Stance? Identifying and Classifying Reasons in Ideological Debates](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 751–762.

- Annette Hautli-Janisz, Zlata Kikteva, Wassiliki Siskou, Kamila Gorska, Ray Becker, and Chris Reed. 2022. [Qt30: A corpus of argument and conflict in broadcast debate](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3291–3300.
- Birger Hjørland. 2001. [Towards a theory of aboutness, subject, topicality, theme, domain, field, content ... and relevance](#). *Journal of the American Society for Information Science and Technology*, 52(9):774–778.
- Kialo. 2020. Kialo. www.kialo.com. Crawled in Jan, 2020.
- Takahiro Kondo, Koki Washio, Katsuhiko Hayashi, and Yusuke Miyao. 2021. [Bayesian argumentation-scheme networks: A probabilistic model of argument validity facilitated by argumentation schemes](#). In *Proceedings of the 8th Workshop on Argument Mining*, pages 112–124.
- Anne Lauscher, Goran Glavaš, and Simone Paolo Ponzetto. 2018. [An argument-annotated corpus of scientific publications](#). In *Proceedings of the 5th Workshop on Argument Mining*, pages 40–46. Association for Computational Linguistics.
- Tamar Lavee, Matan Orbach, Lili Kotlerman, Yoav Kantor, Shai Gretz, Lena Dankin, Michal Jacovi, Yonatan Bilu, Ranit Aharonov, and Noam Slonim. 2019. [Towards Effective Rebuttal: Listening Comprehension using Corpus-Wide Claim Mining](#). In *Proceedings of the Fourth Workshop on Argument Mining 2017 (ArgMining 2017)*, pages 719–724.
- Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. [Context dependent claim detection](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500.
- Ran Levy, Ben Bogin and Shai Gretz, Ranit Aharonov, and Noam Slonim. 2018. [Towards an argumentative content search engine using weak supervision](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2066–2081.
- Yinzi Li, Wei Chen, Zhongyu Wei, Yujun Huang, Chujun Wang, Siyuan Wang, Qi Zhang, Xuanjing Huang, and Libo Wu. 2022. [A structure-aware argument encoder for literature discourse analysis](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7093–7098.
- Marco Lippi and Paolo Torroni. 2016. [Argument Mining from Speech: Detecting Claims in Political Debates](#). In *Proceedings of the 2016 Association for the Advancement of Artificial Intelligence (AAAI 2016)*, pages 2979–2985.
- Siyi Liu, Sihao Chen, Xander Uyttendaele, and Dan Roth. 2021. [Multioped: A corpus of multi-perspective news editorials](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4345–4361.
- Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2018. [Never Retreat, Never Retract: Argumentation Analysis for Political Speeches](#). In *Proceedings of the Thirty-second Association for the Advancement of Artificial Intelligence (AAAI) Conference of Artificial Intelligence*, pages 4889–4896. AAAI Press.
- Rafael Mestre, Razvan Milicin, Stuart E. Middleton, Matt Ryan, Jiatong Zhu, and Timothy J Norman. 2021. [M-arg: Multimodal argument mining dataset for political debates with audio and transcripts](#). In *Proceedings of the 8th Workshop on Argument Mining*, pages 78–88.
- Shachar Mirkin, Guy Moshkovich, Matan Orbach, Lili Kotlerman, Yoav Kantor, Tamar Lavee, Michal Jacovi, Yonatan Bilu, Ranit Aharonov, and Noam Slonim. 2018. [Listening Comprehension over Argumentative Content](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 719–724.
- Amita Misra, Brian Ecker, and Marilyn Walker. 2016. [Measuring the similarity of sentential arguments in dialogue](#). In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287, Los Angeles. Association for Computational Linguistics.
- Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. [Automatic Detection of Arguments in Legal Texts](#). In *Proceedings of the 11th International conference on Artificial Intelligence and Law (ICAIL 2007)*, pages 225–230. Association for Computational Machinery.
- Lily Ng, Anne Lauscher, Joel Tetreault, and Courtney Napoles. 2020. [Creating a domain-diverse corpus for theory-based argument quality assessment](#). In *Proceedings of the 7th Workshop on Argument Mining*, pages 117–126, Online. Association for Computational Linguistics.
- Matan Orbach, Yonatan Bilu, Ariel Gera, Yoav Kantor, Lena Dankin, Tamar Lavee, Lili Kotlerman, Shachar Mirkin, Michal Jacovi, Ranit Aharonov, and Noam Slonim. 2019. [A dataset of general-purpose rebuttal](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5591–5601, Hong Kong, China. Association for Computational Linguistics.
- Matan Orbach, Yonatan Bilu, Assaf Toledo, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. 2020. [Out of the echo chamber: Detecting countering debate speeches](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7073–7086, Online. Association for Computational Linguistics.

- Joonsuk Park and Claire Cardie. 2018. A Corpus of e-Rulemaking User Comments for Measuring Evaluability of Arguments. In *Proceedings of the 2018 International Conference on Language Resources and Evaluation (LREC 2018)*.
- Andreas Peldszus and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *Proceedings of the 2015 European Conference on Argumentation: Argumentation and Reasoned Action (ECA 2015)*.
- Isaac Persing, Alan Davis, and Vincent Ng. 2010. Modeling Organization in Student Essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239.
- Isaac Persing and Vincent Ng. 2017. Lightly-Supervised Modeling of Argument Persuasiveness. In *Proceedings of 2017 International Joint Conference on Natural Language Processing (IJCNLP 2017)*, pages 594–604.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 2227–2237. ACL.
- Prakash Poudyal, Jaromir Savelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. [ECHR: Legal corpus for argument mining](#). In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75, Online. Association for Computational Linguistics.
- Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and Clustering of Arguments with Contextualized Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 567–578. Association for Computational Linguistics.
- Myrthe Reuver, Suzan Verberne, Roser Morante, and Antske Fokkens. 2021. Is Stance Detection Topic-Independent and Cross-topic Generalizable? – A Reproduction Study. In *Proceedings of the 2021 Workshop on Argumentation Mining (ArgMining 2021)*.
- Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show Me Your Evidence - an Automatic Method for Context Dependent Evidence Detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 719–724.
- Stephen E. Robertson, Hugo Zaragoza, and Michael J. Taylor. 2004. [Simple BM25 extension to multiple weighted fields](#). In *Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13, 2004*, pages 42–49. ACM.
- Allen Roush and Arvind Balaji. 2020. [Debatesum: A large-scale argument mining and summarization dataset](#). In *Proceedings of the 7th Workshop on Argument Mining*, pages 1–7, Online. Association for Computational Linguistics.
- Ramon Ruiz-Dolz, Montserrat Nofre, Mariona Taulé, Stella Heras, and Ana García-Fornes. 2021. Vivesdebate: A new annotated multilingual corpus of argumentation in a debate tournament. *Applied Sciences*, 11(15):7160.
- Eyal Schnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it Blend? Blending Weak and Strong Labeled Data in a Neural Network for Argumentation Mining. In *Proceedings of the 2018 Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 599–605.
- Carlos Silla and Alex Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22:31–72.
- Maria Skeppstedt, Andreas Peldszus, and Manfred S Stede. 2018. More or less controlled elicitation of argumentative text: Enlarging a microtext corpus via crowdsourcing. In *Proceedings of the 5th Workshop on Argument Mining 2017 (ArgMining 2017)*, pages 155–163. Association for Computational Linguistics.
- Yangqiu Song and Dan Roth. 2015. [Unsupervised sparse vector densification for short text similarity](#). In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1275–1280.
- Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. ArguementText: Searching for Arguments in Heterogeneous Sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 21–25.
- Christian Stab and Iryna Gurevych. 2017. Parsing Argumentation Structure in Persuasive Essays. *Computational Linguistics*, 43(3):619–659.
- Aixin Sun and Ee-Pen Lim. 2001. Hierarchical Text Classification and Evaluation. In *Proceedings of the 2001 Institute of Electrical and Electronics Engineer (IEEE) International Conference on Data Mining (ICDM 2001)*, pages 521–528. Association for Computational Linguistics.

- Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. [Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions](#). In *Proceedings of the 25th International Conference on World Wide Web (WWW 2016)*, pages 613–624. International World Wide Web Conferences Steering Committee.
- Orith Toledo-Ronen, Matan Orbach, Yonatan Bilu, Artem Spector, and Noam Slonim. 2020. Multilingual argument mining: Datasets and analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 303–317. Association for Computational Linguistics.
- Stephen Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.
- Dietrich Trautmann, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych. 2020. [Fine-grained argument unit recognition and classification](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9048–9056.
- Frans H. van Eemeren, editor. 2015. *Reasonableness and Effectiveness in Argumentative Discourse*, volume 27 of *Argumentation Library*. Springer.
- Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. 2017. Computational Argumentation Quality Assessment in Natural Language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EMNLP 2017)*, pages 176–187.
- Eduardo Xamena, Nélide Beatriz Brignole, and Ana Gabriela Maguitman. 2017. [A structural analysis of topic ontologies](#). *Information Science*, 421:15–29.
- Stephen Yablo. 2014. *Aboutness*. Princeton University Press.
- Justine Zhang, Ravi Kumar, Sujith Ravi, and Cristian Danescu-Niculescu-Mizil. 2016. Conversational Flow in Oxford-style Debates. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*.

A Appendix

A.1 Mapping Topic Labels to Level 2 Topics

For completeness, Figure 4 shows the two graphs that are omitted from Figure 3 of the paper as their fine-grained topics are less relevant for the discussion in Section 5.3.

A.2 Annotation Procedure for Unit Categorization

In order to assess the effectiveness of the approaches and baseline outlined in the paper, we employ a pooled evaluation, as it is standard for information retrieval evaluations, where there are too many instances for a complete manual annotation. We randomly sampled four units from 26 corpora, which were all annotated by three expert annotators. The annotators were instructed to label a topic as about the unit if they could imagine a discussion on the topic for which the unit would be relevant. For each unit, we annotated for aboutness only those topics which are among the five topics with the highest similarity to this unit according to at least one of the approaches. The employed assessment interface (see Figure 5) shows the unit (top left), the current topic (top right), as well as all topics in the pool for that unit (bottom; the current topic is marked blue, whereas already annotated topics are marked green (about) and red (not about)). The same interface has been used for the topic label annotations.

To reduce biases, both the units and the topics were shown in a different and random order to each assessor. The annotation took about 40 hours. The annotation process resulted in an inter-annotator agreement of 0.53 in terms of Krippendorff’s α and produced a total of 34,638 annotations of topic-unit pairs, about 2% of what would have been needed for a complete annotation.

A.3 Additional Unit Categorization Approaches

In addition to the approaches listed in Section 6 we used additional approaches and a baseline which we list here. The additional baseline randomly classifies corpora units as per the prior topic probability of each ontology level.

SI with Word Embeddings (W2V-SI) Adapted from Dense-ESA (Song and Roth, 2015), this approach represents each token by its TFIDF-weighted Word2vec embedding vector, and uses the highest cosine similarity between two vectors, one from each text, as the semantic similarity. To limit this quadratic effort, we use only the 100 tokens of each text with the highest TFIDF-score.

Text2vec-SI As a variant for BERT (Devlin et al., 2018), we embedded the ontology documents and corpora units using ELMo (Peters et al., 2018).

(a) Debatepedia		(b) Wikipedia Level 2	
Topic	Covering units	Topic	Covering units
Islam and the West	50	Irredentism	2
Islam	50	American Civil Liberties Union	2
2008/2009 economic crisis	66	Hezbollah	2
European Union	93	Esports	3
Middle East	134	Separation of church and state	4
Prison	180	Birth defect	5
China	254	Quebec	5
Terrorism	521	Rape	6
Latin America	528	Hurricane Katrina	6
HIV/AIDS	579	Crime in the United States	6
Church and state	654	Sexual abuse	6
US legislation	845	Sex offender	6
Corruption	1,065	Pacifism	7
Welfare	1,369	Cyberstalking	7
Africa	1,435	Brexit	9
Israeli-Palestinian conflict	1,541	Economy of Japan	12
Life and death	1,598	USA PATRIOT Act	12
Languages	1,808	Playboy Magazine	15
Privacy	2,312	Super Bowl XXXVIII	19
Bush administration	2,702	Sexual harassment	20
Iraq	2,720	Media bias	29
Weapons proliferation	2,790	Culture war	35
Third world	3,208	Hip hop culture	35
Taxes	3,562	European culture	35
Disease	3,619	Anime	40
Obama administration	3,765	East Germany	46
Conflict	4,950	Communist state	46
Asia	5,016	Communist Party of China	46
Immigration	5,170	Communist government	46
Race	6,086	Communism	46

(c) World Economic Forum Level 1		(d) World Economic Forum Level 2	
Topic	Covering units	Topic	Covering units
Agile Governance	1	Healthcare Human Capital	2
Institutional Investors	1	Environmentally-Sustainable Consumerism	2
Digital Identity	7	Sustainable Consumption	3
United Kingdom	9	Aquaculture	4
Mexico	12	Urbanization and Circular Practices	5
Behavioural Sciences	15	Accelerating Sustainability	5
Canada	26	Forest Landscape Restoration	5
Corruption	31	Stabilizing Economies, Keeping Protections	10
Illicit Economy	54	The Social Cost of Carbon	13
Future of Economic Progress	66	The Trump Presidency	20
Forests	74	New Leadership	20
European Union	93	Canada and Sustainable Energy	21
Real Estate	132	Economic Institutions	34
Insurance and Asset Management	142	Outbound and Long-Term Investment	34
Humanitarian Action	232	Deepening Interdependence	34
3D Printing	254	Digital Trade	34
China	258	Geopolitical and Geo-economic Recalibration	34
Drones	260	Pricing Climate into Finance	34
Internet Governance	268	Trade and Investment	34
Cybersecurity	298	Trade and the Environment	34
Internet of Things	320	Transnational Actors	34
Precision Medicine	339	Economic Integration	34
Oceans	345	Healthcare Technology	47
Latin America	516	Geo-strategic Competition	54
Financial and Monetary Systems	608	Energy-Related Emission Reduction	61
Arctic	614	Energy Finance and Investment	61
Banking and Capital Markets	634	Energy Access	61
Mining and Metals	656	Environmental Footprint	61
Public Finance and Social Protection	778	Electricity Decentralization	61
Middle East and North Africa	928	Electricity System Integration	61

Table 4: For each ontology except Wikipedia Level 1 the 30 topics with the least (but at least 1) units from the argument corpora covering them. All 14 topics of Wikipedia Level 1 are covered well and thus omitted here.

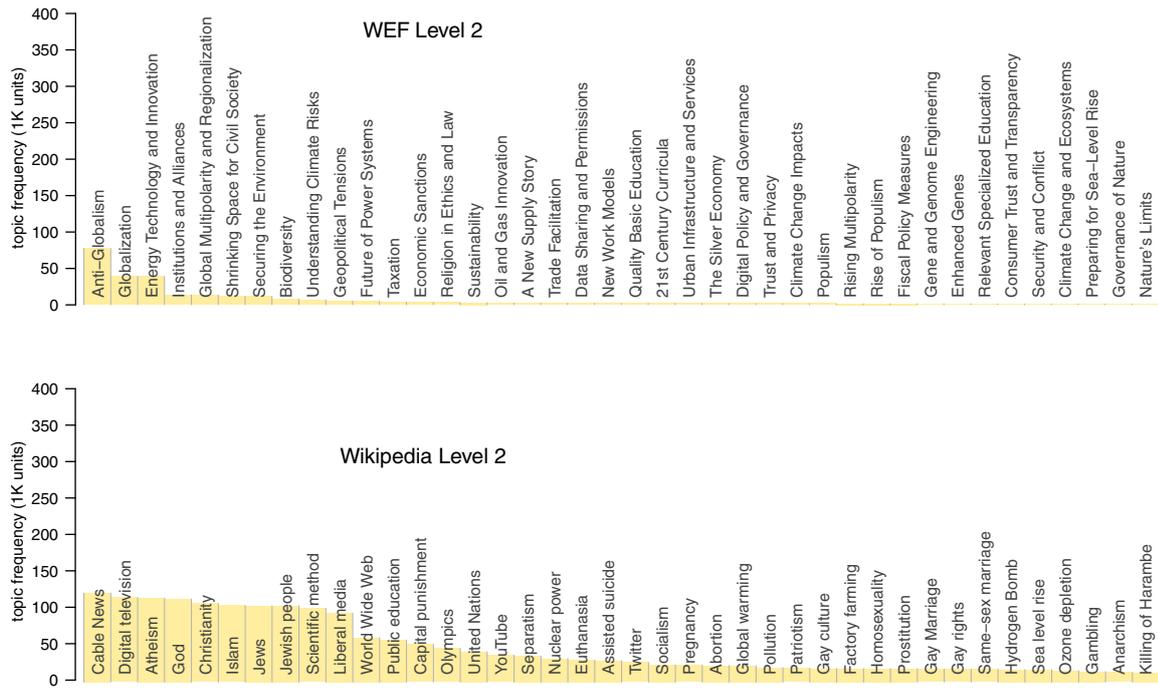


Figure 4: Omitted graphs from Figure 3, Section 5.3

Table 5 lists the results of all approaches for all thresholds and ranks.

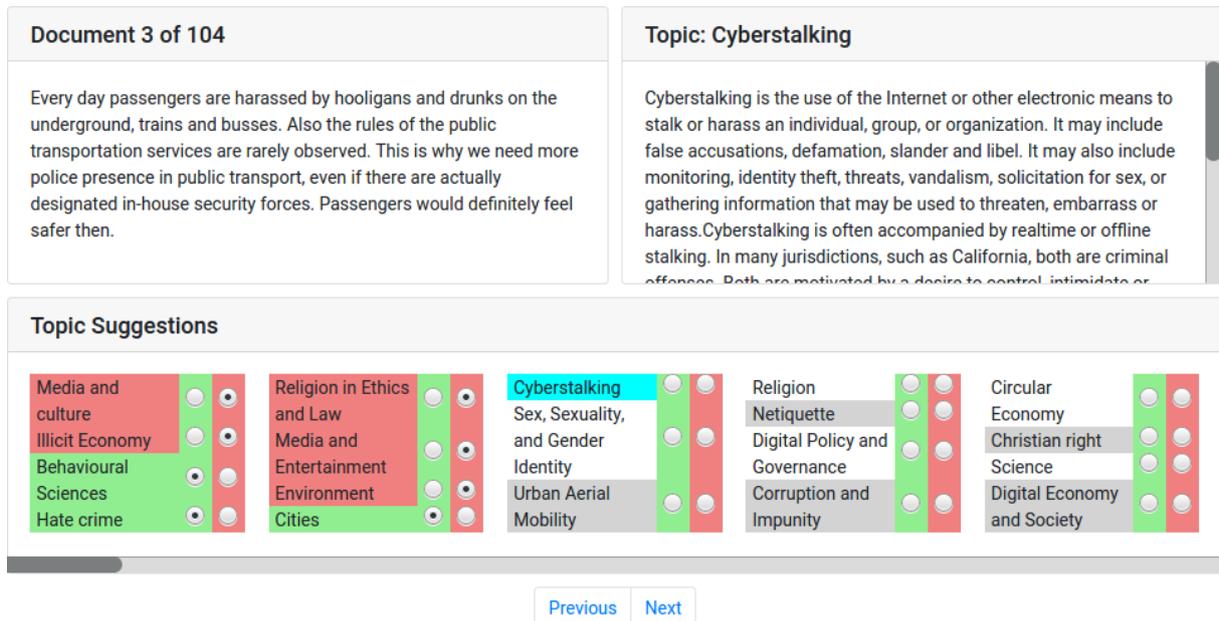


Figure 5: Assessment interface for topic labeling.

Approach	World Economic Forum								Wikipedia								Debatepedia			
	Level 1				Level 2				Level 1				Level 2							
Baselines	P	R	F		P	R	F		P	R	F		P	R	F		P	R	F	
Random	0.02	0.02	0.02		0.01	0.01	0.01		0.11	0.11	0.11		0.01	0.01	0.01		0.07	0.07	0.07	
Direct match	0.38	0.23	0.29		0.59	0.11	0.19		0.12	0.04	0.06		0.47	0.34	0.40		0.49	0.37	0.42	
By threshold	θ	P	R	F	θ	P	R	F	θ	P	R	F	θ	P	R	F	θ	P	R	F
Semantic interpretation	0.02	0.18	0.63	0.28	0.02	0.20	0.58	0.29	0.01	0.29	0.51	0.37	0.05	0.54	0.64	0.59	0.02	0.52	0.61	0.56
W2V-SI	0.20	0.14	0.63	0.23	0.16	0.13	0.63	0.21	0.10	0.12	0.55	0.20	0.11	0.18	0.69	0.29	0.07	0.29	0.81	0.43
Text2vec-SI _{ELMo}	0.87	0.18	0.32	0.23	0.80	0.13	0.65	0.22	0.74	0.23	0.45	0.31	0.76	0.25	0.45	0.32	0.87	0.47	0.46	0.47
Text2vec-SI _{BERT}	0.94	0.19	0.39	0.25	0.93	0.15	0.49	0.23	0.92	0.36	0.22	0.27	0.89	0.22	0.52	0.31	0.92	0.36	0.64	0.46
By rank	k	P	R	F	k	P	R	F	k	P	R	F	k	P	R	F	k	P	R	F
Semantic interpretation	12	0.22	0.75	0.34	30	0.21	0.68	0.32	3	0.32	0.65	0.43	12	0.39	0.70	0.50	19	0.43	0.78	0.56
W2V-SI	83	0.13	0.94	0.24	439	0.12	0.77	0.21	14	0.11	1.00	0.20	290	0.16	0.77	0.27	61	0.27	0.86	0.42
Text2vec-SI _{ELMo}	4	0.25	0.44	0.32	42	0.13	0.68	0.23	2	0.39	0.53	0.45	46	0.18	0.64	0.28	13	0.43	0.71	0.54
Text2vec-SI _{BERT}	7	0.19	0.53	0.28	80	0.11	0.66	0.20	2	0.41	0.55	0.47	80	0.17	0.70	0.28	23	0.36	0.80	0.50

Table 5: Performance of semantic interpretation approaches in human evaluation for each topic ontology level in terms of precision (P), recall (R), and F_1 -score (F) for the “aboutness” label. For methods other than the baselines the table shows the values for both the similarity threshold θ and rank k that lead to the highest F_1 -score respectively. The best F_1 -scores for each ontology level are marked bold.

LongtoNotes: OntoNotes with Longer Coreference Chains

Kumar Shridhar[†] Nicholas Monath^{*‡} Raghuveer Thirukovalluru[◇]
Alessandro Stolfo[†] Manzil Zaheer[¶] Andrew McCallum[‡] Mrinmaya Sachan[†]

[†] ETH Zürich [‡] UMass Amherst [◇] Duke University [¶] Google

Abstract

Ontonotes has served as the most important benchmark for coreference resolution. However, for ease of annotation, long documents in Ontonotes were split into smaller parts. In this work, we build a corpus of coreference-annotated documents of significantly longer length than what is currently available. We do so by providing an accurate, manually-curated, merging of annotations from documents that were split into multiple parts in the original Ontonotes annotation process (Pradhan et al., 2013). The resulting corpus, which we call LongtoNotes contains documents in multiple genres of the English language with varying lengths, the longest of which are up to 8x the length of documents in Ontonotes, and 2x those in Litbank. We evaluate state-of-the-art neural coreference systems on this new corpus, analyze the relationships between model architectures/hyperparameters and document length on performance and efficiency of the models, and demonstrate areas of improvement in long-document coreference modelling revealed by our new corpus. Our data and code is available at: <https://github.com/kumar-shridhar/LongtoNotes>.

1 Introduction

Coreference resolution is an important problem in discourse with applications in knowledge-base construction (Luan et al., 2018), question-answering (Reddy et al., 2019) and reading assistants (Azab et al., 2013; Head et al., 2021). In many such settings, the documents of interest, are significantly longer and/or on wider varieties of domains than the currently available corpora with coreference annotation (Pradhan et al., 2013; Bamman et al., 2019; Mohan and Li, 2019; Cohen et al., 2017).

The Ontonotes corpus (Pradhan et al., 2013) is perhaps the most widely used benchmark for coreference (Lee et al., 2013a; Durrett and Klein, 2013;

^{*}Now at Google.

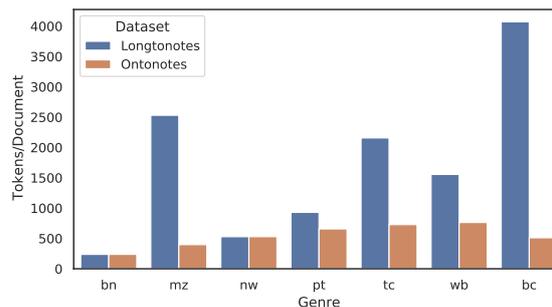


Figure 1: **Comparing Average Document Length.** Long documents in genres such as *broadcast conversations* (*bc*) were split into smaller parts in Ontonotes. Our proposed dataset, LongtoNotes, restores documents to their original form, revealing dramatic increases in length in certain genres.

Sachan et al., 2015; Wiseman et al., 2016; Lee et al., 2017; Joshi et al., 2020; Toshniwal et al., 2020b; Thirukovalluru et al., 2021; Kirstain et al., 2021). The construction process for Ontonotes, however, resulted in documents with an artificially reduced length. For ease of annotation, longer documents were split into smaller parts and each part was annotated separately and treated as an independent document (Pradhan et al., 2013). The result is a corpus in which certain genres, such as *broadcast conversation* (*bc*), have greatly reduced length compared to their original form (Figure 1). As a result, the long, bursty spread of coreference chains in these documents is missing from the evaluation benchmark.

In this work, we present an extension to the Ontonotes corpus, called LongtoNotes. LongtoNotes combines coreference annotations in various parts of the same document, leading to a full document coreference annotation. A carefully trained annotation team merged coreference annotation following the annotation guidelines laid out in the original Ontonotes corpus (§3). The resulting LongtoNotes dataset has an average document length that is over 40% longer than

the standard OntoNotes benchmark. Furthermore, LongtoNotes sees a 25% increase in the average size of coreference chains. While other datasets such as Litbank (Bamman et al., 2019) and CRAFT (Cohen et al., 2017) focus on long documents in specialized domains, LongtoNotes comprises of documents in multiple genres (Table 2).

To illustrate the usefulness of LongtoNotes, we evaluate state-of-the-art coreference resolution models (Kirstain et al., 2021; Toshniwal et al., 2020b; Joshi et al., 2020) on the corpus and analyze the performance in terms of document length (§4.2). We illustrate how model architecture decisions and hyperparameters that support long-range dependencies have the greatest impact on coreference performance and importantly, these differences are only illustrated using LongtoNotes and are not seen in Ontonotes (§4.3). LongtoNotes also presents a challenge in scaling coreference models as prediction time and memory requirement increase substantially on the long documents (§4.4).

2 Our Contribution: LongtoNotes

We present LongtoNotes, a corpus that extends the English coreference annotation in the OntoNotes Release 5.0 corpus¹ (Pradhan et al., 2013) to provide annotations for longer documents. In the original English OntoNotes corpus, the genres such as *broadcast conversations (bc)* and *telephone conversation (tc)* contain long documents that were divided into smaller parts to facilitate easier annotation. LongtoNotes is constructed by collecting annotations to combine within-part coreference chains into coreference chains over the entire long document. The annotation procedure, in which annotators merge coreference chains, is described and analyzed in Section 3.

The divided parts of a long document in Ontonotes are all assigned to the same partition (train/dev/test). This allows LongtoNotes to maintain the same train/dev/test partition, at the document level, as Ontonotes (Table 1). While the content of each partition remains the same, the number of documents changes because the divided parts are merged into a single annotated text in LongtoNotes. We refer to LongtoNotes_s as the subset of LongtoNotes comprising only the merged documents (i.e. documents merged by the annotators).

¹The Arabic and Chinese parts of the Ontonotes dataset are not considered in our study. See Appendix A.3

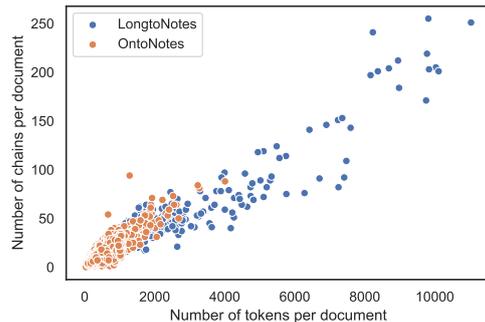


Figure 2: **Document and Coref Chain Length.** The number of coreference chains increases with the increase in token length in LongtoNotes.

Dataset	Train	Dev	Test
OntoNotes	2802	343	348
LongtoNotes	1959	234	222

Table 1: Comparison of the train-test-dev split of documents between OntoNotes and LongtoNotes

2.1 Length of Documents in LongtoNotes

The average number of tokens per document (rounded to the nearest integer) in LongtoNotes is 674, ~44% higher than in Ontonotes (466). Table 2 shows the changes in document length by genre. We observe that the genre with the longest documents is *broadcast conversation* with 4071 tokens per document, which is a dramatic increase from the length of the divided parts in Ontonotes which had 511 tokens per document in the same genre. The number of coreference chains and the number of mentions per chain grows as well. The long documents that were split into multiple parts during the original OntoNotes annotation are not evenly distributed among the genres of text present in the corpus. In particular, text categories *broadcast news (bn)* and *newswire (nw)* consist exclusively of short non-split documents, which were not affected by the LongtoNotes merging process. A list of which documents are merged in LongtoNotes is provided in Table 10 (Appendix).

2.2 Number of Coreference Chains

As a consequence of the increase in document length, LongtoNotes presents a higher number of coreference chains per document (16), compared to OntoNotes (12). Figure 2 shows the length and number of coreference chains for each document in the two corpora. As expected, the number of chains

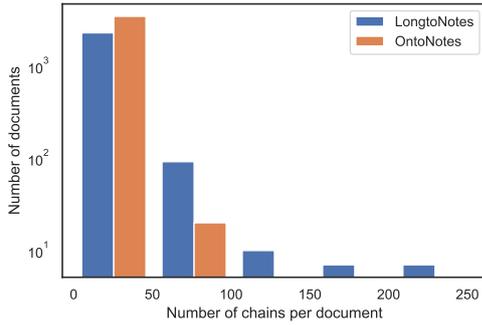


Figure 3: **Number of Chains per Document.** A histogram log plot reveals the long tailed distribution of the number of coreference chains present per document in LongtoNotes. Ontonotes contains more documents with fewer chains.

in a document tends to get larger as the document size increases. For genres with longer average document lengths like *broadcast conversation (bc)*, the increase in the number of chains is as high as 85%, while this increase is only 25% for *pivot (pt)* genre when the document length is comparatively shorter. It is worth noting that the majority of documents had a number of chains in the range of 20 to 50 and only about 20 documents out of 3493 in the OntoNotes dataset had >50 chains per document. For LongtoNotes the number increases to 96 documents. A comparison of the number of chains per document between OntoNotes and LongtoNotes is shown in Figure 3.

2.3 Number of Mentions per Chain

The number of mentions per coreference chain in LongtoNotes is over 30% larger than in OntoNotes. This is primarily because of longer documents and an increase in the number of coreference chains per document. Mentions per chain increase with the increase in document length. For the *broadcast conversation (bc)* genre, the increase in the mentions per chain is highest with 87%, while for the *pivot (pt)* (Old Testament and New Testament text) genre it is only 30% as it has shorter documents.

2.4 Distances to the Antecedents

For each coreference chain, we analyzed the distance between the mentions and their antecedents. The largest distance for a mention to its antecedent grew 3x for LongtoNotes when compared to OntoNotes from 4,885 to 11,473 tokens. Figure 4 shows a detailed breakdown of the mention to

antecedent distance. There are no mentions that are more than 5K tokens distant from its antecedent in OntoNotes. There are 178 such mentions in LongtoNotes.

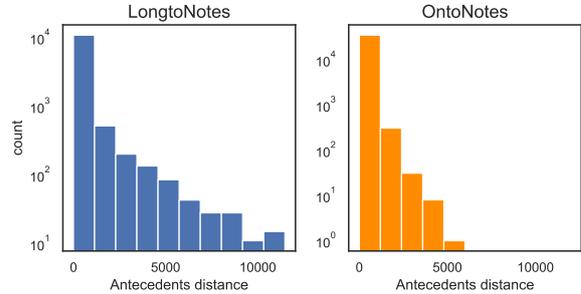


Figure 4: **Distance to Antecedent.** Histogram (log-scale) shows that the largest distance of mention to their antecedents per chain increases in LongtoNotes compared to OntoNotes.

2.5 Comparison with other Datasets

The literature contains multiple works proposing datasets for coreference resolution: Wiki coref (Ghaddar and Langlais, 2016), LitBank (Bamman et al., 2019), PreCo (Chen et al., 2018), Quiz Bowl Questions (Rodriguez et al., 2019; Guha et al., 2015), ACE corpus (Walker et al., 2006), MUC (Chinchor and Sundheim, 1995), MedMentions (Mohan and Li, 2019), inter alia. We compare LongtoNotes to these datasets in terms of number of documents, total number of tokens, and document length (Table 3).

Litbank is a popular long document coreference dataset, presenting a high tokens/document ratio. However, the dataset consists of only 100 documents, rendering model development challenges. Moreover, it focuses only on the literary domain. Other datasets containing long documents (e.g., WikiCoref) are also very small in size. On the other hand, datasets consisting of a larger number of texts tend to contain shorter documents (e.g., PreCo). Thus, by building LongtoNotes, we address the scarcity of a multi-genre corpus with a collection of long documents containing long-range coreference dependencies.

In concurrent work, Gupta et al. (2023) present a generalised annotation platform for coreference with simplified guidelines to users. In the future, such a tool could be used to more easily annotate documents of increased length.

Categories	# Docs		Tokens/Doc		# Chains		Ment./Chains	
	Ont.	Long.	Ont.	Long.	Ont.	Long.	Ont.	Long.
broadcast conversation (bc)	397	50	511	4071	14	85	65	519
broadcast news (bn)	947	947	237	237	8	8	29	29
magazine (mz)	494	78	398	2531	8	41	32	208
newswire (nw)	922	922	529	529	12	12	47	47
pivot (pt)	369	261	657	930	20	27	131	186
telephone conversation (tc)	142	48	728	2157	17	44	108	319
web data (wb)	222	109	763	1555	17	31	73	149
Overall	3493	2415	466	674	12	16	55	80

Table 2: **Genre Comparison.** Comparison of document and coreference chain statistics per genre in OntoNotes 5.0 and our proposed dataset, LongtoNotes.

Dataset	# Docs	Total Size	Tokens/Doc
WikiCoref	30	60K	2000
ACE-2007	599	300K	500
MUC-6	60	30K	500
MUC-7	50	25K	500
QuizBowl	400	50K	125
PreCo	37.6K	12.4M	330
LitBank	100	200K	2105
MedMentions	4392	1.1M	267
OntoNotes	3493	1.6M	466
LongtoNotes	2415	1.6M	674
LongtoNotes _s	283	740K	2615

Table 3: **Coreference Datasets.** A comparison of various coref datasets with our proposed dataset LongtoNotes.

3 Annotation Procedure & Quality

In this section, we describe and assess the annotation procedure used to build LongtoNotes.

3.1 Annotation Task

The annotators merge the coreference annotation in a sequential fashion. That is, they combine annotations from the second split part of an Ontonotes document into the first part, then the third part into the combined first two parts, and so on. Precisely, to build LongtoNotes, annotators successively merge chains in the current part $i + 1$ of the document with one of the chains in the previous parts $1, \dots, i$. We reformulate this annotation process as a question answering task where we ask annotators a series of questions (rather the same coreference determining question for different mentions) using our own annotation tool designed for this task (Figure 5). We display parts $1, \dots, i$ with color-coded mention spans. We then show a highlighted concept (a coreference chain in part $i + 1$) and ask the question: *The highlighted concept below refers to which concept in the above paragraphs?* The anno-

tators select one of the colour-coded chains from parts $1, \dots, i$ from a list of answers or the annotators can specify that the highlighted concept in part $i + 1$ does not refer to any concept in parts $1, \dots, i$, (i.e., a new chain emerging in part $i + 1$). The list of answers here are the merged chains formed in the previous iterations.

The annotation tool proceeds with a question for each coreference chain ordered (sorted by the first token offset of the first mention in the chain). The annotation of all parts of a document comprises an annotation task. That is, a single annotator is tasked with answering the multiple-choice question for each coreference chain in each part of a document. At the end of each part, annotators are shown a summary page that allows them to review, modify, and confirm the decisions made in the considered part. A screenshot of the summary page is provided in the Figure 9 in the Appendix.

From Annotations to Coreference Labels The annotations collected in this way are then converted into coreference labels for the merged parts of a document. The answers to the questions tell us the antecedent link between two coreference chains. These links are used to relabel all mentions in the two chains with the same coreference label, resulting in the LongtoNotes dataset.

Singletons Existing OntoNotes coreference annotation does not include singletons. Considering all parts of a document together might allow mentions that were considered to be singletons in a specific part to be assigned to a coreference chain. To understand the frequency of singletons in a single part of a document that has coreferent mentions in other parts, we manually analysed 500 mentions spread across 10 parts over three randomly selected long documents. We found only 17 instances (~0.03%) where singletons can be merged

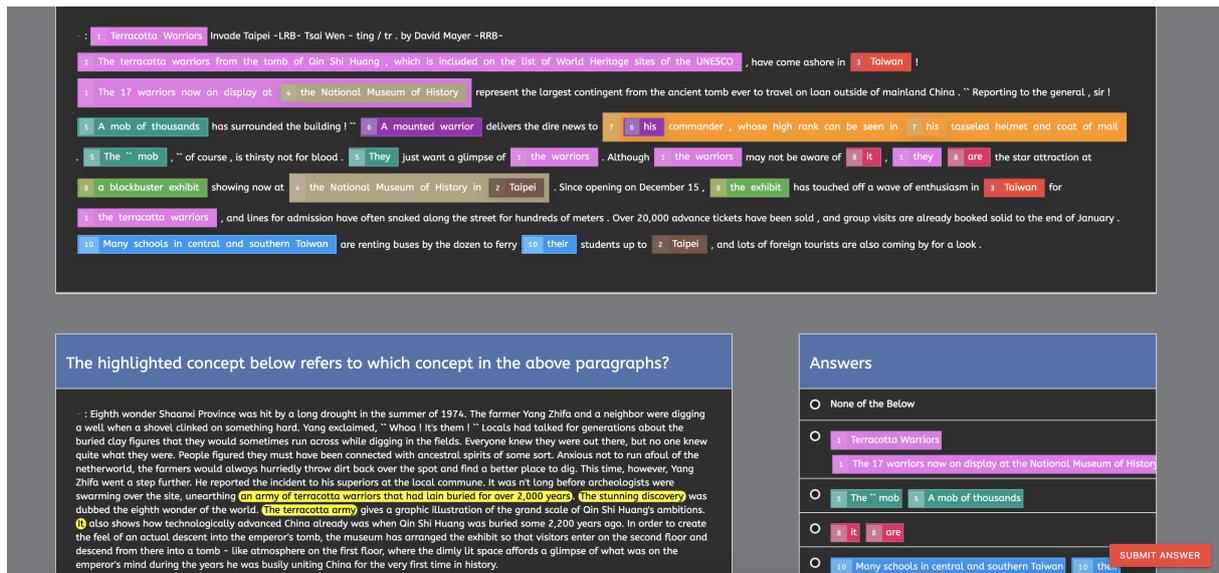


Figure 5: **Annotation Tool Interface.** The upper box represents all the previous paragraphs while the box on the bottom left is the current paragraph. The mentions of the current chain to be merged are shown in yellow. On the right side, the answers are presented which are chains from previous paragraphs and the annotator can select one of them or choose the None of the below option which creates a new chain.

with coreference chains in different parts of the same document. Given that such singletons would constitute only such a small percentage of mentions, we decided it was appropriate to omit them from the annotation process to reduce the complexity of annotation. To merge this small number of singleton mentions, our annotators would have had to label over 50% more mentions per document. We further discuss this in Appendix A.4.

3.2 Annotators and Training

We hired and trained a team of three annotators for the aforementioned task. The annotators were university-level English majors from India and were closely supervised by an expert with experience in similar annotation projects. The annotation team was paid a fair wage of approximately 15 USD per hour for the work. We had several hour-long training sessions outlining the annotation task, setup of the problem, and Ontonotes annotation guidelines. We reviewed example cases of difficult annotation decisions and collaboratively worked through example annotations. We then ran a pilot annotation study with a small number of documents (approx 5% of the total documents). For these documents, we also provided annotations to ensure the training of the annotators and eventual annotation quality. We calculated the inter-annotators' agreement between the annotators and us. After a few rounds of training, we were able to achieve an inter-

annotator agreement score (strict match, defined in the next subsection) of over 95% and we decided to go ahead with the annotation task. This confirmed the annotators' understanding of the task.

After the satisfactory pilot annotation study, the tasks were assigned to the annotators in five batches of 60 documents each. For 10% of the tasks, we had all three annotators provide annotations. For the remaining 90%, a single annotator was used. For the documents with multiple annotators, we used majority voting to settle disagreements. If all annotators disagreed on a specific case, we selected Annotator 1's decision over the others (analysis in the Appendix B).

3.3 Measuring Quality of Annotation

We would like to ensure that LongtoNotes maintains the high-quality standards of OntoNotes. Thus, we compute various metrics of agreement between a pair of annotators. We consider (1) the question-answering agreement (i.e., how similar are the annotations made using the annotation tool), and (2) the coreference label agreement (i.e., at the level of the resulting coreference annotation).

Assume that annotator j receives a set of chains $C_1^{(j)}, C_2^{(j)}, \dots, C_N^{(j)}$. For each chain $C_i^{(j)}$, the annotator links it to a New chain or a chain from their (annotator specific) set of available chains. Let us call $D_i^{(j)}$ the linking decision of the j th annotator, which consists of a pair $(C_i^{(j)}, A_i^{(j)})$, where $A_i^{(j)}$

is the selected antecedent chain. We consider the following question answering metrics:

(i) Strict Decision Matching: When two annotators agreed on merging two chains and there is an exact match between the merged chains. Calculated as $\frac{1}{N} \sum_i \mathbb{I}[D_i^{(1)} = D_i^{(2)}]$.

(ii) Jaccard Decision Match: Jaccard decision calculates the Jaccard similarity between the merged chain: $\frac{1}{N} \sum_i \frac{|A_i^{(1)} \cap A_i^{(2)}|}{|A_i^{(1)} \cup A_i^{(2)}|}$.

(iii) New Chain Agreement: Number of times both annotators select a new chain divided by the number of times at least one selects new chain.

(iv) Not New Chain Agreement: Number of times two annotators agreed on not a *New* chain choice divided by the number of times at least one annotator labels not a *New* chain.

(v) Krippendorff’s alpha: Krippendorff’s alpha (Krippendorff, 2011) is the reliability coefficient measuring inter annotator agreement. We compute Krippendorff’s alpha using a strict decision match as the coding for agreement.

Table 4 presents the results for these metrics. We observed that on average annotators agreed with each other on over 90% of their decisions except when the *No New* chains were considered. Removing *New* chains reduces the total decisions to be made significantly, and hence a lower score on *No New* chains agreement. We found that Annotator 1 agreed most with the experts and hence Annotator 1’s decisions were preferred over the others in case of disagreement between all three annotators.

Where are disagreements found in annotation?

We would like to understand what kinds of mentions lead to the disagreement between annotators. We measure the part of speech of all the disagreed chain assignments between the annotators. We found that the 8% of the mentions within the disagreed chain assignments were pronouns, 8% were verbs, and 9% were common nouns. The number of proper nouns disagreements was lower with just 5%. When considering different genres, it was observed that genres with longer documents like *broadcast conversation (bc)* had more mentions that were pronouns when compared with genres with shorter documents *pivot (pt)*. As expected, the number of disagreements in general increased with the size of the documents. However, we found that the number of disagreements was small even for long document genres such as *broadcast conversation (bc)*. See Appendix B.

Metric	Score
Strict Match	0.90
Jaccard Match	0.95
New Chain	0.88
Not New Chain	0.87
Krippendorff’s alpha	0.90

Table 4: **Annotation Quality Assessment.** We report the average of each metric over all pairs of annotators.

3.4 Time Taken per Annotation

We also recorded the time taken for each annotation. Time taken per annotation increases with the increase in the document length (Appendix Fig. 10). This is expected as more chains create more options to be chosen from and longer document length demands more reading and attention. In total, our annotation process took 400 hours.

3.5 Pitfalls of Automatically Merging Chains

To show the importance of our human-based annotation process, we investigate whether the annotators’ decisions could have been replicated using off-the-shelf automatic tools. We performed two experiments: (i) a simple greedy rule-based string matching system (described in the Appendix A.5) and (ii) Stanford rule-based coreference system to merge chains across various parts. We use the merged chains to calculate the CoNLL F_1 score with the annotations produced by our annotators. We found that our string-matching system achieved a CoNLL F_1 score of only 61%, while the Stanford coreference system reached a score of only 69%. The low scores compared to the annotators’ agreement (which is over 90%) underline the complexity of the task and the need for a human-annotations.

4 Empirical Analysis with LongtoNotes

We hope to show that LongtoNotes can facilitate the empirical analysis of coreference models in ways that were not possible with the original OntoNotes. We are interested in the following empirical questions using the datasets—Ontonotes (Pradhan et al., 2013), and our proposed LongtoNotes and LongtoNotes_s:

- How does the length of documents play a role in the empirical performance of models?

- Does the empirical accuracy of models depend on different hyperparameters in LongtoNotes and Ontonotes?
- Does LongtoNotes reveal properties about the efficiency/scalability of models not present in Ontonotes?

4.1 Models

Much of the recent work on coreference can be organized into three categories: span based representations (Lee et al., 2017; Joshi et al., 2020), token-wise representations (Thirukovalluru et al., 2021; Kirstain et al., 2021) and memory networks / incremental models (Toshniwal et al., 2020b,a). We consider one approach from all three categories.

Span-based representation We used the Joshi et al. (2020) implementation of the higher-order coref resolution model (Lee et al., 2018) with SpanBERT. Here, the documents were divided into a non-overlapping segment length of 384 tokens. We used SpanBERT Base as our model due to memory constraints. The number of training sentences was set to 3. We set the maximum top antecedents, $K = 50$. We used Adam (Kingma and Ba, 2014) as our optimiser with a learning rate of $2e^{-4}$.

Token-wise representation We used the LongFormer Large (Beltagy et al., 2020) version of Kirstain et al. (2021) work, as this approach is less memory demanding and it is possible to fit this model in our memory. The max sequence length was set to 384 or 4096. Adam was used as an optimiser with a learning rate of $1e^{-5}$. A dropout (Srivastava et al., 2014) probability of 0.3 was used.

Memory networks We used SpanBERT Large with a sequence length of 512 tokens. Following Toshniwal et al. (2020b), an endpoint-based mention detector was trained first and then was used for coreference resolution. The number of training sentences was set to 5, 10, and 20. The number of memory cells was selected from 20 or 40. All experiments were performed with AutoMemory models with learned memory type.

4.2 Length of Documents & Performance

Impact of Training Corpus We first investigate whether or not training on the longer documents in LongtoNotes are needed to achieve state-of-the-art results on the dataset. We compare the performance of models trained on Ontonotes to

# Tokens	Training	CoNLL F1
$\leq 2K$	Ontonotes	78.85
	LongtoNotes	78.25
$> 2K$	Ontonotes	65.11
	LongtoNotes	66.20

Table 5: **Performance and Document Length for Span-based Models.** F_1 score across different document length for SpanBERT Base trained model on OntoNotes and LongtoNotes dataset.

those trained on LongtoNotes. We find that by training on LongtoNotes, we can achieve higher CoNLL F1 measures on LongtoNotes than training with Ontonotes for each model architecture (Table 6). This suggests that the longer dependencies formed by merging annotations in various parts of documents in OntoNotes are difficult to model when training on short documents.

We find that to achieve accuracy with hyperparameters such as learning rate/warmup size, we need to maintain a number of steps per epoch consistent with Ontonotes when training with LongtoNotes. A detailed analysis is presented in the Appendix Section C.

Length Analysis - Number of Tokens We break down the performance of the span-based model by the number of tokens in each document. We compare the performance of the model depending on the training set. Figure 2 shows that the majority of the documents in the OntoNotes dataset falls within a token length of 2000 per document. We create two splits of LongtoNotes_s, one having a token length greater than 2000 tokens, the other having a number of tokens smaller than 2000. Table 5 shows that for smaller document length (less than 2000 tokens), the SpanBERT model trained on OntoNotes performed better but the trend reverses for longer documents (more than 2000 tokens), on which the model trained on LongtoNotes outperformed the model trained on OntoNotes by +1%.

Length Analysis - Number of Clusters Table 7 displays the change in F_1 score with the increase in the number of clusters per document. The SpanBERT Base model trained on LongtoNotes outperforms the same model trained on OntoNotes (+0.6%) when the number of clusters is more than 40. Note that, 40 is selected based on the cluster distribution shown in Table 2 with the majority documents in LongtoNotes lying in this range.

	Training	OntoNotes			LongtoNotes _s			LongtoNotes		
		P	R	F ₁	P	R	F ₁	P	R	F ₁
Stanford Coref (Lee et al., 2013b)	-	58.6	58.8	58.6	48.5	58.2	52.7	53.6	57.3	55.2
Span-based (Joshi et al., 2020)	OntoNotes	76.5	77.6	77.4	72.7	69.1	70.8	74.4	73.0	73.7
	LongtoNotes	75.9	77.7	76.8	72.4	70.7	71.5	73.9	74.1	74.0
Token-Level (Kirstain et al., 2021)	Ontonotes	81.2	79.5	80.4	79.6	80.0	79.8	79.7	77.2	78.5
	LongtoNotes	80.0	78.2	79.1	80.3	80.3	80.3	80.2	78.0	79.1
Memory-Model (Toshniwal et al., 2020b)	OntoNotes	73.5	79.3	76.4	63.4	73.8	68.2	67.9	76.6	72.0
	LongtoNotes	73.8	79.4	76.6	66.3	74.6	70.2	69.3	77.0	72.9

Table 6: **Performance Variation by Training Set.** Comparison of F_1 scores on various datasets using different models. All experiments have been performed atleast 2 times and a variance of only ± 0.1 was observed.

# Chains	Training	SpanBERT	Token	Memory
≤ 40	Onto	73.60	79.80	72.80
	Longto	72.86	78.80	71.94
> 40	Onto	68.44	75.60	67.72
	Longto	69.09	76.42	68.60

Table 7: **Performance and Number of Chains for different models.** CoNLL F_1 score across different document length for SpanBERT Base, Token-Level and Memory-Model trained on OntoNotes and LongtoNotes dataset.

4.3 Hyperparameters & Document Length

Each model has a set of hyperparameters that would seemingly lead to variation in performance with respect to document length. We consider the performance of the models on LongtoNotes as a function of these hyperparameters.

Span-based model hyperparameters We consider two hyperparameters: the number of antecedents to use, K and the max number of sentences used in each training example. We found that upon varying K : 10, 25, and 50, there was only a small difference observed in the results for both the models trained on OntoNotes and LongtoNotes (increasing K led to only minor increases). The result is summarized in Table 8. We could not go beyond $K = 50$ due to our GPU memory limitations. However, going beyond 50 might further help for longer documents. Furthermore, we found that the *number of sentences* parameter used to create training batches does not play a significant role in performance either (Figure 8).

Token-wise model hyperparameters Reducing the sequence length when testing from 4096 to 384 leads to a drop in F1 as seen in Figure 6. We observed that longer sequence length (4096) helps

K	OntoNotes	LongtoNotes	LongtoNotes _s
10	77.05	73.44	70.37
25	76.93	73.99	71.61
50	77.60	74.01	71.58

Table 8: **Number of Antecedents vs. Performance** SpanBERT Base model trained on LongtoNotes dataset with varying K value.

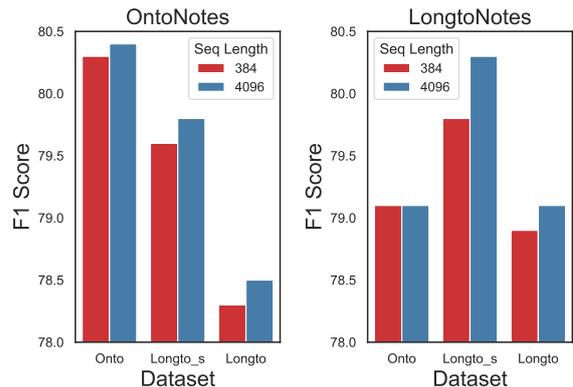


Figure 6: **Sequence Length vs. Performance.** LongFormer is significantly better on LongtoNotes with 4096 sequence length compared to 384. Two sequence lengths perform similarly on Ontonotes.

more for LongtoNotes_s as there are longer sequences than for OntoNotes, which is evident in Figure 6. Furthermore, we analyzed performance on two genres: *magazine (mz)* having 6x longer sequences in LongtoNotes than OntoNotes vs *pivot (pt)* having just 1.4x longer documents. As observed in Figure 7 (and Appendix Table 15), when the document is long as in *magazine (mz)*, there is a significant increase in performance with a longer sequence but the effect is negligible for *pivot (pt)* where the size of the document is almost the same.

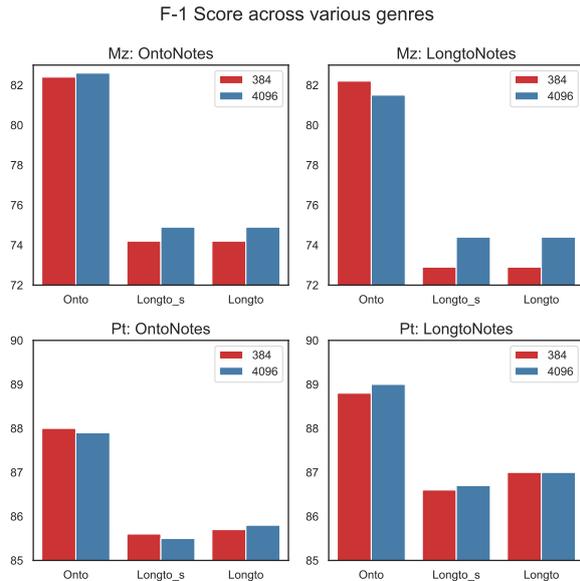


Figure 7: **Sequence Length vs. Performance by Genre** Comparing the effect of sequence length on F1 for two genres: *magazine (mz)*, where LongtoNotes contains 6x longer documents, and *pivot (pt)*, where LongtoNotes has 1.4x longer documents.

Memory model hyperparameters We consider two hyperparameters - the memory size which denotes the maximum active antecedents that can be considered and the max number of sentences used in training. We show that doubling the size of the memory leads to an increase of 0.8 points of CoNLL F_1 for LongtoNotes dataset. (Appendix Table 14). Figure 8 demonstrates that there is no significant improvement in the performance of the model with the increase in the number of training sentences.

4.4 Model Efficiency

We compare the prediction time for the span-based model on the longest length and average length documents in LongtoNotes and Ontonotes in Table 9. We observe that there is a significant jump in running time and memory required to scale the model to long documents on LongtoNotes; this jump is much smaller on Ontonotes. This suggests that our proposed dataset is better suited for assessing the scaling properties of coreference methods.

5 Conclusion

In this paper, we introduced LongtoNotes, a dataset that merges the coreference annotation of documents that in the original OntoNotes dataset were split into multiple independently-annotated

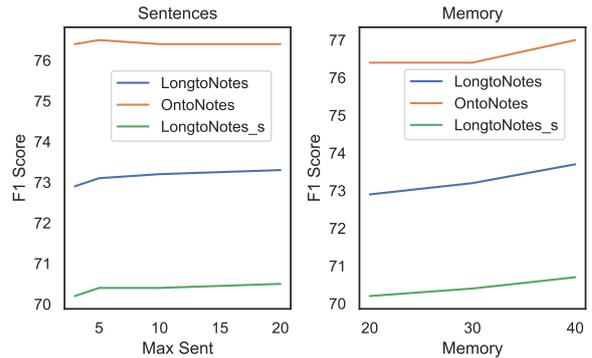


Figure 8: **Max Sentence Length.** Increasing max sentences from 3 to 20 has a small effect on the performance of the SpanBERT large model. On the other hand, the increase is linear with the increase in the memory size alongside the increase in max training sentences.

Dataset	Type	Pred. Time	Pred. Mem
Ontonotes	Average	0.11 sec	1.50 GB
LongtoNotes	Average	0.47 sec	6.50 GB
Ontonotes	Longest	0.37 sec	5.84 GB
LongtoNotes	Longest	2.35 sec	42.68 GB

Table 9: **Model Efficiency of Span-based Models.** We find that LongtoNotes documents have extended length leading to greater variation of prediction time and prediction memory.

parts. LongtoNotes has longer documents and coreference chains than the original OntoNotes dataset. Using LongtoNotes, we demonstrate that scaling current approaches to long documents has significant challenges both in terms of achieving better performance as well as scalability. We demonstrate the merits of using LongtoNotes as an evaluation benchmark for coreference resolution and encourage future work to do so.

Acknowledgements

This material is based upon work supported in part by an ETH Zürich Research grant (ETH-19 21-1), a grant from the Swiss National Science Foundation (project # 201009), the University of Massachusetts Amherst Center for Data Science and the Center for Intelligent Information Retrieval, and in part by the Chan Zuckerberg Initiative under the project “Scientific Knowledge Base Construction”. Alessandro Stolfo is supported by armasuisse Science and Technology through a CYD Doctoral Fellowship. We also thank our annotators from Xsaras.com (a data annotation company) for their sincere efforts in making this project possible.

Limitations

Our dataset is comprised solely of English texts, and our analysis, therefore, applies uniquely to the English language. OntoNotes, however, consists of the Arabic and the Chinese annotations too and those languages were not considered in our study due to the limited expertise of the annotators.

Since our models are not tuned for any specific real-world application, the methods should not be used directly in highly sensitive contexts such as legal or health-care settings, and any work building on our methods must undertake extensive quality-assurance and robustness testing before using them.

Ethical Considerations

The annotation was performed with a data annotation service which ensured that the annotators were paid a fair compensation of 15 USD per hour. The annotation process did not solicit any sensitive information from the annotators.

Replicability We have released the model checkpoints and data at: <https://github.com/kumar-shridhar/LongtoNotes>.

References

- Mahmoud Azab, Ahmed Salama, Kemal Oflazer, Hideki Shima, Jun Araki, and Teruko Mitamura. 2013. [An NLP-based reading tool for aiding non-native English readers](#). In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 41–48, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.
- Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.
- David Bamman, Sejal Popat, and Sheng Shen. 2019. [An annotated dataset of literary entities](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2138–2144, Minneapolis, Minnesota. Association for Computational Linguistics.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Hong Chen, Zhenhua Fan, Hao Lu, Alan L Yuille, and Shu Rong. 2018. Preco: A large-scale dataset in preschool vocabulary for coreference resolution. *arXiv preprint arXiv:1810.09807*.
- Nancy A Chinchor and Beth Sundheim. 1995. Message understanding conference (muc) tests of discourse processing. In *Proc. AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 21–26.
- K Bretonnel Cohen, Arrick Lanfranchi, Miji Joo-young Choi, Michael Bada, William A Baumgartner, Natalya Panteleyeva, Karin Verspoor, Martha Palmer, and Lawrence E Hunter. 2017. Coreference annotation and resolution in the colorado richly annotated full text (craft) corpus of biomedical journal articles. *BMC bioinformatics*, 18(1):1–14.
- Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982.
- Abbas Ghaddar and Phillippe Langlais. 2016. [Wiki-Coref: An English coreference-annotated corpus of Wikipedia articles](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 136–142, Portorož, Slovenia. European Language Resources Association (ELRA).
- Anupam Guha, Mohit Iyyer, Danny Bouman, and Jordan Boyd-Graber. 2015. [Removing the training wheels: A coreference dataset that entertains humans and challenges computers](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1108–1118, Denver, Colorado. Association for Computational Linguistics.
- Ankita Gupta, Marzena Karpinska, Wenlong Zhao, Kalpesh Krishna, Jack Merullo, Luke Yeh, Mohit Iyyer, and Brendan O’Connor. 2023. [ezcoref: Towards unifying annotation guidelines for coreference resolution](#). *Findings of ACL: EACL*.
- Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S. Weld, and Marti A. Hearst. 2021. [Augmenting Scientific Papers with Just-in-Time, Position-Sensitive Definitions of Terms and Symbols](#). Association for Computing Machinery, New York, NY, USA.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Yuval Kirstain, Ori Ram, and Omer Levy. 2021. Coreference resolution without span representations. In *ACL/IJCNLP*.
- Klaus Krippendorff. 2011. Computing krippendorff’s alpha-reliability.

- Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013a. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational linguistics*, 39(4):885–916.
- Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013b. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.
- Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.
- Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.
- Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Xiaoqiang Luo. 2005. On coreference resolution performance metrics. HLT '05, page 25–32, USA. Association for Computational Linguistics.
- Sunil Mohan and Donghui Li. 2019. Medmentions: a large biomedical corpus annotated with umls concepts. *arXiv preprint arXiv:1902.09476*.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.
- Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.
- Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2019. Quizbowl: The case for incremental question answering. *arXiv preprint arXiv:1904.04792*.
- Mrinmaya Sachan, Eduard Hovy, and Eric P Xing. 2015. An active learning approach to coreference resolution. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.
- Raghuvveer Thirukovalluru, Nicholas Monath, Kumar Shridhar, Manzil Zaheer, Mrinmaya Sachan, and Andrew McCallum. 2021. Scaling within document coreference to long texts. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3921–3931, Online. Association for Computational Linguistics.
- Shubham Toshniwal, Allyson Ettinger, Kevin Gimpel, and Karen Livescu. 2020a. Petra: A sparsely supervised memory model for people tracking. *arXiv preprint arXiv:2005.02990*.
- Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. 2020b. Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8519–8526, Online. Association for Computational Linguistics.
- Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, page 45–52, USA. Association for Computational Linguistics.
- Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.
- Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Appendix

A Dataset and Annotation Details

A.1 Annotation tool

Fig. 9 shows our tool’s summary page.

A.2 Comparison with OntoNotes

A detailed genre-wise comparison of the documents from OntoNotes dataset which were merged in LongtoNotes is presented in Table 10. It can be seen that categories like `bn` and `nw` are completely missing in LongtoNotes, while `pt` is partially missing.

Documents in Corpus comparison		
Category	Onto	Longto
<code>bc/cctv</code>	✓	✓
<code>bc/cnn</code>	✓	✓
<code>bc/msnbc</code>	✓	✓
<code>bc/phoenix</code>	✓	✓
<code>bn/abc</code>	✓	✗
<code>bn/cnn</code>	✓	✗
<code>bn/mnb</code>	✓	✗
<code>bn/nbc</code>	✓	✗
<code>bn/pri</code>	✓	✗
<code>bn/voa</code>	✓	✗
<code>mz/sinorama</code>	✓	✓
<code>nw/wsj</code>	✓	✗
<code>nw/xinhua</code>	✓	✗
<code>pt/nt</code>	✓	✓
<code>pt/ot</code>	✓	✗
<code>tc/ch</code>	✓	✓
<code>wb/a2e</code>	✓	✓
<code>wb/c2e</code>	✓	✓
<code>wb/eng</code>	✓	✓

Table 10: Comparison of documents from various sub-categories that exists in OntoNotes 5.0 and our proposed dataset LongtoNotes

A.3 Dataset selection decision

Due to budget constraints and the expertise of our team and annotators in English only (and some training of annotators is required to ensure data quality), we only considered the English parts of the OntoNotes dataset in our work. We think that the dataset can be extended to Arabic and Chinese too, but we leave it for future work.

A.4 Annotating singletons

While manually annotating all singletons, we observed that almost all NPs can be thought of as

mentions and all those NPs that are not part of any chain can be thought of as a singleton. Our analysis suggests that there are over 50% mentions that are not annotated by OntoNotes and can qualify for singletons. To annotate all the singletons, the annotator needs to go through all of them, discard the ones that do not abide by the OntoNotes rules and then make a decision whether to merge each singleton to some chain or other singleton. In our analysis, the number of such singletons is very low and all the efforts were not worth it for the small improvement over the current annotations. So we decide to ignore all the singletons in our study.

A.5 Greedy rule-based matching system

We use a greedy string matching system where we take all the mentions in a chain of the current para $i + 1$ and analyse its part of speech provided in the OntoNotes dataset. We take the first Noun (NN or NP) present in each chain and look for the mentions overlap in all other previous paras $1, \dots, i$ chains. We merged two chains if there is a strict overlap with any of the mentions in a given chain. If there are no strict overlaps, we move to the next noun in the given chain and repeat the process. If we find no strict overlap with any mentions in any other para chains, we keep the chain independent (same as assigning *None of the below* in our annotation tool). We repeat the process with all chains in a given document and constantly update the chain after every para.

B Annotation Disagreement Analysis

B.1 Genre wise disagreement analysis

Table 11 presents the genre-wise disagreement analysis for strict decision matching. Genres with longer documents like `bc`, `mz` have more disagreements compared to genres with smaller document lengths like `tc`, `pt`.

The trend is very similar for new chain assignments where genres with larger documents have more disagreements over new chain assignments. The numbers are presented in Table 13.

B.2 Annotators disagreements analysis

Figure 11 shows the cases (in black) when the annotators disagreed for each part of the speech categories (shown in big coloured bubbles). The size of the bubbles is representative of their occurrence in the dataset, suggesting there are more pronominal mentions in the dataset than nouns or proper nouns.

Figure 9: The summary page of our annotation tool that is shown after all the chains decisions in a paragraph is made. The annotators can look and verify all the decisions and confirm answers and proceed to the next para or can change their answers if they want.

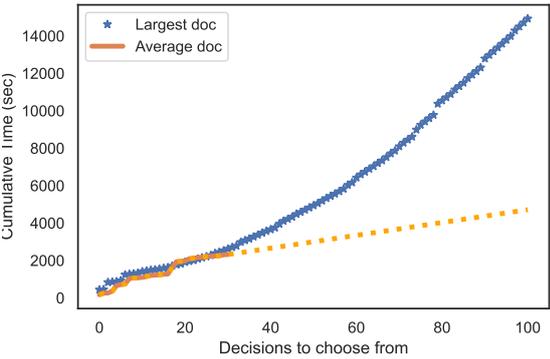
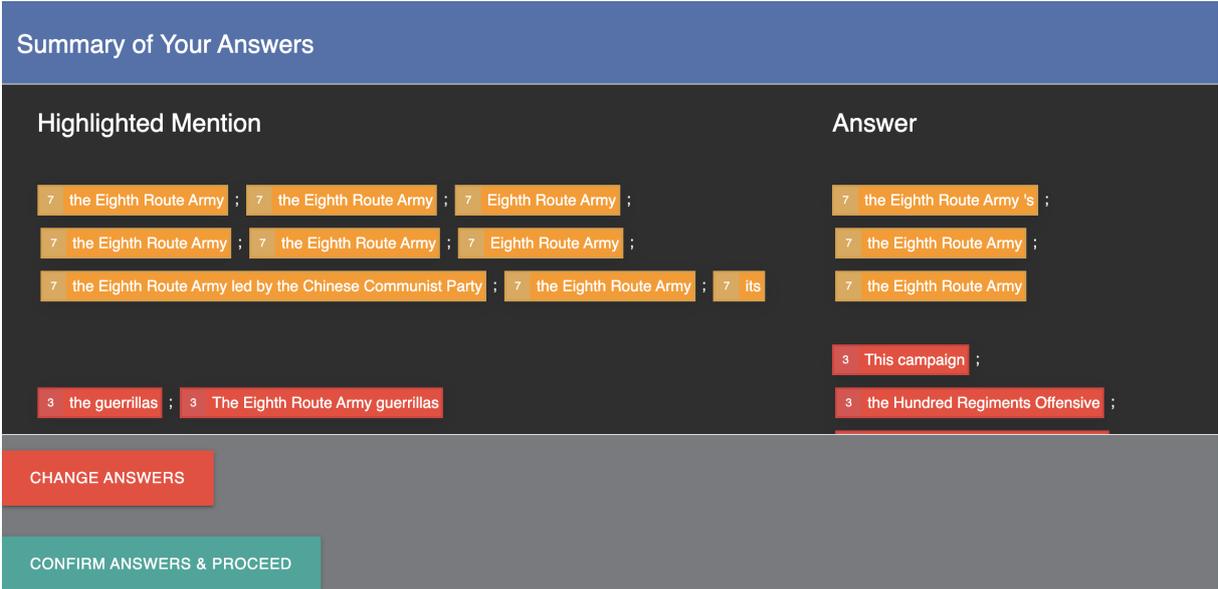


Figure 10: **Annotation Time and Document Length.** Annotation time (cumulative) increases exponentially with the increase in the number of decisions to choose from. A comparison is shown between the longest document in LongtoNotes vs an average document. The dotted lines represent the increase in annotation time if the growth was linear.

B.2.1 Genre wise disagreement analysis

In general, annotators disagree more on pronouns than proper nouns and the trend is consistent for various genres as shown in Table 12.

C Results

C.1 MUC, B³ and CEAFE scores

Tables 16, 17 and 18 present the MUC (Vilain et al., 1995), B³ (Bagga and Baldwin, 1998) and

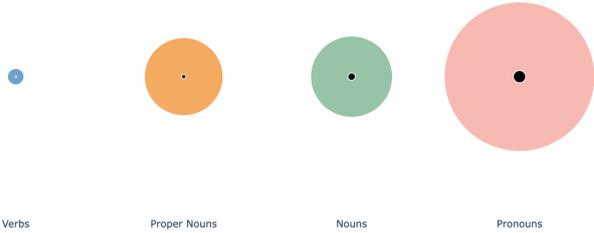


Figure 11: Plot showing the part of speech distribution for the disagreed clusters between annotators.

CEAFE (Luo, 2005) scores for SpanBERT Base (Lee et al., 2017) and LongDocCoref Models (Toshniwal et al., 2020b). On all three metrics, both models trained on LongtoNotes dataset outperforms the models trained on OntoNotes dataset. For SpanBERT base model, we compare three version of the LongtoNotes dataset: LongtoNotes_s and LongtoNotes dataset as mentioned in the paper and LongtoNotes_{eq} where LongtoNotes dataset is reweighted to create the total number of documents equal to the number of documents in OntoNotes dataset. For LongDocCoref model, *n* represents the maximum number of training sentences, while *m* refers to the memory used.

C.2 Genre wise F₁ scores vs sequence length

Table 15 shows that LongFormer Large model with larger sequence length (4096) outperforms the one

bc			
	Ann1	Ann2	Ann3
Ann1	1.0	0.91	0.87
Ann2	0.91	1.0	0.88
Ann3	0.87	0.88	1.0

mz			
	Ann1	Ann2	Ann3
Ann1	1.0	0.91	0.94
Ann2	0.91	1.0	0.93
Ann3	0.94	0.93	1.0

pt			
	Ann1	Ann2	Ann3
Ann1	1.0	0.97	0.98
Ann2	0.97	1.0	0.96
Ann3	0.98	0.96	1.0

tc			
	Ann1	Ann2	Ann3
Ann1	1.0	0.99	0.98
Ann2	0.99	1.0	0.98
Ann3	0.98	0.98	1.0

wb			
	Ann1	Ann2	Ann3
Ann1	1.0	0.93	0.90
Ann2	0.93	1.0	0.92
Ann3	0.90	0.92	1.0

Table 11: Genre wise strict decision based disagreement analysis between the annotators.

PoS type	bc	pt
Pronouns	3.6	0.04
Nouns	3.2	0.05
Proper Nouns	1.9	0.03
Verbs	3.5	1.0

Table 12: Genre wise part of speech comparison for two genres: bc and pt. The numbers are normalized and presented in percentage.

with shorter sequence length (384) for all models. The difference is higher when the documents are longer (as seen in mz genre) than when the documents are shorter (as seen in pt).

bc			
	Ann1	Ann2	Ann3
Ann1	1.0	0.91	0.85
Ann2	0.91	1.0	0.86
Ann3	0.85	0.86	1.0

mz			
	Ann1	Ann2	Ann3
Ann1	1.0	0.89	0.91
Ann2	0.89	1.0	0.90
Ann3	0.91	0.90	1.0

pt			
	Ann1	Ann2	Ann3
Ann1	1.0	0.94	0.95
Ann2	0.94	1.0	0.91
Ann3	0.95	0.91	1.0

tc			
	Ann1	Ann2	Ann3
Ann1	1.0	0.98	0.98
Ann2	0.98	1.0	0.98
Ann3	0.98	0.98	1.0

wb			
	Ann1	Ann2	Ann3
Ann1	1.0	0.92	0.90
Ann2	0.92	1.0	0.91
Ann3	0.90	0.91	1.0

Table 13: Genre wise disagreement analysis between the annotators for new chain assignment.

Dataset	Memory Size	
	20	40
OntoNotes	76.6	77.0
LongtoNotes	72.9	73.7
LongtoNotes _s	70.2	70.7

Table 14: **Memory Size vs. Performance.** We compare two settings of the memory size parameter in memory model (Toshniwal et al., 2020b) and find that the larger memory version achieves better results on each dataset.

	OntoNotes						LongtoNotes _s						LongtoNotes					
	Mention			Coref			Mention			Coref			Mention			Coref		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
LongFormer Large (mz)																		
+ OntoNotes (384)	88.0	87.9	88.0	82.4	82.4	82.4	84.3	86.1	85.2	73.8	75.0	74.2	84.3	86.1	85.2	73.8	75.0	74.2
+ OntoNotes (4096)	87.9	88.3	88.1	82.4	82.9	82.6	84.4	86.7	85.5	74.1	75.9	74.9	84.4	86.7	85.5	74.1	75.9	74.9
+ LongtoNotes (384)	87.0	88.4	87.7	81.4	83.0	82.2	84.4	86.9	85.6	72.4	73.6	72.9	84.4	86.9	85.6	72.4	73.6	72.9
+ LongtoNotes (4096)	86.9	87.8	87.4	80.9	82.0	81.5	85.0	86.7	85.8	74.1	74.8	74.4	85.0	86.7	85.8	74.1	74.8	74.4
LongFormer Large (pt)																		
+ OntoNotes (384)	95.5	94.4	95.0	88.6	87.4	88.0	94.3	95.3	94.8	84.6	86.9	85.7	94.9	94.4	94.7	85.5	85.8	85.6
+ OntoNotes (4096)	95.6	94.2	94.9	88.9	86.9	87.9	94.4	94.8	94.6	84.8	86.8	85.8	94.9	94.0	94.5	85.5	85.2	85.5
+ LongtoNotes (384)	95.1	94.3	94.7	89.2	88.3	88.8	94.2	95.1	94.6	86.0	88.0	87.0	94.6	94.2	94.4	86.5	86.7	86.6
+ LongtoNotes (4096)	95.3	94.2	94.8	89.7	88.2	89.0	94.5	94.5	94.5	86.4	87.4	86.9	94.8	93.7	94.3	87.0	86.4	86.7

Table 15: Comparison of F_1 scores for mz and pt genres.

	OntoNotes						LongtoNotes _s						LongtoNotes					
	Mention			Coref			Mention			Coref			Mention			Coref		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
SpanBERT Base (Lee et al., 2017)																		
+ OntoNotes	86.6	87.5	87.0	83.1	83.6	83.4	88.4	85.0	86.7	84.2	80.8	82.4	86.7	85.4	86.1	83.0	81.3	82.1
+ LongtoNotes _s	73.3	91.0	81.2	70.0	85.7	77.1	78.3	90.5	84.0	73.8	85.5	79.2	73.2	90.4	80.9	69.4	85.1	76.5
+ LongtoNotes	86.6	87.1	86.8	83.0	82.9	86.8	88.1	84.6	86.3	83.3	80.1	81.7	86.6	85.5	86.0	82.4	81.0	81.7
+ LongtoNotes _{eq}	86.1	87.8	87.0	82.8	83.5	83.2	87.7	86.2	87.0	83.4	81.9	82.6	86.1	86.3	86.2	82.3	81.9	82.1
LongDocCoref (Toshniwal et al., 2020b)																		
+ OntoNotes	95.3	85.6	86.4	81.2	85.4	83.2	95.3	85.6	86.4	77.8	86.2	81.8	95.3	85.6	86.4	78.2	85.2	81.6
+ LongtoNotes _s	95.3	85.6	86.4	22.3	66.9	33.5	95.3	85.6	86.4	17.5	65.7	27.6	95.3	85.6	86.4	21.7	66.9	32.8
+ LongtoNotes	95.3	85.6	86.4	81.4	85.0	83.2	95.3	85.6	86.4	79.3	85.8	82.4	95.3	85.6	86.4	79.1	85.0	81.9
+ LongtoNotes _{eq} (n=3)	95.3	85.6	86.4	81.6	85.2	83.4	95.3	85.6	86.4	79.7	86.2	82.8	95.3	85.6	86.4	79.3	85.2	82.2
+ LongtoNotes _{eq} (n=5)	95.3	85.6	86.4	81.4	85.3	83.3	95.3	85.6	86.4	79.7	86.2	82.8	95.3	85.6	86.4	79.2	85.3	82.1
+ LongtoNotes _{eq} (n=10)	95.3	85.6	86.4	81.5	85.1	83.3	95.3	85.6	86.4	79.7	86.2	82.8	95.3	85.6	86.4	79.6	84.8	82.1
+ LongtoNotes _{eq} (n=10, m=40)	95.3	85.6	86.4	81.6	85.6	83.6	95.3	85.6	86.4	79.8	85.9	82.7	95.3	85.6	86.4	79.5	85.2	82.3

Table 16: Comparison of MUC scores

	OntoNotes						LongtoNotes _s						LongtoNotes					
	Mention			Coref			Mention			Coref			Mention			Coref		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
SpanBERT Base (Lee et al., 2017)																		
+ OntoNotes	86.6	87.5	87.0	75.0	75.5	75.3	88.4	85.0	86.7	70.7	65.1	67.8	86.7	85.4	86.1	72.3	69.5	70.9
+ LongtoNotes _s	73.3	91.0	81.2	57.0	76.8	65.4	78.3	90.5	84	54.8	69.7	61.3	73.2	90.4	80.9	53.3	72.8	61.5
+ LongtoNotes	86.6	87.1	86.8	74.6	74.0	74.3	88.1	84.6	86.3	67.5	62.7	65.0	86.6	85.5	86.0	70.6	68.2	69.4
+ LongtoNotes _{eq}	86.1	87.8	87.0	74.9	75.2	75.0	87.7	86.2	87.0	69.7	67.0	68.3	86.1	86.3	86.2	71.7	70.6	71.2
LongDocCoref (Toshniwal et al., 2020b)																		
+ OntoNotes	95.3	85.6	86.4	72.2	77.9	74.9	95.3	85.6	86.4	57.9	71.7	64.0	95.3	85.6	86.4	63.9	74.7	68.9
+ LongtoNotes _s	95.3	85.6	86.4	18.3	61.7	28.2	95.3	85.6	86.4	10.7	53.6	17.9	95.3	85.6	86.4	16.1	58.7	25.2
+ LongtoNotes	95.3	85.6	86.4	73.3	76.7	75.0	95.3	85.6	86.4	61.0	70.1	65.2	95.3	85.6	86.4	65.5	73.7	69.4
+ LongtoNotes _{eq} (n=3)	95.3	85.6	86.4	73.7	76.9	75.2	95.3	85.6	86.4	64.4	70.4	67.3	95.3	85.6	86.4	67.5	73.7	70.5
+ LongtoNotes _{eq} (n=5)	95.3	85.6	86.4	73.4	77.3	75.3	95.3	85.6	86.4	64.5	70.9	67.6	95.3	85.6	86.4	67.5	74.2	70.7
+ LongtoNotes _{eq} (n=10)	95.3	85.6	86.4	73.6	77.0	75.3	95.3	85.6	86.4	64.5	70.9	67.6	95.3	85.6	86.4	68.3	73.5	70.8
+ LongtoNotes _{eq} (n=10, m=40)	95.3	85.6	86.4	73.5	78.1	75.7	95.3	85.6	86.4	65.0	70.5	67.6	95.3	85.6	86.4	67.9	74.4	71.0

Table 17: Comparison of BCUB scores

	OntoNotes						LongtoNotes _s						LongtoNotes					
	Mention			Coref			Mention			Coref			Mention			Coref		
	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁	P	R	F ₁
SpanBERT Base (Lee et al., 2017)																		
+ OntoNotes	86.6	87.5	87.0	71.5	73.7	72.1	88.4	85.0	86.7	63.3	61.6	62.4	86.7	85.4	86.1	68.1	68.4	68.2
+ LongtoNotes _s	73.3	91.0	81.2	53.2	69.5	60.3	78.3	90.5	84.0	51.5	59.2	55.1	73.2	90.4	80.9	50.4	64.2	56.5
+ LongtoNotes	86.6	87.1	86.8	70.8	73.1	71.9	88.1	84.6	86.3	63.4	60.5	61.9	86.6	85.5	86.0	67.7	68.2	67.9
+ LongtoNotes _{eq}	86.1	87.8	87.0	70.2	74.2	72.1	87.7	86.2	87.0	64.0	63.1	63.5	86.1	86.3	86.2	67.5	69.6	68.5
LongDocCoref (Toshniwal et al., 2020b)																		
+ OntoNotes	95.3	85.6	86.4	67.0	74.5	70.5	95.3	85.6	86.4	54.5	63.4	58.6	95.3	85.6	86.4	61.6	69.8	65.4
+ LongtoNotes _s	95.3	85.6	86.4	25.7	60.0	35.9	95.3	85.6	86.4	16.8	47.8	24.8	95.3	85.6	86.4	23.5	57.2	33.3
+ LongtoNotes	95.3	85.6	86.4	65.8	75.3	70.2	95.3	85.6	86.4	53.7	65.9	59.2	95.3	85.6	86.4	60.5	71.7	65.6
+ LongtoNotes _{eq} (n=3)	95.3	85.6	86.4	66.1	76.2	70.8	95.3	85.6	86.4	54.9	67.4	60.5	95.3	85.6	86.4	61.2	72.2	66.2
+ LongtoNotes _{eq} (n=5)	95.3	85.6	86.4	66.7	76.0	71.1	95.3	85.6	86.4	56.0	66.6	60.9	95.3	85.6	86.4	61.9	71.8	66.5
+ LongtoNotes _{eq} (n=10)	95.3	85.6	86.4	66.2	75.9	70.7	95.3	85.6	86.4	56.0	66.6	60.9	95.3	85.6	86.4	61.7	72.2	66.6
+ LongtoNotes _{eq} (n=10, m=40)	95.3	85.6	86.4	68.0	75.9	71.7	95.3	85.6	86.4	56.1	68.9	61.9	95.3	85.6	86.4	62.9	72.9	67.5

Table 18: Comparison of CEAFF scores

More Robust Schema-Guided Dialogue State Tracking via Tree-Based Paraphrase Ranking

Alexandru Coca[†], Bo-Hsiang Tseng[‡], Weizhe Lin[†], Bill Byrne[†]

[†]Department of Engineering, University of Cambridge, United Kingdom

[‡]Apple

{ac2123, wl356, wjb31}@cam.ac.uk

bohsiang_tseng@apple.com

Abstract

The schema-guided paradigm overcomes scalability issues inherent in building task-oriented dialogue (TOD) agents with static ontologies. Instead of operating on dialogue context alone, agents have access to hierarchical schemas containing task-relevant natural language descriptions. Fine-tuned language models excel at schema-guided dialogue state tracking (DST) but are sensitive to the writing style of the schemas. We explore methods for improving the robustness of DST models. We propose a framework¹ for generating synthetic schemas which uses tree-based ranking to jointly optimise lexical diversity and semantic faithfulness. The generalisation of strong baselines is improved when augmenting their training data with prompts generated by our framework, as demonstrated by marked improvements in average joint goal accuracy (JGA) and schema sensitivity (SS) on the SGD-X benchmark.

1 Introduction

DST is concerned with tracking user goals in task-oriented conversations. The goals are represented as key-value pair sequences, with the keys known as *slots* (e.g. hotel name). Pre-trained language models (PLMs) (Devlin et al., 2019; Raffel et al., 2020) have helped shift focus from systems that can only track slots drawn from a database or *domain ontology* (Henderson et al., 2014) to models that do not require re-training to parse goals in new domains. The Schema-Guided Dialogue (SGD) dataset (Rastogi et al., 2020) facilitates this shift with a large-scale set of conversations grounded in 45 *service APIs* or *schemas* that describe the domains, slots and user intents that annotate the conversations (Appendix A). Test set dialogues are grounded in 6 schemas *seen* during training and 15 *unseen* ones.

Neural models perform impressively on the difficult schema-guided DST task (Rastogi et al., 2020),

¹Code will be released here: <https://bit.ly/3WYB7F1>

but Lee et al. (2022) show that the uniformity of the descriptive language of the schemas facilitates this. They create the SGD-X benchmark to evaluate robust zero-shot generalisation of DST models. This is achieved by grounding the SGD test set conversations in five *schema variants* increasingly dissimilar to the SGD schemata². To perform well, a DST model should correctly track the state of a dialogue when conditioned, in turn, on prompts constructed from the five variants.

We show how to improve DST robustness by introducing controlled variability in the data. We contribute to robust DST research by (1) a flexible framework for generating and ranking diverse outputs of a paraphrase model based on a tree-clustering algorithm designed to control lexical diversity and semantic similarity; (2) combine state-of-the-art paraphrase models and language generation metrics to generate increasingly diverse schemata paraphrases; (3) show that augmenting the training dataset with these schemata improves the robustness and generalisation performance of strong DST baselines.

2 Related Work

Input variety, data scarcity and domain shifts affect the robustness of DST models. Liu et al. (2021) investigate the former. They employ word-level data augmentation (DA) (Wei and Zou, 2019), turn paraphrasing and speech disfluency modeling to approximate their field performance. Turn and dialogue generation are effective in low-resource settings (Campagna et al., 2020; Hou et al., 2018) but are very difficult to scale to new domains and are not effective in the high-resource setting we consider (Campagna et al., 2020; Mohapatra et al., 2021). This also applies to word- and sentence-level meth-

²Variants are ordered according to their lexical similarity to the SGD schemas. The v1 variant is the most similar whereas v5 is the most dissimilar. See Appendix A for details and examples and the schemata here: <https://bit.ly/3Ev0KrV>.

ods (Quan and Xiong, 2019; Louvan and Magnini, 2020). Lee et al. (2022) find word-order changes and deletions to be ineffective in the high-resource, schema-guided setting we consider.

Schema-guided DST tackles both data scarcity and novel domains by using API definitions to prompt PLMs (Zhao et al., 2022). Yet Lee et al. (2022) demonstrate the lack of robustness of schema-guided DST models to prompt styles and vocabulary, creating a new research direction. They show that augmenting the training data with synthetic prompts obtained via backtranslation significantly improves models’ ability to track states under meaning-preserving prompt transformations. Backtranslation is also applied to improve DST robustness to linguistic variation inherent in user communication (Ma et al., 2019; Einolghozati et al., 2019), which is orthogonal to the prompt style and vocabulary robustness setting we consider. Reinforcement learning has also been applied (Yin et al., 2020), but works only in the very constrained single-domain, ontology-driven setting. Other TOD-relevant DA approaches apply to policy learning (Gritta et al., 2021) and response-generation (Gao et al., 2020; Zhang et al., 2020b).

Addressing the dearth of augmentation methods designed to ensure prompt robustness of schema-guided DST models, we propose to generate schemas by ranking large paraphrase candidate lists with learned metrics in a tree ranking scheme.

3 Tree-Based Paraphrase Ranking

Tree construction A large pool of schema candidates is created by generating paraphrases given grids of generation parameters (eg temperature, number of beams). The set is filtered to address generation failures (eg toxic and hallucinated words). We optionally filter candidates with an entailment model to increase semantic faithfulness (Narayan et al., 2022) (see Appendix B.1).

The tree constructor (Algorithm 1) takes as input an object (Node) that stores a metric value, *val*, and the candidate paraphrases which are split at that node, *sents*. A list of metrics to be computed between each candidate and the input is provided by the user. This enables our framework to build arbitrary-depth trees with custom user metrics. Each unique list of metric values describing the distance between the input and a candidate generates a path in the tree (lines 5-13). The *n*-ary tree constructed in this way has the property that level-

order traversal of the first level can yield diverse candidates with respect to the metric it encodes. In practice, the metrics measure lexical and semantic distances between their inputs.

Algorithm 1: Tree building

```

1: def build_tree(root: Node, inp: str,
   cands: list[str], metrics: list[Callable]):
   Data: root, inp input, cands input
           paraphrases, metrics objects to
           eval. dist. between input & cand.
   Result: tree splitting cands according
           to metrics
2:   curr ← root ;
3:   for c in cands:
4:     curr ← root ;
5:     for m in metrics:
6:       m_val = m(inp, c) ;
7:       next ← get_child
           (curr.children, m_val) ;
8:       if next is NULL:
9:         next ← Node (val=m_val,
           sents=[c]) ;
10:      curr.children.add(next)
11:     else:
12:       next.sents.add(c) ;
13:     curr ← next
14:   return root

```

Ranking Our ranker input is the tree and a list of decision functions, with elements corresponding to each level in the tree, *f_{dec}*. Without loss of generality, we assume that the first level encodes a metric with respect to which the user wishes to maximise diversity (eg lexical distance). As shown in Figure 1, our algorithm traverses breadth-first the level for which diversity is to be maximised. Each subtree returned in the traversal is traversed depth-first, guided by the decision functions. For example, in Figure 1 we show that the node $B = 0.77$ is selected by applying the max decision function to the children of $J = 66$, and that applying min to the children of $B = 0.77$ selects the leaf $S = 77$. See Algorithm 3 (Appendix B) for details.

4 Experiments

4.1 Schema generation

Our paraphrase model is PegasusParaphrase³, a fine-tuned Pegasus model (Zhang et al., 2020a).

³Available at <https://bit.ly/3vgY7EZ>.

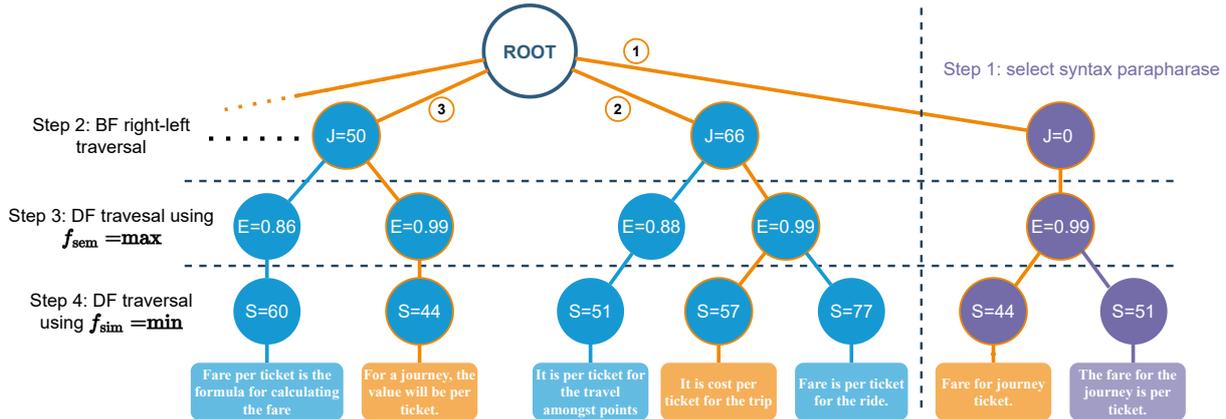


Figure 1: Ranking paraphrases of *Fare per ticket for journey* using a tree. Top level node split is by Jaccard distance (J), middle nodes split by entailment score (E) and leaf nodes store string similarities (S). Here $f_{\text{dec}} = [\text{None}, \text{max}, \text{min}]$. By using J we guarantee that candidates with $J = 0$ are syntactic paraphrases if S is constrained. Orange leaves show top ranked candidates. Numbers on paths show ranking order.

Using 10 settings for the number of beams and temperature, we generate 500 candidates for each input. For efficiency purposes, these are filtered heuristically (Appendix B). We construct a depth $d = 3$ tree which splits the candidates by Jaccard distance J , entailment E , and string similarity S . That is, the input to our tree constructor (Algorithm 2) is $\text{metrics} = [J, E, S]$. Here entailment is computed using BART (Lewis et al., 2020) as described in Appendix B. We prune nodes with $J > 0.75$ to limit hallucination.

We select $k = 5$ lexically-diverse paraphrases that maximise entailment given the constraint that the returned candidates should be lexically diverse. First, we select a syntactic paraphrase by traversing the subtree rooted at $J = 0$ and minimising S . The remainder of the candidates are selected by constraining the breadth-first traversal of the first level, which encodes lexical distance, to return the nodes sorted from high to low. This procedure is depicted schematically in Figure 1. We sort the ranked candidate lists for each description according to the Jaccard distance between them and the SGD descriptions. Hence we obtain $k = 5$ synthetic schema variants, with v_1 being the most similar to the SGD schema and v_5 the most dissimilar. We refer to this scheme as Pegasus + BART.

4.2 State tracking data augmentation

4.2.1 Baseline models

D3ST The Description-Driven Dialogue Modelling (D3ST) model (Zhao et al., 2022) is a state-of-the-art DST model that performs intent tracking, requested slots prediction, and state tracking in a

single pass. See Appendix C for a visual representation of inputs and targets. We process the data and train the model as described in Zhao et al. (2022) and Appendix C, selecting models that maximise the development set JGA.

T5DST We follow Lee et al. (2022) to implement a simplified T5DST (Lee et al., 2021a). It predicts the value of each slot iteratively, requiring a number of decoding passes equal to the number of slots in an API to predict the dialogue state given a dialogue history. Training and inference with this model is very expensive and we train the models with a fixed computational budget of 20,000 gradient steps⁴ for the baseline and 40,000 steps for all augmented data experiments. See Appendix C for prompt structure and implementation details.

4.2.2 Evaluation

On SGD, the JGA (JGA_{orig}) is computed for the 4,201 test set dialogues. 77% of these have a turn span where the agent calls an API unseen in training. Only 6 out of 21 schemata are seen in training.

Evaluation on SGD-X proceeds as follows. First, the SGD descriptions in the prompt are replaced, in turn, with descriptions taken from the five SGD-X variants. The DST model then predicts the state of a given dialogue 5 times, conditioned on prompts that are increasingly dissimilar to the SGD test set. Hence, the JGA_{v_1-5} figures reported are averages over approximately 21,000 conversations. For all experiments except *oracle* (see Section 4.2.3), *none* of the test time prompts are seen during training:

⁴This is the number of steps required for maximising the development set JGA, for all three runs.

the *seen* superscript in the metric names reported in Section 5 identifies conversations where the *SGD test set prompt is seen* during training. Therefore, it quantifies whether the model can robustly identify slots seen in training by interpreting the meaning of the descriptions rather than relying on linguistic patterns in the training schema. Meanwhile *unseen* measures the ability of the model to generalise to new APIs, which may describe new slots and domains, notwithstanding the language used by developers to phrase the descriptions. The JGA coefficient of variation (ie *schema sensitivity*, SS_{JGA}) as the prompt changes measures the sensitivity of a model to the prompt (Lee et al., 2021b).

Metric	Ranking	v1	v2	v3	v4	v5
Jaccard Dist	Pegasus + BART	17.1	63.0	69.5	72.0	76.6
	Backtranslation	18.2	29.9	43.9	-	-
	EDA	3.9	4.1	6.1	16.0	32.9
	SGD-X	55.6	65.6	71.2	78.1	85.7
Entailment	Pegasus + BART	99.0	96.7	94.9	94.6	94.4
	Backtranslation	97.5	96.5	95.9	-	-
	EDA	99.1	98.5	96.6	93.2	86.4
	SGD-X	89.7	88.0	88.4	86.8	87.5
BLEU	Pegasus + BART	13.4	12.5	12.5	13.2	12.7
	Backtranslation	36.4	26.0	18.9	-	-
	EDA	72.0	63.3	47.2	42.3	44.2
	SGD-X	20.4	15.3	10.8	8.3	5.2
self-BLEU	Pegasus + BART	-	12.0	11.4	11.0	10.9
	Backtranslation	-	49.3	41.7	-	-
	EDA	-	87.0	68.8	58.0	53.1
	SGD-X	-	13.5	11.2	9.9	8.6

Table 1: Automatic synthetic schema evaluation. J is multiplied by 100 for readability.

4.2.3 Experimental setup

We show our approach is effective by augmenting the DST training data with synthetic prompts composed from our generated schemata. To study the effect of controlling prompt diversity augmented datasets are two (2x) to six times (6x) the SGD size. For 2x, augmented data contains prompts constructed from the v1 synthetic schema, whereas for 6x we use all five generated schemas⁵.

Baselines We create three synthetic schema by **backtranslation**. Our pivot languages are Korean, Japanese and Chinese (Lee et al., 2022). The augmented DST training dataset is four times (4x) larger than SGD. Following Huang et al. (2021), we also consider French and Russian as pivot languages to generate two more synthetic schemas and obtain an augmented dataset six times (6x) larger than SGD. We also compare with **easy data augmentation** (EDA) (Wei and Zou, 2019), a word-level DA approach based on synonym replacement

⁵Ordering is from most (v1) to least (v5) similar to SGD.

(SR), random insertion, deletion and substitution. We perform SR with probability 0.25 and the other operations with equal probability of 0.05. Just like for backtranslation, we generate 3 or 5 synthetic schemas with this method via the public API. Augmentation with the **SGD-X** human schemata paraphrases is considered an *oracle* because these models see the SGD-X schemata at training time.

5 Results and Discussion

5.1 Synthetic schema generation

Our ranking method generates increasingly lexically diverse schemata as shown by the increase in Jaccard distance across schema variants (Table 1). This aspect is much more difficult to achieve with EDA without significantly affecting semantics. Furthermore, self-BLEU (Zhu et al., 2018) scores indicate EDA is the least effective in ensuring candidate diversity compared to other approaches. The BLEU difference between the SGD-X variants v1 and v5 is 15.2 but smaller (0.66) for our approach. Hence, the PEGASUS + BART copies n -grams from the input and includes additional information. This information is not always meaning-preserving: *City where the event is happening* is paraphrased as *The bustling city where the event is taking place* (v5) but *End date for the reservation or to find the house* is paraphrased as *End date for hotel reservation to allow time for a replacement both at the struck and in the run up to the event* (v5). The self-BLEU of the SGD-X schemas decreases faster compared to the automatically generated paraphrases, suggesting that Jaccard distance increases partly due to hallucination.

Entailment scores show that backtranslation is effective in preserving semantics. For EDA, the semantic similarity drops significantly as more candidates are generated since more dissimilar schemas are generated with more edit operations which are likely to affect meaning. The entailment scores for the SGD-X paraphrases are also lower since they do not always perfectly semantically overlap with the input by construction (Lee et al., 2022) and because of entailment model errors.

5.2 Dialogue state tracking

D3ST Both the robustness and robust generalisation are improved by augmentation with our synthetic schemas, as demonstrated by maximum JGA_{v1-5}^{seen} (12.35%) and JGA_{v1-5}^{unseen} (5.85%) increases and 23.6% drop in SS_{JGA} (rows 1&4, Ta-

Model	#	Generation method - Dataset size	JGA _{orig} ↑	JGA _{v1-5}	JGA _{v1-5} ^{seen}	JGA _{v1-5} ^{unseen}	SS _{JGA} ↓
D3ST	1	None - 1x	69.8	56.5	73.6	50.8	70.1
	2	Pegasus + BART - 2x	72.8	61.3	80.9	54.8	56.4
	3	Pegasus + BART - 4x	72.6	62.5	81.7	56.1	51.0
	4	Pegasus + BART - 6x	71.2	63.9	85.9	56.6	46.5
	5	EDA - 4x (Wei and Zou, 2019)	71.0	59.0	78.5	52.5	63.0
	6	EDA - 6x (Wei and Zou, 2019)	71.4	62.3	83.3	55.3	53.2
	7	Backtranslation - 4x (Lee et al., 2021b)	72.1	62.2	84.0	54.9	53.1
	8	Backtranslation - 6x (Huang et al., 2021)	71.5	61.0	82.5	53.8	54.4
	9	SGD-X - 6x (Lee et al., 2021b) (Oracle)	73.8	69.7	92.5	62.1	27.9
T5DST	10	None	70.0	50.4	58.5	47.7	87.0
	11	Pegasus B + BART - 4x	71.3	55.1	71.2	49.7	70.1
	12	Pegasus + BART - 6x	68.7	52.5	71.6	46.5	77.6
	13	EDA - 6x (Wei and Zou, 2019)	72.2	51.1	55.6	49.6	84.1
	14	Backtranslation - 4x (Lee et al., 2021b)	72.8	53.9	67.0	49.6	76.4
	15	SGD-X - 6x (Lee et al., 2021b) (Oracle)	74.2	67.2	91.8	59.0	36.6

Table 2: SGD and SGD-X dialogue state tracking performance when training with augmented data. Best performance (excluding the oracle setup) is in **bold**. Dataset size is the number of times the augmented dataset is larger than SGD.

ble 2). The most benefit is obtained by training with syntactically diverse prompts (Pegasus + BART 2x). Adding more diverse data (rows 3&4) improves DST performance. Part of this improvement may arise because paraphrasing leaves out domain-dependent information: *Average review rating of the doctor* is paraphrased as *The rating is average, so it's not perfect*, so the model can learn to identify ratings more generally⁶. Moreover, inputs are noisy due to hallucination, so the models trained with our augmentation are less likely to overfit to the linguistic patterns of the training schemas. BLEU scores indicate high lexical overlap between EDA-generated and SGD schemas (Table 1). This limits the magnitude of EDA improvement (row 5) and we perform better with less data (rows 3&6, 2&5).

Backtranslation is comparable with our method given the same data quantity (rows 3&7). When we also backtranslate via French and Russian (Huang et al., 2021) the data diversity does not significantly increase (Table 5, Appendix D.1). This negatively impacts the DST performance, while our method improves it (rows 4&8). We can control the schema generation process to match SGD backtranslation performance (Appendix E).

T5DST⁷ We outperform EDA (rows 12&13) but not backtranslation (row 14). This may be due to (1) the larger computational budget needed to maximise T5DST performance⁸ and (2) T5DST's sensitivity to noisy descriptions owing to its prompt format (Appendix C). We control hallucination by pruning candidates with $J > 0.5$ and entailment

smaller than 0.58 and maximise J while minimising S to produce an augmented dataset 4x larger than SGD (Pegasus B + BART 4x). Limiting lexical diversity improves entailment compared to Pegasus + BART 6x (Table 6, Appendix D.1), and the scheme improves DST robustness compared to the backtranslation baseline (rows 11&14).

The best augmentation schemes fail to improve robustness and generalisation relative to the human baseline (rows 9&15). This is due to the intrinsic challenge of generating diverse yet semantically faithful paraphrases but also due to the fact that humans use common sense and schema information when paraphrasing, so the SGD-X paraphrases are not strictly semantically equivalent. However, the proposed automatic process of paraphrase generation enhances DST, yielding non-trivial improvements in model robustness, while being less costly and more scalable compared to gathering human-written schemata paraphrases.

6 Conclusion and Future Work

We presented a simple tree-based ranking algorithm for optimising lexical diversity and semantic faithfulness during schema generation. The synthetic schemas improve both the DST models' robustness to schemata writing style and their generalisation. Our framework will allow researchers working on paraphrase generation and semantic faithfulness to measure the generalisation of their models in a way that may be difficult to capture by existing benchmarks: it can generate schemata paraphrases and train SOTA dialogue state trackers which were shown to benefit from augmentation with high quality, crowdsourced paraphrases.

⁶It appears in 4 unseen services in the test set.

⁷Lee et al. (2021b) report 72.6% JGA on SGD and 64.0% SGD-X but we could reproduce only 69.98% and 50.42%.

⁸Each training example is seen only once.

Limitations

The optimality of our ranking method depends on the ability of the underlying paraphrase model to generate a search space that contains paraphrases which are lexically and syntactically diverse and preserve the meaning of the input description. This is sometimes challenging with schema inputs which tend to be short (e.g. *name of event*) and contain little information. Our future work will focus on addressing this by contextualising these inputs to enable the paraphrase model to produce a richer space of candidates. Secondly, our method requires that the semantic faithfulness metrics capture semantic similarity well even as the vocabulary of the candidates and their syntax are very diverse. Previous work on abstractive summarisation (Narayan et al., 2022; Maynez et al., 2020; Kryscinski et al., 2019) finds entailment scores to be best correlated with human judgment of faithfulness. However, the correlations are not perfect so the output of the ranking algorithm is still expected to contain noisy candidates. For slot description paraphrases, this is challenging because different inputs are very closely semantically related and the entailment model may not identify paraphrase model errors that map a slot description (e.g. departure time) to one with related semantics (e.g. arrival time). We intend to address this in future work by developing finetuning schemes for semantic faithfulness metrics.

Ethics Statement

Our work is concerned with the use of language generation models to augment training datasets for schema-guided dialogue datasets. The generation phase is unconstrained, so the model may generate candidates that exhibit biases inherited from the C4 (Raffel et al., 2020) and HugeNews (Zhang et al., 2020a) pre-training datasets. In our experiments, we did not observe toxic or harmful outputs, but on one occasion the model did generate the word *apartheid* as part of an incoherent sentence. For this reason, our filtering stack rejects any candidates containing sensitive words. The list of words that parameterize the sensitive words filter is defined by the user.

Acknowledgements

Alexandru Coca was supported supported by EP-SRC grant EP/R513180/1. He would like to acknowledge Harrison Lee and Raghav Gupta from

Google Research for support and guidance with D3ST and T5DST implementation. Weizhe Lin was supported by a Research Studentship funded by Toyota Motor Europe (RG92562(24020)). We also thank Howard Mei from University of Cambridge for help with editing the final draft. Authors would like to acknowledge the improvement suggestions made by anonymous reviewers and the EACL program committee.

References

- Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein-Dor. 2022. [Quality controlled paraphrase generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 596–609. Association for Computational Linguistics.
- Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S. Lam. 2020. [Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 122–132. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Arash Einolghozati, Sonal Gupta, Mrinal Mohit, and Rushin Shah. 2019. [Improving robustness of task oriented dialog systems](#). *CoRR*, abs/1911.05153.
- Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. [Paraphrase augmented task-oriented dialog generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 639–649. Association for Computational Linguistics.
- Milan Gritta, Gerasimos Lampouras, and Ignacio Iacobacci. 2021. [Conversation graph: Data augmentation, training and evaluation for non-deterministic dialogue management](#). *Trans. Assoc. Comput. Linguistics*, 9:36–52.
- Matthew Henderson, Blaise Thomson, and Steve J. Young. 2014. [Word-based dialog state tracking with recurrent neural networks](#). In *Proceedings of the*

- SIGDIAL 2014 Conference, The 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 18-20 June 2014, Philadelphia, PA, USA*, pages 292–299. The Association for Computer Linguistics.
- Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. [Sequence-to-sequence data augmentation for dialogue language understanding](#). In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1234–1245. Association for Computational Linguistics.
- Shuo Huang, Zhuang Li, Lizhen Qu, and Lei Pan. 2021. [On robustness of neural semantic parsers](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 3333–3342. Association for Computational Linguistics.
- Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. [Neural text summarization: A critical evaluation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.
- Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021a. [Dialogue state tracking with a language model using schema-driven prompting](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4937–4949. Association for Computational Linguistics.
- Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2021b. [SGD-X: A benchmark for robust generalization in schema-guided dialogue systems](#). *CoRR*, abs/2110.06800.
- Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2022. [SGD-X: A benchmark for robust generalization in schema-guided dialogue systems](#). In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10938–10946. AAAI Press.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.
- Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2021. [Robustness testing of language understanding in task-oriented dialog](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2467–2480. Association for Computational Linguistics.
- Samuel Louvan and Bernardo Magnini. 2020. [Simple is better! lightweight data augmentation for low resource slot filling and intent classification](#). In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation, PACLIC 2020, Hanoi, Vietnam, October 24-26, 2020*, pages 167–177. Association for Computational Linguistics.
- Yue Ma, Zengfeng Zeng, Dawei Zhu, Xuan Li, Yiyang Yang, Xiaoyuan Yao, Kaijie Zhou, and Jianping Shen. 2019. [An end-to-end dialogue state tracking system with machine reading comprehension and wide & deep classification](#). *CoRR*, abs/1912.09297.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1906–1919. Association for Computational Linguistics.
- Biswesh Mohapatra, Gaurav Pandey, Danish Contractor, and Sachindra Joshi. 2021. [Simulated chats for building dialog systems: Learning to generate conversations from instructions](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 1190–1203. Association for Computational Linguistics.
- Shashi Narayan, Gonçalo Simões, Yao Zhao, Joshua Maynez, Dipanjan Das, Michael Collins, and Mirella Lapata. 2022. [A well-composed text is half done! composition sampling for diverse conditional generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1319–1339. Association for Computational Linguistics.
- Jun Quan and Deyi Xiong. 2019. [Effective data augmentation approaches to end-to-end task-oriented dialogue](#). In *International Conference on Asian Language Processing, IALP 2019, Shanghai, China, November 15-17, 2019*, pages 47–52. IEEE.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. [Towards](#)

- scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8689–8696. AAAI Press.
- Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. **BLEURT: learning robust metrics for text generation**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7881–7892. Association for Computational Linguistics.
- Jason W. Wei and Kai Zou. 2019. **EDA: easy data augmentation techniques for boosting performance on text classification tasks**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6381–6387. Association for Computational Linguistics.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. **A broad-coverage challenge corpus for sentence understanding through inference**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. **Huggingface’s transformers: State-of-the-art natural language processing**. *CoRR*, abs/1910.03771.
- Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. **Dialog state tracking with reinforced data augmentation**. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9474–9481. AAAI Press.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. **PEGASUS: pre-training with extracted gap-sentences for abstractive summarization**. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.
- Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020b. **Task-oriented dialog systems that consider multiple appropriate responses under the same context**. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9604–9611. AAAI Press.
- Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. **Description-driven task-oriented dialog modeling**. *CoRR*, abs/2201.08904.
- Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. **Texygen: A benchmarking platform for text generation models**. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM.

A The SGD and SGD-X datasets

SGD As mentioned in Section 1, the conversations in the SGD dataset are grounded in schemata, which describe a set of service APIs. The most important schema *elements* are⁹:

- a *service name* (e.g. *Messaging_1*) followed by a *service description* (e.g. *Connect and share locations with your contacts*)
- one or more API functions to be invoked as users solve tasks, referred to as (*user*) *intents*; each intent has a *name* (e.g. *ShareLocation*) and an *intent description* (e.g. *Send your location to a contact*)
- optional and required arguments for each API function, or *slots*; each slot has a *name* (e.g. *location*) and a *slot description* (e.g. *Location to share with the contact*)

SGD-X Lee et al. (2022) observe that 71% of intent names and 65% of slot names from unseen APIs exactly match the train set. Furthermore, descriptions are stylistically uniform across the train and test sets. For example, all boolean slots begin with the phrase *Boolean flag ...* or *Whether...*. Therefore, they create the SGD-X dataset as follows:

- crowdsource schema element paraphrasing to more than 400 authors via Amazon Mechanical Turk. Each crowdworker either paraphrases all names or all descriptions for a given schema

⁹Examples below are taken from the SGD test set.

- manually vet responses for quality and correctness.

The slot names collected are sorted in increasing order of their Levenshtein distance to the SGD slot names whereas the descriptions are sorted according to the Jaccard distance between their lemmatized forms (excluding stop words). An example of SGD-X description paraphrases is shown in Table 3.

Variant	Description
SGD	Category to which the attraction belongs
v1	The category that describes what kind of attraction it is
v2	Category of place of interest
v3	Type of tourist attraction
v4	Choose the kind of tourist landmark
v5	The kind of tourist hotspot

Table 3: Example of descriptions paraphrases from the SGD-X test schemas. The more similar v1 description contains overlapping vocabulary with the SGD test set description, whereas v4 and v5 variants are dissimilar both stylistically and lexically

While the examples above are paraphrases of the SGD input, in general, the semantic content of the schema element paraphrases is not perfectly overlapping with the input as the crowdworkers use information from the wider service context when creating new elements.

B Ranking Framework

B.1 Candidate generation

Algorithm 2 summarises the candidate generation procedure, which takes any paraphrase model, a list of model-specific generation parameters and, optionally, a list of filters as an input (line 1). These parameters are temperature and number of beams for Pegasus, or a grid of lexical, semantic and syntactic distances for the Quality Controlled Paraphrase Generation (QCPG) (Bandel et al., 2022) model presented in Appendix E. The model generates one or more paraphrases, which are filtered before returning (lines 3-8). We describe the filtering process next.

Heuristic filtering Our main motivation for implementing heuristic filters is to filter the majority of poor quality candidates, without making use of the large GPU cards required to run the entailment model. We also address the fact that the model is free to generate a very large number of candidates and therefore is expected to hallucinate significantly. These filters are general purpose and are

implemented in few lines of code using the spaCy and nltk libraries. Table 4 lists active filters along with typical examples filtered.

Entailment filtering We implement our entailment filter using BART (Lewis et al., 2020)¹⁰. This model is pre-trained on the MNLI dataset (Williams et al., 2018). To measure entailment this model consumes a premise and hypothesis in the format premise <SEP> hypothesis. In our implementation we replace premise with the description to be paraphrased. By default, the hypothesis is a template of the form This example is {}, where {} is a placeholder for the user hypothesis, in our case the paraphrased description. We find that considering alternative templates improves the reliability of the model, so we consider {}, This example has the same meaning as {}, This text is about {}, and This example implies that {}, averaging the entailment scores across templates to calculate the entailment score. The same procedure is followed when computing the entailment of candidates during ranking.

Algorithm 2: Candidate generation

```

1: def generate_candidates(model: Any,
   inp: str, params: dict, filters:
   Optional[list[Callable]]):
   Data: model text generation model, inp
           input sentence, params model
           specific parameters, filters a
           list of boolean functions
   Result: cand_s list of inp paraphrases
2: cand_s ← [] ;
3: for p in params:
4:   c ← model.forward(inp, **p) ;
5:   c ← [p for p in c if not any(f(p,
   inp) for f in filters)]
6:   cand_s.extend(c)
7: return cand_s

```

B.2 Ranking

Ranking Algorithm 3 summarises the tree-ranking procedure. This procedure takes as an input the tree constructed as described in Algorithm 1, along with a list of decision functions f_{dec} . Our algorithm starts by selecting a paraphrase via depth first traversal of the subtree rooted at $J = 0$ (line 2). The remainder of the candidates are selected by

¹⁰Available at <https://huggingface.co/facebook/bart-large-mnli>.

Filter name	Filtered example
contains advice	An appointment is necessary for your hair .
describes action	They commemorate the number of flights to the airport.
has named entities	Enter the doctor’s Leningrad address.
has low frequency words	The address is ofadvisory .
discard multiple sentences	The address is the dentist’s box. Guidelines for hiring a dentist.
has repeated ngrams	The dentist is Address of the dentist .
has repeated similar bigrams	The type of event is stated in the title of the event .
has consecutive repeated words	Average review rating for a hotel hotel.
is past tense sentence	It was the dentist’s address.
is passive voice sentence	The address was given by abrasives from the dentist.
is question	Is there a balance of the account?
has alphanumeric words	400 baths in an apartment.

Table 4: Filters implemented along with sample examples they discard

Algorithm 3: Tree ranking

```

1: def tree_rank(root: Node, n: int, f_dec:
    list[Callable]):
    Data: root, n number of candidates, f_dec
        decision functions
    Result: list of n ranked candidates
2: ranked ← syntax_select (root);
3: n ← n - len(ranked);
4: while len(ranked) ≠ n:
5:     for next in level_order (root):
6:         for f in f_dec:
7:             cand ← select (next.sents);
8:             ranked.add(cand);
9:             prune (next, cand);
10: return ranked

```

traversing the first level in a breadth-first manner (line 5) and depth-first traversal of each subtree returned during the level-order traversal (lines 6-8). Here the semantics of $f(\text{next.children})$ is that the decision function f takes all the children of next as input and returns a single node which is next in the traversal. A candidate is selected from the leaf¹¹ (line 8) and subsequently removed from the candidates list (line 10). This is to avoid selecting the same candidate multiple times in situations where the paraphrase model generates few distinct candidates.

C State Tracking Baselines

D3ST We process the data as described by Zhao et al. (2022) with the following differences:

- The indices are separated by the = symbol in both the inputs and the targets, to avoid a

¹¹There can be multiple, possibly repeated candidates in a leaf because the generative model may generate the same output given different parameter settings. We select the most common one if there are repeated candidates and randomly otherwise.

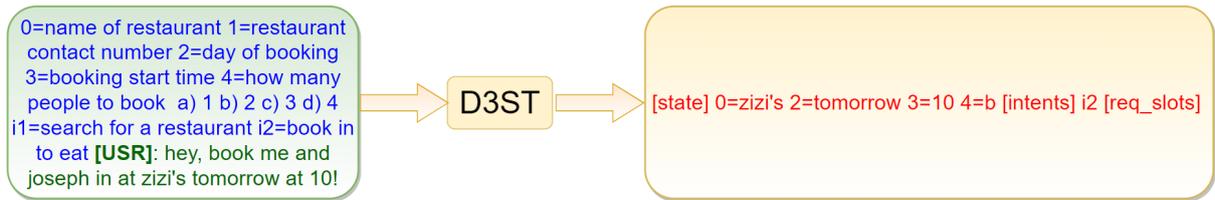
parsing ambiguity which occurs for time slots if : is used as a separator for targets

- For categorical slots which take the dontcare special value, our output contains slot_index: dontcare substring and we do not include the dontcare value in the prefix together with the other options
- We lowercase the inputs and the targets¹².

We obtain 175,780 examples from the original SGD dataset, which are truncated to the last 1,024 tokens on the input side. See Figure 2a for a visual representation of the model inputs and outputs. We optimise the model using the Adafactor optimizer and effective batch size 32, starting from the initial weights google/t5-v1_1-base published by huggingface (Wolf et al., 2019). We interpolate the learning rate linearly between 0 and 10^{-4} over the first 1000 steps and keep it constant thereafter. We select the model by evaluating the development set JGA every 5000 gradient updates, stopping the training if said metric fails to improve after 3 consecutive evaluations. All numbers in Table 2 are averages of 3 runs, except the SGD-X experiment for T5DST which is a single run.

T5DST Given a dialogue in the SGD training set we consider all partial dialogue histories $\{u_1, s_1, \dots, s_{t-1}, u_t\}$ with $t \in \overline{0, T}$ where T is maximum index of the user turn in a dialogue. The turns in each dialogue history are lowercased and separated [usr] and [sys] tokens, *not treated* as special tokens. For each dialogue history we create a training example for each slot in the ground truth schema, which contains the concatenated turns suffixed with the string

¹²This appears in illustrations but is not explicitly stated by (Zhao et al., 2022).



(a) Visual representation of D3ST inputs and targets. Blue font `blue`, preceding the `[USR]` special token represents the prompt, consisting of slot descriptions extracted from the schema. Each slot is assigned an index, which is used to recover the slot value pairs during post-processing. Note that slot 4 is categorical, and so the string `a) 1 b) 2 c) 3 d) 4` is appended to the description to indicate the model that it should output one of the choices. The dialogue history follows the `[USR]` token. The **model outputs** only active slots, in this case omitting slot 1 because it has not been mentioned. The entire dialogue state (as opposed to turn state) is generated at every turn.

```
input: " [user] can you find me something economical to eat in pleasanton. [slot] name of the restaurant"
output: none
```

```
input: " [user] can you find me something economical to eat in pleasanton. [slot] price range for the
restaurant a) expensive b) moderate c) inexpensive d) very expensive"
```

```
output: b
```

```
input: " [user] can you find me something economical to eat in pleasanton. [slot] city in which the restaurant
is located"
```

```
output: pleasanton
```

...

(b) Visual representation of T5DST inputs and targets. The string `[slot]` separates the dialogue history from the slot description. In the first example the model outputs the special value `none` to indicate that the slot was not mentioned by the user. For categorical slots (second row from the top), the slot description is concatenated with options containing all possible slot values and the model predicts the correct option. For non-categorical slots (third row), the exact value is predicted. The ellipsis indicates that `none` is predicted for all other slots in the `Restaurant_1` schema that are not mentioned in the dialogue history

Figure 2: Prompt formats for a) D3ST b) T5DST

`[slot] [slot_description]` where the placeholder `[slot_description]` is replaced by the lowercase descriptions extracted from the SGD schemata. This yields 1,601,356 examples for the SGD training dataset. See Figure 2b for a representation of the model inputs and outputs.

We optimise the model using the Adafactor optimizer and effective batch size 256, starting from the initial weights `google/t5-v1_1-base` published by huggingface. We interpolate the learning rate linearly between 0 and 10^{-4} over the first 1000 steps and keep it constant thereafter. We perform 20,000 optimisation steps¹³, limiting the number of training steps to 40,000 steps¹⁴ for all augmented data experiments. All numbers in Table 2 are averages of 3 runs, except the Oracle experiment on T5DST which is a single run.

¹³For a single run, this is approximately 6 hours of computation on 8 nvidia A100-80GB cards. Moreover, decoding a single run on SGD and SGD-X takes 6 hours.

¹⁴This is sufficient so that the model sees every example once when working with an augmented training set six times the size of SGD.

D Additional Results

D.1 Increasing backtranslation dataset size

We include Table 5 to substantiate our intuition that the training with the Backtranslation 6x scheme does not yield further improvement compared to the Backtranslation 4x scheme as the additional data does not significantly increase the prompt diversity. Most clearly, this is indicated by the fact that the `v5` variant has similar BLEU to variant `v3` in Backtranslation 4x, indicating that a large proportion of additional data has some overlaps more with the SGD distribution than the data backtranslated to Chinese, Korean and Japanese. This is also indicated by how self-BLEU decays as more data is added, comparatively, between Backtranslation 4x and Backtranslation 6x.

D.2 Controlling schema generation diversity

Table 6 shows that the alternative schema scheme generates schemas with lower average Jaccard distance and higher entailment with respect to the SGD schemata. We find this effectively controls the noise in the data, leading to improved performance

Metric	Ranking	v1	v2	v3	v4	v5
Jaccard Dist	Backtr. 4x	18.2	29.9	43.9	-	-
	Backtr. 6x	12.9	22.7	27.8	35.6	46.7
Entailment	Backtr. 4x	97.5	96.5	95.9	-	-
	Backtr. 6x	98.0	97.5	95.2	94.8	95.5
BLEU	Backtr. 4x	36.4	26.01	18.9	-	-
	Backtr. 6x	51.3	37.2	29.5	23.4	18.2
self-BLEU	Backtr. 4x	-	49.3	41.7	-	-
	Backtr. 6x	-	55.3	49.7	44.6	39.6

Table 5: Effect of using French and Russian as additional pivot languages on automatic metrics

Metric	Ranking	v1	v2	v3	v4	v5
Jaccard Dist	Pegasus + BART	13.0	61.2	68.8	71.2	76.2
	Pegasus B + BART	10.2	38.3	46.9	55.1	54.5
	SGD-X	55.6	65.6	71.2	78.1	85.7
Entailment	Pegasus + BART	99.1	96.3	94.6	94.2	94.2
	Pegasus B + BART	98.8	98.2	96.4	96.2	96.7
	SGD-X	89.7	88.0	88.4	86.8	87.5

Table 6: Comparison of diversity and semantic faithfulness metrics for slot description paraphrases

Index	Augmentation	JGA _{orig}	JGA _{v1-5}	JGA _{v1-5} ^{seen}	JGA _{v1-5} ^{unseen}	SS _{JGA}
1	Pegasus+BART 6x	71.2	<u>63.9</u>	<u>85.9</u>	56.6	46.5
2	Pegasus+BLEURT 6x	<u>72.4</u>	64.0	86.6	<u>56.4</u>	<u>46.6</u>
3	QCPG+BLEURT 6x	72.7	63.2	85.2	55.9	47.3
4	Backtranslation 4x (Lee et al., 2021b)	72.1	62.2	84.0	54.9	53.1

Table 7: Ranking with a more accurate semantic faithfulness metric (row 2) or generating candidates with a controllable paraphrase model (row 4) can be used to boost SGD performance over our Pegasus+BART approach (row 1). Bold font marks column maximum, underlined second largest number.

for T5DST and similar performance to PEGASUS + BART for D3ST.

E Schema Generation with BLEURT and QCPG

BLEURT (Sellam et al., 2020) is a BERT-based natural metric commonly used in translation, so it is expected to be highly sensitive to semantic differences. In Table 7 we show that simply re-ranking the Pegasus output space with BLEURT improves SGD performance comparably with backtranslation (rows 2&4) and the robustness and generalisation improvements are maintained.

Bandel et al. (2022) exploit high quality examples in paraphrase corpora by conditioning the model with a string *quality parameters string* outlining target semantic, syntactic and lexical distances of the generated paraphrase during finetuning. At inference one must specify these parameters to obtain diverse yet high quality paraphrases. We could not apply the quality parameter selection method proposed by QCPG authors at inference time as the code had not been fully released at the time of writing. Instead, we generated a large number of paraphrases with different quality targets and greedy decoding, and re-ranked the candidates using our framework. This demonstrates the versatility of our framework. In Table 7 we show that this model can equally achieve improved performance on SGD. The improvement on SGD-X is slightly less than achieved by PEGASUS+BART 6x, as expected since greedy decoding and better semantic faithfulness optimisation generate schemata closer

to the SGD distribution so less out-of-distribution improvement is achieved.

This experiments in this section and Appendix D.2 demonstrate the versatility of our framework and its usefulness as a tool for generating synthetic schema prompts.

Language Model Decoding as Likelihood–Utility Alignment

Martin Josifoski,[◇] Maxime Peyrard,[◇] Frano Rajic,[◇] Jiheng Wei,[♣]
Debjit Paul,[◇] Valentin Hartmann,[◇] Barun Patra,[♣] Vishrav Chaudhary,[♣]
Emre Kıcıman,[♣] Boi Faltings,[◇] Robert West[◇]

[◇]EPFL [♣]Microsoft Corporation [♣]PSL University
{martin.josifoski, maxime.peyrard, robert.west}@epfl.ch

Abstract

A critical component of a successful language generation pipeline is the *decoding algorithm*. However, the general principles that should guide the choice of a decoding algorithm remain unclear. Previous works only compare decoding algorithms in narrow scenarios, and their findings do not generalize across tasks. We argue that the misalignment between the model’s *likelihood* and the task-specific notion of *utility* is the key factor to understanding the effectiveness of decoding algorithms. To structure the discussion, we introduce a taxonomy of misalignment mitigation strategies (MMSs), providing a unifying view of decoding as a tool for alignment. The MMS taxonomy groups decoding algorithms based on their implicit assumptions about likelihood–utility misalignment, yielding general statements about their applicability across tasks. Specifically, by analyzing the correlation between the likelihood and the utility of predictions across a diverse set of tasks, we provide empirical evidence supporting the proposed taxonomy and a set of principles to structure reasoning when choosing a decoding algorithm. Crucially, our analysis is the first to relate likelihood-based decoding algorithms with algorithms that rely on external information, such as value-guided methods and prompting, and covers the most diverse set of tasks to date. Code, data, and models are available at <https://github.com/epfl-dlab/understanding-decoding>.

1 Introduction

Large transformer-based *language models* (LMs) have been pushing the boundaries on tasks ranging from natural language generation (Radford et al., 2018) to information extraction (Josifoski et al., 2022), theorem proving (Polu and Sutskever, 2020), code generation (Zügner et al., 2021), and even protein generation (Ferruz et al., 2022). At inference time, these models rely on a decoding algorithm to generate an output. The goal of decoding algorithms is to select an output of high utility from

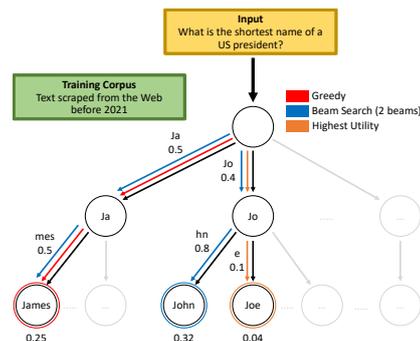


Figure 1: **Example of likelihood–utility misalignment.** Imagine a fictional LM trained before Joe Biden became the US president. The input asks for the shortest name of a US president. After Joe Biden’s inauguration, this is ‘Joe’, but before, it was ‘John’. Greedy search returns ‘James’ since its first token ‘Ja’ has the highest likelihood. Beam search manages to find the highest likelihood sequence ‘John’. Both fail to find the correct answer ‘Joe’ with the highest utility since ‘Joe’ has a very low likelihood.

the exponentially large output space. In contrast to the generic language modeling training objective, which is based on the data likelihood, the notion of utility is task-specific. The potential gap between the two can create a *misalignment* between *model likelihood* and *task utility*; see Fig. 1 for an illustration of this concept.

Indeed, across different tasks, researchers noticed that high likelihood is often not associated with desired properties of the output (Stahlberg and Byrne, 2019; Zhang et al., 2021; Klein et al., 2017). Naturally, this has led to the development of decoding strategies aimed at mitigating this problem. In the context of natural language generation (NLG), Nucleus Sampling (i.e., top- p) (Holtzman et al., 2020) has been proposed to avoid dull or degenerate text. Similarly, in the context of machine translation (MT), solutions ranging from simple ad-hoc tweaks like enforcing a minimal sequence length (Stahlberg and Byrne, 2019) to leveraging a value model to directly optimize for utility in de-

coding (Leblond et al., 2021) have been developed. These methods for alignment are task-specific and have been tested only in narrow domains, making it difficult for practitioners to compare them.

Recently, Meister et al. (2022) and Wiher et al. (2022) explored the likelihood–utility misalignment across tasks. However, these studies still largely focus on a group of similar tasks — NLG tasks — and, more crucially, do not include decoding strategies that make use of external sources of information at inference time. Therefore, a general framework to structure our thinking about decoding algorithms is still missing.

Our work makes the first step towards filling this gap. We propose a unified perspective of decoding as a tool for mitigating the likelihood–utility misalignment without modifications to the model. Looking at decoding through this functional lens, in Sec. 3, we provide a taxonomy of misalignment mitigation strategies (MMSs). The taxonomy groups decoding algorithms based on the implicit assumptions about likelihood–utility misalignment that need to hold for them to be effective.

Equipped with this taxonomy, we conduct a comprehensive empirical analysis in which we choose a representative set of decoding algorithms and a representative set of tasks to cover the relevant types of misalignment. We identify three main sources of misalignment: *training imperfections* (finite dataset, differentiable surrogate loss), *distribution shift*, and changes in the model’s intended usage, which we call *utility drift*. Then, we measure the likelihood–utility misalignment across tasks (RQ1) by estimating: the correlation between likelihood and utility for the generated outputs after decoding (RQ1-a) and the correlation between likelihood and utility among candidate outputs explored by the decoding algorithm (RQ1-b). We proceed by investigating the benefits of decoding algorithms that leverage external information at inference (RQ2). Finally, we experiment with large generalist LMs (LLMs) and show that prompting can be seen as means for improving the alignment at inference time (RQ3).

Our experiments reveal that: (i) When no distribution shift or utility drift happens, decoding based solely on the likelihood is enough to provide high utility, i.e., likelihood is a strong predictor of utility. (ii) In such cases, there is no significant difference between different kinds of decoding algorithms, and we would recommend keeping beam search.

(iii) In the presence of distribution shift or utility drift, value-guided beam search is both an effective and efficient decoding algorithm that leverages a value model at inference time to fix misalignment. Finally, (iv) for LLMs, prompting is a mechanism that sets the model in a state where the likelihood is well-aligned with the utility. This perspective provides a tentative explanation for the empirical success of prompting LLMs.

This work studies the fundamental problem in decoding, which involves a complex interaction between models, tasks, and data. Our unifying conceptual framework (the MMS taxonomy), accommodating all known decoding algorithms, enables the systematic study of decoding in a considerably broader scope than previously. By sharing the taxonomy and open-sourcing our implementations, we hope to pave the way for a more structured discussion in the future.

2 Background

High Utility Is the Goal. For a task t and input x , the utility function assigns a score $u_t(y|x)$ to each element y in the output space \mathcal{Y} . This score quantifies the goodness, or quality, of the output y with respect to a specific input x . For instance, in translation, the utility quantifies the extent to which the output conveys the same message as the input. For question answering, the utility simply quantifies the correctness of the answer. These task-specific notions of utility are operationalized in the evaluation metrics. The development of evaluation metrics that correlate with the human-defined notion of utility is a very active research area (Sai et al., 2022) and beyond the scope of this work. In our analysis, we use the canonical evaluation metric of each task as the utility function. For partially-decoded sequences, the utility can be approximated using a *value function* (see Sec. 3.3).

Given some input x , an *ideal model* would generate the element from the output space corresponding to the highest utility score: $\operatorname{argmax}_{y \in \mathcal{Y}} u_t(y|x)$.

Unfortunately, most of the practically relevant utility functions are not amenable to optimization, forcing us to work with proxy functions, such as the canonical likelihood.

Language Models. A language model corresponds to a probability distribution p over $y \in \mathcal{Y}$, where \mathcal{Y} is the set of all sequences that can be constructed using a vocabulary \mathcal{V} . In this work, we focus on conditional LMs $p(\cdot|x)$. Usually, these

conditional distributions are modeled autoregressively (parametrized by θ): $p_\theta(y|x) = \prod_{i=1}^{|y|} p_\theta(y_i | y_{<i}, x)$. The model is trained to maximize the target sequence’s conditional log-likelihood with teacher forcing, using the cross-entropy loss $\mathcal{L}(\theta) = -\sum_{(x,y) \in \mathcal{D}} \log p_\theta(y|x)$, where \mathcal{D} is the training corpus (Sutskever et al., 2011, 2014).

Once an LM is trained, it provides a next-token probability distribution across the output vocabulary. Decoding algorithms define how tokens are chosen during generation.

3 Proposed MMS Taxonomy

In this section, we propose a taxonomy of misalignment mitigation strategies (MMSs). This work focuses on decoding-based MMSs that mitigate the misalignment without modifying the model. As a primary signal, they rely on the model’s likelihood. However, additional components (e.g., value model, knowledge base, etc.) can be leveraged. Decoding algorithms, which we define as procedures that take in an input — and potentially some context (e.g., a prompt with a task description or examples) — and return a sequence from the output space, can be seen as specific implementations of an MMS. Apart from fixing the misalignment problem at inference time, it is also possible to retrain or finetune the model with newly collected data that better reflect the intended utility and target testing distribution. We leave the detailed treatment of this part of the taxonomy for future work. See the Limitations section at the end of the writing for further discussion of these alternatives.

3.1 Greedy Likelihood-Based Strategy

Given the LM’s probabilistic formulation, one could strive to select the most likely sequence under the model: $\operatorname{argmax}_{y \in \mathcal{Y}} p_\theta(y|x)$. However, due to the exponentially large state space, this optimization problem is intractable.

The class of algorithms following the greedy likelihood-based MMS approximate the intractable argmax by following the greedy heuristic of making *locally* optimal choices at each decoding step w.r.t. the *likelihood* under the language model. However, reaching a globally optimal solution may require locally sub-optimal steps. When this happens, we say that the likelihood landscape is *greedy adversarial* (Meister et al., 2020). For a likelihood model that is not greedy adversarial, greedy heuristics will retrieve the highest-likelihood solution.

Therefore, the algorithms’ effectiveness depends on the likelihood–utility alignment.

Contrarily, greedy decoding algorithms may fall arbitrarily short of the global maximum for likelihood models that are greedy adversarial. Indeed, greedy decoding algorithms implicitly optimize a different objective function — a *tampered* version of the *likelihood* objective in which a term that encourages locally optimal solutions is added (Meister et al., 2020). Therefore, the ability of greedy decoding algorithms to retrieve high-utility sequences even from a likelihood model that is perfectly aligned with the utility is inversely proportional to how greedy adversarial the likelihood landscape is. In some cases, the particular bias induced by the greedy heuristic *mitigates* the likelihood–utility *misalignment*, and makes the tampered likelihood objective better aligned with the utility than the original (Meister et al., 2020, 2023; Su et al., 2022).

The decoding algorithms in this category can be further divided into two subgroups: (i) deterministic ones, such as greedy search (GS) and beam search (BS); and (ii) stochastic ones, such as top- k sampling (Fan et al., 2018), top- p sampling (Holtzman et al., 2020), and stochastic beams (SB) (Kool et al., 2019). For more details, see Appendix A.1.

3.2 Greedy Likelihood-Based Strategy with Pruning

An understated fact in the literature is that even for tasks for which the canonical decoding algorithms (e.g., BS) perform well, a non-negligible portion of the performance relies on some bespoke, ad-hoc tweaks on the likelihood scores (Stahlberg and Byrne, 2019). These tweaks are usually based on either: (i) post-hoc observations that likelihood-based decoded sequences often contain specific undesirable patterns (e.g., empty or short sequences, repetitive patterns, etc.); or (ii) problem-specific knowledge about the utility landscape suggests that high-utility sequences have a specific property, which can be explicitly enforced by the decoding strategy (e.g., sequences should correspond to triplets of elements from a predefined set). Conceptually, all these tweaks employ mechanisms that *discourage* the generation of *high-likelihood* patterns that are known (or expected) to be associated with *low utility*.

This category includes: (i) decoding algorithms with ad-hoc heuristics such as the n -gram repetition

penalty (Klein et al., 2017); (ii) constrained beam search (CBS) (Scholak et al., 2021; De Cao et al., 2022; Josifoski et al., 2022); (iii) NeuroLogic (Lu et al., 2022). For more details, see Appendix A.2.

3.3 Greedy Likelihood- and Value-Based Strategy

Heuristics such as the ones used in the previous category can capture some properties associated with high utility but are limited to properties that can be easily expressed explicitly. While the utility function is generally defined for complete sequences only, to guide decoding algorithms, one can rely on the general concept of a value model. A value model approximates, for partially decoded sequences, the expected utility of the final sequence if the decoding keeps following the same policy. The most prominent algorithm in this category is value-guided beam search (VGBS) (He et al., 2017; Ren et al., 2017; Krishna et al., 2022). It uses a greedy strategy similar to BS but selects the next token using a linear combination of the LM’s likelihood and the value model’s scores. See Appendix A.3 for details.

3.4 Simulation-Based Strategy

Even though VGBS considers both likelihood and value, it remains greedy by only looking one step ahead. Simulation-based decoding algorithms explore further into the future before making the next decision. When the value landscape is complex and constructing a good value model is hard, such strategies with a look-ahead may become particularly effective. By turning the knob controlling the number of simulations, one can trade off computational efficiency for obtaining better value estimates. Monte-Carlo Tree Search (MCTS) is the canonical example of simulation-based tree exploration informed by a value model. For details, see Appendix A.4.

3.5 Prompting-Based Strategy

The decoding algorithms described in the last two sections address the likelihood–utility misalignment post-hoc. An alternative is to change the conditioning of the model’s probability distribution such that the misalignment never happens. The effort now goes into choosing a context that aligns the likelihood landscape with the task-specific utility. The strength of this class of decoding algorithms lies in the fact that they can readily be applied to a new task without requiring any modifications to

the model or increasing the computation cost of inference (beyond the processing of the prompts’ tokens). However, they only work for large generalist LMs (Chowdhery et al., 2022). The most prominent members are the few-shot (FS) and the chain-of-thought (CoT) prompting methods, described in Appendix A.5.

4 Experimental Setup

4.1 Research Questions

In contrast to previous works that have studied the misalignment problem in constrained settings, we propose quantifying it in a unified and large-scale analysis across tasks (RQ1). We investigate the benefits of a diverse set of previously proposed solutions to the misalignment problem. Our study includes value-guided approaches (RQ2) and prompting (RQ3), covering each class of the MMS taxonomy with at least one representative. Specifically, we ask the following research questions.

RQ1: How correlated are the utility and the likelihood across tasks? As argued in Sec. 3, greedy likelihood-based strategies only require the likelihood to be a strong predictor of utility. We investigate whether this holds across tasks. Specifically, we measure two important aspects of the likelihood–utility alignment: (a) **Post-decoding alignment**. For each data point, the decoding algorithm chooses one output; we measure the likelihood and utility of the prediction and analyze their relation. Is high likelihood associated with high utility in the same way across tasks? (b) **During-decoding alignment**. Decoding algorithms typically explore a set of high-scoring candidates (e.g., BS returns one candidate per beam). We measure the correlation between likelihood and utility among these candidate outputs to analyze the likelihood landscape of the model.

RQ2: How effective are value-guided MMSs? In particular, we investigate the benefit of value-guided decoding algorithms as a function of the value model’s quality.

RQ3: Is prompting an MMS? We investigate the efficacy of prompting as a likelihood–utility alignment tool for generalist LMs (LLMs).

4.2 Tasks and Datasets

To organize the discussion, we propose a simple classification of the sources of misalignment: (a) **Training imperfections (TI)**, when the model is trained on a different objective than the true util-

Tasks	Utilities	Misalignment Types	Model	Dataset
Closed Information Extraction (cIE)	F1 score [M]	TI	GenIE (BART)	REBEL
Machine Translation (MT)	BLEU [M]	TI, DS	mBART50	WMT14
Non-Toxic Text Generation (NTTG)	Non-Toxicity [T]	TI, DS, UD	GPT2	RTP
Non-Soluble Protein Generation (NSPG)	Non-Solubility [T]	TI, DS, UD	ProtoGPT2	SwissProt
Sports Understanding	Solve Rate [M]	TI, DS, UD	MT-NLG 530B	Sports

Table 1: **Overview of the tasks.** Utility functions are categorized into: (a) [M]: Metric-based and (b) [T]: Trained model-based. The three misalignment types are TI: Training imperfection, DS: Distribution shift, UD: Utility drift.

ity, because of the finite size of the dataset and the approximation error in training (e.g., via stochastic gradient descent); (b) **Distribution shift (DS)**, when the training and testing data distributions differ; (c) **Utility drift (UD)**, when the utility used in development differs from the utility at test time.

While we can expect TI to affect all machine learning tasks (due to the finiteness of datasets and approximations resulting from gradient-based training), DS and UD are task-specific. DS typically occurs when the distribution of the data changes between the training and testing scenarios. UD occurs when the notion of utility changes, i.e., the labels for the same data points are changing.

We carefully selected a variety of generation tasks covering (a) different *notions of utility*, and (b) different expected types of (*mis*)*alignment between utility and likelihood*. Table 1 gives a high-level overview of these tasks, their utility functions, and associated datasets.¹ In closed information extraction, the training and testing data come from the same distribution, and we expect only TI-type of misalignment. In machine translation, the mBART model is pretrained on a different dataset, inducing some DS as the texts used for training may come from different domains. For non-toxic text and non-soluble protein generation, the task definition changed from generating low perplexity sequences to generating non-toxic sequences. Therefore, UD is expected to be the main driver of misalignment. Similarly, for the sports understanding task, since MT-NLG was not trained for this specific task, we also expect UD to be the main source of misalignment, but here VGBS and MCTS are too expensive due to the size of the LM. Instead, we use this setting to investigate prompting-based MMSs.

4.3 Decoding Algorithms

To cover the full space of MMSs, we experiment with at least one representative from each class from the taxonomy defined in Sec. 3. From

¹For more details about the models, data, and utility functions, see Appendix B.1.

the Greedy Likelihood-based category, we include the canonical GS and BS, as well as the sampling-based SB. From the Greedy Likelihood-based Strategies with pruning, we use CBS. VGBS and MCTS are representatives of the Greedy Likelihood- and Value-based, and Simulation-based decoding classes, respectively. For prompting, we consider the FS and CoT methods. The hyper-parameters for each algorithm are given in Appendix B.2. Appendix B.3, provides a complexity analysis in terms of LM and value model calls.

4.4 Value Models

The quality of a value model reflects its ability to approximate the expected utility. To determine the relationship between the value model’s quality and the benefit of leveraging it in decoding, we craft models that allow us to instantiate versions with varying levels of quality, ranging from a random predictor to an oracle.

Non-Toxic Text Generation. The state-of-the-art method for detecting toxicity is via classification (Hanu and Unitary team, 2020). Such a classifier can readily be used as a value model in decoding. We reproduce the training procedure from Hanu and Unitary team (2020) and save checkpoints at regular intervals until the training is complete. Due to the gradually decreasing under-fitting, these checkpoints give us a sequence of classifiers that systematically improve in terms of quality.

Machine Translation. To achieve a similar effect for MT, we start by assigning to each data point in the dataset another randomly chosen data point which will serve as a false target. This assignment is fixed across all runs. During inference, the value model calculates the BLEU score for both correct and incorrect targets and returns a linear combination between the points. By gradually increasing the weight assigned to the false target from zero to one, the perfect value model (oracle) slowly degrades to a random predictor.

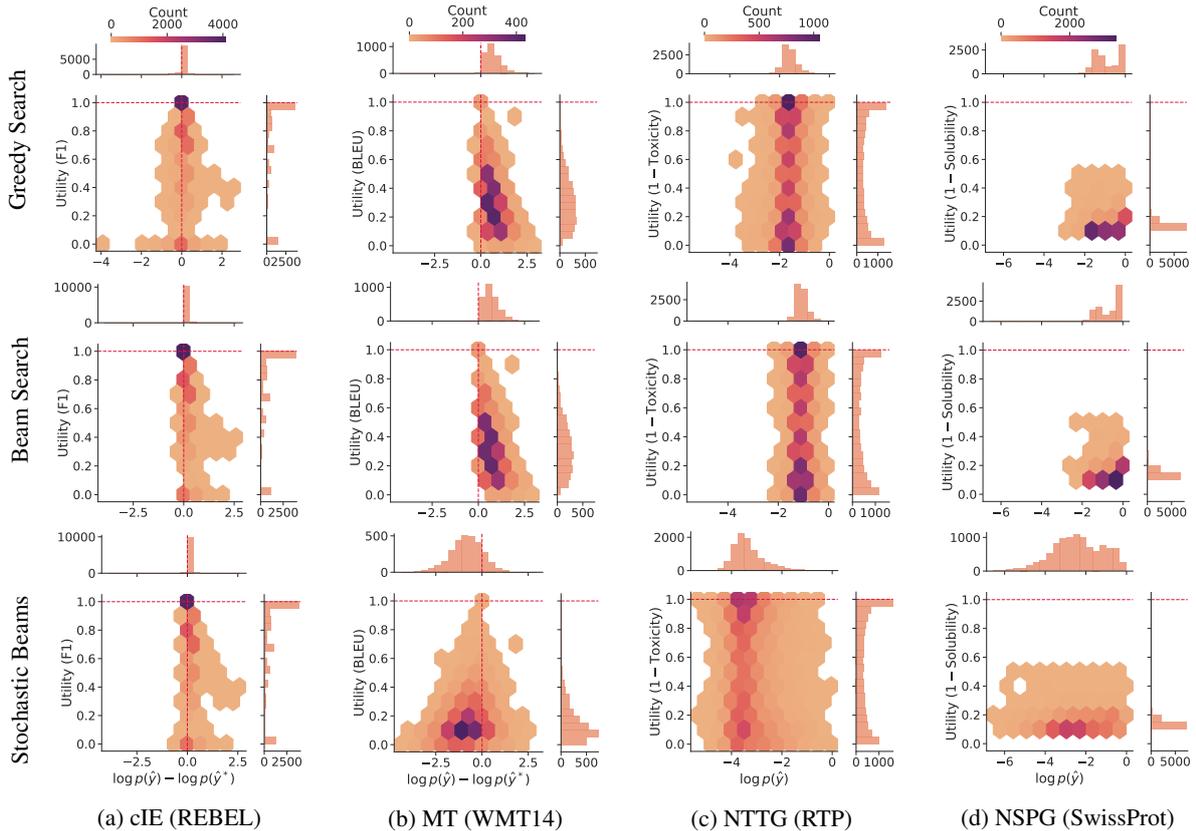


Figure 2: **RQ1 (post-decoding alignment)**: Each decoding algorithm is applied to each dataset-model pair. For each subplot, the x -axis represents the outputs’ log-likelihood under the model, and the y -axis the output’s task-specific utility score. The plots are frequency heatmaps counting the number of decoded outputs pertaining to a given hexagon. For MT and cIE, the x -axis is normalized such that 0 is the log-likelihood of the target answers². These plots show where the outputs are located in the likelihood-utility landscape across tasks and decoding algorithms.

5 Experiments and Results

5.1 RQ1: The Likelihood–Utility Relationship

We present two analyses, one measuring the likelihood–utility misalignment after decoding and one measuring it during decoding.

Post-decoding alignment. In this experiment, we first run each likelihood-based decoding algorithm (GS, BS, and SB) for each dataset–model pair. Then, for each output, we compute both the model likelihood and the task-specific utility.³ We report the results in Fig. 2.

For cIE (first column), most outputs have a likelihood close to the targets’ likelihood. The majority of the outputs have perfect utility — the decoded output is exactly the target, and the model is well-calibrated. This confirms the intuition that when the UD and DS are small, greedy likelihood-based MMS are very effective and can cope with the TI.

²See Fig. 6 for a plot without the target normalization.

³We also ran experiments with top- p and top- k but did not observe a behavior different from BS.

However, in tasks with larger DS and UD, the story is different. In MT (second column), the combined effect of TI and DS gives rise to a negative global correlation ($-.56$ for BS and $-.52$ for GS in terms of Pearson’s correlation; $p < 10^{-3}$) between the predictions’ utility and likelihood after decoding. This is an instance of *Goodhart’s law*, where a surrogate metric (likelihood), when being optimized heavily, becomes a poor approximation of the original property it is supposed to track (utility).

In tasks with large UD, NTTG (third column), and NSPG (fourth column), decoding according to likelihood does not guarantee utility. For example, in NTTG, the likelihood–utility correlation is $-.10$ for GS, $.10$ for SB, and $.03$ for GS in terms of Pearson’s correlation; $p < 10^{-3}$. These scenarios require external information that can guide the decoding towards high-utility outputs.

During-decoding alignment. Now, we investigate the likelihood–utility alignment where it matters: for outputs close to being extracted by the decoding

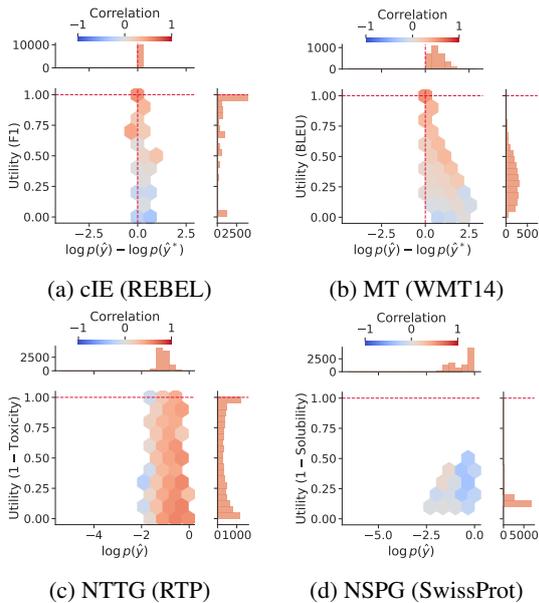


Figure 3: **RQ1 (during-decoding alignment)**: For each dataset-model pair, we run BS and analyze the correlation between likelihood and utility of the top-5 candidate hypotheses. The y-axis represents the task-specific utility score, and the x-axis the log-likelihood under the model. The plots are generated as follows: (i) take the BS outputs from Fig. 2 with their log-likelihood and utility scores, which indicate the x and y coordinate of each data point; (ii) for each data point measure the Kendall’s τ correlation between likelihood and utility of the top-5 candidate hypotheses; and (iii) average the correlation across the points belonging to the same hexagon.

algorithms. BS maintains k candidate hypotheses, one per beam, before returning the top-scoring one as the final output. In this experiment, we analyze the correlation between the likelihood and utility of the top-5 candidates. The results are reported in Fig. 3. There are three dimensions to this problem: (i) the likelihood (x-axis); (ii) the utility (y-axis); (iii) the correlation (color). Ideally, we would like to see red everywhere, indicating that failure to retrieve a high-utility output is due to the decoding algorithm, but the likelihood of the model is still a good predictor of utility. However, this is not what we observe.

For MT and cIE, we see a clear picture, red color (high likelihood–utility correlation) occurs at the top of the plot (high-utility): high likelihood-utility correlation among candidate outputs is enough to yield close to perfect-utility outputs.

For the NTTG (Fig. 3c), the correlation between utility and likelihood among the beams increases as the likelihood increases. When the model generates high-likelihood outputs, there is a positive correlation between being more likely and being less

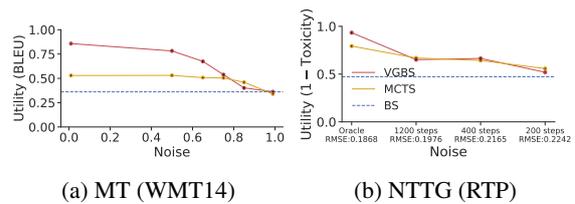


Figure 4: **RQ2**: For MT and NTTG, we ran VGBS and MCTS with value models displaying various levels of noise. We report the average utility of outputs on the y-axis (with 95% confidence interval). The noisy value models are described in Sec. 4.4.

toxic. However, the likelihood mass is assigned to low-utility regions of the output space, which cannot be resolved with decoding based only on the likelihood. For NSPG (Fig. 3d), the correlation across all bins is negative, indicating that high likelihood is a very bad predictor of high utility.

Takeaways. When TI is the only cause of misalignment, the likelihood is a strong predictor of utility; then, likelihood-based decoding algorithms are expected to retrieve high-utility outputs. When UD and/or DS are present, the correlation between likelihood and utility post-decoding plummets, indicating that likelihood-based decoding algorithms are ill-suited. When UD is present (bottom row of Fig. 4), good correlation among the beams does not necessarily mean good utility. However, without UD (top row of Fig. 4), higher correlation among the beams is associated with high utility.

5.2 RQ2: The Benefits of Value Models

We now analyze the benefits of value-guided decoding algorithms (VGBS and MCTS) as a function of the value model’s quality (see Sec. 4.4). Due to the high computational cost of running the experiments with both VGBS and MCTS, we focus on two tasks: MT and NTTG. For each version of the value model, we first perform a hyperparameter search on a small subset of the data and use the best hyperparameters on the test set. The results are reported in Fig. 4. VGBS and MCTS are always at least as good as BS, even with random value models, as the small-scale hyperparameter search selects parameters that ignore the values when they are not useful. However, when there is some signal in the value model, both VGBS and MCTS effectively leverage it and quickly start outperforming BS. When the value model is accurate, very high-utility outputs are discovered. Interestingly, VGBS mostly outperforms MCTS, and can extract almost

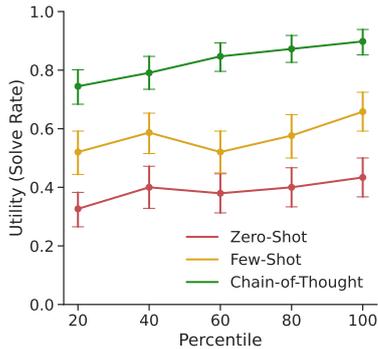


Figure 5: **RQ3:** ZS, FS, and CoT prompting, on the Sports understanding dataset. We report the utility (y-axis) of outputs binned according to the empirical percentiles of their likelihood (x-axis). The lineplot is the average utility per bin with 95% confidence intervals.

perfect outputs, whereas MCTS plateaus. This is significant because VGBS has a substantially lower complexity than MCTS (see Appendix B.3).

Takeaways. Value-guided decoding algorithms can overcome the likelihood–utility misalignment and significantly outperform likelihood-based decoding algorithms even with noisy value models, as long as a small-scale hyperparameter search is done. VGBS offers a better trade-off between performance and computation cost than MCTS.

5.3 RQ3: Prompting as an MMS

Recently, Wei et al. (2022) showed how the simple and broadly applicable idea of including in the prompt a few examples where the targets contain a sequence of steps that lead to the answer can greatly enhance the reasoning capabilities of LMs. We mimic the Sports Understanding task in their work by taking the same in-context examples and evaluating the same two prompting methods: CoT and standard FS prompting. Additionally, we evaluate the model in the standard zero-shot (ZS) setting, without any examples in the context, as a baseline. This results in unstructured answers that need to be labeled manually, see Appendix C.2.

To test our hypothesis that prompting is a means of addressing misalignment, we measure the utility and the likelihood under the model for all the testing data points. The results, summarized in Fig. 5, provide three insights. First, similarly to Wei et al. (2022), CoT outperforms FS and ZS with an accuracy of 83%, versus 57.3% and 38.9% for FS and ZS, respectively. Second, (and complementary to the information visualized on the plot) the average log-likelihood of the outputs generated

by CoT is significantly higher than the FS and ZS generated outputs, -0.067 as opposed to -0.17 and -2.472 . Third, the correlation between the likelihood and the utility when decoding with CoT is higher: 0.11 Pearson’s correlation compared to 0.07 and 0.09 for FS and ZS, respectively.⁴ Referring back to the observation made in RQ1b on Fig. 3, the value-guided MMSs studied in Sec. 5.2 address the misalignment post-hoc. However, the hidden representations building up to that misalignment are not modified, and the undesired information will still be attended to in predicting the next token probability distribution. In contrast, an effective prompting strategy addresses the misalignment before it affects the hidden representations, thereby (i) forcing the model to assign high likelihood to high-utility regions of the output space and (ii) improving the likelihood–utility alignment, making it easier to find high-utility outputs with greedy likelihood-based decoding algorithms.

Takeaways. Effective prompting methods put the model in a state where the generated outputs’ likelihood is well-aligned with the desired utility.

6 Discussion

RQ1 reveals that decoding based solely on the likelihood gives poor expected utility whenever DS or UD occurs. DS and UD make the likelihood a poor predictor of utility. When only TI is present, these decoding algorithms perform well because the likelihood is a strong predictor of utility.

Then, in RQ2 and RQ3, we saw that methods bringing in external information at decoding time manage to effectively solve the likelihood–utility misalignment problem. While finetuning (or re-training) would be an obvious and apparently ideal MMS, this is often neither possible nor necessary. Indeed, our experiments show that if a value model can be crafted and we can afford the extra compute for the value model calls, then VGBS becomes a strong decoding algorithm capable of fixing misalignment problems at inference time. It is more efficient than MCTS and performs better than BS, even if the value model is only a poor approximator of the utility. When crafting a useful value model is difficult (e.g., protein function depends on the 3D structure, which cannot be easily approximated from partial amino-acid sequences), MCTS with a large number of simulations with roll-outs can be

⁴The differences are statistically significant ($p < 10^{-3}$).

used to “estimate” one. However, the price to pay is a higher computational cost at inference time. Finally, for large, generalist LMs, decoding algorithms such as MCTS or VGBS are prohibitively expensive due to the high computational cost of each call to the LM. Prompting methods combined with greedy or top- p decoding can be considered as a way to leverage external information in the form of few-shots prompts to set the model in a state where the likelihood is better aligned with the utility. Our experiments support this explanation of the success of prompting large LMs. For a similar perspective, comparing prompting methods with training-based MMSs see [He et al. \(2021\)](#).

Limitations

Non-exhaustive empirical analysis. This work studies a fundamental problem that involves a complex interaction between tasks, models, and data; and sequence-to-sequence models have been applied to a very broad set of tasks. Covering all possible combinations is impossible, and for our empirical analysis, we chose a subset to evaluate. Our choice is guided by the classification of misalignment sources proposed in Sec. 4.2 and aims to cover different areas of the misalignment space. A seemingly small difference between two choices (e.g., a difference in the loss function used in training the model) can give rise to a considerably different misalignment and, consequently, performance. This is why the goal of the proposed conceptual framework is to make a step toward enabling a more systematic study of decoding. To further help the community investigate the broader space of tasks, models, and datasets through this lens, we open-source the implementation of our analysis.

Alternative ways of fixing misalignment. Apart from value-based decoding, other techniques could be considered to fix the misalignment problem: (a) Retrain or finetune the model with data that better reflects the task’s utility. For instance, to generate non-toxic text, one could retrain or finetune GPT2 on curated datasets that contain toxic prompts and non-toxic sentence continuations. (b) Optimize more directly the utility function instead of surrogate differentiable objectives. This could be done via reinforcement learning (see [Wang et al. \(2018\)](#); [Wu et al. \(2018\)](#) for BLEU).

In this work, we focused on decoding algorithms and ways of fixing the likelihood–utility misalignment problem at inference time. Future research

could further investigate the trade-offs involved in finetuning and retraining. Is it better to invest resources in acquiring new data that fits the task for finetuning? Or is it better to fix DS and UD at inference time with VGBS, MCTS or prompt engineering? Where do the inflection points lie?

Acknowledgments

This work was conducted in the context of the Microsoft Turing Academic Program (MS-TAP). West’s lab is partly supported by grants from Swiss National Science Foundation (200021_185043), Swiss Data Science Center (P22_08), H2020 (952215), Microsoft Swiss Joint Research Center, and Google, and by generous gifts from Facebook, Google, and Microsoft.

References

- Amos Bairoch and Rolf Apweiler. 2000. [The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000](#). *Nucleic Acids Research*, 28(1):45–48.
- BIG-bench collaboration. 2021. [Beyond the imitation game: Measuring and extrapolating the capabilities of language models](#). *In preparation*.
- Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ales Tamchyna. 2014. [Findings of the 2014 workshop on statistical machine translation](#). In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Antoine Chaffin, Vincent Claveau, and Ewa Kijak. 2022. [PPL-MCTS: Constrained textual generation through discriminator-guided MCTS decoding](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages

- 2953–2967, Seattle, United States. Association for Computational Linguistics.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. [Palm: Scaling language modeling with pathways](#). *CoRR*, abs/2204.02311.
- Nicola De Cao, Ledell Wu, Kashyap Papat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. [Multilingual autoregressive entity linking](#). *Transactions of the Association for Computational Linguistics*, 10:274–290.
- Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhownik, and Burkhard Rost. 2021. [Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Noelia Ferruz, Steffen Schmidt, and Birte Höcker. 2022. [Protgpt2 is a deep unsupervised language model for protein design](#). *Nature Communications*, 13.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. [RealToxicityPrompts: Evaluating neural toxic degeneration in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.
- Laura Hanu and Unitary team. 2020. [Detoxify](#). Github. <https://github.com/unitaryai/detoxify>.
- Di He, Hanqing Lu, Yingce Xia, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2017. [Decoding with value networks for neural machine translation](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. [Towards a unified view of parameter-efficient transfer learning](#). *CoRR*, abs/2110.04366.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text degeneration](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Pere-Lluís Hugué Cabot and Roberto Navigli. 2021. [REBEL: Relation extraction by end-to-end language generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2022. [GenIE: Generative information extraction](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4626–4643, Seattle, United States. Association for Computational Linguistics.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. [OpenNMT: Open-source toolkit for neural machine translation](#). In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.
- Wouter Kool, Herke Van Hoof, and Max Welling. 2019. [Stochastic beams and where to find them: The Gumbel-top-k trick for sampling sequences without replacement](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3499–3508. PMLR.
- Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. [RankGen: Improving text generation with large ranking models](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 199–232, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Rémi Leblond, Jean-Baptiste Alayrac, Laurent Sifre, Miruna Pislari, Leshpiau Jean-Baptiste, Ioannis Antonoglou, Karen Simonyan, and Oriol Vinyals. 2021. [Machine translation decoding beyond beam search](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8410–8434, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khachabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. [NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.
- Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. [If beam search is the answer, what was the question?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.
- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. [Locally Typical Sampling](#). *Transactions of the Association for Computational Linguistics*, 11:102–121.
- Clara Meister, Gian Wiher, Tiago Pimentel, and Ryan Cotterell. 2022. [On the probability-quality paradox in language generation](#). *CoRR*, abs/2203.17217.
- Stanislas Polu and Ilya Sutskever. 2020. [Generative language modeling for automated theorem proving](#). *CoRR*, abs/2009.03393.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. [Language models are unsupervised multitask learners](#).
- Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. 2017. [Deep reinforcement learning-based image captioning with embedding reward](#). In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1151–1159. IEEE Computer Society.
- Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. 2022. [A survey of evaluation metrics used for nlg systems](#). 55(2).
- Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. [PICARD: Parsing incrementally for constrained auto-regressive decoding from language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Shaden Smith, Mostofa Patwary, Brandon Norrick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. [Using deep-speed and megatron to train megatron-turing nlg 530b, a large-scale generative language model](#). *arXiv preprint arXiv:2201.11990*.
- Felix Stahlberg and Bill Byrne. 2019. [On NMT search errors and model errors: Cat got your tongue?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.
- Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. [A contrastive framework for neural text generation](#). In *Advances in Neural Information Processing Systems*.
- Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. [Generating text with recurrent neural networks](#). In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 1017–1024. Omnipress.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to sequence learning with neural networks](#). In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. [Multilingual translation from denoising pre-training](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.
- William Yang Wang, Jiwei Li, and Xiaodong He. 2018. [Deep reinforcement learning for NLP](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 19–21, Melbourne, Australia. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). *CoRR*, abs/2201.11903.
- Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. [On decoding strategies for neural text generators](#).
- Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. [A study of reinforcement learning for neural machine translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621, Brussels, Belgium. Association for Computational Linguistics.
- Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. [Trading off diversity and quality in natural language generation](#). In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

Daniel Zügner, Tobias Kirschstein, Michele Catasta, Jure Leskovec, and Stephan Günnemann. 2021. [Language-agnostic representation learning of source code from structure and context](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

A Proposed MMS Taxonomy

This section describes some of the most prominent members in each class of the proposed taxonomy.

A.1 Greedy Likelihood-Based Strategy

Deterministic

Greedy Search (GS). The simplest among the decoding algorithms, at each step t , GS selects the token with the highest likelihood under the model.

Beam Search (BS). An extension of GS, BS, maintains not one, but $k \in \mathbb{N}^+$ partially-decoded sequences, called beams, in parallel. At each step t , BS: (i) pre-selects the most likely k tokens for each beam; (ii) from the resulting $k \times k$ nodes, the algorithm selects the k with the highest likelihood and drops the rest.

Stochastic

An alternative that increases the diversity of output sequences is to sample the tokens at each step from the likelihood distribution $\hat{y}_t \sim p(y_t | \hat{y}_{<t}, x)$. Instead of sampling from the full distribution, these decoding algorithms typically focus greedily on the tokens corresponding to the high-probability regions.

Top- k Sampling. Top- k samples the next tokens from a truncated distribution where only the k most probable tokens are considered (Fan et al., 2018).

Top- p Sampling. Top- p samples the next tokens from a truncated distribution where only the smallest set of tokens with a probability mass bigger than (or equal to) p is considered (Holtzman et al., 2020).

Stochastic Beams (SB). SB samples completed outputs without replacements according to the LM’s likelihood. The implementation relies on applying BS on likelihood scores perturbed with Gumbel noise (Kool et al., 2019).

A.2 Greedy Likelihood-Based Strategy with Pruning

Ad-Hoc Heuristics. Currently, most tasks utilize some ad-hoc heuristics. For instance, in MT, it is often necessary to discourage empty (or short) sequences by enforcing a minimal sequence length (Stahlberg and Byrne, 2019). Similarly, state-of-the-art language generation models often get stuck in repetitive loops. Therefore, an n -gram repetition penalty is now part of the standard toolkit (Klein et al., 2017).

Constrained Beam Search (CBS). The idea of constraining the likelihood during decoding can be extended to include task-specific knowledge. For example, in information extraction tasks, the BS decoding strategy has been constrained to only extract outputs satisfying the predefined schema (Scholak et al., 2021; De Cao et al., 2022; Josifoski et al., 2022). Then, BS only searches high-scoring outputs among the smaller subset of valid ones.

NeuroLogic. The NeuroLogic strategy enforces the satisfaction of given lexical constraints by controlling the decoding stage of sequence generation (Lu et al., 2022). While BS aims to maximize the likelihood of the generated sequence, NeuroLogic searches for optimal output sequences among the strings that also satisfy the given constraints. Hard logic constraints are converted into a soft penalty term in the decoding objective, and beam-based search is used to find approximately optimal solutions.

A.3 Greedy Likelihood- and Value-Based Strategy

Value-Guided Beam Search (VGBS). It is the most intuitive example of a greedy decoding algorithm that leverages a value model (He et al., 2017; Ren et al., 2017). It uses a greedy strategy similar to BS but selects the next token using a linear combination of the LM’s likelihood and the scores from the value model.

More specifically, instead of expanding each beam by the m highest-scored tokens according to the likelihood, the algorithm chooses the top m tokens according to the following scoring function:

$$s_{y_{<i},x}(y_i) = \frac{\alpha}{i} \log(p(y_{<i}y_i|x)) + (1 - \alpha)v(y_{<i}y_i,x),$$

where the factor α weights the contribution of the the value model, $y_{<i}$ denotes the partially decoded sequence, and y_i corresponds to the next token under consideration.

A.4 Simulation-Based Strategy

Monte-Carlo Tree Search (MCTS). MCTS is the canonical example of simulation-based tree exploration informed by value. In our setup, it differs from all other decoding algorithms because, at step i , it may explore sequences of length greater than i . It is not tied to committing to local decisions without exploring the tree. In each step, MCTS

has a fixed computational budget that it uses to explore multiple paths before choosing the next token. For more details, we refer to Chaffin et al. (2022), whose implementation we adapt for this work.

A.5 Prompting-Based Strategy

Few-Shot (FS). At inference time, instead of only passing the input x , a context comprised of k examples $(x_i, y_i)_{i=1}^k$ is added as a prefix. The main idea is that the model will build on its semantic understanding of the relation between x_i and y_i and make the “guided” likelihood better aligned with the utility (Brown et al., 2020).

Chain-of-Thought (CoT). The CoT decoding method (Wei et al., 2022) is a conceptual extension of FS which presents the examples’ targets as a sequence of steps that lead to the solution. This format is particularly helpful for tasks that require multi-step reasoning, with which transformers generally struggle.

B Experimental Setup

This section provides additional details about the experimental setup.

B.1 Details about Data, Models, and Utility Functions

In Table 1, we present a summary of the tasks, utility functions, misalignment types, model, and dataset. We now give a brief description of each task:

Closed Information Extraction (cIE) with the REBEL dataset (Huguet Cabot and Navigli, 2021) and GenIE model (Josifoski et al., 2022) (an instance of BART finetuned to extract the exhaustive set of triples in a sentence following the Wikidata schema). The utility is the F1 score between the generated and the target set of triples.

Machine Translation (MT) with the WMT14 dataset (Bojar et al., 2014) and a pretrained mBART50 model (Tang et al., 2021) to translate *English to French*. The notion of utility is the match between the generated and the target translation, as measured by BLEU-4.

Non-Toxic Text Generation based on the Real Toxicity Prompt (RTP) dataset (Gehman et al., 2020) for prompting a GPT2 model. The notion of utility is whether the generated output contains toxic language or not. The utility function is an ALBERT model (Hanu and Unitary team, 2020)

	LM calls	Value calls
Greedy Search	N	–
Beam Search	$N \times B$	–
Stochastic Beams	$N \times B$	–
VGBS	$N \times B$	$N \times B \times K$
MCTS	$N \times S$	$N \times S$

Table 2: Coarse complexity analysis of the decoding algorithms used, in terms of LM calls and Value calls. N is the number of tokens to be generated, B the number of beams, K the number of next tokens considered by the value model per beam in VGBS, S the number of simulations per generated token in MCTS. In all our experiments, B=5, K=20, S=50.

trained on the Jigsaw dataset with an unintended bias to measure the toxicity of a text.

Non-Soluble Protein Generation: We use the SwissProt-EF dataset (Bairoch and Apweiler, 2000) for prompting a ProtoGPT2 model (Ferruz et al., 2022), which is pretrained on sequences of amino acids from protein prompts. The notion of utility is whether the generated protein is *soluble* or not. To measure non-solubility, we use ProtBERT (Elnaggar et al., 2021), which is a BERT-based model trained on a large corpus of protein sequences in a self-supervised fashion. Finally,

Sports Understanding with the Sports Understanding (SU) task, part of the BIG-bench effort (BIG-bench collaboration, 2021), with a 530B parameter pre-trained language model: MT-NLG (Smith et al., 2022). The primary purpose of this task is to test the general understanding of sports by asking the model to discriminate between plausible and implausible statements relating to sports.

B.2 Hyperparameters of Decoding Algorithms

The number of beams for BS, SB, VGBS is fixed to 5 for all tasks, except for cIE where it is 10 — the model’s default; and the number of simulations in MCTS is fixed to 50. Due to the high computational cost, to decide the optimal value for MCTS’s c_{puct} and VGBS’s α in RQ3, we run a hyperparameter search for each level of noise over a small sample of 80 data points (see Appendix C.3 for the ranges of the search). For both of the prompting-based strategies, we use greedy decoding during inference.

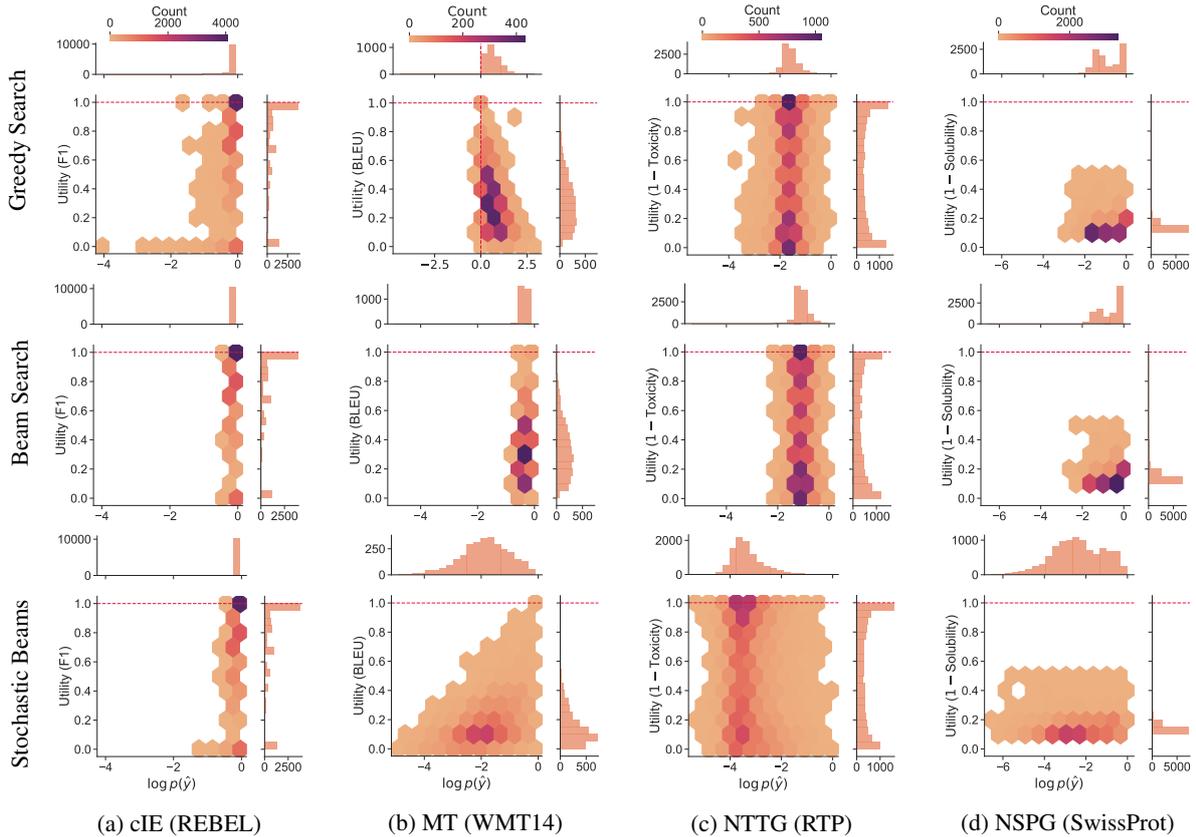


Figure 6: **RQ1 (post-decoding alignment) – without target normalization**: A version of Fig. 2 without the target answer log-likelihood normalization.

ID	Input	Prediction
1	Is the following sentence plausible? Leon Draisaitl grounded out to second base in the National League Championship Series. A:	i'm pretty sure that's not really a grammatical sentence, although one might...
2	Is the following sentence plausible? John Collins threw a touchdown in the NFC divisional round. A:	the nor gates give you (not(a)) and (not(b))
3	Is the following sentence plausible? Jack Eichel dunked the ball. A:	hmm? well, i'm not really sure. let me look it up in

Table 3: Examples of outputs that are not providing an answer. The first and the second row provide an example where the model produces unrelated text, while the third row is an example of an indefinite answer.

B.3 Complexity Analysis of Decoding Algorithms

Most of the compute during decoding is allocated on querying the LM or the value model. Therefore, to show how decoding strategies compare in terms of the computation cost, in Table 2 we provide a coarse complexity analysis in terms of the LM and value model calls.

C Experiments and Results

C.1 RQ1: The Likelihood–Utility Relationship

Fig. 6 is an alternative version Fig. 2 where the x-axis (for cIE and MT) is not normalized using the log-likelihood of the target answers.

C.2 RQ3: Prompting as an MMS

Extracting Labels from Zero-Shot Predictions

The outputs produced with zero-shot prompting do not follow a particular structure that can be used to extract the answer and therefore need to be processed manually. In some cases, it was not possible for an answer to be extracted. The two most common reasons for this were unrelated text as an answer or an indefinite answer. We provide examples of such predictions in Table 3. Overall, 24.9% of the answers could not be parsed. In such cases, we favored putting an indefinite label instead of "yes" or "no", and counting the answer as wrong irrespective of the ground truth label. If the answer and explanation were unrelated, but an answer was given, we did consider the answer.

	Number of Beams	Time (in GPU hours)
cIE + Greedy	1	1.5
cIE + BS	10	10.5
cIE + SB	10	10.5
MT + Greedy	1	0.5
MT + BS	5	1.0
MT + SB	5	1.5
NTTG + Greedy	1	2.0
NTTG + BS	5	3.5
NTTG + SB	5	4.5
NSPG + Greedy	1	3.5
NSPG + BS	5	5.5
NSPG + SB	5	7.0

Table 4: **Parameters for the greedy likelihood-based decoding algorithms.** The default parameters for each model were used, and no hyperparameter search was conducted.

C.3 Computational Infrastructure and Runtime

The evaluation for RQ1 as well as the hyperparameter search for RQ2 were conducted on a single machine with 24 Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz processor cores and 441 GB of RAM, equipped with 4 NVIDIA V100-PCIE-16GB GPUs. Table 4 provides the details for RQ1.

For VGBS, we performed a small hyperparameter search over different values for $\alpha = 0.01, 0.25, 0.5, 0.75, 0.99$ on 80 datapoints for each noise value of the value model. The same procedure was conducted for MCTS, over $c_{puct} = 0.25, 1.25, 3$. Each of these runs took 20 to 30 minutes of wall time, that is, slightly over 1 to 2 hours of GPU time. Table 5 and Table 6 provide the final parameters for VGBS and MCTS respectively.

The evaluation for RQ2 and RQ3, as well as the hyperparameter search for RQ2, were conducted on a single machine with 96 processor cores and 840 GB of RAM, equipped with 8 NVIDIA A100-SXM4-80GB GPUs. Table 5 and Table 6 provide the details for RQ2.

The evaluation for RQ3 was performed following the Sports Understanding setup in Wei et al. (2022), by taking the same in-context examples. We used greedy decoding for all of the prompting methods. The running time for all prompting experiments (ZS, FS, and CoT) was 6 GPU hours.

	α	Time (in GPU hours)
MT ($\lambda = 0.99$)	0.01	4
MT ($\lambda = 0.5$)	0.01	4
MT ($\lambda = 0.35$)	0.25	4
MT ($\lambda = 0.25$)	0.25	4
MT ($\lambda = 0.15$)	0.25	4
MT ($\lambda = 0.01$)	0.75	4
NTTG (oracle)	0.25	32
NTTG (1200 steps)	0.25	30
NTTG (400 steps)	0.25	30
NTTG (200 steps)	0.25	29

Table 5: **Parameters for VGBS.** For all the experiments, the value models consider the top-10 tokens according to the likelihood. The BLEU to the true target is weighted by λ (i.e., high λ translates to high-quality value model).

	c_{puct}	Time (in GPU hours)
MT ($\lambda = 0.99$)	1.25	41
MT ($\lambda = 0.5$)	0.5	41
MT ($\lambda = 0.35$)	0.5	40.5
MT ($\lambda = 0.25$)	0.25	40
MT ($\lambda = 0.15$)	0.25	40
MT ($\lambda = 0.01$)	1.25	40
NTTG (oracle)	1	125
NTTG (1200 steps)	1	96
NTTG (400 steps)	1	100
NTTG (200 steps)	1	98

Table 6: **Parameters for MCTS.** For all the experiments, at each node, we consider the top-20 tokens according to the likelihood and perform 50 simulations. The BLEU to the true target is weighted by λ (i.e., high λ translates to high-quality value model).

Lightweight Spatial Modeling for Combinatorial Information Extraction From Documents

Yanfei Dong^{1,2}, Lambert Deng¹, Jiazhang Zhang¹

Xiaodong Yu¹, Ting Lin¹, Francesco Gelli¹, Soujanya Poria¹, Wee Sun Lee²

¹ PayPal ² National University of Singapore

 DeCLaRe Lab, Singapore University of Technology and Design
{dyanfei, yuadeng, jiazhang, xiaodyu, tinlin, fgelli}@paypal.com
sporia@sutd.edu.sg, leews@comp.nus.edu.sg

Abstract

Documents that consist of diverse templates and exhibit complex spatial structures pose a challenge for document entity classification. We propose *KNN-Former*, which incorporates a new kind of spatial bias in attention calculation based on the K-nearest-neighbor (*KNN*) graph of document entities. We limit entities' attention only to their local radius defined by the *KNN* graph. We also use combinatorial matching to address the one-to-one mapping property that exists in many documents, where one field has only one corresponding entity. Moreover, our method is highly parameter-efficient compared to existing approaches in terms of the number of trainable parameters. Despite this, experiments across various datasets show our method outperforms baselines in most entity types. Many real-world documents exhibit combinatorial properties which can be leveraged as inductive biases to improve extraction accuracy, but existing datasets do not cover these documents. To facilitate future research into these types of documents, we release a new ID document dataset that covers diverse templates and languages. We also release enhanced annotations for an existing dataset.¹

1 Introduction

Structured document information extraction (IE) attracts increasing research interest due to the surging demand for automatic document processing, with practical applications in receipt digitization, workflow automation, and identity verification etc.

Recent state-of-the-art methods for processing documents with complex layouts extensively exploit layout information, such as position, relative distance, and angle, with transformer-based models. Spatial modelling is a key contributing factor to the success of these methods (Xu et al. 2020, Appalaraju et al. 2021, Xu et al. 2021, Hwang et al. 2021). However, absolute coordinates, pair-wise

relative Euclidean distance, and angle are insufficient to capture the spatial relationship in complex layouts. Two document entity pairs could carry different importance despite having the same position and distance, due to the presence or absence of other entities positioned between the pairs. We believe that spatial information can be better exploited for document entity classification.

We propose *KNN-Former*, a parameter-efficient transformer-based model that extracts information from structured documents with combinatorial properties. In addition to relative Euclidean distance and angle embeddings as inductive biases (Hwang et al., 2021), we introduce a new form of spatial inductive bias based on the K-Nearest Neighbour (*KNN*) graph which is constructed from the document entities and integrate it directly into the attention mechanism. Specifically, we first construct a *KNN* graph based on the relative Euclidean distance of document entities. Then we incorporate hop distance between entities, which is defined as the shortest path between two entities on the *KNN* graph, in training their pair-wise attention weight. For entity pairs with the same Euclidean distance but different hop distance, the difference in hop distance would still contribute to different attention weights. We limit an entity's attention calculation only to its local radius of neighborhood defined by the *KNN* graph. This also strengthens the inductive bias as reflected by our experiment results.

Furthermore, many real-world document information extraction tasks come with combinatorial properties, such as one-to-one mapping between field categories and values. Such combinatorial properties can be leveraged as inductive biases to improve the extraction accuracy, but are under-explored because existing datasets do not cover such documents. Current methods that do not address the combinatorial constraints suffer suboptimal performance on these types of documents. We further leverage this inductive bias by treating the

¹<https://github.com/miafei/knn-former>

entity classification task as a set prediction problem and using combinatorial matching to post-process model predictions (Kuhn, 1955; Carion et al., 2020; Stewart et al., 2016).

In addition, *KNN-Former* is parameter-efficient. Recent baseline models are initialized with parameters of pre-trained language models (Xu et al., 2020, 2021; Hwang et al., 2021; Hong et al., 2022), making their model size larger or at least comparable to the language models. *KNN-Former* does not utilize initialized parameters of existing language models, therefore free from the parameter size floor restriction. It is designed to be 100x smaller in trainable parameters compared to prevailing baselines. *KNN-Former*'s parameter efficiency makes it energy-efficient, contributes to faster training, fine-tuning and inference speed and makes mobile deployment feasible.

To encourage the progress of IE research in complex structured documents with combinatorial mapping properties, we release an ID document dataset (named POI). While the existing ID document dataset has only 10 templates (Bulatov et al., 2021), POI exhibits better template and lingual diversity. It also has a special mapping constraint where one field category has only one corresponding entity. In compliance with privacy regulations, the documents in the POI dataset are specimens and do not contain information about real persons.

We conduct extensive experiments to evaluate the effectiveness of our proposed method. *KNN-Former* outperforms baselines on most field categories across various datasets, despite having a significantly smaller model size. Extensive ablation studies show the importance of the *KNN*-based inductive bias and combinatorial matching.

To summarize, our contributions include (1) a highly parameter-efficient transformer-based model that (2) incorporates *KNN*-based graph information in sparsified local attention; (3) combinatorial matching to address the one-to-one mapping constraint; (4) a new ID document dataset with good template diversity, complex layout, and a combinatorial mapping constraint.

2 Related Work

Researchers have tried multiple approaches for document information extraction (Jaume et al., 2019; Mathew et al., 2021; Stanisławek et al., 2021). However, these works do not have spatial cues, such as the position of the information in the origi-

nal document. To address this shortcoming, a number of works introduce the modality of layout information as additional input features. Majumder et al. (2020) adopts positional information as inputs to their method to extract information from receipt documents. LayoutLM (Xu et al., 2020) adds 1-D and 2-D absolute position encodings to text embeddings before passing them to the transformer. Hong et al. (2021) proposes to train a language model from unlabeled documents with area masking, encoding relative positions of texts. StructuralLM (Li et al., 2021) assigns the bounding box cell position as the position coordinates for each word contained in it. DocFormer (Appalaraju et al., 2021) encodes 2D spatial coordinates of bounding boxes for visual and language features. LayoutLMv2 (Xu et al., 2021) uses learnable pair-wise relative positional embeddings as attention bias.

A few works propose to use graphs to represent spatial entity relationships in documents. SPADE (Hwang et al., 2021) uses a three steps graph decoder and formulates the information extraction task as a dependency parsing problem. FormNet (Lee et al., 2022) constructs a k-nearest neighbor graph and applies a 12-layer graph convolutional network (GCN) to get the entity embeddings before feeding them into a transformer network. However, there are some limitations in using GCN to obtain embeddings. It is well established that the message passing-based GCN are limited in their expressive power (Xu et al., 2018; Arvind et al., 2020; Morris et al., 2019; Chen et al., 2020; Loukas, 2019; Dehmamy et al., 2019). In addition, FormNet does not use the hop distance between nodes, which could serve as a strong inductive bias to capture the spatial relationships between document entities.

Datasets with positional information such as Funsd (Jaume et al., 2019), Cord (Park et al., 2019), Sroie (Huang et al., 2019) are released to facilitate research in document understanding. However, they do not contain documents with combinatorial properties which are common in real-world applications. MIDV500 (Arlazarov et al., 2018) and MIDV2020 (Bulatov et al., 2021) are two synthetic ID datasets with combinatorial properties, but are unsuitable for document information extraction tasks due to incomplete annotations. They also lack template diversity.

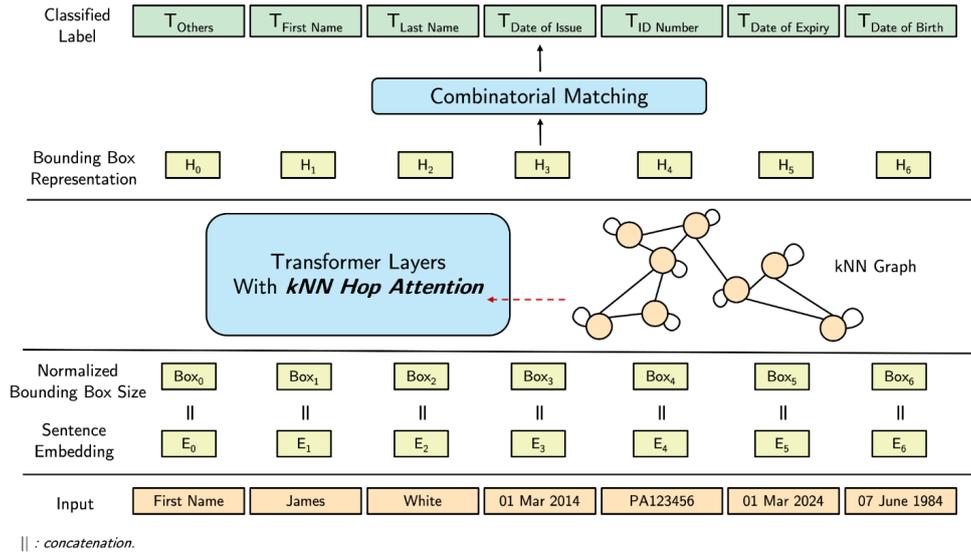


Figure 1: An illustration of *KNN-Former*. Bounding box texts are embedded using sentence transformer, which are concatenated with embeddings of bounding box size to form input embeddings. The concatenated embeddings are then passed to the transformer layers with *KNN Hop Attention*, which incorporates pair-wise relative hop distance between entities on *KNN* graph in attention calculation. The output entity representations of the transformer layers are passed to combinatorial matching for set prediction.

3 Methodology

In this section, we discuss the methodology for our model. We formulate the problem in Sec.3.1 and explain our overall model architecture and the details of each component in Sec.3.2.

3.1 Problem Formulation

Given a document D which consists of multiple entities $\{e_i, \dots, e_j\}$, and the bounding box coordinates and texts $\{x_i, \dots, x_j\}$ detected by Optical Character Recognition (OCR) tool. We measure the relative distance and angle between two entities e_i and e_j as $\sigma_{(i,j)}$ based on the coordinates of bounding boxes. Our task is to map each entity e_i in document D to its field category y_i , which is one of the predefined labels. For each field category y_i , there is only one corresponding entity e_i .

3.2 Model Architecture

We propose *KNN-Former*, a transformer-based model for document entity classification. The architecture of *KNN-Former* is shown in Fig. 1. *KNN-Former* uses K-Nearest Neighbours Hop Attention, which incorporates a new inductive bias into attention computation. *KNN-Former* also treats document entity classification as a set prediction problem and uses combinatorial assignment to address the one-to-one correspondence between entities and fields. *KNN-Former* is highly parameter-

efficient compared to baselines. Details of model size can be found in Tab 4.

3.2.1 K-Nearest Neighbors Hop Attention

One key contribution of *KNN-Former* is the proposed attention mechanism. Following (Lee et al., 2022), we first construct a *KNN* graph based on the Euclidean distance between each pair of entities. We represent entities as nodes and then connect edges between each entity and its K nearest neighboring entities. We also add a self-loop to each entity to improve performance (Kipf and Welling, 2016). While previous works focus on leveraging pair-wise relative Euclidean distance (Xu et al., 2021; Hwang et al., 2021), we propose to incorporate pair-wise relative **hop distance**, which is defined as the shortest path between two entities on the *KNN* graph. Two entities could be in proximity in terms of Euclidean distance but not so in terms of hop distance. For example, in documents with complex layouts, it is common to have two entities that are close to each other in the Euclidean space, but there is a third entity positioned in between. This type of entity pair should be treated differently from pairs that are close to each other in both Euclidean and hop distances. In this case, the spatial attention mechanism based solely on the relative Euclidean distances between entity pairs is insufficient since it neglects this structural information. We argue that the *KNN* graph structure is an

effective way of capturing the structural information and propose to incorporate it as an inductive bias into the attention computation.

Intuitively, different hop distances should carry different weights in calculating pairwise attention. We use $\phi_{(i,j)}$ to represent the hop distance between entity i and j and H to represent a learnable embedding lookup table based on the hop distance $\phi_{(i,j)}$. Inspired by DeBERTa (He et al., 2020) and Transformer-XL (Dai et al., 2019), we integrate the hop distance bias into attention as described in the following equations

$$e_{ij} = [x_i W^Q (x_j W^K + H_{\phi_{(i,j)}}^Q + R_{\sigma_{(i,j)}}^Q) + (H_{\phi_{(i,j)}}^K + R_{\sigma_{(i,j)}}^K) x_i W^K] / \sqrt{d}, \quad (1)$$

$$z_i = \sum_j a_{ij} (x_j W^V + H_{\phi_{(i,j)}}^V + R_{\sigma_{(i,j)}}^V), \quad (2)$$

where $\sigma_{(i,j)}$ is a concatenation of the relative Euclidean distance and angle between entity i and j , and R is a learnable matrix. H could be a learnable matrix or a lookup table that maps $\sigma_{(i,j)}$ to learnable embeddings. e_{ij} is the attention weight between entity i and j . a_{ij} is calculated as the weight of $\exp(e_{ij})$ in the exponential sum of all e_{ik} , as described in Eqn.3.

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}. \quad (3)$$

Similar to how pair-wise relative Euclidean distance is added to attention, we add pair-wise hop distance as three learnable weight matrices, two of which multiply with query and key vectors respectively while the remaining one is added to the value vector. We further limit an entity’s attention only to its local radius of neighborhood defined by the KNN graph. Specifically, we do not calculate e_{ij} if the hop distance between entity i and j exceeds a certain threshold. This also strengthens the inductive bias as supported by our experiment results.

3.2.2 Combinatorial Matching

We hypothesize that combinatorial properties between field categories and entities can be leveraged as inductive biases to improve extraction performance. Different from existing methods that treat the classification of each entity independently (Xu et al., 2021; Hwang et al., 2021; Lee et al., 2022),

we propose to treat the entity classification task as a set prediction problem to exploit the one-to-one mapping constraint, where one field has one and only one corresponding entity. The combinatorial assignment is described in Eqn.4.

$$\tau_{opt} = \underset{\tau}{\operatorname{argmin}} \sum_i^N L_{match}(y_i^{label}, y_{\tau(i)}^{pred}), \quad (4)$$

where τ is an assignment, and L_{match} is the matching cost. N is the number of entities in a document. In practice, N is often much larger than the number of entities of interest. Therefore, we pad the number of ground truths to N in order to perform a one-to-one combinatorial assignment. This can be done with the Hungarian algorithm in polynomial time (Kuhn, 1955; Carion et al., 2020; Stewart et al., 2016).

4 Datasets

Many real-world documents exhibit combinatorial properties, such as a one-to-one mapping between its fields and entities. However, existing public datasets do not cover documents with such properties (Jaume et al., 2019; Park et al., 2019; Huang et al., 2019). To fill the gap, we release a new ID document dataset POI, and enhanced annotations of MIDV2020. We also verify our method on a private dataset PRV. All 3 datasets exhibit combinatorial properties.

In addition, we design the POI dataset to be template-rich with diverse languages. We also design the enhanced MIDV2020 with a difficult split such that templates in testing are unseen during training. BERT alone without spatial information can achieve above 90% F1 on some existing datasets (Hong et al., 2022; Park et al., 2019; Huang et al., 2019), indicating relative sufficiency of leveraging text information alone. Yet in many real-world use cases, using text alone is insufficient. This motivates us to work on more challenging datasets where the exploitation of spatial information is important. Dataset statistics are summarized in 1 and Tab. 2. More details are as follows.

	#Train Doc.	#Test Doc.
POI	421	109
MIDV2020	500	200
PRV	3480	807

Table 1: Number of documents in training and testing.

Dataset	Avg # of Ent. per Doc.	Total # of Ent.	Total # of Doc.
POI	31.79	16850	530
MIDV2020	32.85	23000	700
PRV	24.31	104245	4287

Table 2: Statistics of entity distribution in documents. Ent. stands for entities and Doc. stands for documents.

POI We collect and annotate 530 Proof-of-Identity documents from online sources. We will release this POI dataset which consists of 10 document types, 265 distinct templates, and 131 countries of origin. The template and language diversity of POI create a challenging task for document understanding. All images are specimens with dummy values. The document type distribution is shown in Tab.3.

There are 8 field categories in total: last name, first name, date of birth, date of issue, date of expiry, ID number, key, and others. Key represents entities that indicate the field names for the important entities (e.g. Last Name) that we are interested to extract. The first 6 field categories appear in each document image once and only once, creating a special mapping constraint unseen in other datasets. The last 2 field categories (key and others) are not subject to the constraint. In real-world applications, it is common to extract a set of entities from documents that have combinatorial properties between its field and entities. ID document information extraction is one such use case, where we only expect to extract one entity for each field category of interest. This one-to-one correspondence can be leveraged to improve classification performance. Despite being a common task setting, we notice the lack of method exploration and innovation in this direction, due to the unavailability of such property among existing popular document datasets. More details about the dataset can be found in the Appendix.

Document Type	# Document
Passport	238
Driving License	119
Travel Document	109
ID	30
Resident Permit	21
Seafarer ID	10
Others	3

Table 3: Distribution of document types in POI dataset

MIDV2020 We utilize the 1000 synthesized ID documents from the initial MIDV2020 dataset (Bulatov et al., 2021). These documents are generated from 10 templates, with 100 documents for each template. Each document image is annotated with a list of bounding box coordinates and field values. We find that only artificially generated entities, such as the values of names and ID numbers, are annotated, while entities that belong to the original templates, such as document title and field names are not. We proceed to annotate the remaining entities. The newly annotated ground truths of MIDV2020 will be released alongside POI. These enhanced annotations enable us to perform information extraction task in a setting that is closer to real-world application, where all texts recognized by the OCR engine are used. The train/test split we introduce for MIDV2020 is a split by countries, this ensures that the document templates in the training dataset are unseen in the testing dataset. The country split simulates real-world scenarios where the model extension to new countries or new versions of documents is needed. More details can be found in the Appendix.

PRV Since POI and MIDV2020 only contain specimens or artificially generated images, we run our model on a private dataset (named PRV) that mostly consists of US driver licenses. The documents are protected by strict privacy requirements and massive human annotations are not available as raw images are inaccessible. Therefore, we build automatic fuzzy labeling to annotate the ground truth.

Comparison on Datasets POI exhibits better template and language diversity. POI contains 265 templates from 131 countries, while MIDV2020 has 10 templates from 10 countries. The number of templates in PRV is unknown due to privacy-related limitations. In addition, POI consists of templates in a multitude of languages, whereas MIDV2020 and PRV dataset lack such diversity. Texts in POI and MIDV2020 are made up largely by artificial text which is more readable and clearer, while PRV contains real texts. POI and PRV samples are split randomly. Since MIDV2020 has only 10 templates, we split the samples by country to make the task more challenging. PRV is the easiest dataset among the three due to its lingual monotony and random split.

Dataset	Method	F1 Score						Input Modality	#Parameters	
		L.Name	F.Name	DoB	DoI	DoE	ID No.		Trainable	Total
POI	BERT _{BASE}	67.90	72.73	92.11	70.78	69.06	78.70	text	110 M	110 M
	GCN	45.35	56.08	85.62	62.37	62.32	70.65	text + layout	31.5 K	22.7M
	LayoutLM _{BASE}	87.03	86.88	93.93	86.23	87.72	83.12	text + layout	110 M	110 M
	LayoutLMv2 _{BASE}	90.58	89.26	96.00	94.22	92.59	88.16	text + layout + image	199 M	199 M
	SPADE	73.73	78.63	90.09	89.59	90.27	83.98	text + layout	128 M	128 M
	BROS _{BASE}	82.39	82.76	94.16	91.41	88.32	83.18	text + layout	109 M	109 M
	KNN-former	83.57	82.18	98.37	95.89	94.48	90.06	text + layout	0.5 M	23.2M
MIDV2020	BERT _{BASE}	40.61	52.89	100.00	85.29	80.00	55.62	text	110 M	110 M
	GCN	32.03	43.09	99.50	99.00	79.76	43.82	text + layout	31.5 K	22.7M
	StructuralLM _{LARGE}	25.13	11.83	100.00	89.29	91.53	99.50	text + layout	355 M	355 M
	LayoutLM _{BASE}	47.65	15.10	100.00	97.96	80.16	67.97	text + layout	110 M	110 M
	LayoutLMv2 _{BASE}	47.54	49.91	87.15	97.56	77.24	94.18	text + layout + image	199 M	199 M
	SPADE	48.91	45.54	79.90	63.47	60.85	60.34	text + layout	128 M	128 M
	BROS _{BASE}	23.31	23.78	98.50	70.83	18.27	85.39	text + layout	109 M	109 M
KNN-former	87.88	54.26	100.00	100.00	95.21	69.65	text + layout	0.5 M	23.2M	
PRV	BERT _{BASE}	71.32	76.39	97.72	88.78	86.22	87.21	text	110 M	110 M
	GCN	66.32	81.97	97.59	89.53	87.90	89.38	text + layout	31.5 K	22.7M
	StructuralLM _{LARGE}	93.72	93.27	99.56	98.86	99.21	97.86	text + layout	355 M	355 M
	LayoutLM _{BASE}	95.36	94.71	99.17	98.76	98.61	97.85	text + layout	110 M	110 M
	LayoutLMv2 _{BASE}	95.26	95.31	99.52	99.29	99.36	98.82	text + layout + image	199 M	199 M
	SPADE	65.61	70.65	98.70	98.10	96.43	92.48	text + layout	128 M	128 M
	BROS _{BASE}	93.52	91.68	99.00	98.44	97.53	97.91	text + layout	109 M	109 M
KNN-former	92.03	96.81	91.22	99.68	99.47	98.76	text + layout	0.5 M	23.2M	

Table 4: Entity-level F1 score of *KNN-Former* compared to baselines. Column L.Name, F.Name, DoB, DoI, DoE and ID No. correspond to results of Last Name, First Name, Date of Birth, Date of Issue, Date of Expiry, and ID Numbers. GCN and *KNN-Former* have additional 22.7 M fixed parameters since we employed a light-weighted 6-layer sentence transformer (Reimers and Gurevych, 2019) to get the text embeddings.

5 Experiments

In this section, we conduct extensive experiments to evaluate our proposed *KNN-Former* on aforementioned datasets. We first compare our results with several baselines in Sec. 5.1. Then in Sec. 5.2, we evaluate the generalization ability of our method on unseen templates. We then conduct ablation studies in Sec.5.3 and Sec.5.4 to assess the effects of each component in *KNN-Former* and the impact of different K in the *KNN* graph.

5.1 Comparison with Baselines on Multiple Datasets

We first evaluate the performance of *KNN-Former* against multiple competitive methods. We choose base models for most of the baselines, because these are closest to *KNN-Former* in terms of the number of parameters. Brief description of baseline models as well as the implementation details of all the models can be found in Sec. A.1. We do not have results for StructuralLM on POI dataset because of an OOV error.

Tab.4 shows the entity-level classification performance. The results show that our method outperforms the baselines on most entity types across

various datasets. In particular, *KNN-Former* outperforms LayoutLMv2_{BASE}, a state-of-the-art model that uses additional image features. We also observe that BERT performs poorly on these datasets, indicating the importance of exploiting spatial information.

Secondly, as shown in Trainable Param column in Tab.4, *KNN-Former* is highly parameter-efficient. All baselines except GCN have more than 100 million trainable parameters, while *KNN-Former* has only 0.5 million and is magnitudes smaller than competing methods. Even after adding the sentence transformer, *KNN-Former* has only 23.2 million parameters, still 5x smaller than baselines. The parameter efficiency has 4 benefits. First, it contributes to learning and inference time efficiency, with details illustrated in 5.5. Second, it allows for faster fine-tuning on new datasets and domains, especially in real-world use cases when training datasets are big and re-training requirements are frequent. Third, smaller model size and faster inference time make mobile deployment more feasible. Fourth, training, fine-tuning and inferring smaller models reduces power consumption and carbon footprint. Despite the smaller model size, *KNN-*

Former achieves comparable or better performance across datasets.

Thirdly, we observe that *KNN-Former* underperforms both $\text{LayoutLM}_{\text{BASE}}$ and $\text{LayoutLMv2}_{\text{BASE}}$ for name related entities in both POI and PRV datasets. The robustness of the two baselines in predicting names could be attributed to their extensive pre-training. The two baselines learn common names in pre-training, enabling them to predict names correctly regardless of context. However, despite no extensive pre-training, *KNN-Former* still outperforms BROS and StructuralLM which are also pre-trained on 11 million documents.

Fourthly, we observe all methods suffer performance degradation on MIDV2020, compared to the other two datasets. This is because in MIDV2020, training and testing documents are split by countries, templates in testing are not seen during training. In addition, MIDV2020 has only 6 templates in training data, which easily leads to overfitting. Detailed discussion on the generalization ability can be found in Sec. 5.2. we find that BERT outperforms several baselines with spatial modelling on names, this may be due to overfitting to limited number of training templates. We notice that our method do not perform well on id number entity. We conducted manual inspection on several error cases, and find that in many documents there exist two different types of id numbers(see Fig. 3(b)), but only one of them is labeled as id number according to the provided annotations. Our model sometimes predicts the other one as id number. This also explains the poor performance on id number for some other baselines.

Lastly, we notice that on the PRV dataset, *KNN-Former* performs poorly on DoB field, underperforming even GCN. *KNN-Former*'s performance on DoB drops after combinatorial matching, despite an overall increase macro average F1. This could be due to the presence of noise in groundtruth, since this dataset is annotated by automatic fuzzy labeling logic. Manual examination of a few documents confirms our hypothesis.

5.2 Evaluation of generalization ability on unseen templates

To assess the generalization capability of our model, we test and compare our model with other competitive baselines on MIDV2020 dataset using two train/test settings: random split and split by country. The country split is a more difficult set-

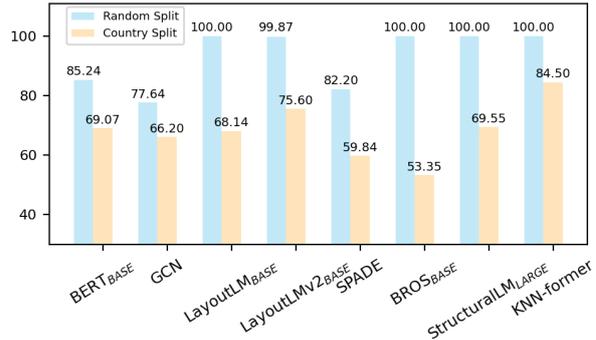


Figure 2: Macro average F1 scores of *KNN-Former* and various baseline models under random split and country split on MIDV2020 dataset.

ting as the templates in testing are unseen during training. Intuitively, we would expect a decline in performance as compared to the random split setting. Fig. 2 shows the Macro average F1 scores comparison of *KNN-Former* and multiple baselines under both the random split and the country split. We observe across-the-board performance degradation for all methods after switching from random split to country split. However, the drop is least significant on *KNN-Former*, enabling it to achieve 10% higher F1 than the best baseline. These experiments indicate that our method is more robust and generalizes better to unseen templates as compared to existing baseline models. This is helpful in real-world applications where models frequently encounter new types of documents.

5.3 Effects of each component in *KNN-Former*

Model	F1
<i>KNN-Former</i>	90.76
(-) <i>KNN</i> hop attention	88.33 (-2.43)
(-) Local attention based on <i>KNN</i> hop & (-) <i>KNN</i> hop attention	85.67 (-5.09)
(-) Relative Euclidean distance & angle attention	87.17 (-3.59)
(-) Relative Euclidean distance & angle attention & (-) <i>KNN</i> hop attention	86.67 (-4.09)
(-) Combinatorial Matching	88.16 (-2.60)
(+) Absolute positional encoding	86.33 (-4.43)

Table 5: Ablation results on POI dataset. (-) indicates the component is absent compared to *KNN-Former*, (+) indicates the component is additional.

To better understand how *KNN-Former* works, we ablatively study the effects of each component and report the results in Tab. 5. Entity-level detailed results can be found in the Appendix.

Firstly, we observe a 2.43% drop in performance with the removal of *KNN* hop attention and an even bigger 5.09% drop when local attention is removed

together with *KNN* hop attention. This demonstrates that the *KNN* graph-based inductive bias is effective in capturing the structural information between document entities. It also shows that local attention, the practice of masking out attention weights when the hop distance between two entities exceeds a pre-defined threshold, further strengthens the inductive bias.

Secondly, we observe that the commonly used spatial inductive bias based on the pairwise relative Euclidean distance and angle also plays an important role. When both relative Euclidean distance attention and *KNN* hop attention are absent, there is a 4.09% drop in performance, an additional decrease of 1.66% compared to when only *KNN* hop attention is ablated (2.43%). The overlap of performance drop suggests some information are captured by both Euclidean distance and hop distance, as some pairs are similarly close/far from each other as measured in both distances. However, each distance also complements the other by capturing additional information. For example, two pairs could carry different importance despite having the same Euclidean distance, due to the presence or absence of other entities positioned between the pairs, signifying the importance of hop distance.

Thirdly, we notice that the F1 score drops drastically by 4.76% when combinatorial matching is ablated. This demonstrates the important contribution of combinatorial matching, as the datasets we experiment on are all subject to a special one-to-one mapping constraint between fields and entities. Combinatorial matching enables our method to treat entity classification as a set prediction problem, instead of predicting each entity’s class independently, which enhances our model robustness.

Lastly, we observe that there is a 4.43% drop in performance when absolute positional encoding is added. Previous works (Hwang et al., 2021) have similar findings that adding absolute positional encoding is not helpful, especially when the test set contains a diverse set of unseen templates. In our experiments, adding absolute positional encoding improves performance in training but generalizes poorly in testing.

5.4 Impact of different *K* in the *KNN* graph

To further study the effect of how the hyper-parameter of the *KNN* graph affects the performance, we conduct experiments with different values of *K* on the POI dataset. As shown in Tab. 6, the

#K	(+) H (-) R	(-) H (+) R	(+) H (+) R
2	90.67	89.33	89.50
5	88.74	90.23	89.51

Table 6: Impact of number of *K* in *KNN-Former* on POI dataset. (+) indicates presence, (-) indicates absence. H refers to the *KNN* hop attention. R refers to relative Euclidean distance and angle attention.

2-NN graph achieves the best performance when *KNN*-based hop distance is used and relative Euclidean distance is removed. This is because when only 2 nearest entities are counted as an entity’s first-hop neighbors, the correlation between hop distance and entity pair’s importance is pronounced. However, a 5-NN graph achieves the best performance when *KNN*-based hop distance is ablated and only relative Euclidean distance is used. This is because the information of who is an entity’s 5 nearest neighbors is less useful in documents with an average of 31.79 annotated bounding boxes per file. Models with 2-NN and 5-NN graphs underperform the 4-NN graph in the POI dataset, underscoring the importance of choosing the correct *KNN* graph hyper-parameter for different datasets.

5.5 Runtime Comparison

In addition to performance evaluation, we also evaluate the runtime of our model against competitive baselines. For fair comparison, we report the total runtime of sentence transformer plus *KNN-Former*, since *KNN-Former* uses sentence transformer for text embeddings. In fact, the sentence transformer takes up half of the time in our pipeline.

Model	Single	Batch
LayoutLM _{BASE}	19.61	237.90
LayoutLMv2 _{BASE}	56.64	2941.32
SPADE	39.47	6091.52
BROS _{BASE}	23.45	646.65
<i>KNN-Former</i>	22.60	77.57

Table 7: Runtime comparison with baselines. Time taken is reported in milliseconds.

We first measure the runtime to process a single document for each method. As shown in Tab. 7, time taken for sentence encoder plus *KNN-former* is comparable to LayoutLM and BROS, and is faster than SPADE, LayoutLMv2. We run StructuralLM(written in tensorflow1.14) on CPU due to cuda version mismatch, hence there is no speed measurement.

Moreover, our method allows for significantly larger batch sizes because of the smaller model size. Therefore, runtime for documents in batch is significantly faster than the baselines. Running with maximum possible batch size for each model using a 16GB V100 GPU, *KNN-Former* is significantly faster than the rest, as shown in Tab. 7. This experiment demonstrates that our model is advantageous when faster execution time is desirable, and this could be attributed to the lightweight property of our model.

6 Conclusion

We propose *KNN-Former*, a parameter-efficient transformer-based model for document entity classification. *KNN-Former* uses *KNN* Hop Attention, a new attention mechanism that leverages *KNN* graph-based inductive bias to capture structural information between document entities. *KNN-Former* utilizes combinatorial matching to perform set prediction. We also release POI, a template-rich ID document dataset subject to combinatorial constraints. Experiments show that *KNN-Former* outperforms baselines in entity classification across various datasets.

Limitations

We identify the following limitations in this work. First, the robust performance of baseline methods that leverage image features (Appalaraju et al., 2021) testifies to the importance of visual cues. The inclusion of image features to *KNN-Former* might contribute to better performance. Second, unlike models that perform extensive pre-training (Xu et al., 2020, 2021), *KNN-Former* might lack generic domain knowledge. Third, *KNN-Former* uses a vanilla sentence transformer to get the text embedding inputs. The sentence transformer model is pre-trained and not fine-tuned on the new datasets. An end-to-end training pipeline that jointly trains the text encoding model and *KNN-Former* could lead to better results. Fourth, there are many design choices we did not explore, such as applying attention directly at the token level and pooling representations at the end. Lastly, *KNN-Former*, along with all baselines used in this work, are subject to OCR failure. All models consume OCR outputs such as bounding box coordinates and texts. In the case of OCR failure, where one bounding box is detected as two or two boxes are merged as one, models that consume OCR results are less likely to

make correct predictions.

Ethics Statement

This work has obtained clearance from author’s institutional review board. The annotators for POI and MIDV2020 are all paid full-time interns and researchers hired by our institute, whose compensation are determined based on the the salary guidelines of our institute. Among the datasets and annotations released, POI only contains specimens with dummy values, while MIDV is a synthetic dataset. External data are accessed and used in compliance with fair use clauses. We conduct experiments on the private dataset PRV in a secure data zone with strict access control, using auto-labeling scripts for annotations.

Acknowledgement

This research is partly supported by the SRG grant id: T1SRIS19149 and the Ministry of Education, Singapore, under its AcRF Tier-2 grant (Project no. T2MOE2008, and Grantor reference no. MOET2EP20220-0017). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

References

- Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 993–1003.
- Vladimir V. Arlazarov, Konstantin B. Bulatov, and Timofey S. Chernov. 2018. MIDV-500: A dataset for identity documents analysis and recognition on mobile devices in video stream. *CoRR*, abs/1807.05786.
- Vikraman Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. 2020. On weisfeiler-leman invariance: Subgraph counts and related graph properties. *Journal of Computer and System Sciences*, 113:42–59.
- Konstantin B. Bulatov, Ekaterina Emelianova, Daniil V. Tropin, Natalya Skoryukina, Yulia S. Chernyshova, Alexander Sheshkus, Sergey A. Usilin, Zuheng Ming, Jean-Christophe Burie, Muhammad Muzzamil Luqman, and Vladimir V. Arlazarov. 2021. MIDV-2020: A comprehensive benchmark dataset for identity document analysis. *CoRR*, abs/2107.00396.
- Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey

- Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.
- Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. 2020. Can graph neural networks count substructures? *Advances in neural information processing systems*, 33:10383–10395.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.
- Nima Dehmamy, Albert-László Barabási, and Rose Yu. 2019. Understanding the representation power of graph neural networks in learning graph topology. *Advances in Neural Information Processing Systems*, 32.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2021. **BROS: A layout-aware pre-trained language model for understanding documents**. *CoRR*, abs/2108.04539.
- Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2022. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10767–10775.
- Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE.
- Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. 2021. **Spatial dependency parsing for semi-structured document information extraction**. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 330–343, Online. Association for Computational Linguistics.
- Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. **Funsd: A dataset for form understanding in noisy scanned documents**. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6.
- Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.
- Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Ren-shen Wang, Yasuhisa Fujii, and Tomas Pfister. 2022. **FormNet: Structural encoding beyond sequential modeling in form document information extraction**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3735–3754, Dublin, Ireland. Association for Computational Linguistics.
- Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021. Structurallm: Structural pre-training for form understanding. *arXiv preprint arXiv:2105.11210*.
- Andreas Loukas. 2019. What graph neural networks cannot learn: depth vs width. *arXiv preprint arXiv:1907.03199*.
- Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley Wendt, Qi Zhao, and Marc Najork. 2020. **Representation learning for information extraction from form-like documents**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6495–6504, Online. Association for Computational Linguistics.
- Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2200–2209.
- Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609.
- Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. {CORD}: A consolidated receipt dataset for post-{ocr} parsing. In *Workshop on Document Intelligence at NeurIPS 2019*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. **Kleister: Key information extraction datasets involving long documents with complex layouts**. In *Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I*, page 564–579. Springer-Verlag.

Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. 2016. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. **LayoutLMv2: Multi-modal pre-training for visually-rich document understanding**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online. Association for Computational Linguistics.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. **Layoutlm: Pre-training of text and layout for document image understanding**. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20*, page 1192–1200. Association for Computing Machinery.

A Appendix

A.1 Implementation details

We briefly describe the baseline models as well as detailed implementation details of all models in this section.

- **BERT_{BASE}** (Devlin et al., 2019): We use the pre-trained BERT base model for token classification.
- **GCN** (Kipf and Welling, 2016): We use sentence transformer (Reimers and Gurevych, 2019) to get the embeddings of text inputs and use them as the node features for the constructed *KNN* graph. Then we train a 2-layer graph convolutional network to classify the nodes/entities.

- **LayoutLM_{BASE}** (Xu et al., 2020): LayoutLM is a transformer-based model for document image understanding. It is pre-trained on IIT-CDIP Test Collection with 11 million scanned images.

- **LayoutLMv2_{BASE}** (Xu et al., 2021): In addition to LayoutLM, the LayoutLMv2 adds a new multi-modal task during pre-training to take in the visual cues and incorporates a novel spatial-aware self-attention mechanism.

- **StructuralLM_{LARGE}** (Li et al., 2021): On top of LayoutLM, Structural LM uses cell position for each word, and introduces a new pre-training task that predicts the cell position. It is also pre-trained on the IIT-CDIP dataset.

- **SPADE** (Hwang et al., 2021): SPADE builds a directed graph of document entities and extracts and parses the spatial dependency using both linguistic and spatial information.

- **BROS** (Hong et al., 2022): Similar to LayoutLM, BROS is also pre-trained on the IIT-CDIP dataset, but with a different area masking pre-training task, and a different method to encode the 2D positions of bounding boxes.

- **DocFormer** (Appalaraju et al., 2021): DocFormer is a multi-modal transformer that takes in both text and visual cues. It proposes a multi-modal attention mechanism and is pre-trained with several tasks involving both text and image input.

All models are trained on 16G V100 GPUs and implemented with Pytorch, except for StructuralLM_{LARGE}, for which we use their official repository ² that is implemented in Tensorflow1.14 and we train it on cpu because of cuda version mismatch. We use APIs open-sourced by Huggingface ³ for Bert, LayoutLM_{BASE} and LayoutLMv2_{BASE}. SPADE is implemented using the official implementation released by ClovaAI⁴. BROS is implemented using their released official repository ⁵. Only text inputs are passed to BERT_{BASE} for classification while bounding box coordinates are neglected. Results are obtained after training for 100 epochs. We trained the SPADE

²<https://github.com/alibaba/AliceMind/StructuralLM>

³<https://huggingface.co>

⁴<https://github.com/clovaai/spade>

⁵<https://github.com/clovaai/bros>

model for 10 to 20 hours up to 1000 epochs depending on the datasets. All settings of $\text{LayoutLM}_{\text{BASE}}$ and $\text{LayoutLMv2}_{\text{BASE}}$ are from the authors. For BROS, we use the same tokenizer as LayoutLM , same learning rate in their paper and fine-tuned BROS on each dataset for at least 100 epochs, and made sure it converged. We report results for epoch 80. For $\text{StructuralLM}_{\text{LARGE}}$, we were only partially successful to reproduce it due to OOV error when running on POI dataset. In addition, this is the only baseline that we use the large version because there was an error with the base version. We train the model with 25 epochs with all other hyperparameters following their paper. We reproduced DocFormer from an unofficial repository⁶ since there is no official repository available. There is no released pretraining weights for DocFormer, but DocFormer uses plain ResNet50 (He et al., 2016) as the first step for image feature extraction, and the language embedding weights are initialized with $\text{LayoutLMv1}_{\text{BASE}}$ pre-trained weights. We trained DocFormer for at least 100 epochs and used hyperparameters for fine-tuning setting mentioned in the paper. We report results for epoch 100.

For *KNN-Former*, we use 8 layers, 8 heads, and 80 hidden dimensions for the architecture. Results are obtained after training for 400 epochs. We use a 6-layer sentence transformer to extract text features in for both *KNN-Former* and GCN baseline model implementation. We use Adam optimizer with learning rate of $5e-3$. We perform a grid search in choosing hyper-parameters, with learning rate in [$5e-3$, $1e-3$, $5e-4$], the number of layers in [4, 8], local attention threshold in [1,2,3], and the number of attention heads in [4,8]. To incorporate relative Euclidean distance and angle, we tried both real and quantized angles in our initial exploration and did not find a significant difference. We use real angle values throughout the experiments. In the implementation of combinatorial matching, we choose class probabilities as matching cost following (Carion et al., 2020). Despite no theoretical justification, they observe better performance than log probabilities. We conduct experiments comparing class and log probabilities but do not observe significant differences in POI dataset ($<0.005\%$). Reported results are the average performance of 3 runs. The sentence transformer we used is paraphrase-MiniLM-L6-v2 from hugging face.

⁶<https://github.com/shabie/docformer>

A.2 Experimental Results on MIDV2020 random split

Tab 8 shows the additional experimental results on MIDV2020 random split. Column L.Name, F.Name, DoB, DoI, DoE and ID No. correspond to results of Last Name, First Name, Date of Birth, Date of Issue, Date of Expiry, and ID Numbers. GCN and *KNN-Former* have additional 22.7 M fixed parameters since we employed a light-weighted 6-layer sentence transformer (Reimers and Gurevych, 2019) to get the text embeddings. MIDV dataset has 10 templates, and each template has 100 images. As a result, this random split is an easy setting where performance results are generally good. $\text{BERT}_{\text{BASE}}$ still produces relatively poor performance, which reiterates the point that spatial information is important.

A.3 Experimental Results on DocFormer

Tab 9 shows the experimental results of DocFormer on various datasets. On POI, PRV dataset and MIDV2020 dataset random split, DocFormer performs reasonably well. On POI dataset, it only falls behind $\text{LayoutLMv2}_{\text{BASE}}$ and *KNN-Former*; on PRV dataset, it outperforms $\text{BERT}_{\text{BASE}}$, GCN and SPADE; on MIDV2020 dataset random split, it achieves 100% F1 score for every field like *KNN-Former*, $\text{StructuralLM}_{\text{LARGE}}$, $\text{LayoutLM}_{\text{BASE}}$ and $\text{BROS}_{\text{BASE}}$. However, on MIDV2020 dataset country split, we cannot get reasonable performance for DocFormer although we made sure our training was converged.

We also measured the runtime of DocFormer, results shown in Tab. 10.

A.4 POI Dataset Details

All images are publicly available specimen ID documents and do not contain information about real persons. Despite that, due to the sensitivity of the subject and increasing societal concerns about the role artificial intelligence should play in protecting people’s privacy, we will only release the annotated JSON file instead of the actual images to comply with fair use of specimens.

We store a list of objects in the annotated file; each object contains annotations for an image. The annotations include bounding box coordinates, text, and category.

The released dataset is subject to fair use clause and should only be used for academic purposes.

We implement quality control during the annota-

Dataset	Method	F1 Score						Trainable Param
		L.Name	F.Name	DoB	DoI	DoE	ID No.	
MIDV	BERT _{BASE}	72.09	81.35	100.00	92.99	88.48	76.52	110 M
	GCN	51.48	61.66	98.68	91.59	88.55	73.90	31.5 K
	StructuralLM _{LARGE}	100.00	100.00	100.00	100.00	100.00	100.00	355M
	LayoutLM _{BASE}	100.00	100.00	100.00	100.00	100.00	100.00	110 M
	LayoutLMv2 _{BASE}	99.47	99.74	100.00	100.00	100.00	100.00	199 M
	SPADE	88.14	86.82	70.63	80.33	79.71	87.55	128 M
	BROS _{BASE}	100.00	100.00	100.00	100.00	100.00	100.00	109 M
	<i>KNN-Former</i>	100.00	100.00	100.00	100.00	100.00	100.00	0.5 M

Table 8: Experimental Results on MIDV2020 Random Split.

Dataset	Method	F1 Score						Input Modality	#Parameters	
		L.Name	F.Name	DoB	DoI	DoE	ID No.		Trainable	Total
POI	DocFormer _{BASE}	78.22	78.87	95.15	90.99	91.82	81.65	text + layout + image	110M	110M
PRV		78.21	84.86	98.17	96.42	97.38	91.89			
MIDV2020 (random split)		100.00	100.00	100.00	100.00	100.00	100.00			
MIDV2020 (country split)		1.50	0.00	0.00	1.91	0.00	0.00			

Table 9: Experimental Results on DocFormer.

Model	Single	Batch
LayoutLM _{BASE}	19.61	237.90
LayoutLMv2 _{BASE}	56.64	2941.32
SPADE	39.47	6091.52
BROS _{BASE}	23.45	646.65
DocFormer _{BASE}	71.57	7485.10
<i>KNN-Former</i>	22.60	77.57

Table 10: Runtime comparison with baselines. Time taken is reported in milliseconds.

tion process by having annotators cross-check each other’s results to affirm the correctness of labels.

A.5 Sample documents of POI and MIDV2020

In Fig. 3, we show samples documents with bounding boxes and annotations.

A.6 PRV Dataset Details

Since POI and MIDV2020 only contain specimens or artificially generated images, we run our model on a private (PRV) dataset that consists of actual ID documents. The documents are protected by strict privacy requirements and massive human annotations are not available as raw images are inaccessible. Therefore, we build automatic labeling to annotate the ground truth. Specifically, we map personal information in the existing database to OCR-ed text outputs. The matched bounding box is classified as the corresponding entity if a match is found. All bounding boxes that are not matched are classified as ‘others’.



(a) POI document



(b) Original MIDV2020 document



(c) Enhanced MIDV2020 document

Figure 3: Example documents with bounding boxes and annotations. There is only one entity box corresponding to one field of interest.

On the Generalization Ability of Retrieval-Enhanced Transformers

Tobias Norlund^{1,4*} Ehsan Doostmohammadi² Richard Johansson^{1,3} Marco Kuhlmann²

¹ Chalmers University of Technology ² Linköping University

³ University of Gothenburg ⁴ Recorded Future

Abstract

Recent work on the Retrieval-Enhanced Transformer (RETRO) model has shown that off-loading memory from trainable weights to a retrieval database can significantly improve language modeling and match the performance of non-retrieval models that are an order of magnitude larger in size. It has been suggested that at least some of this performance gain is due to non-trivial generalization based on both model weights and retrieval. In this paper, we try to better understand the relative contributions of these two components. We find that the performance gains from retrieval largely originate from overlapping tokens between the database and the test data, suggesting less non-trivial generalization than previously assumed. More generally, our results point to the challenges of evaluating the generalization of retrieval-augmented language models such as RETRO, as even limited token overlap may significantly decrease test-time loss. We release our code and model at <https://github.com/TobiasNorlund/retro>

1 Introduction

Large-scale generative language models have shown promising results toward creating a general-purpose foundation for many natural language applications. While sheer scale-up has resulted in better language modeling performance, the immense costs are an inhibiting factor towards further improvements (Sharir et al., 2020).

Recent work on retrieval-augmented language models, such as the Retrieval-Enhanced Transformer (RETRO; Borgeaud et al., 2022), suggests that *memory* can be effectively off-loaded from the model parameters to an external database. In RETRO, the information retrieved from the database is used to augment the context from which the model predicts new tokens, reducing the need to memorize this information in the model parameters. This opens up for smaller language models with retained performance. Specifically, Borgeaud et al. (2022) report that, with a large enough retrieval

database, RETRO can achieve a performance comparable to GPT-3 (Brown et al., 2020) and Jurassic-1 (Lieber et al., 2021) on the Pile (Gao et al., 2020), at only 4% of the parameters. Similarly, RETRO achieves significantly lower bits-per-byte performance compared to a baseline of the same size without retrieval.

Borgeaud et al. (2022) conclude that RETRO has the capacity for non-trivial generalization based on both the model parameters and the retrieval database, even though they find that part of the performance gains can be attributed to lexical overlap between retrieval and test data. In this work, we want to better understand the nature and magnitude of this effect. Our findings indicate that performance gains¹ originate *almost exclusively* from RETRO’s ability to copy tokens verbatim from retrieved data, effectively exploiting any (small or large) overlap between training and test data. This suggests that the ability of RETRO to fuse retrieved and in-parameter information may be more limited than previously assumed.

2 Method

To investigate gains from retrieval, we re-implement the RETRO model described by Borgeaud et al. (2022) (with a few deviations; see below). We present the model here in brevity.

2.1 The RETRO Model

RETRO is an autoregressive language model trained with the next-token prediction objective, where the prediction probability is conditioned on additional context retrieved from a database.

Retrieval Retrieval occurs at the granularity of contiguous token chunks with a fixed size m . More specifically, assume that RETRO has already generated a sequence of tokens $x_{1:t}$. Each token x_i

¹Results on RETRO were originally reported in bits-per-byte, while we report results in loss.

*Corresponding author, tobiasno@chalmers.se

belongs to a chunk $C_{c(i)}$, where $c(i) = \lceil i/m \rceil$. The probability of the next token x_{t+1} depends on the previously generated tokens and the context retrieved from the previously seen chunks:

$$P(x_{t+1} | x_{1:t}, \text{RET}(C_1), \dots, \text{RET}(C_{c(t+1)-1}); \theta)$$

Database RETRO’s database takes the form of a key-value storage $R(N) \mapsto [N, F]$, where N is a chunk from one of the indexed documents, F is the immediately following chunk, and the key $R(N) \in \mathbb{R}^d$ is the embedding of N according to some embedding model R . This database is used to retrieve the k nearest neighbors of a chunk C , based on the embedding $R(C)$:

$$\text{RET}(C) = ([N^1, F^1], \dots, [N^k, F^k])$$

Architecture RETRO is based on the original Transformer architecture (Vaswani et al., 2017). Chunk neighbors are encoded by the encoder and attended to by the decoder. Due to the quadratic complexity in self-attention, each neighbor is encoded separately; all representations are then concatenated and made available to the decoder (Izacard and Grave, 2021). The original decoder is modified such that for the prediction of token x_{t+1} , cross-attention (CA) can only attend to the neighbor representations retrieved based on the previous chunk $C_{c(t+1)-1}$. This is called *chunked cross-attention* (CCA). Furthermore, the encoder is modified to include a restricted form of cross-attention to the decoder. Specifically, the encoder CA attends to the decoder hidden states immediately before the first CCA. We refer to Borgeaud et al. (2022) for more details.

Implementation Details For tokenizing documents, we use the pre-trained T5 tokenizer. The retrieval was performed using approximate nearest neighbor search with the high-performant faiss library (Johnson et al., 2019). We implement RETRO in PyTorch (Paszke et al., 2019) and use PyTorch Lightning for distributing the training and validation data across GPUs and compute nodes. Our implementation deviates from that of Borgeaud et al. (2022) only in that we

- use learnable relative positional biases as in T5 (Raffel et al., 2020), with a bucket for each unique relative position; and
- instantiate the chunk embedding model R by a pre-trained Sentence-BERT (SB) model (Reimers and

Gurevych, 2019) instead of BERT. We deemed SB to be preferable over BERT as it is smaller (i.e. cheaper to compute) and produces embeddings of lower dimensionality (i.e. saves disk space).

2.2 Dataset

Borgeaud et al. (2022) used a multi-lingual version of *MassiveText* (Rae et al., 2021) for both training and retrieval data. To replicate the English portion of this data, we sought open-source alternatives. *MassiveText* comprises text from the categories web text, news, code, books, and Wikipedia. By pooling matching categories from Pile (Gao et al., 2020) and adding the RealNews dataset (Zellers et al., 2019), we obtain a large dataset composed of all five categories, consisting of 36M documents and 52B tokens. We keep the training/validation splits from the Pile categories. For RealNews, we use the provided training set and a subsample of 16,400 documents from the validation set. The full description of our dataset is shown in Table 1.

2.3 Model Training

For our experiments, we train a RETRO model that resembles the 425M model² in Borgeaud et al. (2022), as shown in Table 2. We train and test on our open-source version of *MassiveText* as described in Section 2.2. During training, we retrieve neighbors from the training set, while at validation time, we retrieve from the union of training and validation sets. We filter out neighbors that originate from the same source document as the query chunk. Each model is trained on sequences of no more than 1,024 tokens; longer sequences are truncated. We use a chunk size of 64 and retrieve two neighbors during both training and validation. We train the model for 140k training steps with a batch size of 16. This means that only 6% of the training documents are actually used during training, excluding retrieved neighbors. We use the Adam optimizer with a fixed learning rate of $1e-4$.

3 Experiments

Borgeaud et al. (2022) observed that retrieval increases language modeling performance. To validate this observation, we compare two configurations of our model: RETRO[ON], where we enable retrieval, and RETRO[OFF], where we remove the CCA layers, thereby reducing RETRO to a standard decoder-only language model. As we can see in

²The 425M parameters exclude embeddings.

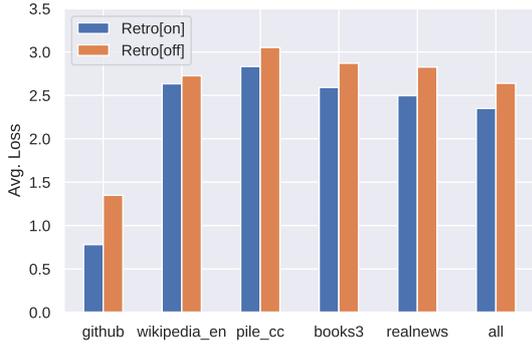


Figure 1: Comparing loss on validation set categories, when using retrieval vs. no retrieval.

Figure 1, retrieval reduces the loss across all data categories, and with 11% across the full validation set. GitHub data has the lowest validation loss among all categories and is also where we see the largest reduction in loss, at 42%. Wikipedia sees the smallest reduction in loss, at only 3%. A closer comparison to the results from Borgeaud et al. (2022) is available in Appendix D.

3.1 Loss per Degree of Overlap

As Borgeaud et al. (2022) note, retrieval-based models such as RETRO may more easily exploit evaluation dataset leakage. To quantify how much of the positive effect of retrieval on language modeling performance can be attributed to such leakage, the authors computed bits-per-byte (bpb) for evaluation chunks with different amounts of consecutive token overlap relative to their retrieved neighbors. This analysis showed that, while the positive effect of retrieval decreased with smaller overlaps, it was still significant at overlap levels of at most 8 contiguous tokens, which the authors considered small enough to conclude that while RETRO actually learns to *generalize* from retrieval data, not merely copy-and-paste it. Here we investigate the hypothesis that the bpb reductions observed by Borgeaud et al. (2022) are localized exclusively in the overlapping tokens. If this was true, it would challenge the conclusion that RETRO learns non-trivial generalizations based on retrieval data.

To test our hypothesis, we sort the validation set tokens into buckets based on their leftward overlap. Specifically, we put a token x_i into a bucket $\Phi(n)$, where n is the largest number such that x_i and the $n - 1$ tokens preceding it consecutively overlap with some neighboring chunk in $\text{RET}(C_{c(i)-1})$. For example, the bucket $\Phi(1)$ contains all tokens x_i for

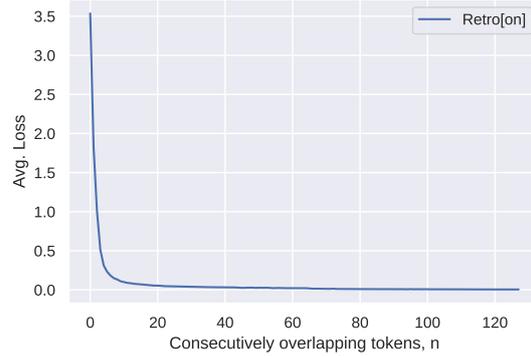


Figure 2: Average loss from RETRO[ON] over tokens in $\Phi(n)$. Note the drastic decrease with increasing overlap.

which the unigram x_i appears in some neighbor, but not the bigram $x_{i-1}x_i$; the bucket $\Phi(2)$ contains all x_i for which $x_{i-1}x_i$ overlaps but not $x_{i-2}x_{i-1}x_i$, and so on. As a special case, the bucket $\Phi(0)$ contains all tokens that do not overlap with any of its neighbors. This includes all tokens that occur in a first chunk C_1 , which lacks neighbors.

In Figure 2 we plot the average loss per bucket,

$$\frac{1}{|\Phi(n)|} \sum_{x_i \in \Phi(n)} \mathcal{L}_{x_i}^{\text{RETRO[ON]}}, \quad (1)$$

as a function of n . Here, $\mathcal{L}_{x_i}^{\text{RETRO[ON]}}$ is the loss when predicting token x_i using RETRO[ON]³. We see that the loss drastically decreases as the consecutive overlap increases. For example, at an overlap of $n = 5$ tokens, the loss is only 6% of the loss for non-overlapping tokens. This suggests that RETRO enters “copy mode” when the previous tokens overlap with those from a neighbor.

3.2 Loss Reductions per Degree of Overlap

For a more detailed analysis of the effect of overlap on predictive performance, we look at the token-specific loss differences between the two configurations RETRO[OFF] and RETRO[ON]:

$$\Delta \mathcal{L}_{x_i} = \mathcal{L}_{x_i}^{\text{RETRO[OFF]}} - \mathcal{L}_{x_i}^{\text{RETRO[ON]}}$$

Note that a loss difference $\Delta \mathcal{L}_{x_i}$ is positive if the access to the retrieved context reduces the token-specific loss for x_i . The overall reduction in loss visible in Figure 1 is the average of the loss differences across all tokens in the validation data. By aggregating loss differences per bucket $\Phi(n)$, we get a fine-grained picture of how the reductions

³The sizes of each bucket (accumulated over the validation data) are shown in the appendix, Figure 4.

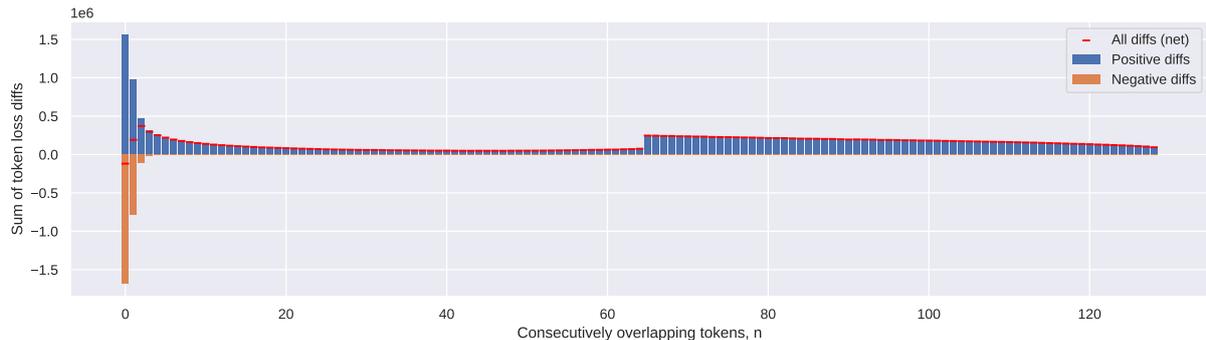


Figure 3: Token-specific loss differences, as distributed over different degrees of overlap. *Positive diffs* shows the sum of all positive loss differences, $\sum_{x_i \in \Phi(n)} \max(0, \Delta \mathcal{L}_{x_i})$, and *Negative diffs* shows the sum of negative loss differences, $\sum_{x_i \in \Phi(n)} \min(0, \Delta \mathcal{L}_{x_i})$. *All diffs* shows the total sum. We see that the vast majority of loss reductions comes from overlapping tokens, e.g. $n > 0$.

are distributed with respect to different degrees of consecutive overlap. This is illustrated in Figure 3.

For non-overlapping tokens ($n = 0$), we can see that there are both positive and negative differences, with a small negative net. For all overlapping tokens ($n > 0$), the net differences are positive, and for buckets with 3 or more overlapping tokens, there are almost no negative differences at all.⁴ This shows that the largest share of all loss reductions originates from tokens that are consecutively overlapping in neighbors. Interestingly, the net differences are positive even for very small degrees of overlap. Borgeaud et al. (2022) considered reductions in bits-per-byte from chunks with up to 8 consecutively overlapping tokens as evidence of a non-trivial generalization capacity. However, our results suggest that even a small number of overlapping tokens may cause a large reduction in loss, which we take as an argument against this conclusion.

4 Related Work

Equipping language models with a retrievable external memory has been extensively studied (Guu et al., 2020; Karpukhin et al., 2020; Lewis et al., 2020; Izacard and Grave, 2021; Li et al., 2022). Explicitly leveraging the training data through retrieval to reduce perplexity is proposed in kNN-LM (Khandelwal et al., 2020). kNN-LM matches the leftward context with the leftward context of all training data tokens, and explicitly interpolates between generating and copying the next token. A recent study analyzes kNN-LM to better understand

⁴We note a sudden increase in accumulated loss difference for $n > 64$ which is expected considering the way in which we construct the buckets; see Appendix C for more details.

the causes of performance gains (Xu et al., 2023). Similar to our findings in RETRO, lexical overlap has also been found to play a significant role in explaining retrieval performance gains in kNN-LM as well (Drozdo et al., 2022). The idea of kNN-LM is extended in SPALM (Yogatama et al., 2021) to instead learn a gating function that facilitates more dynamic interpolation.

In both kNN-LM and SPALM, retrieval is incorporated at the top of the network. This might induce a bias towards surface-level rather than semantic augmentation. In contrast, retrieval in RETRO is incorporated in lower layers of the network, which opens up for more sophisticated integration of the retrieved information. Our results suggest, however, that retrieval in RETRO also contributes at the surface rather than at the semantic level, similar to the previous works.

5 Conclusions and Future Work

The capacity of language models for generalization is often measured intrinsically using perplexity, loss or bits-per-byte on held-out validation data. Low perplexity language models perform well as few-shot learners on many downstream tasks due to their capacity to both memorize and non-trivially combine textual information from many sources (Brown et al., 2020; Rae et al., 2021; Lieber et al., 2021; Chowdhery et al., 2022). The hope is that we can externalize memory to reduce the footprints of language models without reducing generalization and downstream task performance.

Our results show that the low loss in RETRO almost exclusively originates from tokens overlapping between retrieval and validation data, rather than from more sophisticated generalization. To better

understand this effect, it would be interesting to modify the retrieval component and deliver semantically similar but lexically different context during training. If the retrieved context is uninformative, the model will learn to ignore it, but if the context is too specific (e.g. literal overlap) the model will learn to copy. By better balancing between these two modes, models may become better at utilizing retrieved information at a deeper and more generalizable level.

Limitations

We have made our best effort in trying to reproduce the model and results of [Borgeaud et al. \(2022\)](#). Nonetheless, our experiments were performed on one of the smaller model sizes and with a dataset that is only $\sim 2.5\%$ of their size (52 billion vs. 2 trillion tokens). This was due to computational constraints and lack of larger open datasets. However, as was also shown by [Borgeaud et al. \(2022\)](#), the performance gain of retrieval is constant with respect to model size. We speculate that larger RETRO models mostly improve with respect to loss on tokens that are not overlapping, which would not change our conclusions here.

One noteworthy limitation of our work is the fact that we compare to a non-retrieval baseline (RETRO[OFF]) that was trained with access to retrieved context. We were not able to train a separate non-retrieval baseline due to computational constraints, but note that the bits-per-byte results of RETRO[OFF] and the baseline in [Borgeaud et al. \(2022\)](#) were close to identical.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Alvis partially funded by the Swedish Research Council through grant agreement no. 2022-06725, and by the Berzelius resources provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste

Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. [Improving language models by retrieving from trillions of tokens](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pilla, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Aleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. [PaLM: Scaling language modeling with pathways](#).

Andrew Drozdov, Shufan Wang, Razieh Rahimi, Andrew McCallum, Hamed Zamani, and Mohit Iyyer. 2022. You can’t pick your neighbors, or can you? when and how to rely on retrieval in the k nn-lm. *arXiv preprint arXiv:2210.15859*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. [The Pile: An 800gb dataset of diverse text for language modeling](#). *arXiv preprint arXiv:2101.00027*.

- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. [Realm: Retrieval-augmented language model pre-training](#). In *International Conference on Machine Learning*, pages 3929–3938. PMLR.
- Gautier Izacard and Edouard Grave. 2021. [Leveraging passage retrieval with generative models for open domain question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. [Billion-scale similarity search with GPUs](#). *IEEE Transactions on Big Data*, 7(3):535–547.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. [Generalization through Memorization: Nearest Neighbor Language Models](#). In *International Conference on Learning Representations (ICLR)*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. [A survey on retrieval-augmented text generation](#). arXiv preprint 2202.01110.
- Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. [Jurassic-1: Technical details and evaluation](#). Technical report, AI21 Labs.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d’Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorraine Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. [Scaling language models: Methods, analysis & insights from training gopher](#).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21(1).
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Or Sharir, Barak Peleg, and Yoav Shoham. 2020. [The cost of training NLP models: A concise overview](#). arXiv preprint 2004.08900.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Frank F. Xu, Uri Alon, and Graham Neubig. 2023. [Why do nearest neighbor language models work?](#)
- Dani Yogatama, Cyprien de Masson d’Autume, and Lingpeng Kong. 2021. [Adaptive semiparametric language models](#). *Transactions of the Association for Computational Linguistics*, 9:362–373.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. [Defending against neural fake news](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

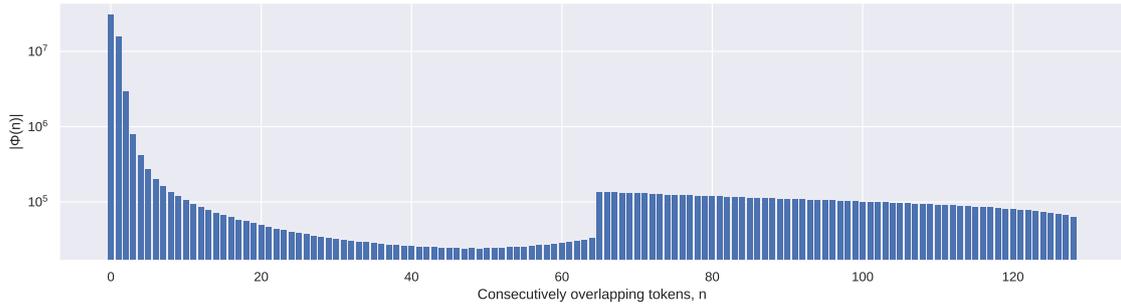


Figure 4: Number of validation set tokens in each bucket $\Phi(n)$. Since the neighbors have a maximal length of 128 tokens, this is also the longest possible overlap n .

		Documents	Chunks	Tokens
Training	Pile-CC	15,728k	269M	16.7B
	Wikipedia En	5,082k	61M	3.8B
	GitHub	5,417k	181M	11.4B
	Books3	83k	191M	12.2B
	RealNews	9,360k	130M	8.0B
	Total	35,670k	833M	52.2B
Validation	Pile-CC	52.8k	900.4k	56.0M
	Wikipedia En	17.4k	215.9k	13.3M
	GitHub	18.3k	598.4k	37.7M
	Books3	0.3k	727.6k	46.5M
	RealNews	16.4k	234.5k	14.5M
	Total	105.3k	2,676.8k	168.0M

Table 1: Statistics for our MassiveOpenText dataset. We use the web text, Wikipedia, GitHub and Books3 corpora from the Pile, and news text from RealNews.

A MassiveOpenText statistics

Statistics on the number of documents, chunks and tokens for each split and text category are shown in Table 1.

B RETRO model details

We show hyperparameters of our RETRO model in Table 2.

	Param	
Encoder	Num layers	2
	Num heads	14
	Hidden size	896
	FFN	3584
	CA layers	[2]
Decoder	Num layers	12
	Num heads	12
	Hidden size	1536
	FFN	6144
	CCA layers	[6,9,12]

Table 2: Hyperparameters of our trained Retro model.

C Consecutively overlapping tokens

As explained in Section 3.1, we sort validation set tokens into buckets denoted $\Phi(n)$ depending on the longest overlapping leftward context.

In Figure 4 we show the number of tokens in each bucket. We note a big “jump” from $n = 64$ to $n = 65$, which can be explained by the following rationale. A neighbor $[N, F]$ to a chunk C_i is retrieved based on the similarity between C_i and N . In the case where both $C_i = N$ and $C_{i+1} = F$, tokens in C_{i+1} will be put into $\Phi(n)$ with $n = 65, \dots, 128$. The jump in Figure 4 indicates such duplicates are common in our data.

D Model validation

As we aim to reproduce the 425M model trained in Borgeaud et al. (2022), it is important to validate that the implementations are equivalent and that their evaluation results are comparable. However, evaluations of the 425M model in Borgeaud et al. (2022) on the Pile are not available, making it hard to make direct comparisons. Borgeaud et al.

(2022) report evaluation results on the C4 (Raffel et al., 2022) dataset, with various sizes of retrieval datasets. For their setup with 36B retrieval tokens, which is the most similar to our own retrieval size, they report that bits-per-byte is reduced by $\sim 2\%$ (from 0.92 to 0.90) when using retrieval. That could be compared to our results on Pile-CC, as both datasets originate from Common Crawl. In our experiments, loss is reduced by 7% (from 3.05 to 2.83) on Pile-CC.

Evaluations on the Pile in Borgeaud et al. (2022) are only reported for their largest model (7B params) and largest retrieval set (2T tokens). For example, on Pile-GitHub their reduction is $\sim 53\%$ whereas our reduction is 42% .

While these numbers are not directly comparable, we believe they indicate that our reimplementation of the RETRO model is working as expected.

Assessing Monotonicity Reasoning in Dutch through Natural Language Inference

Gijs Wijnholds

Institute for Language Sciences

Utrecht University

g.j.wijnholds@uu.nl

Abstract

In this paper we investigate monotonicity reasoning in Dutch, through a novel Natural Language Inference dataset. Monotonicity reasoning shows to be highly challenging for Transformer-based language models in English and here, we corroborate those findings using a parallel Dutch dataset, obtained by translating the Monotonicity Entailment Dataset of Yanaka et al. (2019). After fine-tuning two Dutch language models BERTje and RobBERT on the Dutch NLI dataset SICK-NL, we find that performance severely drops on the monotonicity reasoning dataset, indicating poor generalization capacity of the models. We provide a detailed analysis of the test results by means of the linguistic annotations in the dataset. We find that models struggle with downward entailing contexts, and argue that this is due to a poor understanding of negation. Additionally, we find that the choice of monotonicity context affects model performance on conjunction and disjunction. We hope that this new resource paves the way for further research in generalization of neural reasoning models in Dutch, and contributes to the development of better language technology for Natural Language Inference, specifically for Dutch.

1 Introduction

Natural Language Inference (NLI) is one of the standard benchmark tasks for current-day NLP architectures. In this task a model takes two sentences as input, and has to classify the relationship between the former (premise) sentence and the latter (hypothesis) sentence, typically between Entailment, Contradiction, and Neutral. NLI makes for an interesting task as drawing the correct inference may require subtle aspects of syntax, lexical semantics, and even pragmatics. While many NLI datasets exist like SICK (Marelli et al., 2014), SNLI (Bowman et al., 2015) and its extensions (MNLI Williams et al. (2018), XNLI Conneau et al. (2018),

e-SNLI Camburu et al. (2018)), much is still unknown about how and why neural language models (LMs) like BERT (Devlin et al., 2019) perform on the task. Evidence shows that fine-tuned LMs don't generalize well across NLI benchmarks (Talman and Chatzikyriakidis, 2019), and other investigation shows that LMs may be exploiting dataset heuristics to solve the task (Naik et al., 2018; McCoy et al., 2019). More generally, LMs do seem to encode a certain amount of syntactic structure (Rogers et al., 2020), but the relation to NLI remains unclear.

In order to shed light on the performance of large-scale LMs, specific datasets have been developed to understand what models do and don't understand. Specifically in the context of NLI, Yanaka et al. (2019) introduce the Monotonicity Entailment Dataset (MED), which targets models' capacity for understanding *monotonicity reasoning* (Icard III and Moss, 2014). Monotonicity reasoning is a staple test of human reasoning which requires lexical knowledge, as well as syntactic knowledge, making it suitable for an NLI benchmark.

In cases of monotonicity reasoning, a particular lexical item in the sentence licenses inferences by means of substituting specific syntactic constituents by either more general (upward context) or more specific (downward context).

- (a) *Every* [*man* ↓] [*sung and danced* ↑].
- (b) *Every bald man sung and danced.* ✓
- (c) *Every man danced.* ✓
- (d) *Every human sung and danced.* ✗

Figure 1: Example cases of monotonicity reasoning as natural language inference.

In Figure 1, the quantifier *Every* is downward entailing in its first argument, and upward entailing in its second arguments, meaning that either *man* may be replaced by a more specific instance to obtain an inference pair – as in 1(b) – while *sung and*

danced ought to be substituted for a more general constituent to preserve inference, as in 1(c). Violating the entailment context leads to a hypothesis for which there is no entailment (but not necessarily a contradiction), as in 1(d).

While the field of research into NLI is lively, it is largely focused on English. In this article, we work with Dutch, a language that has a relatively high digital prevalence, while at the same time being underrepresented in terms of typical sentence-level NLP benchmarks.

For Dutch there is the Lassy corpus, which contains a smaller gold standard, and a larger silver standard syntactically annotated corpus of written (van Noord et al., 2013), and the SONAR corpus of written Dutch (Oostdijk et al., 2013). Given the availability of these corpora combined with a rich Wikipedia dump, two transformer-based language models have been developed for Dutch, based on the BERT architecture (BERTje, de Vries et al. (2019)) and the RoBERTa architecture (RobBERT, Delobelle et al. (2020)), both available through HuggingFace’s transformers library.¹ In terms of investigations into these Dutch language models, de Vries et al. (2020) argues that BERTje encodes a typical ‘NLP pipeline’, which had been argued for BERT before (Tenney et al., 2019), whereas Kogkalidis and Wijnholds (2022) show through probing that long-distance dependencies are hard to recognize for both Dutch language models.

In order to extend the research done on NLI and on Dutch NLP, we add a benchmark for monotonicity reasoning in Dutch by translating the MED dataset of Yanaka et al. (2019). We perform an evaluation of large-scale language models for Dutch on this novel benchmark, that we dub MED-NL. We corroborate the findings of Yanaka et al. (2019), observing that the Dutch LMs similarly have difficulty with inferences coming from downward entailing contexts. Further inspection suggests that the main problem comes from inference pairs containing negation. In what follows, we first detail the creation of the dataset and the experimental setup for the evaluation, after which we report results and inspect the model predictions.

2 Dataset Creation & Evaluation

The dataset is obtained by translation from the English MED dataset of (Yanaka et al., 2019). First,

¹There is also a distilled version of RobBERT (Delobelle et al., 2021) which we did not include in our experiments.

all 5241 unique sentences are collected and lexicographically sorted to ensure consistency among sentence translations. These sentences are given to a native Dutch speaker for translation who could ensure quality and naturality of the translated examples. Using the translated sentences, we populate the original dataset with its Dutch incarnation. Since the original Entailment/Neutral labels derive from monotonicity properties, the entailment labels are preserved in Dutch. It is important to note that the labelling is binary, since MED only considers entailment and non-entailment (or neutral).

	MED	MED-NL
No. of tokens	81209	83809
No. of unique tokens	3614	3883
Avg. sentence length	7.54	7.79
Avg. word overlap	74.60%	73.25%

Table 1: Basic statistics of MED vs MED-NL.

Table 1 shows that in the translation, there is a 3% blowup in the number of words used in Dutch, with the corresponding increase in average sentence length. However, the number of unique tokens in the Dutch dataset increased, owing to a plurality of interpretation of English source words that may get disambiguated in the translation process.

To evaluate, we then performed a standard language model fine-tuning routine. We use two state of the art Dutch neural language models; BERTje (de Vries et al., 2019), a BERT-based model pre-trained for Dutch, and RobBERT (Delobelle et al., 2020), a RoBERTa-based model for Dutch. For multilingual comparison, we furthermore train multilingual BERT (Devlin et al., 2019). Each model was trained on the SICK-NL dataset (Wijnholds and Moortgat, 2021), which is the only existing NLI benchmark for Dutch. We binarize the labels in SICK-NL by conflating all Neutral and Contradiction labels into one class, as to make the training data compatible with the binary format of MED-NL. Training proceeds for 20 epochs, and the model is saved for the epoch for which highest development accuracy is obtained.² We test on the SICK-NL for validation purposes, after which testing is performed on MED-NL. To reduce any potential influences of performance perturbation due

²For BERTje, highest development accuracy was achieved at epochs 3, 5, and 5, whereas RobBERT achieve peak development set performance at epochs 5, 11, and 13.

to model seed initialization, we train each model thrice and report seed-averaged accuracy.

3 Results & Analysis

Table 2 displays the average development and test accuracy on SICK-NL, and test performance on MED-NL.

	SICK-NL _d	SICK-NL _t	MED-NL
BERTje	86.89	87.40	47.56
RobBERT	86.43	85.79	46.07
mBERT	71.20	71.38	49.74

Table 2: Seed-averaged (over 3 runs) accuracy results for two Dutch BERT models and multilingual BERT, trained on SICK-NL, evaluated on both SICK-NL and the new MED-NL dataset.

Performance on the development and test set of SICK-NL are slightly higher than reported in related work (Wijnholds and Moortgat, 2021; Delobelle et al., 2021), which may be due to the fact that the classification labels have been binarized. The high drop in accuracy on MED-NL is however on par with reported results on its English counterpart (Yanaka et al., 2019), despite the models and training dataset being different. In terms of difference between the models, overall accuracy barely distinguishes BERTje and RobBERT in terms of pure performance. Interestingly, multilingual BERT has a performance decline of ca 15% compared to RobBERT, yet reached highest performance on MED-NL. The multilingual model has more trouble with the Dutch training data, although all three runs reached peak validation accuracy after one epoch of training.

Monotonicity Contexts A breakdown of accuracy results by the type of monotonicity context is given in Table 3, which shows that non-upward entailing contexts typically represent a challenge to the language models’ predictions.

	Total	Up	Down	Non
(Support)	(5382)	(1818)	(3272)	(292)
BERTje	47.56	64.76	38.72	39.50
RobBERT	46.07	61.13	39.22	28.30
mBERT	49.74	65.57	36.67	97.60

Table 3: Seed-averaged (over 3 runs) accuracy results on the MED-NL dataset, by monotonicity category.

Specifically, these results contrast the performance of the monolingual Dutch LMs with multilingual BERT, the latter doing the worst on downward entailing contexts while trumping the former models on non-monotone contexts.

Linguistic Features In order to delve deeper in the results, we make use of the annotations in the dataset that indicate specific linguistic features for premise/hypothesis pairs. Table 4 displays detailed scores for linguistic features that have a significant overall occurrence in the MED-NL dataset, where we display the number of occurrences next to the name of the feature.

Phenomenon		BERTje	RobBERT	mBERT
<i>Lexical</i>	743	62.72	58.73	77.39
<i>Conjunction</i>	177	65.16	61.77	58.76
<i>Disjunction</i>	96	24.31	29.86	53.12
↑ <i>Conditionals</i>	24	48.61	44.44	70.83
<i>NPI</i>	64	33.33	36.98	64.06
<i>Reverse</i>	235	52.91	51.63	50.21
<i>Other</i>	698	74.79	69.91	58.74
<i>Lexical</i>	477	33.47	34.45	29.98
<i>Conjunction</i>	106	34.91	32.08	23.58
<i>Disjunction</i>	138	49.76	49.52	40.58
↓ <i>Conditionals</i>	125	45.60	43.47	18.40
<i>NPI</i>	266	36.59	39.10	32.71
<i>Reverse</i>	9	29.63	33.33	33.33
<i>Other</i>	2249	39.6	40.15	39.88
<i>Lexical</i>	182	37.73	31.32	98.35
= <i>Disjunction</i>	20	56.67	31.67	100.0
<i>NPI</i>	8	66.67	37.50	100.0
<i>Other</i>	90	39.26	23.70	95.56

Table 4: Seed-averaged (over 3 runs) accuracy results on the MED-NL dataset, by monotonicity category and phenomenon.

These results start to highlight an interesting pattern: with an overall performance on upward entailing contexts of 64.76 (BERTje), we see that cases of disjunction, conditionals, negative polarity items and reverse (e.g. double negation) are most challenging in this context. The surprising result here is that such cases are much more on par with the rest in a downward entailing context. Most strikingly, cases of disjunction become easier to deal with than conjunction in a downward entailing context, even though the situation was converse in the case of upward entailing contexts.

Model Comparison Although the results in Table 4 give some insight into the difference between models – e.g. RobBERT appears to perform higher at cases with negative polarity items, whereas BERTje performs better at cases of conjunction –, the models seem to be relatively equal in their overall accuracy. To better distinguish the models we analyse the overlap between model predictions.

Phenomenon	\cap	Shared	BERTje	RobBERT
Lexical	75%	47.61	55.26	44.74
Disjunction	81%	38.58	48.98	51.02
Conjunction	82%	52.74	59.16	40.84
\forall Conditionals	81%	43.69	57.73	42.27
NPI	85%	35.49	43.04	56.96
Reverse	94%	51.6	57.21	42.79
Other	86%	46.63	54.34	45.66
Lexical	71%	65.05	57.47	42.53
Disjunction	79%	20.74	39.35	60.65
Conjunction	77%	67.56	57.69	42.31
\uparrow Conditionals	77%	45.39	63.99	36.01
NPI	79%	31.05	39.98	60.02
Reverse	95%	52.4	59.05	40.95
Other	76%	79.4	60.33	39.67
Lexical	87%	31.56	46.37	53.63
Disjunction	86%	49.51	51.75	48.25
Conjunction	91%	31.75	62.55	37.45
\downarrow Conditionals	82%	43.37	57.56	42.44
NPI	87%	35.98	41.14	58.86
Reverse	89%	27.78	33.33	66.67
Other	90%	38.76	47.62	52.38
Lexical	58%	23.71	56.09	43.91
Disjunction	61%	41.55	75.14	24.86
NPI	71%	61.38	100.0	0.0
Other	76%	26.56	75.98	24.02

Table 5: Seed-averaged overlap accuracy results on the MED-NL dataset, between BERTje and RobBERT, by monotonicity category and phenomenon.

Table 5 displays the average overlap between the two monolingual models by feature, together with their shared and individual accuracy, to shed light on where the models differ, color-coded for clarification purposes.

We first observe that the overlap between model predictions overall (the \forall rows) is relatively high with a minimum of 75% and a maximum of 94%. Generally speaking, given that the overlap between model predictions is high, the shared ac-

curacy shows whether models make the same correct/incorrect decisions. This is particularly pronounced in the low accuracy on disjunctions in upward entailing contexts, where models make a lot of shared mistakes, but in their diverging decisions RobBERT has a significantly higher accuracy. The converse is true for conjunction in a downward entailing context where BERT is individually stronger than RobBERT. For the sake of completeness, in Tables 8 and 9 we report overlap results between the Dutch models and multilingual BERT.

The Role of Negation One explanation for the fact that the models perform significantly worse on downward entailing contexts may be that such cases are often constructed through the use of negation words. Table 6 displays the percentages of sentence pairs containing at least one negation word, with specification for conjunction and disjunction.

	Total	Up	Down	Non
	(5382)	(1818)	(3272)	(292)
% Negation	58.86	22.5	84.2	1.37
	\uparrow Conj.	\uparrow Disj.	\downarrow Conj.	\downarrow Disj.
	(177)	(96)	(106)	(138)
% Negation	22.60	18.75	92.45	73.19

Table 6: Percentage of premise/hypothesis pairs in MED-NL containing negation words (*geen, niet, zonder, nooit, niemand*).

Indeed, negation is highly represented in downward monotone contexts, indicating that part of the reason why the models perform so poorly in such context is that they are not sensitive (enough) to negation. Inspection of the distribution of negation in SICK-NL (train set) and MED-NL, displayed in Table 7, shows that models may have learnt to incorrectly classify cases involving negation.

% Negation	SICK-NL	MED-NL
Entailment	1.26	69.80
Non-entailment	31.94	47.81

Table 7: Distribution of negation in cases of entailment and non-entailment in SICK-NL and MED-NL.

However, this explanation can't be replicated in the case of conjunction and disjunction, leaving a further inspection into these cases to future work.

Phenomenon	\cap	Shared	BERTje	mBERT
<i>Lexical</i>	67%	60.08	28.12	71.88
<i>Disjunction</i>	66%	43.03	35.47	64.53
<i>Conjunction</i>	57%	49.44	59.7	40.3
\forall <i>Conditionals</i>	53%	24.23	70.3	29.7
<i>NPI</i>	79%	35.37	40.16	59.84
<i>Reverse</i>	91%	50.91	63.7	36.3
<i>Other</i>	76%	45.67	54.2	45.8
<hr/>				
<i>Lexical</i>	60%	83.4	31.99	68.01
<i>Disjunction</i>	42%	24.97	23.32	76.68
<i>Conjunction</i>	41%	78.91	55.6	44.4
\uparrow <i>Conditionals</i>	67%	64.57	16.19	83.81
<i>NPI</i>	51%	47.87	18.07	81.93
<i>Reverse</i>	91%	51.72	64.94	35.06
<i>Other</i>	44%	88.28	64.48	35.52
<hr/>				
<i>Lexical</i>	90%	29.57	69.73	30.27
<i>Disjunction</i>	85%	44.52	78.21	21.79
<i>Conjunction</i>	83%	24.94	85.0	15.0
\downarrow <i>Conditionals</i>	50%	13.9	77.12	22.88
<i>NPI</i>	87%	32.24	66.81	33.19
<i>Reverse</i>	89%	27.78	33.33	66.67
<i>Other</i>	87%	38.14	50.03	49.97
<hr/>				
<i>Lexical</i>	36%	100.0	2.69	97.31
<i>Disjunction</i>	57%	100.0	0.0	100.0
<i>NPI</i>	67%	100.0	0.0	100.0
<i>Other</i>	37%	96.53	5.63	94.37

Table 8: Seed-averaged overlap accuracy results on the MED-NL dataset, between BERTje and multilingual BERT, by monotonicity category and phenomenon.

4 Conclusion

In this paper we provided MED-NL, a novel NLI dataset for Dutch, which specifically targets monotonicity reasoning. The evaluation of two Dutch language models on this test set shows that the models specifically struggle with cases in downward entailing contexts, which had earlier been established for English as well (Yanaka et al., 2019). However, we indicate specifically that the role of negation words may play a large role in the poor model performance on such cases, giving way for future research into language models and negation.

On the other hand, the evaluation also shows that disjunction is much easier to handle by the models than conjunction, for which no explanation was found. In future investigations, we hope to provide more analysis of these language models, specifically regarding negation.

Phenomenon	\cap	Shared	RobBERT	mBERT
<i>Lexical</i>	63%	58.58	27.24	72.76
<i>Disjunction</i>	62%	42.34	37.75	62.25
<i>Conjunction</i>	60%	46.73	56.58	43.42
\forall <i>Conditionals</i>	55%	23.38	68.16	31.84
<i>NPI</i>	74%	35.7	46.86	53.14
<i>Reverse</i>	90%	50.29	57.88	42.12
<i>Other</i>	72%	44.75	51.0	49.0
<hr/>				
<i>Lexical</i>	59%	80.78	27.73	72.27
<i>Disjunction</i>	46%	31.68	27.8	72.2
<i>Conjunction</i>	44%	73.34	53.04	46.96
\uparrow <i>Conditionals</i>	62%	62.47	13.33	86.67
<i>NPI</i>	55%	50.49	19.74	80.26
<i>Reverse</i>	90%	51.01	57.88	42.12
<i>Other</i>	42%	84.93	59.71	40.29
<hr/>				
<i>Lexical</i>	82%	28.22	62.06	37.94
<i>Disjunction</i>	77%	43.46	70.11	29.89
<i>Conjunction</i>	85%	24.01	78.27	21.73
\downarrow <i>Conditionals</i>	54%	14.57	76.64	23.36
<i>NPI</i>	80%	32.28	65.59	34.41
<i>Reverse</i>	100%	33.33	n/a	n/a
<i>Other</i>	84%	38.1	50.48	49.52
<hr/>				
<i>Lexical</i>	30%	100.0	2.37	97.63
<i>Disjunction</i>	32%	100.0	0.0	100.0
<i>NPI</i>	38%	100.0	0.0	100.0
<i>Other</i>	21%	94.71	4.25	95.75

Table 9: Seed-averaged overlap accuracy results on the MED-NL dataset, between RobBERT and multilingual BERT, by monotonicity category and phenomenon.

5 Limitations

This study was performed with monolingual Dutch models and with multilingual BERT, yet comparison with multilingual BERT on the original MED dataset could be insightful. Given that the distribution of cases of negation is skewed between the dataset used for training and the introduced evaluation dataset, another experiment could have been included in which models are trained to deal with cases of negation in a uniformly distributed way.

6 Acknowledgements

The author wishes to acknowledge support from the Dutch Research Council (NWO) under the scope of the project ‘‘A composition calculus for vectorbased semantic modelling with a localization for Dutch’’ (360-89-070). Furthermore, the author thanks Lois Dona for help with the translation.

References

- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. [XNLI: Evaluating cross-lingual sentence representations](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.
- Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. [BERTje: A Dutch BERT model](#). *arXiv preprint arXiv:1912.09582*.
- Wietse de Vries, Andreas van Cranenburgh, and Malvina Nissim. 2020. [What’s so special about BERT’s layers? a closer look at the NLP pipeline in monolingual and multilingual models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4339–4350, Online. Association for Computational Linguistics.
- Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. [RobBERT: a Dutch RoBERTa-based Language Model](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.
- Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2021. [Robbertje: A distilled dutch bert model](#). *Computational Linguistics in the Netherlands Journal*, 11:125–140.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Thomas F Icard III and Lawrence S Moss. 2014. Recent progress on monotonicity. *Linguistic Issues in Language Technology*, 9:167–194.
- Konstantinos Kogkalidis and Gijs Wijnholds. 2022. [Discontinuous constituency and BERT: A case study of Dutch](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3776–3785, Dublin, Ireland. Association for Computational Linguistics.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. [A SICK cure for the evaluation of compositional distributional semantic models](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 216–223, Reykjavik, Iceland. European Languages Resources Association (ELRA).
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. [Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.
- Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. [Stress test evaluation for natural language inference](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Nelleke Oostdijk, Martin Reynaert, Véronique Hoste, and Ineke Schuurman. 2013. *The Construction of a 500-Million-Word Reference Corpus of Contemporary Written Dutch*, pages 219–247. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. [A primer in BERTology: What we know about how BERT works](#). *Transactions of the Association for Computational Linguistics*, 8:842–866.
- Aarne Talman and Stergios Chatzikiyriakidis. 2019. [Testing the generalization power of neural network models across NLI benchmarks](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 85–94, Florence, Italy. Association for Computational Linguistics.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- Gertjan van Noord, Gosse Bouma, Frank Van Eynde, Daniël de Kok, Jelmer van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste. 2013. *Large Scale Syntactic Annotation of Written Dutch: Lassy*, pages 147–164. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Gijs Wijnholds and Michael Moortgat. 2021. [SICK-NL: A dataset for Dutch natural language inference](#). In *Proceedings of the 16th Conference of the European*

Chapter of the Association for Computational Linguistics: Main Volume, pages 1474–1479, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. [Can neural networks understand monotonicity reasoning?](#) In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40, Florence, Italy. Association for Computational Linguistics.

Noisy Parallel Data Alignment

Ruoyu Xie, Antonios Anastasopoulos

Department of Computer Science, George Mason University

{rxie, antonis}@gmu.edu

Abstract

An ongoing challenge in current natural language processing is how its major advancements tend to disproportionately favor resource-rich languages, leaving a significant number of under-resourced languages behind. Due to the lack of resources required to train and evaluate models, most modern language technologies are either nonexistent or unreliable to process endangered, local, and non-standardized languages. Optical character recognition (OCR) is often used to convert endangered language documents into machine-readable data. However, such OCR output is typically noisy, and most word alignment models are not built to work under such noisy conditions. In this work, we study the existing word-level alignment models under noisy settings and aim to make them more robust to noisy data. Our noise simulation and structural biasing method, tested on multiple language pairs, manages to reduce the alignment error rate on a state-of-the-art neural-based alignment model up to 59.6%.¹

1 Introduction

Modern optical character recognition (OCR) software achieves good performance on documents in high-resource standardized languages, producing machine-readable text which can be used for many downstream natural language processing (NLP) tasks and various applications (Ignat et al., 2022; Van Strien et al., 2020; Amrhein and Clematide, 2018). However, attaining the same level of quality for texts in less-resourced local and non-standardized languages remains an open problem (Rijhwani et al., 2020).

The promise of OCR is particularly appealing for endangered languages, for which material might exist in non-machine-readable formats, such as physical books or educational materials. Digitizing such

to protinò ppetì tōxi ' aforammèna ? ' "

[Clean text]

to protino ppeti toxi ' aforammiena ? ' "

[OCR'd text]

to protino ppeti toxi ` aforammèna ? ' "

[Synthetic text]

Figure 1: Synthetic data example in Griko with character differences highlighted. Our synthetic data manage to mimic the real OCR noise.

material could lead to the creation of NLP technologies for such otherwise severely under-served communities (Bustamante et al., 2020).

Beyond the primary goal of digitizing printed material in endangered languages, the need for robust alignment tools is wider. The majority of the world’s languages are being traditionally oral (Bird, 2020), which implies that to obtain textual data at scale one would need to rely on automatic speech recognition (ASR), which in turn would produce invariably noisy outputs. It is worth noting that the availability of translations can significantly improve systems beyond machine translation (MT), such as OCR (Rijhwani et al., 2020) or ASR (Anastasopoulos and Chiang, 2018). This creates a chicken-and-egg situation: on one hand, OCR and ASR can be used to obtain noisy parallel data; on the other hand, having good quality aligned data can improve OCR or ASR.

In this vein, We focus on the scenario of digitizing texts in a less-resourced language along with their translations (usually high-resource and/or widely spoken) similar to Rijhwani et al. (2020). Digitizing parallel documents can also be beneficial for educational purposes, as one could then create dictionaries through word- and phrase-level alignments, or ground language learning on another language (a learner’s either L1 or L2). Also, as Ig-

¹Data and code are available online: https://github.com/ruoyuxie/noisy_parallel_data_alignment

nat et al. (2022) showed in recent work, such parallel corpora can be meaningfully used to create MT systems. However, the process that transforms digitized books or dictionaries into parallel sentences for training MT systems requires painstaking manual intervention.

In theory, the process could be semi-automated using sentence alignment methods, but in practice, the situation is very different: OCR systems tend to generate very noisy text for endangered languages (Alpert-Abrams, 2016, *inter alia*), which in turn leads to poor alignments between two parallel sides. As we show, alignment tools are particularly brittle in the presence of noise.

In this work, we take the first step towards solving the above issue. We investigate the relationship between OCR noise and alignment results and build a probabilistic model to simulate OCR errors and create realistic OCR-like synthetic data. We also manually annotate a total of 4,101 gold alignments for an endangered language pair, Griko-Italian, in order to evaluate our methods in a real-world setting. We leverage structural knowledge and augmented data, greatly reducing the alignment error rate (AER) for all four high- and low-resource language pairs up to 59.6%.

2 Problem Setting

Our work is a straightforward extension of previous word-level alignment work. Given a sequence of words $\mathbf{x} = (x_1, \dots, x_n)$ in a source language and $\mathbf{y} = (y_1, \dots, y_m)$ in a target language, the alignment model produces alignment pairs:²

$$\mathcal{A} = \{(x_i, y_j) : x_i \in \mathbf{x}, y_j \in \mathbf{y}\}$$

The difference with previous work is that the starting data will be the output of an OCR pipeline, hence producing noisy parallel data ($\mathbf{x}^*, \mathbf{y}^*$) instead of “clean” (\mathbf{x}, \mathbf{y}) ones. The level of noise may vary between the two sides.

Hence, our goal is to produce an alignment

$$\mathcal{A}^* = \{(x_i, y_j) : x_i \in \mathbf{x}^*, y_j \in \mathbf{y}^*\}$$

that will be as close to the alignment \mathcal{A} that we would have obtained without the presence of noise. We measure model performance using the alignment error rate (AER; Och and Ney, 2003) against the gold alignments.³

²Sometimes denoted with a latent variable, but we use an equivalent notation for simplicity.

³Lower AER means a better alignment. More details on the metric in Appendix A.

3 Method

We create synthetic data that mimic OCR-like noise, that can be used to train/finetune alignment models. Our *simple yet effective* method mainly consists of (i) building probabilistic models based on edit distance measures and capturing real OCR errors; (ii) creating synthetic (noisy) OCR-like data by applying our error-introducing model on clean parallel data; (iii) training or finetuning alignment models on synthetic data.

3.1 OCR Error Modeling

Error types For OCRed text, different types of texts, languages, and corpora will lead to different error distributions. At the character level, there are generally three types of OCR errors: insertions, deletions, and substitutions. In most cases, deletions and substitutions are more common, with spurious insertions being rarer.

Noise model By comparing the OCRed text with its post-corrected version, we use Levenshtein distance to compute the edit distances and the probability distributions of edits/errors over the corpus with a straightforward count-based approach.

We treat deletion error as part of substitution error. Given a sequence of characters x_i, \dots, x_j from a clean corpus \mathbf{x} and a sequence of characters y_i, \dots, y_j from its OCRed noisy version \mathbf{y} , we simply count the number of times a correct character x_i is recognized as character y_i (or recognized as the empty character ϵ if it is erroneously deleted). We can then compute the probability of an erroneous substitution or deletion as follows:

$$P_{sub}(x_i \rightarrow y_i) = \frac{\text{count}(x_i \rightarrow y_i)}{\text{count}(x_i)}$$

and the overall substitution error distribution is conditioned on the correct character x_i :

$$\mathcal{D}_{sub}(x_i) \sim P_{sub}(x_i \rightarrow y_i).$$

For insertion errors, we consider that insertion occurs when ϵ becomes another character y_i and count the number of times that insertion occurs after its previous character. A special token `<begin>` is used when insertion occurs at the beginning of the sentence. In general, we calculate the insertion error probability with:

$$P_{ins}(x_{i-1}\epsilon \rightarrow x_{i-1}y_i) = \frac{\text{count}(x_{i-1}\epsilon \rightarrow x_{i-1}y_i)}{\text{count}(x_{i-1}\epsilon)}$$

and the insertion error distribution for x_{i-1} :

$$\mathcal{D}_{ins}(x_{i-1}\epsilon) \sim P_{ins}(x_{i-1}\epsilon \rightarrow x_{i-1}y_i).$$

3.2 Data Augmentation

Synthetically noised data can be created by leveraging the calculated probability distributions from the previous section and traversing through the clean corpus for every character.

For each character c , we obtain its probability to be erroneous in the OCR output by sampling from the distribution of the substitution and insertion probabilities $\mathcal{D}_{ins}(c), \mathcal{D}_{sub}(c)$.⁴ We randomly decide whether to add an error here based on its error distribution.

If an error will be introduced on c , we then randomly choose its corresponding error based on $P_{sub}(c)$ or $P_{ins}(c)$ depending on either substitution or insertion operation receptively.

Our method attempts to mimic the real OCR errors in given languages and corpus, resulting in very similar noise distributions. Figure 1 shows a side-by-side comparison of three versions of the same sentence, to showcase how realistic our synthetic text is.

3.3 Model Improvement

Given our synthetically noised parallel data, and potentially along with the original clean parallel data, we can now train or finetune a word alignment model to improve the model performance.

In the case of unsupervised models like the IBM translation models (Brown et al., 1993), fast-align (Dyer et al., 2013), or Giza++ (Och, 2003), we simply train on the concatenation of all available data.

We also work with the state-of-the-art neural alignment model of Dou and Neubig (2021), which is based on Multilingual BERT (mBERT) (Devlin et al., 2019).⁵ For this model, we distinguish two cases: supervised and unsupervised finetuning.⁶ Under a supervised setting, we first obtain silver alignments from the clean dataset and use them as targets for the synthetic noisy data. The unsupervised setting is conceptually similar to training models like Giza++: we feed synthetically-noised sentence pairs into the alignment model, without

⁴Including a third option for not inserting an error

⁵See Section 5.1 for more details.

⁶We use provided default parameters for both cases, which can be found on <https://github.com/neulab/awesome-align>

Language	Total CER	Sub. %	Ins. %
English	7.4	79	21
German	4.9	87	13
French	4.8	85.7	14.3
Griko	3.3	96.8	3.2
Ainu	1.4	91.9	8.1

Table 1: Total character error rate (CER) and percentage of substitution and insertion errors. Generally, substitution is the most common error in OCR output.

using the target alignment as supervision. In low-resource scenarios, we leverage a diagonal bias to further improve the model’s performance.

4 Languages and Datasets

We study four language pairs with varying amounts of data availability: English-French, English-German, Griko-Italian, and Ainu-Japanese.⁷

4.1 Dataset for Error Extraction

The ICDAR 2019 Competition on Post-OCR Text Correction (Rigaud et al., 2019) dataset provides both clean and OCRed text for English-French and English-German, which we use our noisy model to learn and mimic OCR errors for English, French, and German.

For Griko-Italian and Ainu-Japanese, Rijhwani et al. (2020) provide around 800 OCRed noisy and clean (post-corrected) sentences for both Griko and Ainu, from which we extract error distributions; for Italian and Japanese, only OCRed text is provided.⁸

To understand the characteristics of our datasets, we report the observed CER in Table 1. Generally, substitutions are the most common errors. Notice that Griko and Ainu have seemingly lower scores than any high-resource languages; that’s because both use the Latin alphabet, the data that were digitized are typed in books with high-quality scans.⁹

4.2 Synthetic Data

We create synthetic data by applying captured OCR noise on clean text. For English, French, and German, the clean text comes from Europarl v8 corpus (Koehn, 2005). For Ainu, there are 816 clean sentences from Rijhwani et al. (2020), from which

⁷Griko and Ainu are both under-resourced endangered languages.

⁸The quality of the OCR model on these high-resource languages are generally reliable.

⁹The English, French, and German data from ICDAR have lower-quality scans.

Language	Real CER	Syn. CER	Diff.
English	7.4	6.5	0.9
German	4.9	6.7	1.8
French	4.8	5.3	0.5
Griko	3.3	3.3	0
Ainu	1.4	1.0	0.4

Table 2: Our synthetically-noisy data have similar CER compared to the real OCR outputs, which implies that the real OCR noisy data can be mimicked by our noise simulation model.

we keep the first 300 lines as test set and use the rest to create synthetic data. Anastasopoulos et al. (2018) provide 10,009 clean sentences for Griko. Table 2 shows the CER comparison between our synthetic data and real OCR data.

4.3 Test Set and Gold Alignment

The test set and gold alignment for English-French come from Mihalcea and Pedersen (2003). For English-German, the test set and gold alignments come from Europarl v7 corpus (Koehn, 2005) and Vilar et al. (2006), respectively. To study the effect of OCR-like errors on alignment, we create synthetically-noised test sets for both languages pairs by applying noise on one side or both, which results in four copies of the same test set: clean-clean, clean-noisy, noisy-clean, and noisy-noisy.

For low-resource language pairs, Rijhwani et al. (2020) provide about 800 parallel sentence pairs for each. We use the first 300 sentence pairs as our test sets. For the purpose of fair evaluation in our method, we annotate a total of 4,101 *gold* word-level alignment pairs for Griko-Italian test set. On the other hand, we obtain *silver* alignments from awesome-align for Ainu-Japanese as there is no existing gold alignment data available.¹⁰

5 Experiments

In this section, we present multiple experiments and demonstrate that our method results in significant AER reductions.

5.1 Experimental Setup

Models We study the following models:

- IBM model 1&2 (Brown et al., 1993): the classic statistical word alignment models. They underpinned many other statistical machine translation and word alignment models.

¹⁰While not ideal, we can still measure how different results are when comparing alignments on clean versus noisy data.

Model	Clean	OCRed	Diff.
IBM 1	43.7	49.2	5.5
IBM 2	37.3	43.4	6.1
Giza++	14.5	20.8	6.3
fast-align	19.8	25.7	5.9
awesome-align	45.1	48.8	3.7

Table 3: AER comparison for Griko-Italian. Giza++ performs best on both settings, but it exhibits the largest drop in performance.

- Giza++ (Och, 2003): a popular statistical alignment model that is based on a pipeline of IBM and Hidden Markov models (Vogel et al., 1996).
- fast-align (Dyer et al., 2013): a simple but effective statistical word alignment model that is based on IBM Model 2, with an additional bias towards monotone alignment.
- awesome-align (Dou and Neubig, 2021): a neural word alignment model based on mBERT. It fine-tunes a pre-trained multilingual language model with parallel text and extracts the alignments from the resulting representations.

5.2 The Effect of OCR-like Noise

We use Griko-Italian as our main evaluation pair due to the presence of its gold alignments, which can most accurately reflect the model’s performance under a low-resource scenario.

We first benchmark model performance on clean and OCRed parallel text to quantify OCR-error effects on alignment (Table 3). We compute AER for the clean and OCRed versions of Griko-Italian by comparing their alignment against our manually created gold alignment. We benchmark five different models that lead to several observations. First, note that clean text always results in a better alignment for all models. Overall, Giza++ performs best among the models, but note that it also suffers the largest drop in performance when faced with noisy text. On the other hand, a vanilla awesome-align, which is otherwise a state-of-the-art model for languages that were included in the pre-training of its underlying model, performs the worst, not being better than a simple IBM 1.

We can thus conclude that OCR error does impact alignment quality for both statistical and neural based alignment models.

It is of note that for Griko-Italian every statistical model outperforms awesome-align in almost all cases. We hypothesize that this is due to the lack

	Griko-Italian		Ainu-Japanese	
	Clean	OCRed	Clean	OCRed
BASE	45.1	48.8	28.2	29.2
UNSUP-FT (A)	23.2±1.6	28±1.1	21.1 ±1	22.2±1.2
SUP-FT (B)	22.3±2	26.6±1.1	30.9±2.1	31.5±1.2
+ structural bias				
UNSUP-FT (A)	18.7±1	24.2±0.6	15.3±3.9	13.8±4.2
SUP-FT (B)	18.2±2.6	22.9±2.4	26.3 ±2.1	27.4±1.8
AER reduction	59.6%	53.1%	45.7%	52.7%

Table 4: For both endangered languages, our approach greatly reduces AER for both clean and OCRed data.

of structural knowledge; we deal with this in Section 5.3.1. awesome-align’s low performance can also be explained by the fact that Griko is not well supported by its underlying representation model: Griko was not part of the pre-training language mix, and it does not use the same script as its closest language that was included in pre-training (Greek),¹¹ an important factor according to Muller et al. (2021). Compared to statistical models, we also observe considerably fewer alignment pairs are produced by awesome-align (Appendix 8), which might also be a contributing factor.

5.3 Making awesome-align Robust

The performance of awesome-align raises an intriguing question - Is the state-of-the-art neural based model capable to align noisy text, especially from low-resource languages. Given its general higher performance on many popular languages (Dou and Neubig, 2021) and the stability between clean and noisy text,¹² we make awesome-align as our main experiment target.

5.3.1 Low-Resource Setting

We introduce structural bias and propose two models: model (A) and model (B) finetuned in unsupervised and supervised settings respectively.

Structural Bias Structural alignment biases are widely used in statistical alignment models such as Brown et al. (1993); Vogel et al. (1996); Och (2003); Dyer et al. (2013). However, it is a missing component in awesome-align. Following by Dyer et al. (2013), we introduce diagonal bias and apply it on the top of awesome-align’s attention layer. We create (i) a bias matrix M_b based on

¹¹Modern Greek uses the Greek alphabet, while Griko uses the Latin alphabet.

¹²Lowest AER difference between clean and noisy text amount to all models.

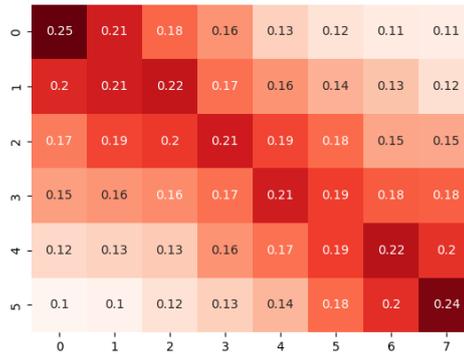


Figure 2: A sample 6×8 diagonal bias matrix. Darker color means stronger bias emphasis. We follow the same steps from Dyer et al. (2013) to calculate each position based on given rows and columns.

the position of the alignment, where the positions near the diagonal of the alignment matrix have the higher weights (See Figure 2); (ii) a tune-able hyper-parameter λ represents the weight of the bias. We set $\lambda=1$ for all low-resource language experiments; (iii) an average matrix M_{avg} that is the average of the original attention score, which is used for smoothing λ to make it where 1 represents maximum bias and 0 means no bias at all. We update the original awesome-align attention score A_{sc} :

$$A_{sc} = \lambda * A_{sc} + (1 - \lambda) * A_{sc} * M_b * M_{avg}$$

Our proposed models For our unsupervised-finetuned model (A), we create the synthetically-noised data by introducing OCR-like noise on clean parallel data, and then simply finetune the baseline model with all available data from both clean and synthetic text.

For the supervised-finetuned model (B), we first finetune an out-of-the-box awesome-align with the clean data from Anastasopoulos et al. (2018) and Rijhwani et al. (2020) for Griko-Italian and

Test Set	English-French				English-German		
	Baseline	Unsup-FT	Sup-FT	Reduction	Baseline	Unsup-FT	Reduction
CLEAN-CLEAN	5.6	4.6	15.9	17.0%	17.9	15.2	15.1%
CLEAN-SYNTH	40.5	36.3	29.4	27.4%	43.8	39.4	10.0%
NOISY-CLEAN	39.2	34.2	28.6	27.0%	52.8	50	5.3%
NOISY-NOISY	53.6	46.1	37.3	30.4%	66.6	63.5	4.7%

Table 5: Result of awesome-align on English to French and German alignment. Both unsupervised and supervised finetuning with noise-induced data leads to big AER reduction when aligning noisy data. Reductions in German are less pronounced. Unsupervised finetuning with noisy data also improves clean-data alignment.

	IBM 1		IBM 2		Giza++		fast-align	
	Clean	OCRed	Clean	OCRed	Clean	OCRed	Clean	OCRed
Baseline	43.7	49.2	37.3	43.4	14.5	20.8	19.8	25.7
Train w. clean	40.2	45.7	32.7	38.1	13.1	19.0	17.9	24.2
Train w. noise	84.2	84.6	80.0	80.8	19.7	25.8	22.8	27.9

Table 6: Experiment on Griko-Italian, every statistical model benefits from training with additional clean data but suffers significant performance drops with synthetic noisy data, suggesting that traditional statistical models rely on clean text.

Aiun-Japanese respectively, which produces silver alignment. Next, we use the silver alignment as supervision to finetune awesome-align with synthetic noisy data.

We report the average plus-minus standard deviation of three runs for each model. Table 4 summarizes the results for our proposed models. We end up with around **50%** AER reduction for both endangered language pairs.

5.3.2 High-Resource Setting

We evaluate our data augmentation method on high-resource language pairs. Up to 400K synthetically noised English-French data was used for unsupervised finetuning. We also offer an additional reference data point, using 100K synthetic noised English-German data for unsupervised fine-tuning.

For supervised finetuning, we use up to 1M synthetic data. As before, we use silver alignments from clean data as supervision to finetune its synthetic noisy version, which does not require any additional human annotation effort.

Under both settings, model performance will plateau when adding more data. The results are summarized in Table 5. Both unsupervised and supervised finetunings with synthetically-noised data significantly improve alignment quality, especially for noisy test sets, in line with our previously presented results in low-resource settings.

5.4 Additional Data on Statistical Models

We conduct additional experiments to find out whether training with additional data aids statistical models for endangered languages. We evaluate model performance on Griko-Italian.

We concatenate additional data to the examples comprising the test set. We first train the models with all 800 clean sentence pairs taken from [Rijhwani et al. \(2020\)](#) (which include the 300 sentences of the test set). Next, instead of using clean data, we substitute it with synthetically noised data and train the models.

The result is presented in Table 6. For every statistical model, training with additional clean text reduces AER. However, training with additional noisy text considerably hurts the models. The result shows that these statistical models rely on *clean text* to improve, which is almost always *unavailable* for endangered languages. This also implies that investing time in manually cleaning OCR data could be effective for these models; however, it is not always possible and contradicts the goal of reducing the human effort in this work.

6 Analysis and Discussion

In this section we conduct several analyses to better understand our method.

Incorporating Diagonal Bias As shown in Table 4, our diagonal bias markedly improves ev-

Test set	En-Fr	En-De
CLEAN-CLEAN	5.6	17.9
CLEAN-NOISY	40.5	43.8
NOISY-CLEAN	39.2	52.8
NOISY-NOISY	53.6	66.6

Table 7: awesome-align baseline on En-Fr and En-De. OCR-like noise dramatically degrades the performance.

ery test case for both endangered language pairs. Note that the attention score will be increased significantly by adding bias, which will still be a valid input for the final alignment matrix due to its alignment extraction mechanism (Dou and Neubig, 2021). In this work, we only apply diagonal bias under low-resource settings since it was shown in Dou and Neubig (2021) that growing heuristics such as grow-diag-final (Koehn et al., 2005; Och and Ney, 2000) do not achieve promising results for multiple high-resource language test sets.

Degradation of Alignment Table 7 presents the evaluation of four test sets for awesome-align in English-French and English-German. We observe a significant decline in performance when OCR-like noise is introduced. For example, with clean parallel text, the AER for English-French is 5.6%, but when OCR-like noise is added, the AER jumps to 53.6%, almost a tenfold increase.

Size of synthetic data We conduct quantitative analyses as shown in Figure 3 to examine awesome-align with different sizes of English-French synthetic data under both unsupervised and supervised settings. For space economy reasons, here we only discuss the results of the more challenging noisy-noisy test set. Note that dramatic degradation of alignment is observed when applying OCR-like noise to clean text (see Table 7). In general, the model produces better alignment as more data are used. However, there is also a trade-off on the clean-clean test set as its performance worsens in the supervised scenario. Keep in mind, though, that this situation is only observed in high-resource language pairs; for a low-resource language pair like our Griko-Italian, in limited ablation experiments we found that we have not reached the data saturation point yet as more data simply resulted in better performance for both clean and noisy text.

Varying degrees of CER In a real-world scenario, the CER of OCRed data is typically un-

known due to the absence of clean text. We investigate how different degrees of CER affect alignments by creating several English-French synthetic data with varying degrees of CER, testing them on awesome-align. We elaborate on the process and results in Appendix B.1. The main finding is that higher CER leads to greater AER, which is expected. However, we also find that mixing with different degrees of CER generally produces better results than a fixed CER throughout the corpus, suggesting that our augmentation approach could also work on the unknown CER real-world scenario.

Statistical Model vs Neural Model The question of which model to use in practical scenarios, though, remains tricky to answer. Due to similarities between Griko and Italian and prolonged language contact over centuries, the two languages follow very similar syntax; as a result, their alignment is largely monotone, which benefits models like Giza++ and fast-align. They outperform, in fact, the vanilla neural awesome-align model by a large margin (see Table 3). However, this will not always be the case. For example, most books with parallel data in the Archive of Indigenous Languages of Latin America (AILLA) mostly contain data between indigenous languages and one of Spanish or English. Now the two sides of the data come from different language families and a monotone alignment is not necessarily to be expected. In such cases, it could indeed be the case that a more adaptable neural model like awesome-align, aided by our data augmentation and diagonal biasing methods, could indeed be the best option.

Different side of OCR noise An important insight derived from Table 5 is that the performance of awesome-align deteriorates significantly more when both sides of the parallel data are noisy, as compared to when only one side is noisy. This is in fact encouraging for our envisioned application scenarios, since, as in the AILLA examples described above, we expect that OCRed parallel data in endangered languages will come with one side in a high-resource standardized language like English and Spanish which in turn we expect the OCR model to be able to adequately handle.¹³

¹³Rijhwani et al. (2020) and Rijhwani et al. (2021) make similar observations on all endangered language datasets they work with.

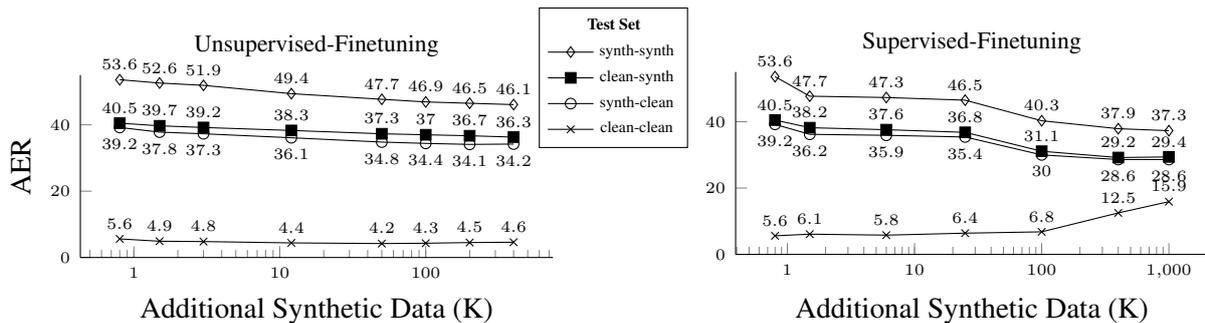


Figure 3: Ablation on English-French with varying degrees of additional synthetically noised data. Notice the log scale on the x-axis. The left-most point corresponds to no additional synthetic data (baseline). More data reduce AER for noisy test sets, especially in the supervised finetuning setting.

7 Related Work

Our work is a natural extension of previous word alignment work. A robust alignment tool for low-resource languages benefits MT systems (Xiang et al., 2010a; Levinboim and Chiang, 2015; Beloucif et al., 2016a; Nagata et al., 2020), or speech recognition (Anastasopoulos and Chiang, 2018), especially if sentence-level alignment tools like LASER (Artetxe and Schwenk, 2019; Chaudhary et al., 2019) do not cover all languages, so one may need to fall-back to word-level alignment heuristics to inform sentence-alignment models like Hunalign (Varga et al., 2007).

Research on word-level alignment started with statistical models, with the IBM Translation Models (Brown et al., 1993) serving as the foundation for many popular statistical word aligners (Och and Ney, 2000, 2003; Och, 2003; Tiedemann et al., 2016; Vogel et al., 1996; Och, 2003; Gao and Vogel, 2008; Dyer et al., 2013). In recent years, different neural network based alignment models gained in popularity including end-to-end based (Zenkel et al., 2020; Wu et al., 2022; Chen et al., 2021), MT-based (Chen et al., 2020), and pre-training based (Garg et al., 2019; Dou and Neubig, 2021). As awesome-align achieves the overall highest performance, we choose to focus on awesome-align in this work.

Some works involve improving word-level alignment for low-resource languages such as utilizing semantic information (Beloucif et al., 2016b; Pourdamghani et al., 2018), multi-task learning (Levinboim and Chiang, 2015), and combining complementary word alignments (Xiang et al., 2010b). None of the previous work, though, to our knowledge, tackles the problem of aligning data with

OCR-like noise on one or both sides. The idea of augmenting training data is not new and has been applied in many areas and applications. Marton et al. (2009) augment data with paraphrases taken from other languages to improve low-resource language alignments. While potentially orthogonal to our approach, this idea is largely inapplicable to our endangered language settings, as we often have to work with the only available datasets for these particular languages. Applying structure alignment bias on statistical and neural models is also a well-studied area (Cohn et al., 2016; Brown et al., 1993; Vogel et al., 1996; Och, 2003; Dyer et al., 2013). However, to the best of our knowledge, we are the first to apply it to low-resource languages, proving that such an approach can greatly aid the real endangered language data.

8 Conclusion

In this work, we benchmark several popular word alignment models under OCR noisy settings with high- and low-resource language pairs, conducting several studies to investigate the relationship between OCR noise and alignment quality. We propose a simple yet effective approach to create realistic OCR-like synthetic data and make the state-of-the-art neural awesome-align model more robust by leveraging structural bias. Our work paves the way for future word-level alignment-related research on underrepresented languages. As part of this paper, we also release a total of 4,101 ground truth word alignment data for Griko-Italian, which can be a useful resource to investigate word- and sentence-level alignment techniques on practical endangered language scenarios.

9 Limitations

Using AER as the main evaluation metric could be a limitation of our work as it might be misleading in some cases (Fraser and Marcu, 2007). Another limitation, of course, is that we only manage to explore the tip of the iceberg given the sheer number of endangered languages. While we are confident in the results of both low-resource language pairs, our experiments on Ainu-Japanese could potentially lead to inaccurate AER since we use the automatically generated silver alignment. In the future, we hope to eventually annotate it with either the help of native speakers or dictionaries. We also plan to explore other alternative metrics and expand our alignment benchmark on as many endangered languages as possible.

Acknowledgements

We are thankful to Shruti Rijhwani and Graham Neubig, as well as the anonymous reviewers, for their valuable comments on the early stages of this work. We would also like to thank Sina Ahmadi for his useful feedback and the GMU Office of Research Computing for the computing resources. This work was supported by NEH Award PR-276810-21 as well as through a GMU OSCAR award for undergraduate research.

References

Hannah Alpert-Abrams. 2016. Machine reading the primeros libros. *Digital Humanities Quarterly*, 10(4).

Chantal Amrhein and Simon Clematide. 2018. Supervised ocr error detection and correction using statistical and neural machine translation methods. *Journal for Language Technology and Computational Linguistics (JLCL)*, 33(1):49–76.

Antonios Anastopoulos and David Chiang. 2018. Leveraging translations for speech transcription in low-resource settings. *Proc. Interspeech 2018*, pages 1279–1283.

Antonios Anastopoulos, Marika Lekakou, Josep Quer, Eleni Zimianiti, Justin DeBenedetto, and David Chiang. 2018. Part-of-speech tagging on an endangered language: a parallel griko-italian resource. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2529–2539.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Meriem Beloucif, Markus Saers, and Dekai Wu. 2016a. Improving word alignment for low resource languages using English monolingual SRL. In *Proceedings of the Sixth Workshop on Hybrid Approaches to Translation (HyTra6)*, pages 51–60, Osaka, Japan. The COLING 2016 Organizing Committee.

Meriem Beloucif, Markus Saers, and Dekai Wu. 2016b. Improving word alignment for low resource languages using english monolingual srl. In *Proceedings of the Sixth Workshop on Hybrid Approaches to Translation (HyTra6)*, pages 51–60.

Steven Bird. 2020. Sparse transcription. *Computational Linguistics*, 46(4):713–744.

Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Gina Bustamante, Arturo Oncevay, and Roberto Zariquiey. 2020. No data to crawl? monolingual corpus creation from pdf files of truly low-resource languages in peru. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2914–2923.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. *WMT 2019*, page 263.

Chi Chen, Maosong Sun, and Yang Liu. 2021. Mask-align: Self-supervised neural word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4781–4791, Online. Association for Computational Linguistics.

Yun Chen, Yang Liu, Guanhua Chen, Xin Jiang, and Qun Liu. 2020. Accurate word alignment induction from neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 566–576, Online. Association for Computational Linguistics.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the*

- North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Zi-Yi Dou and Graham Neubig. 2021. [Word alignment by fine-tuning embeddings on parallel corpora](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.
- Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.
- Alexander Fraser and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303.
- Qin Gao and Stephan Vogel. 2008. [Parallel implementations of word alignment tool](#). In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio. Association for Computational Linguistics.
- Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. [Jointly learning to align and translate with transformer models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.
- Oana Ignat, Jean Maillard, Vishrav Chaudhary, and Francisco Guzmán. 2022. OCR Improves Machine Translation for Low-Resource Languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1164–1174.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86.
- Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. [Edinburgh system description for the 2005 IWSLT speech translation evaluation](#). In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.
- Tomer Levinboim and David Chiang. 2015. [Multi-task word alignment triangulation for low-resource languages](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1221–1226, Denver, Colorado. Association for Computational Linguistics.
- Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. [Improved statistical machine translation using monolingually-derived paraphrases](#). In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 381–390, Singapore. Association for Computational Linguistics.
- Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond*, pages 1–10.
- Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When Being Unseen from mBERT is just the Beginning: Handling New Languages with Multilingual Language Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462.
- Masaaki Nagata, Katsuki Chousa, and Masaaki Nishino. 2020. [A supervised word alignment method based on cross-language span prediction using multilingual BERT](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 555–565, Online. Association for Computational Linguistics.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics*, pages 160–167.
- Franz Josef Och and Hermann Ney. 2000. [Improved statistical alignment models](#). In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong. Association for Computational Linguistics.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.
- Nima Pourdamghani, Marjan Ghazvininejad, and Kevin Knight. 2018. [Using word vectors to improve word alignments for low resource machine translation](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 524–528, New Orleans, Louisiana. Association for Computational Linguistics.
- Christophe Rigaud, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2019. ICDAR 2019 competition on post-OCR text correction. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 1588–1593. IEEE.
- Shruti Rijhwani, Antonios Anastasopoulos, and Graham Neubig. 2020. [OCR Post Correction for Endangered Language Texts](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

- Processing (EMNLP)*, pages 5931–5942, Online. Association for Computational Linguistics.
- Shruti Rijhwani, Daisy Rosenblum, Antonios Anastasopoulos, and Graham Neubig. 2021. Lexically aware semi-supervised learning for ocr post-correction. *Transactions of the Association for Computational Linguistics*, 9:1285–1302.
- Jörg Tiedemann, Fabienne Cap, Jenna Kanerva, Filip Ginter, Sara Stymne, Robert Östling, and Marion Weller-Di Marco. 2016. [Phrase-based SMT for Finnish with more data, better models and alternative alignment and translation tools](#). In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 391–398, Berlin, Germany. Association for Computational Linguistics.
- Daniel Van Strien, Kaspar Beelen, Mariona Coll Ardanuy, Kasra Hosseini, Barbara McGillivray, and Giovanni Colavizza. 2020. Assessing the impact of OCR quality on downstream NLP tasks. *SCITEPRESS-Science and Technology Publications*.
- Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292:247.
- David Vilar, Maja Popović, and Hermann Ney. 2006. AER: Do we need to “improve” our alignments? In *Proceedings of the Third International Workshop on Spoken Language Translation: Papers*.
- Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.
- Di Wu, Liang Ding, Shuo Yang, and Mingyang Li. 2022. Mirroralign: A super lightweight unsupervised word alignment model via cross-lingual contrastive learning. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 83–91.
- Bing Xiang, Yonggang Deng, and Bowen Zhou. 2010a. Diversify and combine: Improving word alignment for machine translation on low-resource languages. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 22–26.
- Bing Xiang, Yonggang Deng, and Bowen Zhou. 2010b. [Diversify and combine: Improving word alignment for machine translation on low-resource languages](#). In *Proceedings of the ACL 2010 Conference Short Papers*, pages 22–26, Uppsala, Sweden. Association for Computational Linguistics.
- Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. [End-to-end neural word alignment outperforms GIZA++](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

A Evaluation metric

We calculate precision, recall and alignment error rate as described in [Och and Ney \(2003\)](#), where A is a set of alignments to compare, S is a set of gold alignments, and P is the union of A and possible alignments in S . We then compute AER with:

$$\text{Precision} = \frac{|A \cap P|}{|A|} \quad \text{Recall} = \frac{|A \cap S|}{|S|}$$
$$\text{AER}(S, P; A) = 1 - \frac{|A \cap S + A \cap P|}{|A + S|}$$

B Additional Analyses

B.1 Varying degrees of CER

We create eight 100k English-French synthetic datasets with different CER for each: “unified” datasets with exactly the same CER on both sides: 2, 5, 10, and one with mixed CER with equally shared portions with 2, 5 and 10 CER; and “varying” datasets with slightly different CER between the English and French side, but in the same range as the others. We then finetune the model with these synthetic training datasets and compare against the “skyline” result presented before, where the augmentation matched the level of the true CER.

The results are presented in [Table 9](#) for both the unsupervised- and the supervised-finetuning setting. Encouragingly, despite different CER in the augmentation data, there are no significant performance differences in most cases, especially for the unsupervised setting. Of course, levels of noise that match the true level tend to perform better or close to best overall. On the other hand, high levels of noise that lead to very high word error rate (WER)¹⁴ cause a large degradation in the performance of the supervised finetuning approach, but do not seem to significantly affect the unsupervised approach.

Even more encouragingly, an augmented dataset that uses a mixture of different target CER (such as having a third of the dataset having a CER around 2, a third with CER around 5, and a third around 10 – named “mixed” in [Table 9](#)) in the supervised setting further outperforms the *informed skyline* which uses additional knowledge that might not be available (the true CER of the data to be aligned). For instance, in the clean-noisy test set this model reduces AER by a further 5% (from 31.1 to 29.3)

and on the clean-clean test set it reduces AER by 19% (from 6.8 to 5.5). This means that our augmentation approach with varying levels of noise could be applied to any scenario, even if one does not know the level of noise present in the data-to-be-aligned.

¹⁴For example, a CER of around 10 translates to a WER of more than 70, meaning that (approximately) only 3 out of 10 words are correct.

	IBM 1		IBM 2		Giza++		fast-align		awesome	
	Clean OCR		Clean OCR		Clean OCR		Clean OCR		Clean OCR	
# of pairs	3844	3839	3833	3855	3810	3813	3801	3794	2978	2969
Precision	58.2	52.5	64.9	58.6	88.7	82.2	83.4	77.3	65.3	64
Recall	54.5	49.2	60.7	54.8	82.4	76.4	77.3	71.5	47.4	44.2

Table 8: Comparing the number of alignment pairs produced by models on Griko-Italian. awesome-align produces almost 25% less alignment pairs, resulting in markedly lower precision/recall and higher AER.

CER (WER) on Synthetic Data	Clean-Clean		Clean-Noisy		Noisy-Clean		Noisy-Noisy		
	UNSUP-FT	SUP-FT	UNSUP-FT	SUP-FT	UNSUP-FT	SUP-FT	UNSUP-FT	SUP-FT	
Skyline: Using exactly the CER of the test set									
7.4-4.8 (59.7-51.7)	4.3	6.8	37	31.1	34.4	30	46.9	40.3	
Unified: Exactly the same CER on both sides									
2-2 (32.2-29.1)	4.1	5.1	37.5	33.3	35.1	30.1	48.4	41.8	
5-5 (55.1-52.4)	4.3	7.4	37.2	30.4	34.6	29.8	47.4	40.0	
10-10 (72.2-71)	4.5	32.8	36.8	36.2	34.5	47.7	46.8	47.3	
mixed (55-54.1)	5.4	5.5	37.3	39.6	35.3	39.8	47.7	54.2	
Varying CER between the two parallel sides									
1.6-2.1 (27.8-29.6)	4.0	6.2	37.6	31.6	35	30.6	48.4	41.9	
4.1-5.1 (49.6-52.5)	4.3	7.7	37	29.8	34.7	30.1	47.2	40.2	
8.1-9.6 (66.9-69.5)	4.5	31.3	37.1	36.2	34.3	46.3	46.7	46.9	
mixed (47.9-50)	4.2	5.6	37	29.3	34.6	29.7	47.2	40.4	

Table 9: AER comparison for varying CER in 100K English-French augmented data used for either unsupervised or supervised finetuning. We highlight the best result under each setting and test set. Overall, most models’ performance is close to the baseline, but varying amounts of noise (mixed) lead to generally the best results. Too high amounts of noise (e.g. CER around 10 with WER approaching 70) hurts the supervised approach.

Enhancing Dialogue Generation with Conversational Concept Flows

Siheng Li^{1*}, Wangjie Jiang^{1*}, Pengda Si^{1*}, Cheng Yang¹
Yao Qiu², Jinchao Zhang², Jie Zhou², Yujiu Yang^{1†}

¹Shenzhen International Graduate School, Tsinghua University

²Tencent Inc, Beijing, China

{lisiheng21, jwj20, spd18}@mails.tsinghua.edu.cn

yang.yujiu@sz.tsinghua.edu.cn

Abstract

Human conversations contain natural and reasonable topic shifts, reflected as the concept flows across utterances. Previous researches prove that explicitly modeling concept flows with a large commonsense knowledge graph effectively improves response quality. However, we argue that there exists a gap between the knowledge graph and the conversation. The knowledge graph has limited commonsense knowledge and ignores the characteristics of natural conversations. Thus, many concepts and relations in conversations are not included. To bridge this gap, we propose to enhance dialogue generation with conversational concept flows. Specifically, we extract abundant concepts and relations from natural conversations and build a new conversation-aware knowledge graph. In addition, we design a novel relation-aware graph encoder to capture the concept flows guided by the knowledge graph. Experimental results on the large-scale Reddit conversation dataset indicate that our method performs better than strong baselines, and further analysis verifies the effectiveness of each component.

1 Introduction

With the remarkable development of conversation artificial intelligence (Shang et al., 2015; Adiwardana et al., 2020; Thoppilan et al., 2022), response generation has been improved in many ways, e.g., human-like persona (Zhang et al., 2018a), empathetic expression (Rashkin et al., 2019) and knowledge injection (Dinan et al., 2019), etc. However, there still exists a series of challenges (Gao et al., 2019; Xu et al., 2020a; Huang et al., 2020). One of the most noticeable is that humans are good at naturally switching topics during conversations, while machine-generated responses are relatively dull and tend to keep the topic still (Fang et al.,

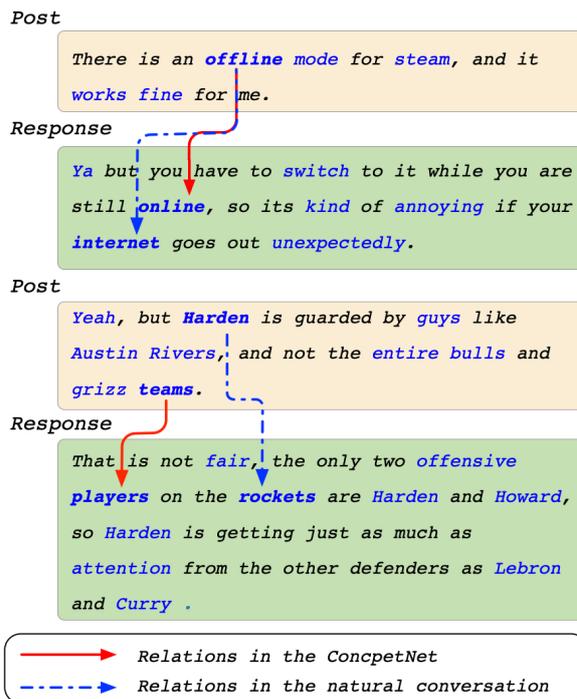


Figure 1: Two cases in the Reddit dataset. We use ConceptNet as the external knowledge graph to show concept flows in conversations. Concepts are marked in blue. Relations in the graph and those in the natural conversation are marked with red solid lines and blue dashed lines, respectively.

2018) or throw unexpected topics (Wang et al., 2018; Tang et al., 2019).

To overcome this challenge, previous works treat the topic shifts as concept flows (Zhang et al., 2020a; Zhou et al., 2018b, 2021a), which means traversing in the concept¹ space along relations in an external commonsense knowledge graph. Experimental results have shown that explicitly modelling concept flows effectively improves the relevance and engagingness of responses. However, we argue that there is a gap between the external knowledge graph and natural conversations. The most

* Equal contribution.

† Corresponding author (yang.yujiu@sz.tsinghua.edu.cn).

¹Concept is the node in knowledge graph.

frequently used ConceptNet² (Speer et al., 2017) is limited to mostly (90%) taxonomic (e.g., *IsA*) or lexical (e.g., *Synonym*) knowledge, while contains relatively small portion of commonsense knowledge (Hwang et al., 2021). In addition, concepts and relations in natural conversations are more colloquial and timely. Thus, many concepts and relations are not included in the knowledge graph, which has also been verified in our experiments. As in Figure 1, the concept flows from “offline” to “internet” and from “Harden” to “rockets” are frequently observed in human conversations, while they are both not included in the most frequently used ConceptNet.

To bridge the above gap and capture more concept flows, we propose to Enhance Dialogue Generation with Conversational Concept Flows (ECCF). Specifically, we construct an enhanced knowledge graph that consists of concepts and relations in both commonsense knowledge graph and natural conversations. First, we extract new concepts as new nodes and the high-frequency relations between concepts as new edges from a large-scale dialogue corpora. Then, we add these new nodes and new edges to the commonsense knowledge graph to construct a Conversation-Aware Knowledge Graph (CAKG). To effectively guide concept flows in conversations with CAKG, we further propose a novel Relation-Aware Graph Encoder (RAGE), which reasonably considers concepts and their relations in the graph encoding process for response generation.

We conduct a series of experiments on the large-scale Reddit conversation dataset (Zhou et al., 2018b; Baumgartner et al., 2020). Both automatic evaluation and human evaluation demonstrate that our method ECCF improves the relevance and diversity of responses, and outperforms strong baselines. Further analysis verifies the effectiveness of both CAKG and RAGE. Our research sheds light on explicitly modeling topic shifts with natural conversations.

2 Method

2.1 Overview

Given a dialogue context X , we aim to guide the topic shifts with the concepts and relations in a

²ATOMIC (Sap et al., 2019) is also frequently used, while they focus more on human emotion and reaction in the generation of empathetic responses (Sabour et al., 2021; Tu et al., 2022), which we leave for future work.

knowledge graph. Our method ECCF is shown in Figure 2, and can be summarized as follows:

1. Considering the abundant topic shifts in natural conversations, we enhance a commonsense knowledge graph G with conversational concept flows extracted from large-scale conversation data. Then we get a conversation-aware knowledge graph G_c (CAKG), which is more informative.
2. For response generation, we first encode the dialogue context X with a context encoder. Then, to capture the concept flows defined in the knowledge graph G_c , we use a graph encoder for encoding the retrieved subgraph g from G_c , which is based on the concepts in the dialogue context and their neighbor nodes. Last, we adopt a decoder with copy mechanism to generate a response and it can directly copy concepts from the subgraph g .

2.2 Knowledge Graph Enhancement with Conversational Concept Flows

We construct CAKG G_c on the basis of the commonsense knowledge graph G and a large-scale dialogue corpora Reddit (Baumgartner et al., 2020), so that G_c contains more concept flows in natural conversation. Formulating $G = \{V, E\}$ where V and E represent nodes and edges respectively, we extract new nodes V' and new edges E' from the corpora, then reconstruct $G_c = \{V \cup V', E \cup E'\}$.

To obtain conversational concepts as much as possible, we have two principles when extracting new nodes: common and concrete. First, we set a frequency threshold m and words with a frequency higher than it are regarded as candidate concepts. Second, we choose nouns as new nodes from candidate concepts because nouns have richer semantic information than other types of words³.

We utilize the GIZA++ tool to extract⁴ (Och and Ney, 2003) new edges, which represent concept flows in the conversations. The GIZA++ tool is designed to align words in the machine translation field. Its main idea is that utilize the EM algorithm to iteratively train the bilingual corpus and obtain word alignment from sentence alignment. We choose the toolkit here since concept alignments from source sentences to target sentences in

³We use the NLTK toolkit in python3 for POS tagging <https://www.nltk.org/>

⁴<http://www.statmt.org/moses/giza/GIZA++.html>

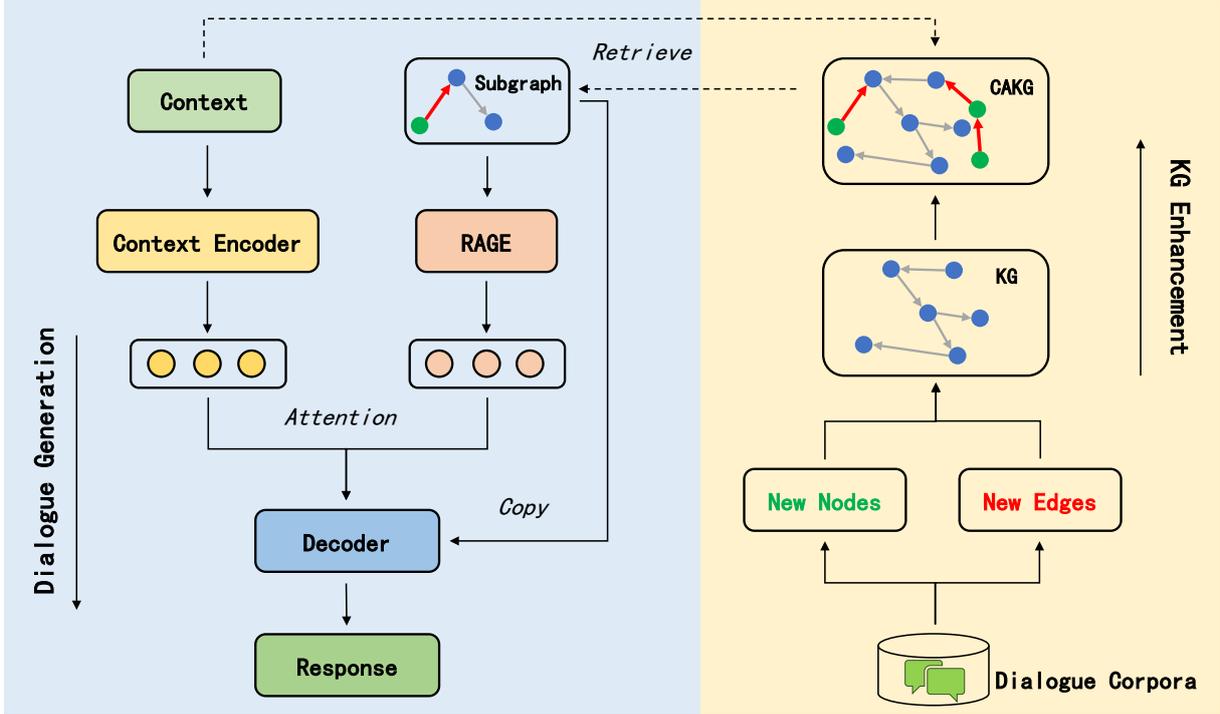


Figure 2: The pipeline of ECCF, which contains two parts. First, as in the right part, we extract new nodes and new edges from the dialogue corpora, then merge them with commonsense knowledge graph (KG) to construct conversational-aware knowledge graph (CAKG). Second, we use CAKG to guide the concept flows during the response generation process. For graph encoding, we use a relation-aware graph encoder (RAGE).

conversations are similar to bilingual word alignment. In practice, we first clean the corpora by removing all words except $V \cup V'$. Then we run the GIZA++ toolkit to get the alignment probabilities. Finally, we arrange the probabilities to select the top k alignments as new edges. More details of the alignment process can be found in their original paper (Och and Ney, 2003).

An example is presented in Figure 3. For the source concept “nurse”, we rank all the target concepts according to the alignment probabilities. The relations from “nurse” to the top k concepts are regarded as new edges, such as “nurse \rightarrow hospital”, and we attribute these edges to a new category: “DialogFlowTo”.

2.3 Response Generation with Conversation-Aware Knowledge Graph

2.3.1 Context Encoder

Given the dialogue context $X = (x_1, x_2, \dots, x_m)$, we utilize a bi-directional encoder to get the contextual representation $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m)$.

$$\mathbf{H} = \text{Encoder}(X). \quad (1)$$

The encoder can be Transformer (Vaswani et al., 2017) or GRU (Cho et al., 2014), to be consistent

<i>source</i>	<i>target</i>	<i>alignment prob</i>	
<i>nurse</i>	<i>nurse</i>	0.0917	} <i>Top k</i> <i>add new edges</i>
	<i>hospital</i>	0.0346	
	<i>nurses</i>	0.0306	
	<i>nursing</i>	0.0254	
	<i>medical</i>	0.0231	
	\vdots	\vdots	
	<i>tests</i>	$1.096 \times 1e-7$	
	<i>expert</i>	$1.087 \times 1e-7$	

Figure 3: Extract concepts and relations from natural conversations.

with previous methods (Zhang et al., 2020a; Zhou et al., 2018b, 2021b), we utilize GRU in our experiments and choose the last word hidden states \mathbf{h}_m as the representation of dialogue context.

2.3.2 Relation-Aware Graph Encoder

Since introducing the whole graph to the generation process is unpractical and unnecessary, we retrieve a subgraph g from G_c and encode g with the relation-aware graph encoder (RAGE), which is based on the Transformer Encoder (Vaswani et al., 2017). The subgraph g derives from the concepts in the dialogue history and their one-hop and two-

hop neighbor nodes⁵. To model the interactions between the dialogue context X and subgraph g , we set a special node \mathcal{X} to connect with all nodes of g , which represents the relations between dialogue and concepts. Then, we initialize the embedding of \mathcal{X} with \mathbf{h}_m , and the embedding of g with TransE embedding (Bordes et al., 2013). To model the graph structure of subgraph g , we design a graph mask matrix M :

$$m_{ij} = \begin{cases} 0 & \text{if } i = \mathcal{X} \text{ or } j = \mathcal{X}, \\ 0 & \text{if } i \in \text{Neighbor}(j), \\ -\infty & \text{otherwise,} \end{cases} \quad (2)$$

where $m_{ij} = 0$ indicates that node i and node j are connected, while $m_{ij} = -\infty$ represents the disconnect. Further, we replace the original Multi-Head Attention (MHA) with Relation-Aware Concept Attention (RACA), which incorporates the graph structure and node relations in the attention process. The differences are as follows:

$$\begin{aligned} \text{MHA} &= \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \\ \text{RACA} &= \text{softmax}\left(\frac{QK^T}{\sqrt{d}} + M + R\right)V, \end{aligned} \quad (3)$$

where Q, K, V is the query, key, and value vectors, more details in the original paper (Vaswani et al., 2017). M represents the graph mask matrix and R denotes edge relation bias:

$$r_{ij} = q^T \times e_{ij}, \quad (4)$$

where $e_{ij} \in \mathcal{R}^d$ is edge embedding⁶, $q \in \mathcal{R}^d$ is used to transform the vector to scalar which represents relation importance in the attention process. We employ different q in different heads and layers of the graph encoder, so that we can capture abundant and diverse relation-aware concept interactions. The output of the last layer is selected as the concept representations \mathbf{G} .

2.3.3 Decoder

The decoder generates response Y based on the dialogue context and subgraph. At t -th time step, the decoder state s_t is updated as follows:

$$s_t = \text{Decoder}(s_{<t}, y_{t-1}, \mathbf{H}, \mathbf{G}) \quad (5)$$

⁵As the two-hop neighbor nodes are extensive, we select 100 two-hop nodes for each concept. For the fairness of the experiment, we use the same two-hop nodes set as in as in Zhang et al. (2020a).

⁶For the edges from a node to itself, we give them a new category: ‘‘SelfTO’’. For edges from and to \mathcal{X} , we give them two new categories: ‘‘FromText’’ and ‘‘ToText’’.

To be consistent with previous works, we utilize GRU in this paper. We employ attention mechanism to capture useful information from \mathbf{H} and \mathbf{G} , more details in (Bahdanau et al., 2015).

In addition, we also apply the copy mechanism to directly copy concepts from subgraph g . The process can be formulated as follows:

$$\begin{aligned} \sigma_t &= \text{Sigmoid}(v_s^\top s_t), \\ p_t^v &= \text{Softmax}(\mathbf{W} \cdot s_t), \\ p_t^c &= \text{Softmax}(\mathbf{G} \cdot s_t), \\ p_t &= (1 - \sigma_t) \cdot p_t^v + \sigma_t \cdot p_t^c, \end{aligned} \quad (6)$$

where p_t^v and p_t^c are the probability of generation and copy, respectively.

2.3.4 Objective Function

Our objective function has two parts, the first is the negative log likelihood of response generation:

$$\mathcal{L}_1 = - \sum_{t=1}^n \log p(x_t | x_{<t}, X, H, G). \quad (7)$$

We also supervise the copy gate as in Zhou et al. (2018a); Chen et al. (2022), so that the decoder can accurately copy concepts from the subgraph:

$$\mathcal{L}_2 = \sum_{t=1}^n q_t \cdot \log \sigma_t + (1 - q_t) \cdot \log(1 - \sigma_t), \quad (8)$$

where $q_t \in \{0, 1\}$ indicates whether x_t is a concept word from the subgraph. The final objective function is $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$.

3 Experiment

3.1 Dataset

Follow Zhou et al. (2018b); Zhang et al. (2020a), we conduct experiments based on Reddit conversation dataset processed by (Zhou et al., 2018b). It contains 3,384,160 training pairs and 10,000 testing pairs. We use the commonsense knowledge graph ConceptNet (Speer et al., 2017) processed by Zhou et al. (2018b), which includes 21,471 nodes, 120,850 edges, and 44 types of edge relation.

3.2 Baselines

The baselines can be divided into three groups:

- **Standard seq2seq model**(Sutskever et al., 2014). The model is based on the classical encoder-decoder framework. The encoder and decoder are GRU as our model.

Model	Bleu-3	Bleu-4	Nist-3	Nist-4	Rouge-1	Rouge-2	Rouge-L	Meteor	PPL	Ent-4
Seq2seq	0.0226	0.0098	1.1056	1.1069	0.1441	0.0189	0.1146	0.0611	48.79	7.6650
MemNet	0.0246	0.0112	1.1960	1.1977	0.1523	0.0215	0.1213	0.0632	47.38	8.4180
CopyNet	0.0226	0.0106	1.0770	1.0788	0.1472	0.0211	0.1153	0.0610	43.28	8.4220
CCM	0.0192	0.0084	0.9082	0.9095	0.1538	0.0211	0.1245	0.0630	42.91	7.8470
ConceptFlow	0.0495	0.0239	1.8838	1.8896	0.2241	0.0457	0.2032	0.0956	29.44	10.2390
GPT-2(lang)	0.0162	0.0162	1.0840	1.0844	0.1321	0.0117	0.1046	0.0637	29.08*	11.6500
GPT-2(conv)	0.0262	0.0124	1.1745	1.1763	0.1514	0.0222	0.1212	0.0629	24.55*	8.5460
DialoGPT	0.0189	0.0095	0.9986	0.9993	0.0985	0.0117	0.0971	0.0546	18.65*	9.8163
ECCF	0.0644	0.0331	2.2573	2.2661	0.2592	0.0601	0.2340	0.1091	25.98	10.8173

Table 1: Automatic Evaluations. We highlight the best scores on each metric. The PPL scores of pre-trained models are not comparable because of different tokenization. The results indicate that our ECCF gets the highest scores on most metrics.

- **Knowledge enhanced models:** MemNet(Ghazvininejad et al., 2018), CopyNet(Zhu et al., 2017), CCM(Zhou et al., 2018b) and ConceptFlow(Zhang et al., 2020a). These models explore knowledge information during the generation process.
- **Pretrained models:** GPT-2 lang(Zhang et al., 2020a), GPT-2 conv(Zhang et al., 2020a), DialoGPT(Zhang et al., 2020b). These models have a large number of parameters and have been pretrained on large corpus. GPT-2 lang and GPT-2 conv are built based on GPT-2(Radford et al., 2019).

For seq2seq, MemNet, CopyNet, CCM, GPT-2 lang and GPT-2 conv, we directly use results in ConceptFlow paper (Zhang et al., 2020a). For ConceptFlow, we run their public codes⁷. For DialoGPT, we finetune it on the dataset⁸.

3.3 Evaluation Metrics

We use the following metrics for evaluation:

- **PPL (Serban et al., 2016):** Perplexity measures the fluency of the responses.
- **Bleu (Chen and Cherry, 2014), Nist (Dodgington, 2002), Rouge(Lin, 2004) :** These metrics measure the overlap between the generated response and the ground truth.
- **Meteor (Lavie and Agarwal, 2007):** Meteor measures the relevance between generated responses and ground truth.
- **Entropy (Zhang et al., 2018b):** Entropy measures the diversity of generated responses.

⁷<https://github.com/thunlp/ConceptFlow>.

⁸<https://huggingface.co/microsoft/DialoGPT-medium>

We implement the above metrics based on the code of Galley et al. (2018)⁹.

3.4 Implementation Details

For constructing CAKG, we utilize the training dataset for extracting conversational concept flows, which includes 3,384,160 utterance pairs. The frequency threshold m is set as follows: we first arrange the frequencies of V (original concepts in ConceptNet) in the dialogue corpora as $f_1, f_2, \dots, f_{|V|}$, then, $f_{0.2 \times |V|}$ is set as m . Noun words with frequency higher than m is selected as new concepts. Further, we choose the top 20% concept relations for each concept as new edges.

For response generation, we use 2-layer GRU as context encoder and decoder, 3 layers of Transformer encoder with relation-aware concept attention as graph encoder. We choose Adam as the optimizer, the batch size, learning rate, max gradients norm, and dropout are set to 30, 1e-4, 5, 0.2, respectively. We use TransE embedding (Bordes et al., 2013) and Glove embedding (Pennington et al., 2014) to initialize the embedding of concepts and words, respectively. We train our method on 8 V100 GPUs, and it takes about 1.5 hours for one-epoch training.

4 Evaluation

4.1 Automation Evaluation

The experimental results are shown in Table 1. Except for pre-trained models, our method achieves the lowest PPL score, indicating that the responses generated by our model are more fluent. Furthermore, Bleu, Nist, Rouge, and Meteor measure the

⁹<https://github.com/DSTC-MSR-NLP/DSTC7-End-to-End-Conversation-Modeling>

Graph	Nodes	Edges	Response Nodes	0-hop Nodes		1-hop Nodes		2-hop Nodes	
				amount	golden	amount	golden	amount	golden
G	21471	120850	5.691	5.8129	0.5998	90.5138	1.2064	99.7706	0.8823
G_c	21754	218478	6.192	6.3223	0.6352	100.6227	1.4114	99.7706	0.8823

Table 2: Statistics of graphs coverage on the conversation dataset. The amount and golden are the numbers of total concepts and concepts appearing in responses, respectively. Obviously, G_c has higher coverage than G .

	Fluency		
	Average	Best @1	kappa
ConceptFlow	2.2875	0.24	0.563
ECCF	2.4325	0.30	0.603
Golden	2.6975	0.69	0.665
	Appropriateness		
	Average	Best @1	kappa
ConceptFlow	1.6200	0.12	0.480
ECCF	1.6850	0.16	0.563
Golden	2.3275	0.81	0.603

Table 3: Evaluation results by human annotators. We also present Fleiss’ Kappa in the table. Kappa values range from 0.4 to 0.6, indicating fair agreement.

relevance between generated responses and ground truth responses in different ways. Our method outperforms all baselines by large margins on these metrics, demonstrating that the responses generated by our method are more relevant to the contexts and topic-consistent with humans. For diversity, our method gets the second-highest score, only lower than GPT-2. This proves that our proposed method can generate diverse responses. It is worth noticing that, although pre-trained models are slightly better at fluency and diversity, they perform much worse in relevance (Bleu, Nist, Rouge, Meteor) compared with our method and ConceptFlow. This indicates the superiority of explicitly modeling conversational topic shifts based on a knowledge graph.

4.2 Human Evaluation

To evaluate model performances more comprehensively, we follow Zhang et al. (2020a) and hire four human annotators to judge the quality of generated responses. Specifically, we randomly sample 100 cases for ConceptFlow, ours, and ground truth responses¹⁰. Annotators are required to score responses from 1 to 3 on two aspects: fluency and appropriateness. Fluency evaluates whether a response is fluent or contains grammar errors, while

¹⁰Zhang et al. (2020a) have proved that ConceptFlow outperforms a series of baselines including GPT-2 based methods. Therefore, we only use ConceptFlow for comparison here in the case of limited human resources.

appropriateness measures whether a response is relevant and reasonable to its dialogue context.

As in Table 3, ECCF is better than the strong baseline ConceptFlow in terms of both fluency and appropriateness, the best @1 ratios of ECCF are also higher than ConceptFlow, demonstrating the superiority of our method. However, there is a large gap between ours and humans, indicating that there is still plenty of room for improvement.

5 Analysis

5.1 Conversation-Aware Knowledge Graph

Table 2 presents the statistics of ConceptNet G and our CAKG G_c . Thanks to the conversational concept flows extracted from large-scale dialogue corpora, G_c has more concepts and relations. Thus, more concepts in the responses are covered, especially for 0-hop and 1-hop concepts. This further proves the limitation of the external commonsense knowledge graph. We conduct an ablation study by replacing CAKG with ConceptNet (Ours w/o CAKG). As in Table 4, the performance drops in both relevance and diversity, which proves the effectiveness of conversational concept flows.

To further explore the relation between commonsense knowledge graph and conversational concept flows, we remove some edges in ConceptNet when constructing CAKG. As shown in Table 4, our method performs worse on relevance, fluency, and diversity, much worse when more edges are removed. Therefore, we can infer that concepts and relations in commonsense knowledge graph are also of great necessity for guiding topic flows in natural conversation. Further, both commonsense and conversation knowledge are beneficial to response generation, a reasonable way is to combine them as in our method.

5.2 Conversational Concept Flows

We conduct a human evaluation to verify the quality of the extracted conversational concept flows. Specifically, we randomly sample 100 extracted edges, and hire four human annotators to judge

Model	Bleu-3	Bleu-4	Nist-3	Nist-4	Rouge-L	Meteor	PPL	Ent-4
ECCF	0.0644	0.0331	2.2573	2.2661	0.2340	0.1091	25.98	10.8173
w/o CAKG	0.0615	0.0319	2.1448	2.1541	0.2307	0.1055	26.40	10.7081
w/o 20% edges in CN	0.0634	0.0328	2.2102	2.2194	0.2322	0.1070	27.17	10.7391
w/o 50% edges in CN	0.0502	0.0249	1.8466	1.8528	0.2044	0.0938	30.77	10.2637
w/o RAGE	0.0529	0.0267	1.9270	1.9340	0.2115	0.0976	27.81	10.4316
w/o graph mask	0.0573	0.0290	2.0694	2.0771	0.2201	0.1025	26.81	10.6822
w/o relation aware	0.0589	0.0295	2.1394	2.1472	0.2246	0.1050	26.46	10.6871
w/o dialogue node	0.0595	0.0305	2.1316	2.1402	0.2237	0.1044	27.00	10.7731

Table 4: Analysis studies for conversation-aware knowledge graph (CAKG) and relation-aware graph encoder (RAGE), CN represents ConceptNet.

whether the target concept is relevant to the source concept. The results show that 68 edges are voted as relevant, of which 47 edges that all four annotators reach an agreement. According to our manually checking, these edges mainly have three categories, as shown in Figure 4. The first type corresponds to pairs that have realistic relations, such as “nurse” and “hospital”. The second type corresponds to pairs in the same kind, such as both “ps4” and “pc” are electronic devices. The third type corresponds to pairs with POS relations, such as “perception” is the noun form of “perceptive”. These three categories are meaningful, which proves that our method can obtain beneficial knowledge from natural conversations.

5.3 Relation-Aware Graph Encoder

We further investigate the effectiveness of the proposed relation-aware graph encoder (RAGE), and conduct several ablation studies as follows:

- **w/o RAGE.** To explore the superiority of our graph encoder, we replace it with a GNN-based architecture named GRAFT-Net (Sun et al., 2018), which is used by the strong baseline ConceptFlow (Zhang et al., 2020a).
- **w/o graph mask.** We remove the graph mask to explore the effectiveness of graph structure.
- **w/o relation aware.** We remove the relation bias in relation-aware concept attention, which aims to explore the effects of relation for graph encoding.
- **w/o dialogue node.** We remove the node \mathcal{X} to study the necessity of the interactions between dialogue context and knowledge graph.

The results are shown in Table 4, and there are several findings. First, the performance drops largely

when replacing our RAGE with traditional GNN used by previous work (Zhang et al., 2020a), which proves the effectiveness of transformer architecture and relation-aware concept attention. Second, the ablation study of each component further verifies the rationality of our design. Third, removing the graph mask results in worse performance compared with removing relation aware and removing dialogue node, which shows the importance of graph structure for graph encoding.

related things	the same kind of things	form change
nurse → hospital	ps4 → pc	perceptive → perception
digest → eat	jews → arabs	piping → pipe
rubble → buildings	bling → shiny	anarchist → anarchy

Figure 4: Several examples of high-quality concepts and relations we extracted from the conversation corpora. We classify them into three categories.

5.4 Case Study

To further investigate the performance, we present three cases in Table 5. In case 1, though the responses generated by DialogPT and ConceptFlow are proper, they are not as informative as ours. We can see that our response contains concept “episode” from G_c , which is consistent with the post and achieves a reasonable and natural topic shift. In case 2, it seems that DialogPT and ConceptFlow don’t understand the context and give wrong responses, while our method gives high-quality response that contains concepts “source”, “server” and “bug”, which are relevant to the context. In case 3, for the post about football and super bowl, DialogPT gives a short and dull response while ConceptFlow gives an unreasonable sentence. In contrast, our response is more consistent with the post. These cases show that our method generates more suitable responses, which is consistent with the automatic evaluation and human evaluation.

#1 post	i drove home last night , saw my dad for the first time in 6th months , and slept all day today . woke up and poured myself a rum and coke then started watching archer . today was a good day .
DialoGPT	i'm glad you had a good day.
ConcpetFlow	i 'm going to the same day . have a good night , man .
ECCF	i 'm going to watch the first episode of archer and see if i can get a chance to sleep .
#2 post	what do you mean ? from the alpha or from the beta ? his uav was nerfed like crazy in the beta , not being able to ping or find a monster if they 're sneaking at all .
DialoGPT	i mean from the alpha. i'm not sure if it was nerfed in the beta, but i'm pretty sure it was.
ConcpetFlow	you can see the alpha of the alpha , but it 's a bug .
ECCF	you can see the source on the server . i think he was just a bug .
#3 post	nowhere . fuck the patriots . fuck the seahawks . fuck football and fuck all these super bowl questions today .
DialoGPT	i'm with you man.
ConcpetFlow	you 're right . it 's a great way to get a new one . you can also see the seahawks and the seahawks .
ECCF	how about you ask me to watch the game ? you want to watch the world cup and see how much of a rivalry it takes to get to the point ?

Table 5: Three cases on the testset. We present responses generated by three different models. To study the impact of the knowledge graph, we mark concepts in the original ConceptNet in blue and concepts introduced by the enhanced graph in magenta.

6 Related Work

Dialogue Generation. Recently, dialogue generation (Adiwardana et al., 2020; Thoppilan et al., 2022) has achieved great progress in many aspects. Pre-trained dialogue models (Zhang et al., 2020c; Roller et al., 2021) improve the response quality largely, even reaches human performance in single-turn dialogue generation. Persona-based dialogue system (Zhang et al., 2018a) possesses a human persona and is able to converse in a more captivating way. Rashkin et al. (2019) propose empathetic response generation, which aims to recognize partner feelings and reply accordingly. To bridge the gap between human utterances and dialogue system utterances, Chen et al. (2022) propose to enhance empathetic response generation with human-like intents. In this paper, we focus on the topic shifts during conversations and propose to enhance dialogue generation with conversational concept flows.

Knowledge-Aware Dialogue Generation. One of the most crucial challenges in dialogue generation is the lack of knowledge. Plentiful works have been proposed to inject reasonable knowledge into responses. One kind of these works utilizes unstructured knowledge, e.g., Wikipedia articles (Dinan et al., 2019), goal-related documents (Feng et al., 2021) etc. Another kind of work focuses on structured knowledge. Zhou et al. (2018a) exploit concept relations in commonsense knowledge graph to imitate concept shifts in human conversation. Zhang et al. (2020a) develop this idea and propose

to explicitly model the concept flows in conversation. As we notice the gap between commonsense knowledge graph and natural conversations, we further propose to enhance dialogue generation with conversational concept flows.

There are also researches that extract information from natural conversations. Some of them extract relationships among persons on a domain-specific dataset (Yu et al., 2020; Xue et al., 2021; Long et al., 2021), while they focus on relation extraction not response generation. Others construct conversational graph from natural conversations to improve response generation (Xu et al., 2020b; Zou et al., 2021). However, their graphs only contain knowledge in conversations, while ignores the rich knowledge in commonsense knowledge graph. As shown in our analysis experiments, both types of knowledge are beneficial to response generation.

7 Conclusion and Future Work

In this paper, we argue the limitation of using external commonsense knowledge graph for response generation. To better capture topic shifts in natural conversation, we propose to enhance dialogue generation with conversational concept flows and construct conversation-aware knowledge graph. We further design a novel relation-aware graph encoder to capture the concept relations in knowledge graph. Extensive experiments on the large-scale Reddit dataset show the superiority of our method, and further analysis demonstrates the rationality of each

component. In future work, we expect to capture more structural information from natural conversations to improve dialogue generation.

Limitations

In this paper, we propose to enhance dialogue generation with conversational concept flows. Experimental results have shown that our method performs better than strong baselines. However, there are several major limitations. First, we use GIZA++ toolkit to extract concept relations, which is efficient but less expressive, as we cannot confirm the relations between concepts while they are quite different. For example, the relation between “nurse” and “hospital” is different to the relation between “thirsty” and “drink”. These relations have certain semantics and can be beneficial for response generation. Second, the experimental results in this paper are only based on one dataset Reddit. Although Reddit is large and contains 3,384,160 examples, more datasets can further verify the generalization ability of our methods. Third, we only combine conversational concept flows with ConceptNet (Speer et al., 2017), while other knowledge graphs (e.g., ATOMIC (Sap et al., 2019)) should be considered in future work to further explore the relations between conversational concept flows and commonsense knowledge.

Acknowledgements

This work was partly supported by the National Key Research and Development Program of China (No. 2020YFB1708200) and the Shenzhen Key Laboratory of Marine IntelliSense and Computation under Contract ZDSYS20200811142605016.

References

- Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020, Held Virtually, Original Venue: Atlanta, Georgia, USA, June 8-11, 2020*, pages 830–839. AAAI Press.
- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.
- Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 362–367. The Association for Computer Linguistics.
- Mao Yan Chen, Siheng Li, and Yujiu Yang. 2022. Emphi: Generating empathetic responses with human-like intents. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1063–1074. Association for Computational Linguistics.
- Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.
- Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145.
- Hao Fang, Hao Cheng, Maarten Sap, Elizabeth Clark, Ari Holtzman, Yejin Choi, Noah A. Smith, and Mari Ostendorf. 2018. Sounding board: A user-centric and content-driven social chatbot. In *Proceedings of NAACL-HLT 2018: Demonstrations*, pages 96–100, New Orleans, Louisiana.
- Song Feng, Siva Sankalp Patel, Hui Wan, and Sachindra Joshi. 2021. Multidoc2dial: Modeling dialogues grounded in multiple documents. In *Proceedings of the 2021 Conference on Empirical Methods in*

- Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6162–6176. Association for Computational Linguistics.
- Michel Galley, Chris Brockett, Xiang Gao, B. Dolan, and Jianfeng Gao. 2018. End-to-end conversation modeling : Moving beyond chitchat dstc 7 task 2 description (v 1 . 0).
- Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. Jointly optimizing diversity and relevance in neural response generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1229–1238. Association for Computational Linguistics.
- Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5110–5117. AAAI Press.
- Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Trans. Inf. Syst.*, 38(3):21:1–21:32.
- Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6384–6392. AAAI Press.
- Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.
- Xinwei Long, Shuzi Niu, and Yucheng Li. 2021. Position enhanced mention graph attention network for dialogue relation extraction. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1985–1989. ACM.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5370–5381. Association for Computational Linguistics.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 300–325. Association for Computational Linguistics.
- Sahand Sabour, Chujie Zheng, and Minlie Huang. 2021. CEM: commonsense-aware empathetic response generation. *CoRR*, abs/2109.05739.
- Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3027–3035. AAAI Press.
- Iulian Vlad Serban, Alessandro Sordani, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3776–3784. AAAI Press.
- Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the*

- 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, pages 1577–1586. The Association for Computer Linguistics.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.
- Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4231–4242. Association for Computational Linguistics.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.
- Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric P. Xing, and Zhiting Hu. 2019. Target-guided open-domain conversation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5624–5634. Association for Computational Linguistics.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Quan Tu, Yanran Li, Jianwei Cui, Bin Wang, Ji-Rong Wen, and Rui Yan. 2022. MISC: A mixed strategy-aware model integrating COMET for emotional support conversation. *CoRR*, abs/2203.13560.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Wenjie Wang, Minlie Huang, Xin-Shun Xu, Fumin Shen, and Liqiang Nie. 2018. Chat more: Deepening and widening the chatting topic via A deep model. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 255–264. ACM.
- Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020a. Recipes for safety in open-domain chatbots. *CoRR*, abs/2010.07079.
- Jun Xu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020b. Conversational graph grounded policy learning for open-domain conversation generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1835–1845. Association for Computational Linguistics.
- Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2021. Gdpnet: Refining latent multi-view graph for relation extraction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14194–14202. AAAI Press.
- Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4927–4940. Association for Computational Linguistics.
- Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020a. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2031–2043. Association for Computational Linguistics.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2204–2213. Association for Computational Linguistics.
- Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018b. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1815–1825.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 270–278. Association for Computational Linguistics.

- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020c. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 270–278. Association for Computational Linguistics.
- Hao Zhou, Minlie Huang, Yong Liu, Wei Chen, and Xiaoyan Zhu. 2021a. EARL: informative knowledge-grounded conversation generation with entity-agnostic representation learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2383–2395. Association for Computational Linguistics.
- Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018a. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 730–739. AAAI Press.
- Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018b. Commonsense knowledge aware conversation generation with graph attention. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4623–4629. ijcai.org.
- Pei Zhou, Karthik Gopalakrishnan, Behnam Hedayatnia, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2021b. Think before you speak: Using self-talk to generate implicit commonsense knowledge for response generation. *CoRR*, abs/2110.08501.
- Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *CoRR*, abs/1709.04264.
- Yicheng Zou, Zhihua Liu, Xingwu Hu, and Qi Zhang. 2021. Thinking clearly, talking fast: Concept-guided non-autoregressive generation for open-domain dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2215–2226. Association for Computational Linguistics.

SMHD-GER: A Large-Scale Benchmark Dataset for Automatic Mental Health Detection from Social Media in German

Sourabh Zanwar

RWTH Aachen University
sourabh.zanwar@rwth-aachen.de

Daniel Wiechmann

University of Amsterdam
d.wiechmann@uva.nl

Yu Qiao

RWTH Aachen University
yu.qiao@rwth-aachen.de

Elma Kerz

RWTH Aachen University
elma.kerz@ifaar.rwth-aachen.de

Abstract

Mental health problems are a challenge to our modern society, and their prevalence is predicted to increase worldwide. Recently, a surge of research has demonstrated the potential of automated detection of mental health conditions (MHC) through social media posts, with the ultimate goal of enabling early intervention and monitoring population-level health outcomes in real time. Progress in this area of research is highly dependent on the availability of high-quality datasets and benchmark corpora. However, the publicly available datasets for understanding and modeling MHC are largely confined to the English language. In this paper, we introduce SMHD-GER (Self-Reported Mental Health Diagnoses for German), a large-scale, carefully constructed dataset for MHC detection built on high-precision patterns proposed for English. We provide benchmark models for this dataset to facilitate further research and conduct extensive experiments. These models leverage engineered (psycho-)linguistic features as well as BERT-German. We also examine nuanced patterns of linguistic markers characteristics of specific MHC.

1 Introduction

Mental health is a major challenge in healthcare and in our modern societies at large (Rehm and Shield, 2019; Santomauro et al., 2021). The World Health Organization estimates that 970 million people worldwide suffer from mental health conditions¹², with the rate of undiagnosed mental disorders estimated to be as high as 45% (La Vonne et al., 2012).

¹<https://www.who.int/news-room/fact-sheets/detail/mental-disorders>

²Mental disorders' can also be referred to as 'mental health conditions'. The latter is sometimes used as a broader term encompassing mental disorders, psycho-social disabilities, and mental conditions, include different types of depression, bipolar disorder, schizophrenia, anxiety disorders, chronic stress etc.. In this work, the two terms are used interchangeably.

The enormous societal impact of mental health conditions (MHC) requires prevention and intervention strategies that focus primarily on screening and early detection. The last decade has seen a surge in digital mental health research, an interdisciplinary line of research that brings together insights from computational linguistics, cognitive psychology and computational social sciences to understand the relationship between patterns of language use and mental health conditions (D'Alfonso, 2020; Schindler and Domahidi, 2022). Natural language processing, in particular, is increasingly recognized as having transformative potential to support healthcare professionals in the diagnosis and treatment of mental disorders and enable people to lead healthy lives (see Guntuku et al. 2017; Thieme et al. 2020; Chancellor and De Choudhury 2020; Zhang et al. 2022 for recent overviews of this research).

Progress in this area of research is highly dependent on the availability of high-quality datasets and benchmark corpora. Social media has emerged as an increasingly vital resource for obtaining such data, as it is now a central place for individuals to participate in discussions, share information, and seek advice. Based on data drawn from platforms such as Twitter and Reddit, recent work has developed scalable methods for constructing mental health datasets based on self-reported diagnoses or grouping individuals based on activity patterns (Coppersmith et al., 2015; Yates et al., 2017; Cohan et al., 2018; Kumar et al., 2015). However, recent reviews on the state of data used for mental health status on social media show that the vast majority of the publicly available datasets for understanding and modeling MHC are on the English language: For example, of the 102 datasets reviewed in Harrigian et al. (2021) 83% were on English with the remaining 17% distributing over five other languages: Chinese (9.8%), Japanese (3.9), Korean (1.9), Spanish and Portuguese (each < 1%). Zhang

et al. (2022) report that 81% of all the datasets are in English, followed by datasets in Chinese (10%), and Arabic (1.5%). While an overwhelming focus on English data is a theme throughout the NLP community, it is a specific concern in this domain where culture often influences the presentation of mental health disorders (De Choudhury et al., 2017; Loveys et al., 2018). Thus, there is an urgent need for publicly available, high-quality mental health datasets and benchmark models to support early detection of MHC in languages other than English.

The main contributions of this work are three-fold: (1) We introduce SMHD-GER (Self-reported Mental Health Diagnosis for German), a new large dataset of social media posts for mental health detection in the German language, and make it publicly available; (2) We provide benchmark models for the detection of four mental health conditions based on a comprehensive set of text-based features that pertain to multiple levels of language use, the German BERT-based model, and hybrid models that combine the two; and (3) We uncover nuanced patterns of linguistic markers characteristic of specific mental health conditions.

The remainder of this paper is organized as follows: In Section 2 we briefly review available social media datasets and NLP classification methods for MHC detection. Section 3 details the construction of the SMHD-GER dataset along with an ethics and privacy statement. Section 4 presents the results of a analysis of linguistic markers of specific MHC. In Section 5, we describe the modeling approach for our benchmark models, and in Section 6, we present and discuss the results. Finally, we conclude with directions for future work in Section 6.

2 Related work

In this section, we provide a concise overview of some of the most widely used self-disclosure social media datasets along with the classification methods used in the detection of mental health conditions. The self-disclosure approach to obtaining labeled data from social media was introduced in Coppersmith et al. (2014) and further refined in consecutive work (Yates et al., 2017; Cohan et al., 2018). In this approach, public self-reports of mental health diagnoses are identified through the use of carefully designed ‘diagnosis patterns’ combined with ‘diagnosis keywords’ mapped to particular mental health conditions: A user is included for

a specific MHC if one of the condition keywords occurs within a certain distance of the diagnosis pattern. Coppersmith et al. (2014) originally applied this approach to Twitter data and identified approximately 1,200 users with four MHC (bipolar, depression, PTSD, SAD) by matching diagnosis patterns in their tweets (e.g., “I was diagnosed with depression”). This dataset was employed in the shared task at the 2nd Computational Linguistics and Clinical Psychology Workshop (CLPsych 2015) that focused on identifying depression and PTSD users on Twitter (Coppersmith et al., 2015). Submissions to the task used traditional (shallow) classification models trained on unigram vectors, character language models, closed-vocabulary approaches (e.g. LIWC, Pennebaker et al., 2001) and supervised topic models. The leading systems reached average precision rates over 85% for both MHC. However, the dataset had a balanced distribution between the classes, rather than one that accurately reflect the user population. This hampered the reliable estimation of actual false alarm rates, as the number of false alarms in the general population is estimated to be 7-15 times higher than in the CLPsych 2015 test sample (Coppersmith et al., 2015).

The text content of a Tweet can contain up to 280 characters or Unicode glyphs. Thus, this format presents a barrier to capturing mental health related language signals. Recent work on compiling datasets for mental health is increasingly turning to Reddit for long-form content that can provide additional linguistic insights³: Yates et al. (2017) applied the self-disclosure approach to create the Reddit Self-reported Depression Diagnosis (RSDD) dataset, which contains 9,210 users with depression and 107,274 control users. Apart from increasing the dataset size by an order of magnitude – 969 posts per user with mean post length of 148 words, the RSDD dataset displays a realistic number of control users matched with each diagnosed user.

The main limitation of the RSDD dataset is its focus on a single mental health condition, depression. In what is to our our knowledge the most comprehensive, carefully constructed mental health dataset based on the self-disclosure approach, Cohan et al. (2018) expand on RSDD by including for eight additional MHC: The Self-reported Mental Health Diagnoses (SMHD) dataset, whose design

³Reddit (<https://www.reddit.com/>) is a social news aggregation, content rating, and discussion website without any length constraints.

underlies the current work, comprises 20,406 diagnosed users and 335,952 matched controls. Diagnosed users were identified using a refined version of the high precision diagnosis patterns used in RSDD, which incorporated synonyms in matching patterns from two synonym mapping ontologies (MedSyn, Yates and Goharian, 2013, Behavioral, Yom-Tov et al., 2013). Control users were selected based on a similar Reddit posting activity, i.e. each diagnosed user was matched with an average of 9 control users with a similar number of posts and a similar range of subreddits they posted in. Importantly, SMHD does not contain any posts that contain any mental health terms or that have been posted in a mental health-related subreddits. The detection of MHC can thus not be based on terms associated with specific mental health conditions. Along with the dataset itself, Cohan et al. (2018) provided benchmarks for both binary (MHC vs. control) and multi-class classification settings. The classification methods included several traditional (shallow) machine learning models (logistic regression, XGBoost (Chen and Guestrin, 2016), support vector machine with linear kernel) trained on tf-idf bag-of-words features, a shallow neural net model trained on character ngrams (Supervised FastText, Joulin et al., 2016), and a Convolutional neural network trained on ngram sequences represented by the FastText embeddings. Subsequent work has improved MHC detection accuracy using Hierarchical Attention Networks (Sekulic and Strube, 2019) and attention-based model using BERT representations (Jiang et al., 2020). Recently, (Zanwar et al., 2022) leveraged transformer language models (BERT Devlin et al., 2019 and RoBERTa Liu et al., 2019) in combination with attention-based BLSTM models trained on engineered language features for MHC detection.

3 Data

3.1 Data construction

In this section we describe the construction and characteristics of the SMHD-GER dataset. SMHD-GER comprises data on seven mental health conditions that correspond to branches in the DSM-5 (APA, 2013): Five conditions are top-level DSM-5 disorders: schizophrenia spectrum disorders (schizophrenia), bipolar disorders (bipolar), depressive disorders (depression), anxiety disorders (anxiety), obsessive-compulsive disorders (OCD). The remaining two conditions are one rank lower: post-

traumatic stress disorder (ptsd) is classified under trauma- and stress-related disorders, and attention-deficit/hyperactivity disorder (ADHD) under neurodevelopmental disorders. The construction of the dataset is an adaptation of the general procedure underlying the construction of the SMHD dataset described in Cohan et al. (2018): The textual data were obtained from Reddit using the Pushshift.io API Wrapper by searching for all posts mentioning any mental health (MH) terms, such as the name of a condition. The list of MH-terms was derived from the corresponding materials used for the SMHD dataset using DeepL translator⁴ followed by manual inspection and editing. We then filtered these posts to keep only those that were in German using the 'langdetect' the Python library.⁵

Diagnosed users were identified using high precision diagnosis patterns as in Cohan et al. (2018): Reddit users received a positive label for a specific MHC if and only if at least one of their posts explicitly states that they suffer from a specific condition or are engaging in behaviors indicative of it. These were triangulated with specific expressions, such as "Ich wurde diagnostiziert mit X" ("I was diagnosed with X"), where X would be filled with a specific MH-term (e.g. "Depression"). Like the MH-terms, the diagnosis patterns were derived from the corresponding materials used for the SMHD dataset using DeepL translator followed by manual inspection and editing. We then collected all posts and comments for the users with a positive label and filtered these to keep only those that (i) were in German, (ii) had no mentions of any of the MH-terms and (iii) were not posted in a subreddit related to mental health (MH-subreddit).

Control users: To compile the data used for control we collected 1049202 posts from 24981 users from r/de⁶ subreddit, and filtered out those users who (i) had used any MH-term in any of their posts or (ii) had posted in a MH-subreddits. For all remaining users we collected all the available posts and comments in German. All Reddit posts were made between August 14, 2009, and October 2, 2022 (inclusive). This procedure yielded a dataset containing 5,611 diagnosed users and 22,426 control users. On average each user in the dataset contributed 16.23 posts with a mean post length of 69 word tokens (see Table 1).

⁴<https://www.deepl.com/translator>

⁵<https://pypi.org/project/langdetect/>

⁶r/de is a reddit community for german speakers

MHC	#users	#posts	mean #posts/user	mean #words/post	sd #words/post
ADHD	1055	19212	18.21	59.50	119.35
Anxiety	14	277	19.79	263.85	557.50
Bipolar	1424	23711	16.65	46.46	84.76
Control	22426	361670	16.13	42.08	56.97
Depression	975	15654	16.06	48.12	110.92
OCD	257	3881	15.10	44.02	111.54
Other	1072	17591	16.41	46.67	86.47
PTSD	728	11684	16.05	44.25	74.39
Schizophrenia	86	1380	16.05	44.64	66.60

Table 1: Means (standard deviations) and counts of posts, tokens and characters for diagnosed and control users.

3.2 Ethics and privacy

Although we rely solely on publicly available Reddit data, mental health remains a sensitive issue, and measures to avoid risks to individuals in social media research should always be considered (Hovy and Spruit, 2016; Šuster et al., 2017; Cohan et al., 2018). Following the data handling procedures of the original SMHD (Cohan et al., 2018), we do not publish excerpts from the data, we did not attempt to contact users, and we did not attempt to identify or link users to other social media accounts. We also replace usernames with random identifiers to prevent users’ identities from being revealed without external information. The SMHD-GER dataset is made available through a data usage agreement (DUA) that protects user privacy. Specifically, the DUA specifies that no attempt may be made to publish any part of the dataset (which could lead to user identification), contact users, identify them, or link them to other user information.

An ethical issue raised by an anonymous reviewer concerns the annotation of positive mental health conditions through self-disclosure of users, as those who choose to disclose them might differ from the population of individuals living with such conditions without disclosing them. Another ethical issue concerns the use of psychometric evaluation of large text corpora leveraging LIWC-like features alone, as this approach may lack precision: Since LIWC’s diagnostic scores are based on both computational correlation and human judgment (in determining the system’s dictionaries and word categories), the outcomes may reflect evaluative biases grounded in the context of social, historical, and cultural development (Stark, 2018).

4 Analysis of Linguistic Markers

In this section, we address the exploration of nuanced patterns of linguistic markers that are indica-

tive of specific MHC. We first obtained measurements of 117 engineered language features that can be roughly divided into five groups: (1) features related to morphological and syntactic structural complexity (N=5), (2) features related to lexical sophistication, variety, and richness (N=8), (3) word-level ngram features related to register-specific language use (N=20), (4) features covering the German version of the LIWC (Linguistic Inquiry and Word Count) dictionary (N=68), and (5) word-level dictionary features from three lexicons related to emotion, affect and sentiment (N=16). An overview of these features can be found in Table 5 in the appendix.

The first group of includes surface features related to the length of production units, such as the average length of clauses and sentences, and the type and frequency of embedded structures, such as mean length of sentence or number of dependent clauses per sentence (Lu, 2010). This group also includes an information-theoretic feature based on the Deflate algorithm (Deutsch, 1996).

The second group of features probing lexical density features, such as the ratio of the number of lexical (as opposed to grammatical) words to the total number of words in a text, lexical variation, i.e. the range of vocabulary as manifested in language use as captured by text-size (corrected) type-token ratio (Lu, 2012).

The third group comprises register-based n-gram frequency features that take into account both frequency rank and the number of word n-grams ($n \in [1, 5]$). The latter were derived from four corpora compiled as to represent language use in four language registers (academic, fiction, news, spoken; see Table 6).

The fourth feature group is based on the German version of the LIWC dictionary (Linguistic Inquiry and Word Count) (Pennebaker et al., 2001).

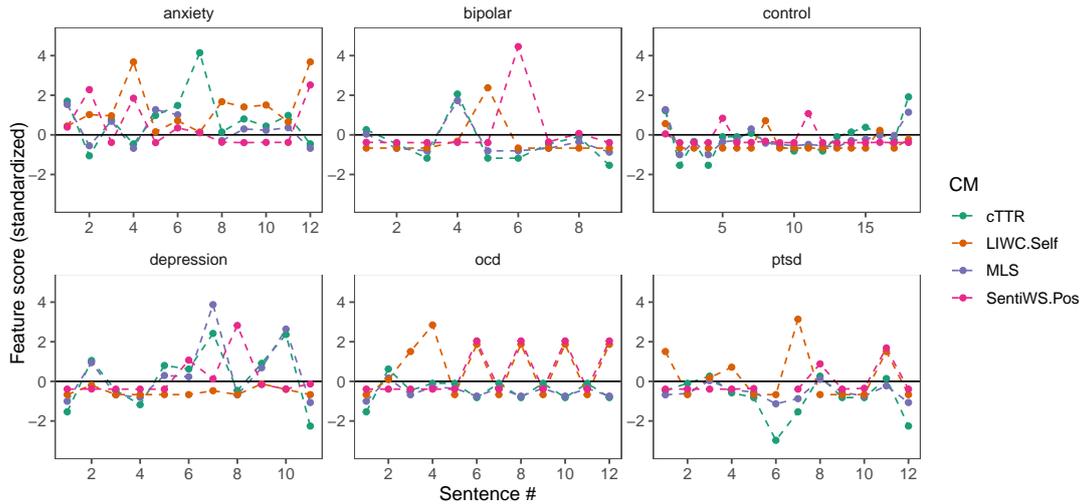


Figure 1: Within-text distributions of four textual features for six randomly selected Reddit post from as many MHC groups (cTTR = corrected Type-Token Ratio, LIWC.Self = self-focused language, MLS = mean length of sentence, SentiWS.Pos = words with positive semantic orientation). All features were z-standardized with 0 representing the corpus average.

The creators of the German version of the LIWC validated this version and demonstrated that the German LIWC categories have a high degree of equivalence to their English counterparts (Wolf et al., 2008). Building on the results of previous studies using LIWC categories for MHC detection in English (e.g. Cohan et al., 2018), we expect that subcategories of particular interest for the MHC classification task will include words with positive or negative emotions, words related to social processes (family/friends/society), pronouns that can capture inclusive (we, us) or exclusionary (you, they, them) language use, and words related to how the person feels (sad, anxious).

The fifth group includes features from three lexicons: MEemoLon (Buechel et al., 2020) is a lexicon comprising eight emotional variables with more than 100k lexical entries for eight emotional variables: Valence, Arousal, Dominance, and Joy, Anger, Sadness, Fear, and Disgust. ANGST is the German adaptation of the Affective Norms for English Words (Schmidtke et al., 2014). It comprises 1,003 German translations of the ANEW material that were rated on a total of six dimensions: the three original scales for valence, arousal, and dominance plus three additional arousal ratings on an adapted scale. SentiWS (Remus et al., 2010) is a dictionary containing 3,468 sentiment bearing German words (1,650 negative and 1,818 positive) across four word classes (adjectives, adverbs, nouns and verbs) along with their weighted senti-

ment scores.

All measurements of these features were obtained using an automated text analytics system that employs a sliding window technique to compute measurements at the level of individual sentences. These measurements capture the within-text distributions of scores for a given feature (for recent applications, see e.g. Wiechmann et al., 2022 or Kerz et al., 2022). Tokenization, sentence splitting, part-of-speech tagging, lemmatization and syntactic PCFG parsing were performed using Stanford CoreNLP (Manning et al., 2014). Examples of these within-text distributions is shown in Figure 1. Each of panels in Figure 1 shows the distributions of four of the 117 textual features for one of six randomly selected texts representing different MHC groups. We note that the distribution of feature values is generally not uniform, but shows large fluctuations over the course of the text. The six texts are characterized by different patterns of spikes of specific features: For example, the bipolar text exhibits a large spike in the SentiWS.Pos feature, which refers to words with positive semantic orientation. The OCD text is characterized by regular peaks of the LIWC.Self feature, which captures self-focused language. The anxiety text displays frequent spikes of high values (>2 standard deviations from corpus average) for three of the four features. In comparison, the control text shows less fluctuation with features scores being closer to the corpus average values. The classification mod-

els described in Section 5.1 are designed to detect and exploit these fluctuations for the detection of specific MHCs. The average scores of all features across all groups are provided in Table 8 in the appendix.

To identify profiles of language use that are characteristic of particular MHC, we compare these feature scores across users in each MHC group using factorial analyses of variance (ANOVA). We focus on those features that display significant differences across groups ($N=16$, for $\alpha = 0.05$). Figure 2 presents a cluster heatmap visualizing the patterns in the data matrix with the MHC groups and the 16 most significant language features.

The results of these analyses revealed some interesting patterns of differential language use: We find that the control group is situated at the margin of the clustering, indicating that the patterns of language use of diagnosed MHC are distinguishable from this baseline.

The language use of anxiety is distinctly different from all other MHC. It is characterized by very high feature scores on five LIWC dimensions related words referring to self-reference, death and sadness. They are further characterized by high scores on the top feature cluster, comprising words referring to anger, fear, disgust, sadness, arousal and negative emotions.

The language use of schizophrenia, is similar to anxiety in that it too displays a larger proportion of words indicating negative emotions. However, it is characterized by low scores on LIWC dimensions related words referring to self-reference. They are also characterized by low scores on the n-gram frequency features, indicating dependence on conventional phrases from specific speech registers.

A striking feature of obsessive-compulsive disorder (OCD) is its heavy reliance on such terms. A characteristic feature of (unipolar) depression is a markedly increased use of words with positive semantic orientation, in stark contrast to bipolar depression, which has significantly lower scores on this dimension. This is intriguing in light of the fact that distinguishing between bipolar disorder and recurrent unipolar depression is a major clinical challenge (de Almeida and Phillips, 2013). In general, conditions of depression and bipolar disorder, attention-deficit/hyperactivity disorder (ADHD) and post traumatic stress disorder (PTSD) display similar patterns of language use.

These findings reflect evidence in the psychiatric

MHC	# posts	mean # words	mean # chars
ADHD	1052	168.78	805.78
Bipolar	1421	150.50	853.66
Depression	974	153.89	872.87
PTSD	728	150.57	902.34
Control	12789	158.55	848.23

Table 2: Description statistics of the data used in benchmark experiments. Note: The size of the control data used in the binary MHC classification tasks were adopted to outnumber the positive cases by a factor of 9. The descriptive statistics of the control categories are based on the entire control corpus.

literature indicating that there is considerable overlap in clinical symptoms and pathophysiological processes and that depressive symptoms may also occur in the context of another psychiatric disorder (e.g., bipolar disorder) (Baldwin et al., 2002). Furthermore, psychiatric data suggest that depressive disorders (i.e., major depressive disorder and dysthymia) are highly comorbid with other common mental disorders (Rohde et al., 1991; Gold et al., 2020).

5 Experiments

5.1 Experimental Setup

In this section, we describe MHC detection experiments performed to obtain benchmark models for the SMHD-GER dataset. We conduct binary classification experiments for the top four most frequently attested MHC in the dataset, namely ADHD, bipolar, depression and PTSD. For each MHC, we use a 1:9 ratio of positive cases to controls to create a more realistic unbalanced classification setting. The size of the textual input to the models was constrained to fall between 110 words, which corresponds to the median number of words all posts, and 512 words, which represents an upper limit to the BERT models. In case no single post of a given user satisfied these constraints, we concatenated several posts from that user so that their total amount fell within the specified boundaries (Figure 5 in the appendix presents a decision tree of the selection method). Table 2 presents the descriptive statistics of the dataset used in classification experiments.

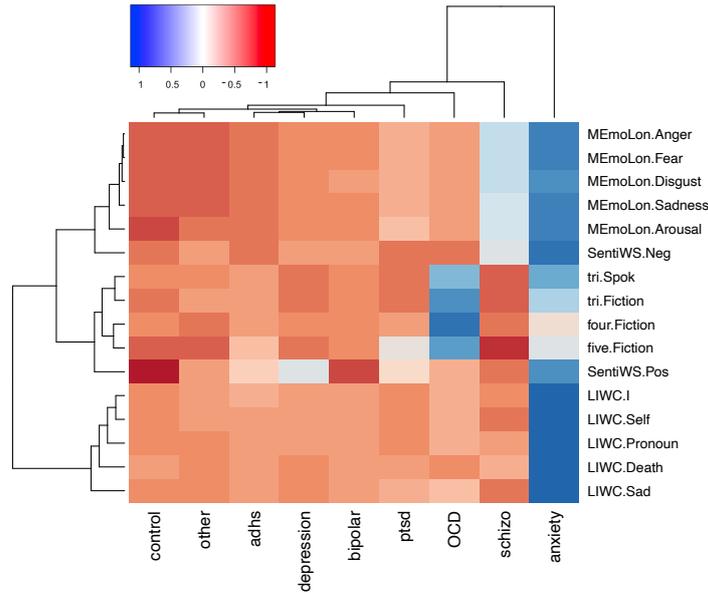


Figure 2: Heatplot of the language profiles of the nine MHC categories (on the x-axis) based on the top-16 language features (on the y-axis). Columns and rows are ordered according to the results of hierarchical clustering, with the dendrograms at the margins showing the groupings of MHC categories and features. Features with the prefix ‘MEemoLon’ refer to emotion categories from the lexicon of the same name. SentiWS.Neg refers to the category of negative words from the SentiWS dictionary. The terms ‘tri’, ‘four’ and ‘five’ refer to the size of word n-gram features from the spoken (‘Spok’) and fiction (‘Fiction’) reference corpora. Features with the prefix ‘LIWC’ refer to categories from the lexicon of the same name. Colors denote z-standardized feature scores.

5.2 Classification models

We performed experiments with three benchmark models: (1) a fine-tuned German BERT model (GBERT), (2) a bidirectional long short-term memory (BLSTM) classifier trained on measurements of linguistic features described in Section 4, and (3) hybrid model integrating GBERT predictions with the engineered language features introduced in Section 4. For (1) we used the pretrained ‘German bert-base-uncased’ (GBERT, Chan et al., 2020) model from the Huggingface Transformers library (Wolf et al., 2020) with an intermediate BLSTM layer with 256 hidden units (Al-Omari et al., 2020). For (2) - the model based solely on linguistic features, we constructed a 5-layer BLSTM with a hidden state dimension of 512. The input to that model is a sequence $CM_1^N = (CM_1, CM_2, \dots, CM_N)$, where CM_i , the output of our text analytics system for the i th sentence of a post, is a 117 dimensional vector and N is the sequence length. To predict the labels of a sequence, we concatenate the last hidden states of the last layer in forward (\vec{h}_n) and backward directions (\overleftarrow{h}_n). The result vector of concatenation $h_n = [\vec{h}_n | \overleftarrow{h}_n]$ is then transformed through a 2-layer feedforward neural network, whose activation function is Rectifier Linear Unit (Agarap,

2018). The output of this is then passed to a Fully Connected Layer FC with ReLu activation function and dropout of 0.2 and it is finally fed to a final FC layer. The output is finally passed through sigmoid function and finally a threshold is used to determine the labels. We trained these models for 100 epochs, with a batch size of 256, a sequence length of 5 and learning rate of 1e-3. The architecture of the hybrid classification model - model (3) - consists of two parts: (i) a pre-trained Transformer-based model with a BLSTM layer and FC layer on top of it and (ii) the linguistic features of the text fed into a BLSTM network and a subsequent FC layer. The FC layers of both parts take the concatenation of last hidden states of the last BLSTM layer in forward and backward direction. We concatenate the outputs of these layers before finally feeding them into a final FC layer with a sigmoid activation function. The model used to generate predictions for the test set was specified as follows: 2-layer BLSTM, 256 hidden units and a dropout of 0.2; BLSTM-PsyLing: 3-layers, hidden size of 512 and dropout 0.2. We trained this model for 12 epochs, saving the model with the best performance (F1-Score) on the development set. The optimizer used is AdamW with a learning rate of 2e-5 and a weight

Model	Metric	ADHD	Bipol.	Depr.	PTSD
Majority Class	Pre	44.85	44.85	44.84	44.82
Baseline	Rec	50.00	50.00	50.00	50.00
	F1	47.29	47.28	47.28	47.27
GBERT	Pre	50.63	50.36	49.74	50.57
	Rec	50.26	51.02	48.13	46.68
	F1	50.44	50.68	48.92	48.54
PsyLing	Pre	56.12	54.78	50.41	50.15
	Rec	52.45	55.31	50.18	49.86
	F1	53.22	53.97	50.26	49.92
Hybrid	Pre	51.29	51.85	51.62	53.20
	Rec	53.47	52.03	50.38	52.44
	F1	53.08	51.91	50.89	53.03

Table 3: Results of MHC prediction experiments (all values of performance metrics are macro averages)

decay of $1e-4$. Structure diagrams of the model based solely on linguistic features and the hybrid architectures are presented in Figures 4 and 3 in appendix. All models were trained using 5-fold CV of the training data as base classifiers and model stacking was performed using logistic regression as a meta-learner to adaptively combine the outputs of the base classifiers.

6 Results and Discussion

Table 3 gives an overview of the results of the MHC prediction experiments. All three baseline models displayed significant improvements in macro F1 scores over the majority baseline for all four MHC. Our PsyLing model consistently outperformed the GBERT baseline in terms of precision, recall and F1 (average improvement F1 = +2.37%; average improvement precision = 2.54%; average improvement recall = +2.93%). This result demonstrates that strong, interpretable mental health detection systems can be built if and when they make full use of the linguistic signals. The PsyLing model achieves highest performance in two of the four MHC, ADHD and bipolar disorder, with improvements over the hybrid model of +2.06% F1 for bipolar and +0.14% F1 for ADHD. However, the hybrid model improves on the performance of the PsyLing model by +3.11% F1 for PTSD and +0.63% F1 for depression.

The results of error analyses shown in Table 4 revealed that these performances were related to the divergent behaviors of the GBERT and PsyLing models for different MHCs: For Depression and PTSD the PsyLing model has a high

Model	MHC	TN	FP	FN	TP
GBERT	ADHD	1734	96	187	23
	Bipolar	2466	112	157	20
	Depression	1374	136	361	17
	PTSD	1168	96	132	14
PsyLing	ADHD	1764	66	192	18
	Bipolar	2344	127	253	31
	Depression	1333	276	231	48
	PTSD	939	268	162	41
Hybrid	ADHD	1500	330	161	49
	Bipolar	2289	260	182	24
	Depression	1633	96	138	21
	PTSD	1160	104	127	19

Table 4: Confusion matrices of the three benchmark models (TN: True Negatives, FP: False Positive, FN: False Negative, TP: True Positive)

false alarm rate, i.e. it classified users as being diagnosed, when they are in fact not (Depression: $FP_{GBERT}=136$, $FP_{PsyLing}=276$; PTSD: $FP_{GBERT}=96$, $FP_{PsyLing}=268$). On the other hand, it also correctly identified a much higher proportion of diagnosed users (Depression: $TP_{GBERT}=17$, $TP_{PsyLing}=48$; PTSD: $TP_{GBERT}=14$, $TP_{PsyLing}=41$). Our results thus indicate that the hybrid model improves on the PsyLing model for depression and PTSD by leveraging the lower false alarm rate of GBERT for these MHC. These results demonstrate that the NLP systems designed to support the diagnosis of mental disorders benefit from employing both interpretable and hybrid approaches.

7 Conclusion and Future Work

We introduced SMHD-GER, a large dataset of Reddit users with diverse mental health conditions and matched control users. The dataset was created using adaptations of the high-precision diagnostic patterns developed for the original English version (Cohan et al., 2018). Furthermore, we investigated the differences in language use between users with mental health conditions and control groups, as measured by a large set of linguistic and psychological cues. We provided strong benchmark models designed to identify diagnosed users for the four most frequently attested MHC. We found that BLSTM networks trained on within-text distributions of interpretable linguistic features consistently outperformed a Transformer-based model based on GBERT. A hybrid model combining the two approaches proved to be the most effective

method for two of the four conditions. We make our dataset available to the community in the hope that it will encourage further research into these problems and improve the reproducibility of suggested approaches.

8 Limitations

In this work, we have framed mental health detection as a binary classification task that aims to distinguish between individuals with a particular mental disorder and control users. In future work, we intend to frame it as a multi-class classification task to determine the extent to which individual mental disorders can be distinguished from one another.

References

- Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.
- Hani Al-Omari, Malak A. Abdullah, and Samira Shaikh. 2020. Emodet2: Emotion detection in English textual dialogue using BERT and BiLSTM models. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, pages 226–232.
- APA. 2013. Diagnostic and statistical manual of mental disorders. *American Psychiatric Association*, 21(21):591–643.
- David S Baldwin, Dwight L Evans, RM Hirschfeld, and Siegfried Kasper. 2002. Can we distinguish anxiety from depression? *Psychopharmacology Bulletin*, 36:158–165.
- Sven Buechel, Susanna Rücker, and Udo Hahn. 2020. Learning and evaluating emotion lexicons for 91 languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1202–1217, Online. Association for Computational Linguistics.
- Branden Chan, Stefan Schweter, and Timo Möller. 2020. German’s next language model. *arXiv preprint arXiv:2010.10906*.
- Stevie Chancellor and Munmun De Choudhury. 2020. Methods in predictive techniques for mental health status on social media: a critical review. *NPJ digital medicine*, 3(1):1–11.
- Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.
- Arman Cohan, Bart Desmet, Andrew Yates, Luca Soldaini, Sean MacAvaney, and Nazli Goharian. 2018. SMHD: a large-scale resource for exploring online language usage for multiple mental health conditions. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Glen Coppersmith, Mark Dredze, and Craig Harman. 2014. Quantifying mental health signals in twitter. In *Proceedings of the workshop on computational linguistics and clinical psychology: From linguistic signal to clinical reality*, pages 51–60.
- Glen Coppersmith, Mark Dredze, Craig Harman, Kristy Hollingshead, and Margaret Mitchell. 2015. CLPsych 2015 shared task: Depression and PTSD on Twitter. In *Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality*, pages 31–39.
- Jorge Renner Cardoso de Almeida and Mary Louise Phillips. 2013. Distinguishing between unipolar depression and bipolar depression: current and future clinical and neuroimaging perspectives. *Biological psychiatry*, 73(2):111–118.
- Munmun De Choudhury, Sanket S Sharma, Tomaz Logar, Wouter Eekhout, and René Clausen Nielsen. 2017. Gender and cross-cultural differences in social media disclosures of mental illness. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, pages 353–369.
- Peter Deutsch. 1996. Rfc1951: Deflate compressed data format specification version 1.3.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics.
- Simon D’Alfonso. 2020. Ai in mental health. *Current Opinion in Psychology*, 36:112–117.
- Stefan M Gold, Ole Köhler-Forsberg, Rona Moss-Morris, Anja Mehnert, J Jaime Miranda, Monika Bullinger, Andrew Steptoe, Mary A Whooley, and Christian Otte. 2020. Comorbid depression in medical diseases. *Nature Reviews Disease Primers*, 6(1):1–22.
- Sharath Chandra Guntuku, David B Yaden, Margaret L Kern, Lyle H Ungar, and Johannes C Eichstaedt. 2017. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49.
- Keith Harrigan, Carlos Aguirre, and Mark Dredze. 2021. On the state of social media data for mental health research. In *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology: Improving Access*, Online. Association for Computational Linguistics.

- Dirk Hovy and Shannon L Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598.
- Zheng Ping Jiang, Sarah Ita Levitan, Jonathan Zomick, and Julia Hirschberg. 2020. Detection of mental health from reddit via deep contextualized representations. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 147–156.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Herve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.
- Elma Kerz, Yu Qiao, Sourabh Zanwar, and Daniel Wiechmann. 2022. Pushing on personality detection from verbal behavior: A transformer meets text contours of psycholinguistic features. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, Dublin, Ireland. Association for Computational Linguistics.
- Mrinal Kumar, Mark Dredze, Glen Coppersmith, and Munmun De Choudhury. 2015. Detecting changes in suicide content manifested in social media following celebrity suicides. In *Proceedings of the 26th ACM conference on Hypertext & Social Media*, pages 85–94.
- A Downey La Vonne, Leslie S Zun, and Trena Burke. 2012. Undiagnosed mental illness in the emergency department. *The Journal of emergency medicine*, 43(5):876–882.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Kate Loveys, Jonathan Torrez, Alex Fine, Glen Moriarty, and Glen Coppersmith. 2018. Cross-cultural differences in language markers of depression online. In *Proceedings of the fifth workshop on computational linguistics and clinical psychology: from keyboard to clinic*, pages 78–87.
- Xiaofei Lu. 2010. Automatic analysis of syntactic complexity in second language writing. *International journal of corpus linguistics*, 15(4):474–496.
- Xiaofei Lu. 2012. The relationship of lexical richness to the quality of esl learners’ oral narratives. *The Modern Language Journal*, 96(2):190–208.
- Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.
- Justus Mattern, Yu Qiao, Elma Kerz, Daniel Wiechmann, and Markus Strohmaier. 2021. Fang-covid: A new large-scale benchmark dataset for fake news detection in german. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 78–91.
- James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.
- Jürgen Rehm and Kevin D Shield. 2019. Global burden of disease and the impact of mental and addictive disorders. *Current psychiatry reports*, 21(2):1–7.
- Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. SentiWS - a publicly available German-language resource for sentiment analysis. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*, Valletta, Malta. European Language Resources Association (ELRA).
- Paul Rohde, Peter M Lewinsohn, and John R Seeley. 1991. Comorbidity of unipolar depression: II. comorbidity with other mental disorders in adolescents and adults. *Journal of abnormal psychology*, 100(2):214.
- Damian F Santomauro, Ana M Mantilla Herrera, Jamileh Shadid, Peng Zheng, Charlie Ashbaugh, David M Pigott, Cristiana Abbafati, Christopher Adolph, Joanne O Amlag, Aleksandr Y Aravkin, et al. 2021. Global prevalence and burden of depressive and anxiety disorders in 204 countries and territories in 2020 due to the covid-19 pandemic. *The Lancet*, 398(10312):1700–1712.
- Max Schindler and Emese Domahidi. 2022. The computational turn in online mental health research: A systematic review. *New Media & Society*, page 14614448221122212.
- David S Schmidtke, Tobias Schröder, Arthur M Jacobs, and Markus Conrad. 2014. Angst: Affective norms for german sentiment terms, derived from the affective norms for english words. *Behavior research methods*, 46(4):1108–1118.
- Ivan Sekulic and Michael Strube. 2019. Adapting deep learning methods for mental health prediction on social media. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. Association for Computational Linguistics.
- Luke Stark. 2018. Algorithmic psychometrics and the scalable subject. *Social Studies of Science*, 48(2):204–231.
- Simon Šuster, Stéphan Tulkens, and Walter Daelemans. 2017. A short review of ethical challenges in clinical natural language processing. *arXiv preprint arXiv:1703.10090*.

- Anja Thieme, Danielle Belgrave, and Gavin Doherty. 2020. Machine learning in mental health: A systematic review of the hci literature to support the development of effective and implementable ml systems. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 27(5):1–53.
- Daniel Wiechmann, Yu Qiao, Elma Kerz, and Justus Mattern. 2022. Measuring the impact of (psycho-)linguistic and readability features and their spill over effects on the prediction of eye movement patterns. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland. Association for Computational Linguistics.
- Markus Wolf, Andrea B Horn, Matthias R Mehl, Severin Haug, James W Pennebaker, and Hans Kordy. 2008. Computergestützte quantitative textanalyse: äquivalenz und robustheit der deutschen version des linguistic inquiry and word count. *Diagnostica*, 54(2):85–98.
- Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.
- Andrew Yates, Arman Cohan, and Nazli Goharian. 2017. Depression and self-harm risk assessment in online forums. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2968–2978, Copenhagen, Denmark. Association for Computational Linguistics.
- Andrew Yates and Nazli Goharian. 2013. ADRTrace: detecting expected and unexpected adverse drug reactions from user reviews on social media sites. In *European Conference on Information Retrieval*, pages 816–819. Springer.
- Elad Yom-Tov, Evgeniy Gabrilovich, et al. 2013. Post-market drug surveillance without trial costs: discovery of adverse drug reactions through large-scale analysis of web search queries. *Journal of medical Internet research*, 15(6):e2614.
- Sourabh Zanwar, Daniel Wiechmann, Yu Qiao, and Elma Kerz. 2022. The best of both worlds: Combining engineered features with transformers for improved mental health prediction from reddit posts. In *Proceedings of The Seventh Workshop on Social Media Mining for Health Applications, Workshop & Shared Task*, pages 197–202.
- Tianlin Zhang, Annika M Schoene, Shaoxiong Ji, and Sophia Ananiadou. 2022. Natural language processing applied to mental illness detection: a narrative review. *NPJ digital medicine*, 5(1):1–13.

A Appendix

Table 5: Overview of the 117 features investigated in the work.

Feature group	Number of features	Features	Example/Description
Morpho-syntactic	5	MLC MLS C/S CoordP/C BaseKolDef	Mean length of clause (words) Mean length of sentence (words) Clauses per Sentence Coordinate phrases per clause Kolmogorov Complexity
Lexical richness	8	MLWc LD NDW cNDW TTR cTTR rTTR log TTR	Mean length per word (characters) Lexical density Number of different words Corrected number of different words Type-Token Ratio (TTR) Corrected TTR Root TTR Logarithmic TTR
Register-based N-gram	20	Spoken ($n \in [1, 5]$) Fiction ($n \in [1, 5]$) News ($n \in [1, 5]$) Academic ($n \in [1, 5]$)	Frequencies of uni-, bi-, tri-, four-, five-grams from four reference corpora (see appendix Table 6)
LIWC	68	LIWC-German	Pennebaker et al. (2001)
Emotion Lexicon	2	SentiWS	Remus et al. (2010)
	6	ANGST	Schmidtke et al. (2014)
	8	MEmoLon	Buechel et al. (2020)

Table 6: Text corpora used to derive register-specific n-gram frequencies

Register	Corpus	Size		
		Vocab	# Words	Items
Academic	Papers from top 100 German publications	477876	12M	2524 papers
Fiction	Gutenberg project German books	907656	49M	2063 books
News	News articles from FANG-Covid corpus (authentic news) (Mattern et al., 2021)	487841	21M	28056 articles
Spoken	OpenSubtitle dataset	1209934	218M	

Table 7: Descriptive statistics of feature groups 1-5 across MHC.

Feature	Control	ADHD	Anxiety	Bipol	Depres.	OCD	PTSD	Schiz.	other
LD	0.56	0.56	0.51	0.56	0.56	0.56	0.55	0.55	0.56
TTR	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
cTTR	4.27	4.27	4.57	4.28	4.26	4.23	4.30	4.29	4.31
logTTR	0.93	0.94	0.97	0.93	0.93	0.93	0.94	0.94	0.93
rTTR	2.92	2.92	3.17	2.91	2.91	2.89	2.94	2.97	2.94
NDW	18.11	17.90	20.05	17.98	17.98	17.48	18.13	18.45	18.42
cNDW	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
MLW	5.26	5.30	5.27	5.28	5.30	5.29	5.31	5.42	5.32

Continued on next page

Table 7 – continued from previous page

Feature	Control	ADHD	Anxiety	Bipol	Depres.	OCD	PTSD	Schiz.	other
CoordPpC	0.25	0.25	0.18	0.24	0.24	0.25	0.25	0.25	0.24
MLC	7.09	7.07	7.07	7.03	6.99	6.91	7.19	7.27	6.99
MLS	20.00	19.69	21.81	19.76	19.95	19.74	19.83	20.26	20.36
ClpS	2.52	2.47	3.14	2.48	2.54	2.53	2.50	2.57	2.61
KD	1.00	1.00	0.91	1.00	1.00	1.01	0.99	0.99	1.00
uni.Acad	108.74	105.51	129.07	107.04	107.04	109.27	106.57	108.90	109.16
bi.Acad	8.88	8.53	10.98	8.73	8.61	9.39	8.64	8.78	8.80
tri.Acad	0.20	0.20	0.23	0.22	0.21	0.23	0.23	0.20	0.21
four.Acad	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01
uni.Fiction	137.06	133.33	165.51	135.23	135.20	139.62	134.44	139.88	137.58
bi.Fiction	19.30	18.52	27.95	18.95	18.55	23.01	18.19	18.87	19.06
tri.Fiction	0.88	0.83	1.26	0.86	0.81	1.47	0.81	0.78	0.87
four.Fiction	0.03	0.03	0.05	0.03	0.03	0.07	0.03	0.03	0.03
uni.News	135.13	131.42	159.84	133.30	133.25	135.85	132.93	137.82	135.76
bi.News	19.17	18.35	27.74	18.79	18.49	21.03	18.19	19.13	18.87
tri.News	0.92	0.85	1.25	0.90	0.84	1.08	0.83	0.86	0.88
four.News	0.04	0.04	0.04	0.06	0.04	0.05	0.04	0.03	0.04
uni.Spok	160.79	156.36	194.64	158.58	159.34	163.27	157.73	163.88	161.56
bi.Spok	27.26	26.07	41.32	26.61	26.43	32.79	25.65	26.10	26.89
tri.Spok	1.78	1.63	3.30	1.73	1.62	3.19	1.59	1.52	1.69
four.Spok	0.12	0.10	0.22	0.15	0.10	0.26	0.10	0.07	0.11
five.Spok	0.01	0.01	0.04	0.03	0.01	0.02	0.01	0.00	0.01

Table 8: Descriptive statistics of feature scores across MHC.

Feature	Control	ADHD	Anxiety	Bipol	Depres.	OCD	PTSD	Schiz.	other
LIWC.Pronoun	0.08	0.08	0.13	0.08	0.08	0.08	0.08	0.08	0.08
LIWC.I	0.04	0.04	0.08	0.04	0.04	0.04	0.04	0.04	0.04
LIWC.Self	0.04	0.04	0.08	0.04	0.04	0.04	0.04	0.04	0.04
LIWC.You	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.01
LIWC.Other	0.02	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
LIWC.Negate	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.02	0.02
LIWC.Assent	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.01
LIWC.Article	0.08	0.08	0.07	0.08	0.08	0.08	0.08	0.09	0.08
LIWC.Preps	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
LIWC.Affect	0.06	0.05	0.07	0.05	0.05	0.05	0.05	0.05	0.05
LIWC.Posemo	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
LIWC.Posfeel	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LIWC.Optim	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
LIWC.Negemo	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.02	0.01
LIWC.Sad	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00
LIWC.Cogmech	0.10	0.10	0.11	0.10	0.10	0.10	0.10	0.10	0.10
LIWC.Cause	0.02	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.02
LIWC.Insight	0.03	0.03	0.04	0.03	0.03	0.03	0.02	0.03	0.03
LIWC.Discrep	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
LIWC.Inhib	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LIWC.Tentat	0.02	0.02	0.02	0.02	0.02	0.01	0.02	0.02	0.02

Continued on next page

Table 8 – continued from previous page

Feature	Control	ADHD	Anxiety	Bipol	Depres.	OCD	PTSD	Schiz.	other
LIWC.Certain	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
LIWC.Social	0.07	0.07	0.09	0.07	0.07	0.07	0.07	0.07	0.07
LIWC.Comm	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.02	0.02
LIWC.Othref	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
LIWC.Friends	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
LIWC.Humans	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.01	0.01
LIWC.Time	0.04	0.04	0.07	0.04	0.04	0.04	0.04	0.04	0.04
LIWC.Past	0.03	0.03	0.04	0.03	0.03	0.02	0.03	0.03	0.03
LIWC.Present	0.08	0.08	0.11	0.08	0.08	0.08	0.08	0.08	0.08
LIWC.Future	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01
LIWC.Space	0.07	0.07	0.06	0.07	0.07	0.07	0.07	0.07	0.07
LIWC.Up	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
LIWC.Incl	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.05
LIWC.Excl	0.02	0.03	0.02	0.03	0.03	0.03	0.03	0.02	0.03
LIWC.Motion	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
LIWC.Occup	0.05	0.05	0.06	0.05	0.05	0.05	0.05	0.05	0.06
LIWC.School	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
LIWC.Job	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02
LIWC.Achieve	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
LIWC.Leisure	0.01	0.02	0.02	0.01	0.02	0.02	0.02	0.01	0.02
LIWC.Home	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.01
LIWC.Money	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01
LIWC.Metaph	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
LIWC.Physical	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
LIWC.Body	0.01	0.00	0.00	0.01	0.01	0.01	0.01	0.00	0.01
Angst.AROANEW	0.15	0.15	0.14	0.15	0.15	0.13	0.14	0.15	0.15
Angst.AROBAWL	0.08	0.08	0.08	0.08	0.08	0.07	0.08	0.08	0.08
Angst.DOM	0.18	0.17	0.16	0.18	0.17	0.16	0.17	0.16	0.17
Angst.IMA	0.14	0.13	0.12	0.13	0.13	0.12	0.13	0.12	0.13
Angst.POT	0.17	0.17	0.16	0.17	0.17	0.15	0.17	0.17	0.17
Angst.VAL	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.02
MEmoLon.Anger	1.38	1.38	1.43	1.38	1.38	1.38	1.39	1.41	1.38
MEmoLon.Arousal	3.70	3.71	3.81	3.71	3.71	3.72	3.73	3.76	3.71
MEmoLon.Disgust	1.37	1.37	1.42	1.38	1.37	1.38	1.38	1.40	1.37
MEmoLon.Dominance	5.06	5.07	5.18	5.07	5.07	5.08	5.10	5.10	5.06
MEmoLon.Fear	1.40	1.40	1.45	1.40	1.40	1.40	1.41	1.43	1.40
MEmoLon.Joy	1.99	1.99	2.05	1.99	1.99	1.99	2.00	1.99	1.99
MEmoLon.Sadness	1.33	1.33	1.39	1.34	1.34	1.34	1.34	1.36	1.33
MEmoLon.Valence	4.98	4.98	5.09	4.98	4.99	5.00	5.01	4.99	4.98
SentiWS.Pos	0.06	0.06	0.07	0.06	0.07	0.06	0.06	0.06	0.06
SentiWS.Neg	0.06	0.06	0.11	0.06	0.06	0.06	0.06	0.09	0.06

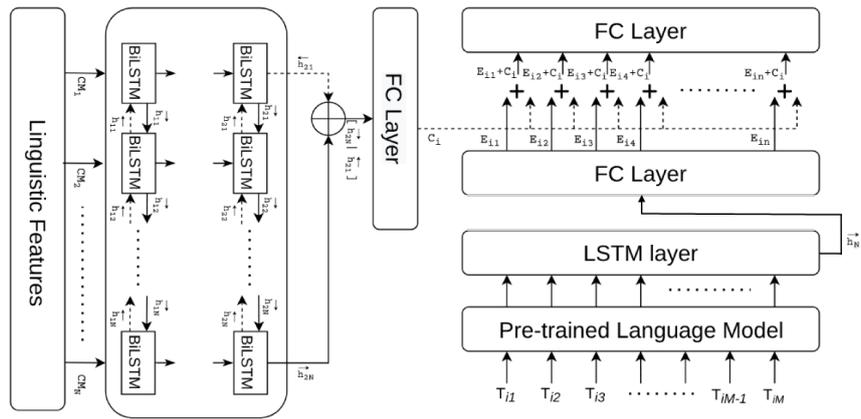


Figure 3: Hybrid model structure

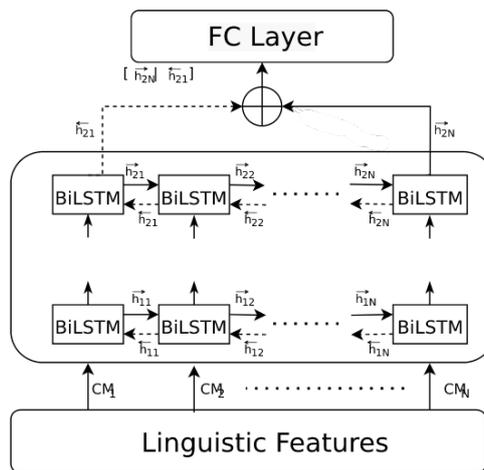


Figure 4: PsyLin model structure

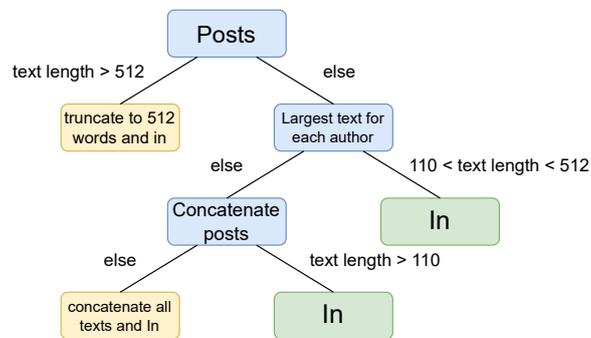


Figure 5: Decision tree for selecting experimental data.

Table 9: Results of MHC prediction experiments (micro scores)

Model type	Metric	Mental Health Condition			
		ADHD	Bipolar	Depression	PTSD
GBERT	Pre	19.12	15.14	11.10	13.24
	Rec	11.43	11.32	4.50	10.07
	F1	14.28	13.29	7.34	11.20
PsyLing	Pre	20.88	19.67	14.81	13.26
	Rec	9.49	11.03	17.18	20.18
	F1	13.34	14.28	15.90	16.00
Hybrid	Pre	13.22	8.45	17.95	15.37
	Rec	22.76	11.84	13.21	12.78
	F1	17.52	10.92	15.72	14.55

Exploring Data Augmentation for Code Generation Tasks

Pinzhen Chen*
School of Informatics
University of Edinburgh
pinzhen.chen@ed.ac.uk

Gerasimos Lampouras
Noah’s Ark Lab
Huawei
gerasimos.lampouras@huawei.com

Abstract

Advances in natural language processing, such as transfer learning from pre-trained language models, have impacted how models are trained for programming language tasks too. Previous research primarily explored code pre-training and expanded it through multi-modality and multi-tasking, yet the data for downstream tasks remain modest in size. Focusing on data utilization for downstream tasks, we propose and adapt augmentation methods that yield consistent improvements in code translation and summarization by up to 6.9% and 7.5% respectively. Further analysis suggests that our methods work orthogonally and show benefits in output code style and numeric consistency. We also discuss test data imperfections.

1 Introduction

Recent years have seen the rapid development of pre-trained models (PLMs) to enable knowledge transfer from generic texts to specific downstream tasks (Devlin et al., 2019; Liu et al., 2019). PLMs have been applied to the programming language domain as well, following the same paradigm of (continuing) training PLMs on code and text data, and then fine-tuning them for specific tasks (Kanade et al., 2020; Feng et al., 2020). PLMs are often adapted to programming languages by including code-specific modalities as part of the input like serialized syntax trees and data flows (Guo et al., 2021, 2022; Tipirneni et al., 2022). Such works have outperformed rule-based tools in various tasks, e.g. the CodeXGLUE benchmark (Lu et al., 2021).

Despite the abundance of raw code available for pre-training, code data that meet downstream needs stay modest in size. This is due to the fact that, unlike texts, code datasets cannot be easily curated by people without programming knowledge. For

example, code translation data in CodeXGLUE is sized at 10K, which is orders of magnitude smaller than their natural language counterparts that often include millions of instances (Kocmi et al., 2022).

We are therefore motivated to enrich data in the fine-tuning phase of code PLMs, using automatic data augmentation (DA) methods like back-translation, monolingual, multilingual, and numeric augmentation. We extensively experiment on code translation, where a programming language is converted to another, and summarization, where a textual description is produced from a code block. Even with limited resources, we can lift performance by 6.9% for translation and 7.5% for summarization compared to baselines. Through manual inspection and extra evaluation measures, we demonstrate that our methods lead to desirable enhancements special to code, namely better output code style and number correctness.

2 Methodology

2.1 Data synthesis

Back-translation (BT, Sennrich et al., 2016) is a data augmentation technique originated from machine translation, where an auxiliary model is used to construct pseudo-parallel data from monolingual resources. It can be straightforwardly applied to code translation. Formally, to train a model $f()$ that converts a programming language PL_x into PL_y , we first train an inverse model $g(PL_y) \rightarrow PL_x$ with the same parallel data. Having the inverse model $g()$, extra monolingual data in PL_y is translated into PL'_x to form pseudo-parallel pairs PL'_x - PL_y that can be used to train $f()$.

For code summarization, back-translation is not applicable as “monolingual” natural language (NL) summaries unaligned to code hardly exist. Hence we propose to use the summaries originally associated with a single programming language as a pivot for other programming languages. After

*Work done during an internship at Huawei Noah’s Ark Lab. Our code will be available at <https://github.com/huawei-noah/noah-research/tree/master/NLP/DA4CodeGeneration>

inverting code-to-text data which has source side code available in multiple programming languages ($PL_1 \rightarrow NL, \dots, PL_n \rightarrow NL$), we train a multilingual text-to-code generator, which outputs a designated programming language given a natural language summary and a target language tag ($NL + tag_{\{1, \dots, n\}} \rightarrow \{PL_1, \dots, PL_n\}$). This generator can iteratively produce code in different PL s by inputting summaries regardless of the original $PL \rightarrow NL$ alignment. These synthesized data, despite having a lower quality, can augment the training data for summarization.

2.2 Utilization of multilinguality

Currey et al. (2017) suggested that including monolingual data in the target language as an additional autoencoding (AE) objective benefits translation models trained on limited data. We migrate this objective to code translation by mixing $PL_x \rightarrow PL_y$ and $PL_y \rightarrow PL_y$ data. This effectively builds a multilingual encoder that enables knowledge transfer, given the high similarity between programming languages, namely the overlap of numerals, syntax tokens, reserved keywords, etc. This process constrains the decoder side to a single programming language PL_y to not add complexity.

In code summarization, as the target NL is fundamentally divergent from the input PL, the autoencoding objective might not be useful. In contrast, we train a “multilingual” code summarization model $\{PL_1, \dots, PL_n\} \rightarrow NL$ where the system takes an arbitrary programming language to produce a natural language summary. Such a many-to-one model allows encoder knowledge sharing too and exposes the decoder to more NL summaries.

2.3 Numeric awareness

Referenced variables and their values are unique components of programming languages; to enhance understanding of these values, previous works on pre-training suggested attending to appropriate modalities, e.g. data flow (Guo et al., 2021). Such sophisticated handling of values might not be necessary for code translation, as copying them over to the target suffices. However, given a small training size, any translation model will still only be exposed to sparse numerical input. To increase model robustness, we augment the data by creating new instances where, in all code tokens containing a number, each digit is randomly replaced with another digit, consistently on both the source and target sides. We do not distinguish

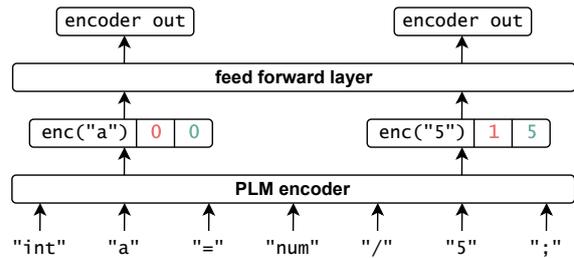


Figure 1: Numeric encoding with a PLM encoder, exemplifying how “a” and “5” are encoded differently.

between purely numerical tokens and tokens including a number. For instance, a variable “num1” could become “num4” in the augmented code pair. The method guarantees that the number-swapped synthetic code is grammatical and compilable.

Apart from numerical augmentation, we propose to include input numbers directly in the encoder output as mathematical values, complementary to their string embedding representations. As illustrated in Figure 1, we append two dimensions to the original encoder output. Particularly, one dimension (red, left) is a binary value (0/1) indicating whether the respective input is a number, while the other dimension (green, right) inherits the input’s value, or 0 if the input is not numeric. The expanded embedding can be reduced to its original size via a feed-forward layer; such a change requires no modification to the pre-trained encoder.

3 Experiments

3.1 Tasks, datasets and evaluation

We benchmark our methods on the code task suite CodeXGLUE (Lu et al., 2021). Its translation task uses code originally developed in Java and then migrated to C#, so the corresponding C#-Java snippets are considered parallel. Training, validation, and test sizes are 10K, 0.5K, and 1K. For back-translation, we translated 377K lines of monolingual Java, albeit out-of-domain, from other CodeXGLUE tasks, into C#. To ensure that the target side consists of genuine data, we only experimented with the C#→Java direction as there is no other C# code in the benchmark for BT.

The summarization task employs CodeSearchNet (Husain et al., 2019) and covers six languages: Ruby, JavaScript, Go, Python, Java, and PHP. Training sizes range from 25K to 250K, totalling 908K; validation and test sets are between 1K and 15K. We performed multilingual back-translation by re-

	BLEU	EM	CodeBLEU [†]
<i>CodeBERT</i>			
paper	72.14	58.0	-
replicate	72.92	57.4	78.93 (72.92 / 73.61 / 87.08 / 82.10)
BT	77.34	61.4	83.36 (77.34 / 78.11 / 90.34 / 87.64)
+ AE	77.60	61.8	83.47 (77.60 / 78.30 / 90.02 / 87.96)
<i>GraphCodeBERT</i>			
paper	72.64	58.8	-
replicate	72.66	58.9	78.55 (72.66 / 73.35 / 87.44 / 80.74)
BT	75.15	60.7	82.13 (75.15 / 75.86 / 90.06 / 87.46)
+ AE	76.15	62.5	82.88 (76.15 / 76.87 / 90.54 / 87.95)

[†]average (n-gram / weighted n-gram / syntax / data flow)

Table 1: Test results for C#→Java translation.

versing the dataset so no external data is introduced; this leads to a five-fold BT data of 4.5M (908K×5). All programming languages share an equal amount of original and synthetic data combined. Moreover, to compare the quality of neural back-translation against hand-written rule-based conversion, we created 80K JavaScript-summary pairs from Python-summary data using `jsbuilder`.

We report code translation results in BLEU-4 (Papineni et al., 2002), exact line matches (EM, in %), and CodeBLEU, a weighted sum of four accuracies: n-grams, weighted n-grams, syntax, and data flow (Ren et al., 2020). Code summarization performance is measured by the de facto choice of BLEU-4 on natural language texts.

3.2 Systems

For all tasks, we use the CodeXGLUE baseline, i.e. CodeBERT with a Transformer decoder, for our base and inverse models (for data synthesis). We continue training PLMs on the augmented data, then fine-tune on the original data, except for numeric augmentation where we mix the synthetic data with the training set. Monolingual and multilingual summarization experiments share the same configurations. For numeric encoding with CodeBERT, we add a feed-forward layer to make the baseline as deep as our proposed network.

To provide results with stronger baselines, we also test with GraphCodeBERT (Guo et al., 2021) for translation and UniXcoder (Guo et al., 2022) for summarization. This helps to verify the stability of data augmentation performance across distinct PLM architectures. We stick to the relevant PLMs’ hyperparameters except for batch size. Model and training details, with links to the preprocessing and evaluation scripts, can be found in Appendix A.

	Ruby	JS	Go	Py	Java	PHP	Avg.
<i>CodeBERT</i>							
paper	12.16	14.90	18.07	19.06	17.65	25.16	17.83
monolingual	12.39	14.13	17.89	18.22	18.66	25.14	17.73
+ rule-trans	-	15.35	-	-	-	-	-
+ BT	13.76	15.00	18.30	18.60	19.64	25.69	18.50
multilingual	14.93	15.53	18.68	18.71	19.70	25.96	18.92
+ rule-trans	14.58	15.65	18.77	18.95	19.86	25.98	18.97
+ BT	14.91	15.81	18.88	18.97	19.69	26.10	19.06
<i>UniXcoder</i>							
paper	14.87	15.85	19.07	19.13	20.31	26.54	19.30
monolingual	14.81	15.28	18.93	19.05	20.22	26.66	19.16
multilingual	15.15	15.64	19.03	19.22	20.45	26.59	19.35
+ BT	14.94	15.85	19.29	19.36	20.43	26.69	19.43

Table 2: Test results for code summarization in BLEU.

3.3 Results and Discussions

We first show translation results in Table 1, where back-translation surpasses baselines by a large margin for both PLMs; on top of it, autoencoding brings a small gain. Table 2 indicates that back-translation also steadily helps code summarization overall. An interesting pattern from both PLMs is that BT helps Ruby and Java less than other languages. Furthermore, learning a single multilingual model is better than learning separate monolingual models, potentially due to transfer learning between programming languages and the increase in natural language data size on the output side.

Table 3 reports the results for numeric augmentation and numeric encoding in translation. Adding number-swapped data to training surpasses the baseline, while our numeric encoding proposal under-performs the baseline. To accommodate the neural network weights which are orders of magnitude smaller than the variable values encountered in code, we investigate linear and logarithmic value scaling. As the scaling gets smaller, result numbers gradually catch up; the optimal is a logarithmic transformation, whereby the model attains the highest performance.

To directly assess our value-aware augmentation, we compute and append output token accuracies to Table 3, with a distinction between numeric and non-numeric tokens. We can observe that the numerical approaches aid number generation without compromising non-numbers, and the improvement in number correctness is generally consistent with the improvement in BLEU and EM. Additional visualization in Appendix B.2 implies that DA models can maintain numeric consistency even when the output is extremely long and complicated.

	BLEU	EM	CodeBLEU	Token Accuracy	
				numeric	non-numeric
<i>CodeBERT</i> + FFN	72.88	58.0	78.07 (72.88 / 73.66 / 86.15 / 79.59)	74.50	86.72
+ numeric augmentation	74.00	59.5	79.43 (74.00 / 74.72 / 87.01 / 82.00)	76.14	87.30
numeric encoding	72.95	58.1	78.77 (72.95 / 73.74 / 86.96 / 81.45)	73.74	86.84
+ numeric augmentation with value scaling					
×10 ²	71.32	51.6	77.71 (71.32 / 72.25 / 86.22 / 81.05)	72.48	85.98
×1 (no scaling)	72.51	57.4	78.45 (72.51 / 73.38 / 86.47 / 81.46)	72.92	86.49
×10 ⁻²	73.48	59.2	79.41 (73.48 / 74.28 / 87.31 / 82.56)	74.11	87.11
×10 ⁻⁴	74.01	58.9	79.73 (74.07 / 74.75 / 87.29 / 82.87)	74.93	87.48
log ₁₀ ()	74.16	59.1	79.84 (74.16 / 74.91 / 87.39 / 82.90)	75.22	87.32

Table 3: Test results for C#→Java translation with numeric augmentation and encoding.

	BLEU	EM	CodeBLEU	Token Accuracy	
				numeric	non-numeric
<i>CodeBERT</i> replicate	72.92	57.4	78.93 (72.92 / 73.61 / 87.08 / 82.10)	74.64	87.54
BT	77.34	61.4	83.36 (77.34 / 78.11 / 90.34 / 87.64)	78.09	88.62
+ num. aug. original only	77.69	61.0	83.44 (77.69 / 78.33 / 90.19 / 87.56)	78.54	88.69
+ num. aug. BT and original	77.37	60.9	83.43 (77.37 / 78.07 / 90.36 / 87.94)	77.16	88.55
BT + AE	77.60	61.8	83.47 (77.60 / 78.30 / 90.02 / 87.96)	77.16	88.64
+ num. aug. original only	77.96	62.0	83.63 (77.96 / 78.62 / 90.15 / 87.82)	78.01	88.79

Table 4: Test results for C#→Java translation with multiple augmentation techniques.

Finally, Table 4 examines if the above methods, namely back-translation and numeric augmentation, work orthogonally. It is observed that better results are achieved when numeric augmentation is applied to the original data, but not to the back-translated data. This is probably because BT is already of inferior quality, so numerical augmentation introduces extra noise. Nevertheless, combining BT and AE with numeric augmentation over the original data leads to the best outcome.

4 Analysis

Upon inspecting the translation test outputs, we find that our data-augmented model is better exposed to the target Java language: it has learned the Java programming conventions instead of following the input code style. We present test instances focused on element retrieval methods, by listing sources, references, and outputs from the CodeBERT baseline and our BT-augmented model in Table 5. Whilst direct retrieval of an element through reference to its position is possible in Java, we observe that the baseline tends to imitate the code style in source C#, but the DA model closely follows the Java coding convention where the inbuilt method `get()` is favoured over directly accessing the attributes by indices.

We should note that in the translation test set a small proportion of code pairs seem to be divergent,

which can lead to an inaccurate estimate of translation performance. We record a few examples of these imperfections in Appendix B.1, but leave in-depth investigation and refinement for future work.

5 Related Works

Recent research at the intersection of natural language processing and programming languages concentrated on pre-training. Kanade et al. (2020) trained CuBERT to obtain embeddings for code understanding tasks. Feng et al. (2020) developed CodeBERT by training RoBERTa on bimodal text-code data with replaced token detection (Clark et al., 2020). In GraphCodeBERT, Guo et al. (2021) incorporated data flow edge prediction and data-variable alignment. Researchers expanded decoder-only models to the code domain too, e.g. CodeGPT, Codex, and Pangu-Coder (Lu et al., 2021; Chen et al., 2021; Christopoulou et al., 2022). Universal encoder-decoder code PLMs have also been presented: PyMT5, CodeT5, PLBART, UniXcoder, and StructCoder (Clement et al., 2020; Wang et al., 2021; Ahmad et al., 2021; Guo et al., 2022; Tipirneni et al., 2022). UniXcoder, which we used, adopts attention masks to control encoder-decoder behaviours in a shared encoder-decoder network.

Datasets for specific tasks concerning code are usually small, so data augmentation can help to boost performance. Roziere et al. (2020) combined

```

// test #85
C# source    ... GetEscherRecord(int index){return escherRecords[index];}
Java reference ... getEscherRecord(int index){return escherRecords.get(index);}
baseline    ... getEscherRecord(int index) {return escherRecords[index];}
DA model    ... getEscherRecord(int index) {return escherRecords.get(index);}

// test #90
C# source    public virtual IQueryNode GetChild(){return GetChildren()[0];}
Java reference public QueryNode getChild() {return getChildren().get(0);}
baseline    public QueryNode getChild() {return getChildren() == 0);}
DA model    public QueryNode getChild() {return getChildren().get(0);}

// test #978
C# source    public virtual SrndQuery GetSubQuery(int qn) { return m_queries[qn]; }
Java reference public SrndQuery getSubQuery(int qn) {return queries.get(qn);}
baseline    public SrndQuery getSubQuery(int qn) {return queries[qn];}
DA model    public SrndQuery getSubQuery(int qn) {return queries.get(qn);}

```

Table 5: C#-Java output translations of element retrieval methods, before and after data augmentation.

cross-lingual masked modelling and iterative back-translation to build an unsupervised code transcompiler. Ahmad et al. (2022) ran code-to-text summarization then text-to-code generation, to obtain translation data. In contrast, we train a text-to-code generation model by reversing the summarization data; our methods differ in both the procedure and the intended task. Also, Yu et al. (2022) crafted rules for source code transformation, whilst our investigation is on automatic neural methods. Finally, techniques like dead code insertion and variable renaming in malware obfuscation (You and Yim, 2010), as well as string manipulation (e.g. token noising, swapping, deletion) can be useful. Nonetheless, these methods are not task-specific, meaning they could be more appropriate for the generic code pre-training stage.

6 Conclusion

We adapt several data augmentation techniques to programming language translation and summarization. Our investigation includes data synthesis, knowledge sharing via multilinguality, and numeric-aware techniques. Enhanced performance is observed in experiments conducted on a variety of pre-trained code language models, and our analysis demonstrates that these methods can benefit output code style and numeric correctness.

7 Limitations

We identify the main limitation to lie in evaluation since we relied on automatic text metrics for both code and text generation. Ideally, code should be treated with software testing practices such as code review, compilation, unit testing, etc. Evaluation is further undermined given the test data issues

revealed in Section 4 and Appendix B.1, so more human analysis should be of interest.

We also do not cover all potential code generation tasks, e.g. code synthesis, where a code snippet is created given a textual description. In this task, the source side carries much less information than the target. We apply a back-translation-style augmentation, but it does not significantly surpass the state-of-the-art PLM. Due to space constraints, we offer some preliminary views in Appendix C.

Acknowledgements

We are grateful to Ignacio Iacobacci for his comments on numeric input scaling, and to the reviewers for their suggestions on qualitative analysis. We also thank the MindSpore team for providing technical support.^{1,2}

Pinzhen Chen is supported by UK Research and Innovation under the UK government’s Horizon Europe funding guarantee [grant number 10052546 – High Performance Language Technologies].

References

- Wasi Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2021. [Unified pre-training for program understanding and generation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2022. [Summarize and generate to back-translate: Unsupervised translation of programming languages](#). *arXiv preprint, abs/2205.11116v1*.

¹<https://www.mindspore.cn/en>

²<https://github.com/mindspore-ai>

- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, ..., and Wojciech Zaremba. 2021. [Evaluating large language models trained on code](#). *arXiv preprint*, abs/2107.03374v2.
- Fenia Christopoulou, Gerasimos Lampouras, Milan Gritta, Guchun Zhang, Yinpeng Guo, Zhong-Yi Li, Qi Zhang, Meng Xiao, Bo Shen, Lin Li, Hao Yu, Li Yan, Pingyi Zhou, Xin Wang, Yu Ma, Ignacio Iacobacci, Yasheng Wang, Guangtai Liang, Jia Wei, ..., and Qun Liu. 2022. [Pangu-coder: Program synthesis with function-level language modeling](#). *arXiv preprint*, abs/2207.11280v1.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [Electra: Pre-training text encoders as discriminators rather than generators](#). In *International Conference on Learning Representations*.
- Colin Clement, Dawn Drain, Jonathan Timcheck, Alexey Svyatkovskiy, and Neel Sundaresan. 2020. [PyMT5: multi-mode translation of natural language and python code with transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. [Copied monolingual data improves low-resource neural machine translation](#). In *Proceedings of the Second Conference on Machine Translation*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.
- Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. 2020. [CodeBERT: A pre-trained model for programming and natural languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*.
- Daya Guo, Shuai Lu, Nan Duan, Yanlin Wang, Ming Zhou, and Jian Yin. 2022. [UniXcoder: Unified cross-modal pre-training for code representation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Daya Guo, Shuo Ren, Shuai Lu, Zhangyin Feng, Duyu Tang, Shujie Liu, Long Zhou, Nan Duan, Alexey Svyatkovskiy, Shengyu Fu, Michele Tufano, Shao Kun Deng, Colin Clement, Dawn Drain, Neel Sundaresan, Jian Yin, Daxin Jiang, and Ming Zhou. 2021. [GraphCodeBERT: Pre-training code representations with data flow](#). In *International Conference on Learning Representations*.
- Hamel Husain, Hongqi Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. [CodeSearchNet challenge: Evaluating the state of semantic code search](#). *arXiv preprint*, abs/1909.09436v3.
- Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2018. [Mapping language to code in programmatic context](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Aditya Kanade, Petros Maniatis, Gogul Balakrishnan, and Kensen Shi. 2020. [Learning and evaluating contextual embedding of source code](#). In *Proceedings of the 37th International Conference on Machine Learning*.
- Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. [Findings of the 2022 conference on machine translation \(WMT22\)](#). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *arXiv preprint*, abs/1907.11692v1.
- Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, ..., and Shujie Liu. 2021. [CodeXGLUE: A machine learning benchmark dataset for code understanding and generation](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.
- Shuo Ren, Daya Guo, Shuai Lu, Long Zhou, Shujie Liu, Duyu Tang, M. Zhou, Ambrosio Blanco, and Shuai Ma. 2020. [CodeBLEU: a method for automatic evaluation of code synthesis](#). *arXiv preprint*, abs/2009.10297v2.
- Baptiste Roziere, Marie-Anne Lachaux, Lowik Chanussot, and Guillaume Lample. 2020. [Unsupervised translation of programming languages](#). In *Advances in Neural Information Processing Systems*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Sindhu Tipirneni, Ming Zhu, and Chandan K. Reddy. 2022. [Structcoder: Structure-aware transformer for code generation](#). *arXiv preprint, abs/2206.05239v1*.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven C.H. Hoi. 2021. [CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Ilsun You and Kangbin Yim. 2010. [Malware obfuscation techniques: A brief survey](#). In *2010 International Conference on Broadband, Wireless Computing, Communication and Applications*.

Shiwen Yu, Ting Wang, and Ji Wang. 2022. [Data augmentation by program transformation](#). *Journal of Systems and Software*, 190:111304.

A Model Configurations

Our training and model configurations are summarized here and in Table 6. We retain the relevant PLMs’ default configurations as much as possible, except for a grid search on the learning rate for code summarization with UniXcoder. We also changed the batch size to utilize our GPUs.

The randomly initialized Transformer decoder attached to CodeBERT and GraphCodeBERT has 6

layers, 12 heads, 768 hidden dimensions, and other hyperparameters as default in PyTorch. For the numeric encoding experiments with CodeBERT, we append 2 dimensions to CodeBERT’s 768d encoder output, then transform it back to 768d using a linear layer. To ensure a fair comparison, a 768d-to-768d layer is added to the baseline to make it as deep.

All experiments are given a fixed budget to run. We save the best checkpoint according to validation BLEU. Results in the paper are based on a single run, but the experiments were benchmarked on PLMs of different architectures to reflect stability.

B More Inspections on Translation Test

B.1 Test imperfections

We show a few translation test instances that are not perfectly parallel in Table 7. In these cases, the code in two languages will not function exactly the same when being executed.

B.2 Numeric consistency

Complementing the number accuracy figures reported in Section 3.3, we list translation outputs containing numbers in Table 9 for visualization. It conveys the idea that our DA models can ensure number consistency even in very long and complicated outputs. In the baseline outputs, for example in test #436, number incorrectness further leads to undesirable hallucinations, which can be prevented in the DA model’s output.

Hyperparameter	Value
PLM checkpoints	CodeBERT: https://huggingface.co/microsoft/codebert-base GraphCodeBERT: https://huggingface.co/microsoft/graphcodebert-base UniXcoder: https://huggingface.co/microsoft/unixcoder-base CodeGPT: https://huggingface.co/microsoft/CodeGPT-small-java-adaptedGPT2 StructCoder: https://github.com/reddy-lab-code-research/structcoder
trainable parameters	CodeBERT: 172.5M + numeric encoding: + 591k GraphCodeBERT: 172.5M UniXcoder: 126.5M CodeGPT: 124.4M StructCoder: 223.4M
learning rate	translation: $5e^{-5}$ summarization: $1e^{-5}$, $5e^{-5}$, $1e^{-6}$, $5e^{-6}$ synthesis: $5e^{-5}$
optimizer	Adam (epsilon= $1e^{-8}$)
training loss	cross-entropy (perplexity)
validation metric	best BLEU
beam size	10
CodeXGLUE	https://github.com/microsoft/CodeXGLUE
jsbuilder	https://github.com/tvst/jsbuilder
javalang	https://github.com/c2nes/javalang

Table 6: Model and training configurations.

```

// test #307
C# source      public override string ToString(){return "IndexSearcher("
                + reader + "; executor=" + executor + ")";}
Java reference public String toString() {return "IndexSearcher("
                + reader + "; executor=" + executor
                + "; sliceExecutionControlPlane " + sliceExecutor + ")";}

// test #518
C# source      public override PushConnection OpenPush() throw
                {new NGit.Errors.NotSupportedException (
                 JGitText.Get().pushIsNotSupportedForBundleTransport);}
Java reference public PushConnection openPush() throws
                {TransportException return new TcpPushConnection();}

// test #892
C# source      public Builder(): base(){lastDocID = -1;wordNum = -1;word = 0;}
Java reference public Builder() {this(true);}

// test #902
C# source      public override string ToString(){return "term="+ term+", field="
                +field+", value="+value;}
Java reference public String toString() {return "term="+term+", field="
                +field+", value="+valueToString()+" ,docIDUpto="+docIDUpto;}

```

Table 7: C#-Java test instances that are not perfectly parallel, with divergence shown in bold.

C Code Synthesis with Augmentation

For code synthesis, while reversing the summarization data is a natural solution, the difficulty lies in forming the class environment (visible and usable variables and methods) because. We parse the code in a summarization instance to obtain positive tokens, as well as randomly sample tokens from other genuine code as negative signals. In other words, from $PL \rightarrow NL$ pairs, we construct code synthesis data $NL + parse(PL) + random(PL') \rightarrow PL$.

We experiment on CodeXGLUE’s code synthesis task, which samples data from CONCODE (Iyer et al., 2018) at 100K/2K/2K for training/validation/test. The source contains a text description as well as the available class variable and function names, and the target is the corresponding Java code. We reverse the Java summary data to create 181K synthetic data; to get available variable and method names, the code is parsed by javalang into tokens. Following CodeXGLUE, we use CodeGPT-adapted as a base model; we further experiment with StructCoder (Tipirneni et al., 2022) which is a more up-to-date code PLM.

The outputs are evaluated by BLEU, EM, and CodeBLEU, similar to translation. Note that the test references are not publicly available, and test predictions need to be sent to the CodeXGLUE authors for evaluation, so we report results on both the validation and test set for reproducibility.

We notice that for CodeGPT, our augmentation

work yields a small gain on validation and test sets. However, it does not improve upon the latest PLM for a few possible reasons: 1) StructCoder is remarkably stronger than CodeGPT, thus the room for improvement is small; 2) the summarization data we used to augment the synthesis task could be different in terms of topic, length, style, etc, resulting in a domain drift.

	BLEU	EM	CodeBLEU
<i>CodeGPT</i> on validation			
replicate	28.13	16.1	31.65
+ augmentation	29.04	16.6	32.35
<i>StructCoder</i> on validation			
replicate	37.30	18.2	40.42
+ augmentation	37.48	18.7	40.47
<i>CodeGPT</i> on test			
paper	32.79	20.1	35.98
replicate	32.66	20.1	35.89
+ augmentation	33.45	19.2	36.47
<i>StructCoder</i> on test			
paper	40.91	22.4	44.77
replicate	41.57	22.6	44.61
+ augmentation	41.32	21.4	44.04

Table 8: Results for code synthesis.

```

// test #131
C# source      public ScaleClusterRequest(): base("CS", "2015-12-15", "ScaleCluster"
                , "cs", "openAPI"){UriPattern = "/clusters/[ClusterId]";
                Method = MethodType.PUT;}
Java reference public ScaleClusterRequest() {super("CS", "2015-12-15", "ScaleCluster"
                , "csk");setUriPattern("/clusters/[ClusterId]");
                setMethod(MethodType.PUT);}
baseline      publicscaleClusterRequest() {super("CS", "2018-12-15", "ScaleCluster"
                , "cs");setUriPattern("/clusters/[ClusterId]");
                setMethod(MethodType.PUT);}
DA model      public ClusterRequest() {super("CS", "2015-12-15", "ScaleCluster"
                , "cs");setUriPattern("/clusters/[ClusterId]");
                setMethod(MethodType.PUT);}

// test #436
C# source      public void CopyTo(byte[] b, int o){FormatHexByte(b, o + 0, w1);
                FormatHexByte(b, o + 8, w2);FormatHexByte(b, o + 16, w3);
                FormatHexByte(b, o + 24, w4);FormatHexByte(b, o + 32, w5);}
Java reference public void copyTo(byte[] b, int o) {formatHexByte(b, o + 0, w1);
                formatHexByte(b, o + 8, w2);formatHexByte(b, o + 16, w3);
                formatHexByte(b, o + 24, w4);formatHexByte(b, o + 32, w5);}
baseline      public void copyTo(byte[] b, int o) {formatHexByte(b, o1);
                formatHexByte(b, o2);formatHexByte(b, o2);
                formatHexByte(b, o3);formatHexByte(b,o + 24, w4);
                formatHexByte(b, o + 32, w5);}
DA model      public void copyTo(int[] b, int o) {formatHexByte(b, o + 0, w1);
                formatHexByte(b, o + 8, w2);formatHexByte(b, o + 16, w3);
                formatHexByte(b, o + 24, w4);formatHexByte(b, o + 32, w5);}

// test #716
C# source      public override void Decode(byte[] blocks, int blocksOffset, int[]
                values, int valuesOffset, int iterations){for (int j = 0;
                j < iterations; ++j){var block = blocks[blocksOffset++];
                values[valuesOffset++] = ((int)((uint)block >> 7)) & 1;
                values[valuesOffset++] = ((int)((uint)block >> 6)) & 1;
                values[valuesOffset++] = ((int)((uint)block >> 5)) & 1;
                values[valuesOffset++] = ((int)((uint)block >> 4)) & 1;
                values[valuesOffset++] = ((int)((uint)block >> 3)) & 1;
                values[valuesOffset++] = ((int)((uint)block >> 2)) & 1;
                values[valuesOffset++] = ((int)((uint)block >> 1)) & 1;
                values[valuesOffset++] = block & 1;}}
Java reference public void decode(byte[] blocks, int blocksOffset, int[]
                values, int valuesOffset, int iterations) {for (int j = 0;
                j < iterations; ++j) {final byte block = blocks[blocksOffset++];
                values[valuesOffset++] = (block >>> 7) & 1;
                values[valuesOffset++] = (block >>> 6) & 1;
                values[valuesOffset++] = (block >>> 5) & 1;
                values[valuesOffset++] = (block >>> 4) & 1;
                values[valuesOffset++] = (block >>> 3) & 1;
                values[valuesOffset++] = (block >>> 2) & 1;
                values[valuesOffset++] = (block >>> 1) & 1;
                values[valuesOffset++] = block & 1;}}
baseline      public void decode(byte[] blocks, int blocksOffset, int[]
                values, int valuesOffset, int iterations) {for (int j = 0;
                j < iterations; ++j) {final byte block = blocks[blocksOffset++];
                values[valuesOffset++] = (block >>> 7) & 1;
                values[valuesOffset++] = (block >>> 6) & 1;
                values[valuesOffset++] = (block >>> 5) & 1;
                values[valuesOffset++] = (block >>> 4) & 1;
                values[valuesOffset++] = (block >>> 4) & 1;
                values[valuesOffset++] = (block >>> 2) & 1;
                values[valuesOffset++] = (block >>> 1) & 1;
                values[valuesOffset++] = block & 1;}}
DA model      public void decode(byte[] blocks, int blocksOffset, int[]
                values, int valuesOffset, int iterations) {for (int j = 0;
                j < iterations; ++j) {final byte block = blocks[blocksOffset++];
                values[valuesOffset++] = (block >>> 7) & 1;
                values[valuesOffset++] = (block >>> 6) & 1;
                values[valuesOffset++] = (block >>> 5) & 1;
                values[valuesOffset++] = (block >>> 4) & 1;
                values[valuesOffset++] = (block >>> 3) & 1;
                values[valuesOffset++] = (block >>> 2) & 1;
                values[valuesOffset++] = (block >>> 1) & 1;
                values[valuesOffset++] = block & 1;}}

```

Table 9: C#.Java output translations containing numbers, before and after data augmentation.

Stabilized In-Context Learning with Pre-trained Language Models for Few Shot Dialogue State Tracking

Derek Chen, Kun Qian, Zhou Yu

Dialogue NLP Lab

Columbia University

{dc3761, kq2157, zy2461}@columbia.edu

Abstract

Prompt-based methods with large pre-trained language models (PLMs) have shown impressive unaided performance across many NLP tasks. These models improve even further with the addition of a few labeled in-context exemplars to guide output generation. However, for more complex tasks such as dialogue state tracking (DST), designing prompts that reliably convey the desired intent is nontrivial, leading to unstable results. Furthermore, building in-context exemplars for dialogue tasks is difficult because conversational contexts are long while model input lengths are relatively short.

To overcome these issues we first adapt a meta-learning scheme to the dialogue domain which stabilizes the ability of the model to perform well under various prompts. We additionally design a novel training method to improve upon vanilla retrieval mechanisms to find ideal in-context examples. Finally, we introduce a saliency model to limit dialogue text length, allowing us to include more exemplars per query. In effect, we are able to achieve highly competitive results for few-shot DST on MultiWOZ.

1 Introduction

Tremendous gains have been made on dialogue state tracking (DST) using large pre-trained language models (PLMs) (Hosseini-Asl et al., 2020; Peng et al., 2021), Fine-tuning such systems though require significant amounts of data, which in turn require substantial effort to collect. Recently, prompting has emerged as a technique for achieving strong performance in a less resource intensive manner (Schick and Schütze, 2021; Liu et al., 2021). Even better performance is possible with in-context exemplars providing a pattern for the model to follow (Brown et al., 2020). Ideally, we should be able to apply these concepts to complex tasks like DST, but results so far have been limited (Madotto et al., 2021).

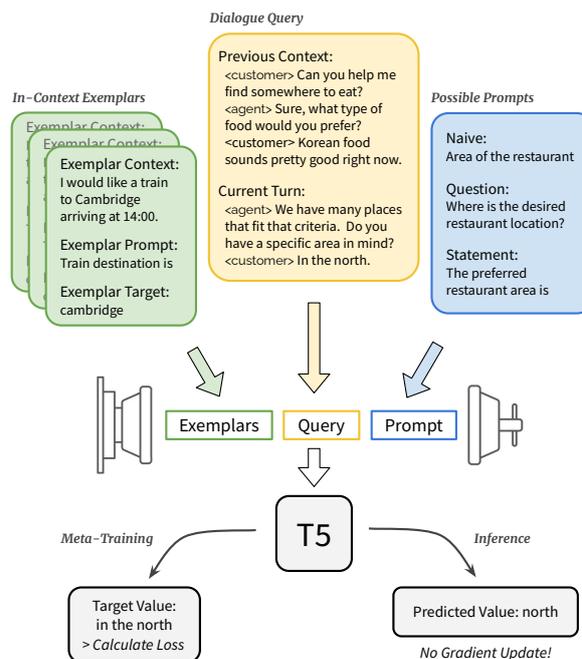


Figure 1: Our system squeezes multiple in-context exemplars, dialogue query with conversational context, and a full prompt into the finite input length of a large PLM to successfully perform few-shot dialogue state tracking, without any need for task-specific training.

One reason for the lack of progress comes from the difficulty of hand-crafting prompts (patterns) and targets (verbalizers), which are highly sensitive to exact phrasing (Lester et al., 2021a). While manually designed prompts have been found to be brittle and unstable (Gu et al., 2021), automatically designed prompts (Gao et al., 2021a) cannot be easily applied to DST since many slots are non-enumerable (Rastogi et al., 2020). A second major hurdle is around dialogue sequence lengths, which are often much longer than those for other tasks (Quan and Xiong, 2020; Kottur et al., 2021) preventing the inclusion of many exemplars for guidance. Full conversations consist of long histories going back many turns, such that the context itself (sans prompt) is already capable of filling

a model’s entire input length. Since state tracking requires carrying over previous dialogue states, naively truncating prior context effectively equates to random guessing (Heck et al., 2020; Kim et al., 2020). A third issue is selecting the exemplars themselves. Prior work recommends choosing a representative example from each class (Gao et al., 2021a), but this is not possible in many cases since most domain-slot-value label combinations simply do not appear in the dataset. Moving to the few-shot scenario further exacerbates this sparsity.

Separately, recall that our main goal is to do well in *few-shot* DST because we purposefully operate in a practical, low-resource data setting. Correspondingly, we aim to achieve good results with a similar low-resource model setting where training should be possible on a single publicly-available commodity server. This precludes the usage of gigantic models such as GPT-3, which are prohibitively expensive to train and bear high economic and environmental costs for inference alone (Strubell et al., 2019; Bender et al., 2021).

We directly tackle each of the three aforementioned issues to achieve state-of-the-art performance on MultiWOZ when restricted to models under 100 billion parameters. To minimize prompt issues, we introduce a meta in-context learning (ICL) framework to stabilize training and reduce variance in prompt performance. To deal with long dialogues, we are inspired by summarization work to condense dialogue histories and filter out non-salient sentences. Our third contribution is designing a novel loss function to train a retrieval model that selects ideal exemplars for priming our downstream model. Our analysis and ablations show that all components help improve our state tracking performance. Finally, we show that unlike other models which only work on specialized LMs, our proposed methods work on any sort of LM, and can be improved with additional training.

2 Related Works

2.1 Few-Shot Dialog State Tracking

Nearly all recent works on dialogue state tracking leverage large pre-trained LMs to achieve good performance (Heck et al., 2020; Kim et al., 2020; Peng et al., 2021). These methods require fine-tuning on large amounts of annotated data, whereas we hope to do well with minimal data.

Few-shot learning can be achieved in many ways, with transfer learning probably being the most pop-

ular, where knowledge is transferred from one domain to another (Wu et al., 2019; Campagna et al., 2020). Data augmentation also supports few-shot learning by generating additional training examples from the few-shot data (Yin et al., 2020; Summerville et al., 2020; Mi et al., 2021). Clustering techniques like prototypical networks have also shown prior success (Snell et al., 2017).

2.2 Meta In-context Learning with Prompting

This work leans on the few-shot techniques of meta-learning (Finn et al., 2017) and prompting with large PLMs (Madotto et al., 2021). Meta-learning allows you to get away with only a few examples at test time by pre-training a model to learn how to learn (Nichol et al., 2018). More recent methods which circumvent the need to calculate second-order gradients (Nichol and Schulman, 2018) have been successfully applied to the task of DST (Dingliwal et al., 2021), but still require fine-tuning on the query set.

Using prompts as natural language instructions have been found to work well on a wide variety of NLP tasks, including dialogue state tracking (Yang et al., 2022). Prompts can be brittle though, so prompt engineering has become its own complex task with numerous ideas on finding discrete prompts (Gao et al., 2021a) or tuning soft prompts, such as through adapters (Xu et al., 2022), prefix tuning (Li and Liang, 2021), or prompt tuning (Lester et al., 2021b). Others have even altered the prompt structure into code in order to fit the capabilities of the network (Lee et al., 2021). Inspired by the success of meta in-context learning on classification tasks (Min et al., 2021; Chen et al., 2022), our work aims to side-step the prompt design issue altogether. Concretely, our method applies meta-learning to teach a model to recognize arbitrary instructions, thereby eliminating the need to rely on domain expertise to craft an optimal prompt.

2.3 Exemplar Retrieval

Lastly, our work is related to retrieval with dense vectors to find good exemplars for in-context learning (Liu et al., 2022). Using dense vectors for similarity search have been applied to dialogue in the past, but mainly in the context of open-domain chat (Adolphs et al., 2021; Komeili et al., 2022) or knowledge-base retrieval (Eric et al., 2017). Lee et al. (2021) is concurrent work which leverages embeddings to search for exemplars in dialogue.

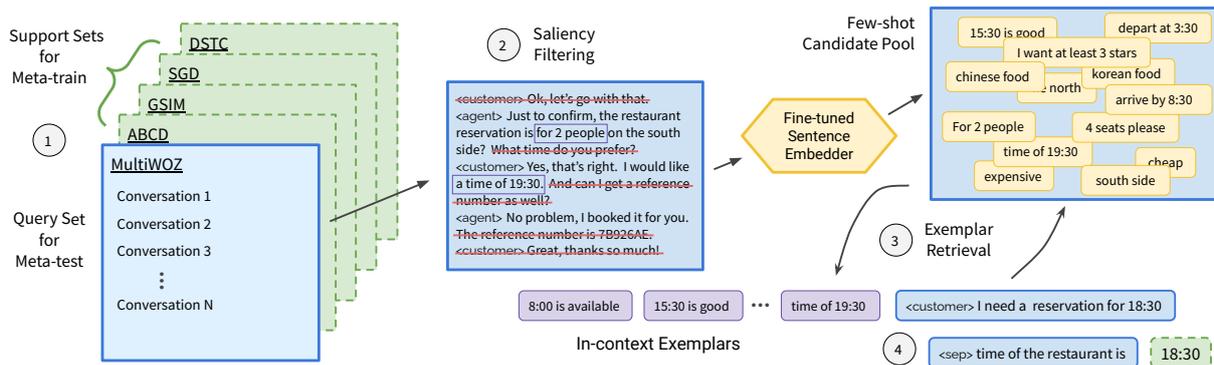


Figure 2: Our method SM2 includes (1) meta-learning with various support sets, (2) saliency filtering to remove irrelevant utterances and (3) improved exemplar retrieval from a few-shot candidate pool. Exemplars are full utterances with dialogue context, which we display as short phrases for illustrative purposes only. They are concatenated and fed into the model for prediction in Step 4. Items in green boxes, including the target value, are only available during meta-training. Purple items are raw text, while yellow ones represent their embedding vectors.

3 Our Method

This section describes our proposal of a Stabilized dialogue state tracker, which leverages Meta in-context learning, dialogue Summarization and a novel Multi-part training loss for fine-tuning a retrieval model, which we refer to as **SM2** for short.

3.1 Preliminaries

The goal of dialogue state tracking (DST) is to extract key information from the conversation as a means of understanding the customer’s intentions in each dialogue turn. More formally, given the dialogue history $H = \{C_1, A_1, C_2, A_2, \dots, C_t\}$ composed of a series of utterances between a customer C_i and an agent A_i , the model should predict the cumulative dialogue state up to current t -th turn. This state is represented as a set of (*domain, slot, value*) tuples, which our system produces by iterating over valid domain-slot pairs and then aggregating all non-null, predicted values for the given turn. A few-shot setup only allows access to $K\%$ of the available labeled data, with $k=[1,5,10]$ for our experiments, where samples are randomly selected from the full labeled dataset. While we compare to models *trained* on k -shot data, our system actually goes a step further since our eventual model receives *no gradient signal* from the task-specific data and instead relies solely on in-context learning to perform inference.

3.2 Stabilized Meta-learning

The intuition behind prompting is that large PLMs understand instructions when written in natural language (Brown et al., 2020). Thus, we write natural

language patterns in an attempt to elicit the dialogue state from the model. However, as previously discussed, minor tweaks in prompt text may cause extreme changes in generated output, leading to highly unstable results (Gu et al., 2021).

Recent works on Meta-ICL (Min et al., 2021; Chen et al., 2022) have shown promise in stabilizing the variance of prompts such that crafting the perfect prompt is no longer necessary, and instead, any reasonable natural language prompt will suffice. Classic meta-learning leverages abundant labeled data from support sets to adapt a model to quickly learn a limited-data target task, denoted as the query set. Finn et al. (2017) proposes MAML that simulates the inner adaptation step during meta-training by conducting a temporary one-step update before computing the loss. Afterwards, a costly second-order gradient is calculated in the outer loop to train the model for faster future adaptations. To get around the expensive loss calculation, variants such as FOMAML have since been developed (Nichol et al., 2018; Nichol and Schulman, 2018). Meta-ICL ingeniously avoids this calculation by replacing the inner adaptation step with in-context learning, which does not require computing gradients! More specifically, in-context learning refers to the use of exemplars to guide the model towards exhibiting ideal behavior. Critically, these exemplars are included as part of the standard model input and thus do not require gradient updates to provide a useful boost.

Following the idea of Meta-ICL, we consider each dataset as a single task and treat MultiWOZ as the held out target task. Specifically, all support datasets are transformed into the DST format for

meta-training, where the in-context inner loop consists of support set training examples. Although the model does not learn about the query set in meta-training, it *is* familiarizing itself with complex DST prompts during that time, allowing it to quickly adapt to the target task in meta-testing. Furthermore, since the prompt meaning is learned during meta-training, theoretically any prompt can be used to instruct the model, including prompts constructed from random tokens (See Table 2).

3.3 Dialogue Compression

Condensing the dialogue context not only fits more exemplars into the model input sequence, but also helps the model focus on more relevant text for predicting dialogue states. We introduce two general ideas under the umbrella of compressing long dialogues into shorter input sequences.

Context Summarization As the task name implies, DST requires tracking dialogue states over long periods of time, including slot-values that were carried over from the start of the conversation. Indeed, initial experiments validated a monotonic decrease in joint goal accuracy as each marginal utterance was removed. Therefore, as an alternative to simply removing prior utterances, we propose summarizing the dialogue history instead. The summary of all prior turns is represented as the predicted dialogue state up to that point, which is represented as a series of (domain, slot, value) tuples. We tried further limiting the input length by only including state tuples directly related to the current slot prediction, but surprisingly found that this formulation of the summary fared worse.

Saliency Filtering Many sentences within a conversation do not contain valuable information, such as "Thanks, that is all I need today." or "Good bye". In order to filter away these lines, the first instinct is to train a large model, but our situation only has access to a few labeled examples, so to keep things simple, we instead gather a small handful of heuristics to identify non-salient utterances. For example, lines that discuss a "reference number" or are excessively terse are targeted for removal. We verify the performance of our heuristics on the limited few-shot examples, where we heavily weight the model's recall of salient utterances over its precision. We take a very conservative approach since accidentally dropping a single relevant sentence can cause a severe penalty in joint goal accuracy.

3.4 Multi-part Retrieval Training

Exemplars are the only guiding signal when dealing with in-context learning, so selecting quality cases is of utmost importance. To do so, we fine-tune the sentence embedder used during retrieval by taking advantage of the limited, few-shot data available.

Exemplar Retrieval Exemplars are retrieved based on their proximity to the query example. Concretely, we first encode all available exemplars into a shared embedding space using a SBERT embedder (Reimers and Gurevych, 2019) where the raw text fed into the embedder is the exemplar's dialogue history. For each incoming query, we encode the instance in the same manner, and then compare their embeddings to rank the closest exemplars in the few-shot candidate pool (Step 3 in Figure 2). Finally, we keep pulling exemplars from the top of the stack to feed into the model until the entire context length of 512 is at capacity. Since the exemplar embeddings are pre-computed, looking for similar exemplars during inference is a very quick operation.

Embedder Fine-tuning To improve the performance of our retrieval model, we explore two categories of training techniques. Inspired by the rise of contrastive learning (Hadsell et al., 2006) as a pre-training method for NLP tasks (Gao et al., 2021b; Karpukhin et al., 2020), we first study a CONTRASTIVE loss which brings positive examples closer together while pushing negative examples further apart. In our case, exemplars sharing the same domain and slot are positive ($Y=0$) while all others are negative ($Y=1$). The loss becomes:

$$\text{Loss}(i, j) = \frac{1 - Y}{2} [\text{dist}(z_i, z_j)]^2 + \frac{Y}{2} \{ \max(0, m - \text{dist}(z_i, z_j)) \}^2$$

where z_i represents the embedding vector for utterance i while m is a margin, set to 1. We explored various distance functions (e.g. euclidean) and found that distance based on cosine similarity worked best:

$$\text{dist}(z_i \cdot z_j) = 1 - \frac{z_i \cdot z_j}{|z_i| \cdot |z_j|}$$

Since we retrieve exemplars based on cosine score, we can directly optimize for this as second technique with a MEAN-SQUARED ERROR loss. More specifically, the positive pair is assigned a target

score of 1 when the two examples share the same domain and slot and 0 otherwise, mirroring the setup of the contrastive loss. The model’s predicted cosine score is then compared against this target to calculate an averaged L2-loss. We generate κ pairs for each of N exemplars, and train our ranker with:

$$L(i, j) = \frac{1}{NK} \sum_{i=1}^N \sum_{j=1}^K \|\text{Target}(i, j) - \text{Pred}(i, j)\|^2$$

Multi-part Modification The standard method for selecting negatives has a few drawbacks since all negatives are treated the same. While this is necessary for unsupervised contrastive learning, our case deals with labeled exemplars. Even binary labels would provide a useful training signal, but we even have varying degrees of similarity. In particular, a positive example would be an exemplar that has a matching domain, slot and value. However, exemplars that contain a matching domain or slot still deserves partial consideration rather than being deemed a pure negative example. Consequently, we introduce a MULTI-CONTRASTIVE loss where the different elements of domain, slot and value are considered positive attributes, weighted with their respective lambdas. These coefficients were chosen by tuning on a held-out development set:

$$\text{Loss}(i, j) = \frac{\lambda_d + \lambda_s + \lambda_v}{4} [\text{dist}(z_i, z_j)]^2 + \frac{\lambda_n}{4} \{\max(0, m - \text{dist}(z_i, z_j))\}^2$$

where:

$$\lambda_d = 3, \quad \lambda_s = 7, \quad \lambda_v = 10 \\ \lambda_n = 1.0, \quad \text{margin} = 1.0$$

For a final loss function, we also test a novel cosine similarity loss where the target label is modified to include multiple parts, MULTI-MSE. The target is altered such that a matching domain for each pair gets $\lambda_d = 0.3$, a matching slot receives another $\lambda_s = 0.3$ boost and matching values get an additional $\lambda_v = 0.4$, where the weights are derived by tuning on the dev set. The final target score is the cumulative sum of the three components - positive pairs sharing all elements get a full score of 1, negative pairs with no matching elements receive a 0, and most pairs lie somewhere in the middle.

$$\text{Target}(i, j) = \sum_e \lambda_e [\mathbb{1}\{e_i = e_j\}], \forall e \in \{d, s, v\} \\ \text{s.t.} \quad \lambda_d + \lambda_s + \lambda_v = 1$$

Dataset	# Dialogs	# Domains	# Slots
MultiWOZ	8,438	7	24
SGD	16,142	16	214
GSIM	1,500	2	13
DSTC2	1,612	1	8
ABCD	8,034	30	231

Table 1: Statistics of involved task-oriented dialogue datasets. Note that the numbers reported are for the training portions for all datasets.

3.5 Model Input

The eventual sequence we feed into the model takes all of the above ideas into account. We start with a context summary represented as the predicted dialogue state, followed by the current turn which consists of two utterances. Each utterance includes a special `<agent>` or `<customer>` token for the respective speaker. Next, a separator token is added, along with a discrete prompt describing the domain and slot. Lastly, we prepend as many exemplars as we can fit into the model maximum token length, truncating from the beginning when necessary. This results in a final model input of:

$[N \text{ exemplars}][\text{prev_dialog_state}][\text{agent_utt}]$
 $[\text{customer_utt}] < \text{sep} > [\text{prompt}][\text{value}]$

Notably, the final `[value]` token is only present during meta-training, and belongs to the support datasets. This value is precisely what we hope to predict when testing the left out query set.

4 Experiments

This section outlines our training implementation details as well as key experiments.

4.1 Training Setup

We consider Schema Guided Dialogue (SGD) (Rastogi et al., 2020), DSTC2 (Henderson et al., 2014), Action-Based Conversations Dataset (ABCD) (Chen et al., 2021), and Google Simulated Chat (GSIM) (Shah et al., 2018) as support sets (listed in Table 1). We then use MultiWOZ 2.1 (Budzianowski et al., 2018; Eric et al., 2019) as a query set, as well as MultiWOZ 2.4 (Zang et al., 2020) which is the cleanest version of MultiWOZ at time of writing. All datasets have dialogue compression techniques applied and use the best performing embedder for exemplar retrieval.

For our training we use T5 (Raffel et al., 2020) with both the three and eleven billion parameters

Prompt Style	Prompt Example
Statement	“The destined location of the taxi is”
Question	“Where is the destination of the taxi ?”
Schema	“<domain> taxi - rent cheap cabs to avoid traffic <slot> destination - what place you want the taxi to take you”
Naive	“destination of the taxi is”
None	“taxi destination”
Random	“blue cobra”

Table 2: Examples for different prompt styles. Here we consider a domain of “taxi” and a slot of “destination”.

versions (T5-3b/T5-11b), where our best models are selected through early stopping on validation data. We set the learning rate as $3e - 4$, employ an Adafactor (Shazeer and Stern, 2018) optimizer and cosine scheduler with warmup of 10,000 steps. Our best system uses an ensemble of exemplar embedders that were trained with of $\kappa = [20, 30, 40]$ and learning rate of $3e - 5$. More details can be found in Appendix C.

4.2 Prompt Variations

Model training can be considered stable if different prompts produce similar outcomes. To test this, we collect six prompts based on common sense and prior work. As much as possible, we use prompts designed by others to avoid biasing the rankings.

Since LMs supposedly operate on prompts as continuation of natural language, the (a) *Statement* prompt takes the form ‘The restaurant cuisine is <blank>’, where we hope the model completes the sentence with the correct slot-value. (b) A *Question* prompt reverses the meaning with ‘What is the restaurant cuisine?’ (c) *Schema* comes from (Lee et al., 2021) and MWOZ 2.2 descriptions, which aims to provide the model with the maximum amount of information. It includes a special token, name, and full description for both the domain and slot. (See Table 2) (d) *Naive* takes the opposite approach by simply following the format of “<slot> of the <domain> is <blank>”. (e) Taken even further, the *None* prompt does not use any natural language at all, instead opting to only include the domain and slot name for evaluation purposes. (f) Finally, we include a *Random* prompt which drops any notion of semantics by replacing the domain with a random color and the slot with a random animal. To empathize with the difficulty of hand-engineering a prompt, note that each option (except for random) seems reasonable, and it is hard to know a priori which one works best.

	MRR@10	NDCG@10	MAP@100
Default	16.7%	9.59%	1.81%
Contrastive	17.4%	10.6%	2.28%
Multi-contrast	17.1%	9.89%	1.90%
Mean Squared	25.1%	15.5%	3.31%
Multi-MSE	26.8%	18.4%	5.24%

Table 3: Results of fine-tuning the sentence embedder with various loss functions. Multi-part cosine is best.

As a baseline, we start with in-context learning without meta-training. We feed in the prompts directly and measure their variance as the standard deviation among scores. Then, we perform meta-learning with all prompts again and measure their results, where we expect that the variance among the scores has now decreased.

4.3 Filtering Threshold

In order to verify that our saliency model successfully removes irrelevant sentences, we employ two experts to annotate 50 dialogs, which is well below the allowed 1% of few-shot data. We then run the saliency model on this tiny evaluation set with different filtering thresholds, ranging from 0.1 to 0.9, with results illustrated in Figure 3. As the threshold increases, only sentences with high relevance are left, as evidenced by high precision and low recall. A maximum F1-score is reached at 0.6, but we would rather keep all relevant sentences at the expense of amassing a handful of irrelevant sentences than to risk missing important information. As a result, we choose 0.4 as the filtering threshold, which achieves a recall of 0.998 and acceptably high precision. Qualitative examples of irrelevant sentences that were removed can be found in section 5.4.

4.4 Retrieval Methods

We adapt SBERT (Reimers and Gurevych, 2019) to our DST task with four different objective functions: standard contrastive loss, multi-part contrastive loss, binary cosine similarity loss and multi-part cosine similarity loss. We test with number of pairs per exemplar in a range from 10 to 100 in increments of ten. We found $\kappa = 30$ to work best, which we use moving forward. As a control, we also include the default SBERT model without any further fine-tuning. We evaluate the results of training on the few-shot examples with Mean Reciprocal Rank (MRR@10), Normalized Discounted Cumulative Gain (NDCG@10) and Maximum Average Precision (MAP@100) as our metrics.

Models	Parameter	MultiWOZ2.1			MultiWOZ2.4		
	Size	1%	5%	10%	1%	5%	10%
TRADE (Wu et al., 2019)	<1B	12.58	31.17	36.18	-	-	-
SGPDST (Lee et al., 2021)		32.11	43.14	46.92	-	-	-
DS2-BART (Shin et al., 2022)		28.25	37.71	40.29	30.55	42.53	41.73
DS2-T5 (Shin et al., 2022)		33.76	44.20	45.38	36.76	49.89	51.05
IC-DST GPT-Neo 2.7b (Hu et al., 2022)	<100B	16.70	26.90	31.65	17.36	29.62	34.38
IC-DST CodeGen 2.7b (Hu et al., 2022)		20.72	29.62	33.81	21.87	33.16	37.45
SM2-3b (Our Method)		38.06	39.94	39.85	37.59	49.22	50.33
- Saliency Filtering		36.11	38.26	38.63	-	-	-
- Context Summarization		37.02	37.83	37.80	-	-	-
- Embedder Fine-tuning		27.15	30.88	31.40	-	-	-
SM2-11b (Our Method)	38.36	44.64	46.02	40.03	51.14	51.97	
IC-DST Codex-davinc 175b (Hu et al., 2022)	>100B	43.13	47.08	48.67	48.35	55.43	56.88

Table 4: DST performance using 1%, 5% and 10% of the training set. Naive prompt used for our method. Bolded numbers indicate highest performance on models under 100 billion parameters. Note that models <1B params fine-tune on task data. Ablation results are also included for dialogue compression and embedder training.

As is shown in Table 3, the multi-part cosine loss showcases the strongest ability to select meaningful exemplars. This shows the benefit of providing partial credit to all elements of the dialogue state. Surprisingly though, the multi-part contrastive loss underperformed. Preliminary error analysis revealed negative examples were successfully separated from positive examples, but the different positive examples were mixed together. We adopt the embedder trained with the MULTI-MSE for all remaining experiments.

5 Results and Analysis

The goal of this work is to achieve strong results on DST without worrying about tedious prompt-engineering. Consequently, we first analyze the ability of the best performing models and then discuss performance stability across different prompts.

5.1 Main Results

Table 4 shows that methods based on in-context learning clearly surpass those based on fine-tuning with few-shot data, as evidenced by the strong performance of SM2 as well as the concurrent work of IC-DST (Hu et al., 2022). In fact, our SM2-11b model is able to achieve the best joint goal accuracy on MultiWOZ 2.1 and 2.4 for most few-shot splits, when focused on models less than 100B parameters. Furthermore, when considering just models operating with in-context learning, SM2-3b greatly outperforms the IC-DST 2.7b models in the same order of magnitude. We note that our method is agnostic to model size, so it is certainly possible to combine them with systems larger than

100B params. Doing so would likely yield strong performance without sacrificing stability.

On that note, Table 5 shows that models trained with SM2 exhibit roughly a 2x reduction in variance over models trained under other regimes. While fine-tuning on certain prompts produces some of the highest scores we observe, other prompts yield some of the lowest, highlighting how hand-crafting prompts are wrought with danger. The instability is most pronounced for the random prompt, which meta-learning is able to smooth over. Also worth noting is that meta-learning from SM2 is able to stabilize prompt performance across multiple model types, including sequence-to-sequence (row 4) or auto-regressive LMs (row 5). This is in contrast to purely in-context models, such as those which were pre-trained on code and must always obey a rigid coding structure during inference.

5.2 Ablation Study

To evaluate the different contributions, we run three ablation experiments, each of which removes one of the key components of SM2. The results presented in Table 4 show that each change makes a noticeable impact. Without saliency filtering, model performance drops by a small, but consistent amount of roughly 1-2%. Disabling context summarization means truncating dialogue history to four utterances and precluding previous dialogue state, which causes an even bigger decrease in accuracy. Using the default SBERT embedder deals the most damage of all, leading to a nearly 10% drop. This suggests that exemplar selection is most critical for in-context learning methods.

Prompt Style	None	Naive	Schema	Statement	Question	Random	STDEV
Fine-Tune	35.3	39.2	38.7	41.1	39.3	24.7	6.02
In-Context	17.5	19.9	14.6	18.9	12.4	4.80	5.58
Pre-train	31.8	35.4	28.2	27.8	34.6	17.2	6.65
SM2 T5-3b	33.9	39.9	30.0	38.2	35.6	33.1	3.58
SM2 GPT-XL	9.70	8.70	8.50	11.4	8.90	1.20	3.53

Table 5: Joint goal accuracy over different prompt styles. Models trained with 5% of training data. The backbone model of Fine-tune and In-Context is T5-3b. Instability is measured as standard deviation of the accuracy scores.

The proposed ideas are also independently applicable to other NLP tasks. For example, compressing inputs to fit more exemplars into an model input sequence can be applied to dialogue generation with large LMs or even reading compression, which requires reasoning over long supporting paragraphs. A multi-part training mechanism can be applied to tasks that contain multiple elements, such as the premise, hypothesis and labels of NLI.

5.3 Additional Discussion

We now turn our attention to the impact of different training regimes, as shown in Table 5. Fine-tuning (row 1) serves as an oracle since it represents training directly on the data in the target domain. Unsurprisingly, SM2 reaches lower average results in comparison. In contrast, SM2 significantly outperforms in-context learning (row 2) since neither perform gradient updates, while SM2 includes a meta-learning stage. Finally, to disentangle the effects of pre-training and meta-ICL, we also compare against a baseline which does not perform in-context learning (row 3). Rather than learning the prompts, this baseline instead simply performs transfer learning from the source datasets to the target dataset. Such a setup does not work as well due to the domain shift from the source distribution to the target distribution.

Digging deeper, we notice that our method displays a meaningful jump in performance when going from 1% to 5% data, but not much when going to 10%. The increased amount of data fails to provide much marginal value since the exemplars being selected did not change much despite choosing from a larger candidate pool. Instead, the finite sequence length became the bottleneck on downstream accuracy.

The performance of the in-context methods are interesting in their own right. Statement prompt does best, while Random does worst, but despite having no training, is well above chance. This surprising result confirms other research on prompt

analysis, which found that large PLMs sometimes perform *too well*, implying that the models are actually paying attention to superficial cues rather than truly understanding the text within a prompt (Webson and Pavlick, 2021; Kavumba et al., 2022).

5.4 Qualitative Analysis

The top half of Table 6 shows an utterance with “*domain=restaurant*” and “*slots=price range, food type*”. Despite having minimal n-gram overlap with the example, the first exemplar E1 receives a high score by matching the same domain and slot of the target utterance. On the other hand, the second exemplar E2 discusses an entirely different topic, producing a low score. This demonstrates the effectiveness of the sentence embedder in distinguishing the value of these exemplars. The bottom half of Table 6 shows how the saliency model successfully conserves a large amount of token space. Short sentences and those void of any dialog state information are safe for removal. When all sentences in an utterance are filtered, then we also remove the associated speaker token. Despite our conservative thresholds, the majority of useless information is successfully trimmed out to allow the model to focus on the most pertinent areas instead.

6 Conclusion

In this paper, we presented a method of performing few-shot dialogue state tracking by leveraging large pre-trained LMs with prompts. Our technique does not require any gradient-based training for the target task and instead relies on in-context learning to guide model generation. To enable success in this low-resource setting, we stabilize training across prompts with Meta-ICL, apply saliency filtering and context summarization to reduce dialogue length, and fine-tune a sentence embedder with a custom loss objective to improve exemplar retrieval. These techniques combined allow us to reach state-of-the-art results on MultiWOZ when limited to models under 100 billion parameters.

Exemplar Retrieval			
Dialog ID	Target Utterance	Exemplar	Score
SSNG0074.json	I am looking for a restaurant in the moderate price range that serves bistro type food .	E1: I would love to help. any particular food you'd like? no , I'd just like for it to be in the east and moderately priced .	0.738
		E2: Seventeen locations meet your criteria. Would you prefer a guesthouse or a hotel? A hotel is fine whichever you recommend.	-0.074
Saliency Filtering			
PMUL0287.json	<Agent>: The phone number is 01223259988. <User>: Perfect. Can you help me with a reservation for 6 people at 14:30 this coming sunday? And please make sure I have a confirmation number to use. <Agent>:our reservation is set!		
PMUL1635.json	<Agent>: What day will you be staying? <User>: Friday and Can you book it for me and get a reference number ?<Agent>:Booking was successful. Reference number is : BMUKPTG6. Can I help you with anything else today? <User>: I am looking to book a train that is leaving from Cambridge to Bishops Stortford on Friday.		

Table 6: Examples of how exemplar retrieval and saliency filtering operate. Same colored text represents matching domain and slots. The strikethrough of text means removal of the irrelevant sentence by the saliency model.

Moving forward, we plan to explore techniques that push model and data efficiency even further. Distillation and pruning can lead to much fewer model parameters, while numerous data augmentation techniques seem promising in maximizing the advantage of limited labeled data. Lastly, rather than meta-learning across different dialog domains, we also would like to explore meta-train model with different prompt styles. With the current framework, the prompt used in inference is required to be the same as the training. However, we might want to use flexible prompts in practice. Consequently, we could meta-train across different prompt styles to allow the model to quickly learn a new prompt style during inference.

7 Limitations

Our method is model-agnostic and can be combined with larger pre-trained model over 100 billion parameters for further improvement on DST task. However, due the budget limit, this is unlikely to be directly validated. Ironically, our method also has the limitation that it cannot be combined with smaller models since the emergent behavior of being to understand prompts only seems to occur with sufficiently large pre-trained models.

Separately, the proposed saliency filtering and the exemplar retrieval module are designed based on the dialog state tracking task, but not specifically for the MultiWOZ dataset. As a result, we planned to apply our framework to other task-oriented dialog datasets, e.g. SGD (Rastogi et al., 2020) to

verify that our framework is generalizable, but have not done so yet due to time constraints. We also ran our experiments with a different model type in GPT-XL, but did not have a chance to properly tune the parameters, leading to low performance.

We would have liked to run our experiments with different random seeds. Considering the stability of our framework among different prompt styles, different random seeds should not cause high variance. However, we still need to run experiments to verify this assumption.

References

- Leonard Adolphs, Kurt Shuster, Jack Urbanek, Arthur Szlam, and Jason Weston. 2021. [Reason first, then respond: Modular generation for knowledge-infused dialogue](#). *CoRR*, abs/2111.05204.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the dangers of stochastic parrots: Can language models be too big?](#) In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21*, page 610–623, New York, NY, USA. Association for Computing Machinery.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). *ArXiv*, abs/2005.14165.
- Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. [Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5016–5026. Association for Computational Linguistics.
- Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. [Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.
- Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. [Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3002–3017, Online. Association for Computational Linguistics.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. [A simple framework for contrastive learning of visual representations](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.
- Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. [Meta-learning via language model in-context tuning](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.
- Saket Dingliwal, Bill Gao, Sanchit Agarwal, Chien-Wei Lin, Tagyoung Chung, and Dilek Z. Hakkani-Tür. 2021. [Few shot dialogue state tracking using meta-learning](#). In *EACL*.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tür. 2019. [Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines](#). *arXiv preprint arXiv:1907.01669*.
- Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. [Key-value retrieval networks for task-oriented dialogue](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.
- Chelsea Finn, P. Abbeel, and Sergey Levine. 2017. [Model-agnostic meta-learning for fast adaptation of deep networks](#). In *ICML*.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. [Making pre-trained language models better few-shot learners](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. [Simcse: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.
- Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. [PPT: pre-trained prompt tuning for few-shot learning](#). *CoRR*, abs/2109.04332.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. [Dimensionality reduction by learning an invariant mapping](#). 2006 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742.
- Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geischauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. [TripPy: A triple copy strategy for value independent neural dialog state tracking](#). In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

- Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL), pages 263–272.
- Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. arXiv preprint arXiv:2005.00796.
- Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking. ArXiv, abs/2203.08568.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16–20, 2020, pages 6769–6781. Association for Computational Linguistics.
- Pride Kavumba, Ryo Takahashi, and Yusuke Oda. 2022. Are prompt-based models clueless? In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2333–2352, Dublin, Ireland. Association for Computational Linguistics.
- Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 567–582, Online. Association for Computational Linguistics.
- Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics.
- Satwik Kottur, Chinnadhurai Sankar, Zhou Yu, and Alborz Geramifard. 2021. DialogStitch: Synthetic deeper and multi-context task-oriented dialogs. In Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 21–26, Singapore and Online. Association for Computational Linguistics.
- Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue state tracking with a language model using schema-driven prompting. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021a. The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021b. The power of scale for parameter-efficient prompt tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582–4597, Online. Association for Computational Linguistics.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In DEELIO.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ArXiv, abs/2107.13586.
- Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems. ArXiv, abs/2110.08118.
- Fei Mi, Wanhao Zhou, Lingjing Kong, Fengyu Cai, Minlie Huang, and Boi Faltings. 2021. Self-training improves pre-training for few-shot learning in task-oriented dialog systems. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 1887–1898. Association for Computational Linguistics.
- Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hananeh Hajishirzi. 2021. Metaicl: Learning to learn in context. ArXiv, abs/2110.15943.
- Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. ArXiv, abs/1803.02999.
- Alex Nichol and John Schulman. 2018. Reptile: a scalable metalearning algorithm. arXiv: Learning.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayan-deh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. Transactions of the Association for Computational Linguistics, 9:807–824.

- Jun Quan and Deyi Xiong. 2020. [Modeling long context for task-oriented dialogue state generation](#). In [Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics](#), pages 7119–7124, Online. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). [Journal of Machine Learning Research](#), 21(140):1–67.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In [Proceedings of the AAAI Conference on Artificial Intelligence](#), volume 34, pages 8689–8696.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In [Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing](#). Association for Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021. [Exploiting cloze-questions for few-shot text classification and natural language inference](#). In [Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume](#), pages 255–269, Online. Association for Computational Linguistics.
- Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. [arXiv preprint arXiv:1801.04871](#).
- Noam M. Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. [ArXiv](#), abs/1804.04235.
- Jamin Shin, Hangeol Yu, Hyeongdon Moon, Andrea Madotto, and Juneyoung Park. 2022. [Dialogue summaries as dialogue states \(DS2\), template-guided summarization for few-shot dialogue state tracking](#). In [Findings of the Association for Computational Linguistics: ACL 2022](#), pages 3824–3846, Dublin, Ireland. Association for Computational Linguistics.
- Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. [ArXiv](#), abs/1703.05175.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and policy considerations for deep learning in NLP](#). In [Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics](#), pages 3645–3650, Florence, Italy. Association for Computational Linguistics.
- Adam Summerville, Jordan Hashemi, James Ryan, and William Ferguson. 2020. [How to tame your data: Data augmentation for dialog state tracking](#). In [Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI](#), pages 32–37, Online. Association for Computational Linguistics.
- Albert Webson and Ellie Pavlick. 2021. [Do prompt-based models really understand the meaning of their prompts?](#) [CoRR](#), abs/2109.01247.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. [Transferable multi-domain state generator for task-oriented dialogue systems](#). In [Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics](#), pages 808–819, Florence, Italy. Association for Computational Linguistics.
- Yan Xu, Etsuko Ishii, Samuel Cahyawijaya, Zihan Liu, Genta Indra Winata, Andrea Madotto, Dan Su, and Pascale Fung. 2022. [Retrieval-free knowledge-grounded dialogue response generation with adapters](#). In [Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering](#), pages 93–107, Dublin, Ireland. Association for Computational Linguistics.
- Yuting Yang, Wenqiang Lei, Juan Cao, Jintao Li, and Tat-Seng Chua. 2022. Prompt learning for few-shot dialogue state tracking. [ArXiv](#), abs/2201.05780.
- Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dialogue state tracking with reinforced data augmentation. In [AAAI](#).
- Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. [Multiwoz 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines](#). [CoRR](#), abs/2007.12720.

A Loss Functions

Gao et al. (2021b) proposes a softmax-based contrastive loss:

$$L_i = -\log \frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^N e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}}$$

which is popular among NLP tasks. However, this loss function requires extremely large batch sizes to work well (Chen et al., 2020). This is especially difficult for us since we specifically target a low-resource setting with small GPU memory requirements. More critically, this softmax contrastive loss views all negatives as being the same. However, in the case of dialog state tracking, where dialog state is represented as (domain, slot, value), the matching is decided at three levels. For example, two dialogue examples can (and should) be considered a negative pair when they have different values for all three elements. In another case though, they might be considered a negative pair by not having matching “value”, but still sharing the same “domain” and “slot”. The softmax contrastive loss considers these two cases as the same, which is not ideal for the DST task. Therefore, we implement the for our experiments. The classic max-margin contrastive loss (Hadsell et al., 2006) is also unable to make a clear distinction for partial credit either, but should be able to when the loss is the sum of multiple elements. Therefore, we use the max-margin loss for our experiments.

B Filtering Results

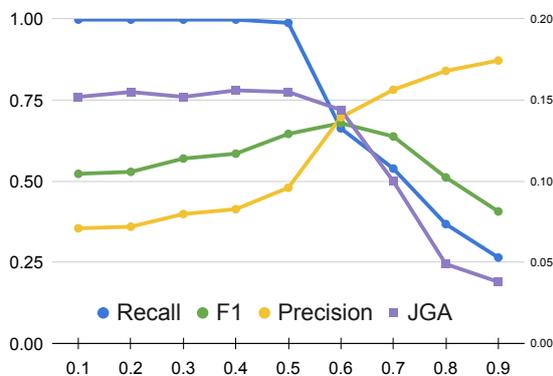


Figure 3: Graph of precision, recall and F1 when varying the acceptance threshold. Joint goal accuracy (JGA) correlates closely with recall due to the nature of DST.

C Other Implementation Details

In this section, we introduce more implementation details. For training, we search the learning rate within the interval [3e-5, 1e-4, 3e-4, 1e-3, 3e-3]. In order to deploy large pre-trained models like T5-3b and T5-11b, we first adjust the batch size. To achieve a balance between GPU memory consumption and batch performance, we alter the number of gradient accumulation steps to maintain a consistent effective batch size of 64 across runs. Furthermore, we also change everything into bitfloat 16 (BF16) and adopt AdaFactor as the optimizer to lower the number of parameters.

We additionally perform ensemble decoding for multiple times using different retrieval embedders. These sentence embedders are distinguished by being trained on different levels of kappa, where we end up choosing embedders trained with kappa of [20,30,40]. These values were selected since they were the models which had the best results as measured by MRR@10 and MAP@10. We run exemplar retrieval with these models and take the majority vote of the system.

In addition to adopting different prompts for our models, we also apply the concept of verbalizers (Schick and Schütze, 2021). More specifically, we use verbalizers to map natural sounding output to the more limited slot-values in the ontology. For example, given the prompt ‘Whether the hotel offers wifi’, we consider both ‘True’ (or ‘False’) and ‘Yes’ (or ‘No’) to be the same answer.

D Input Example

(See next page.)

Exemplar 0 (Truncated)	<pad> options available. Would you like to narrow it down by departure time or arrival time? <customer> I'd like to leave after 21:45, if possible. I won't need to book. I'll just need the arrival time, please? <sep> departure of the train is cambridge</s>
Exemplar 1	taxi destination kamar, taxi departure lovell lodge <agent> when would you like to arrive? <customer> It doesn't matter. I just want to leave there after 10:45 <sep> destination of the taxi is kamar</s>
Exemplar 2	taxi destination riverboat georgina, taxi departure archway house, hotel area north, hotel day thursday, hotel stay 5, hotel people 3, hotel stars 4, attraction name cambridge punter, attraction type boat <agent> what time would you like to leave or arrive by? <customer> I'd like to leave the hotel by 3:15 please. <sep> stars of the hotel is 4</s>
Exemplar 3	train day saturday, train destination cambridge, train departure ely <agent> sure, do you know what time you want to arrive? <customer> I want to arrive by 11:30. <sep> departure of the train is ely</s>
Exemplar 4	restaurant area centre, restaurant people 8, restaurant day thursday, restaurant time 14:00, restaurant food chinese, restaurant price range cheap, taxi destination charlie chan, taxi departure museum of classical archaeology, attraction name museum of classical archaeology <agent> When would you like the leave and arrive by? <customer> I don't mind what time we leave, but I need to arrive at the restaurant by 14:00. <sep> departure of the taxi is museum of classical archaeology</s>
Exemplar 5	restaurant area south, restaurant food asian oriental, restaurant name any, restaurant price range any, train arrive by none, train day wednesday, train destination cambridge, train departure london kings cross, train leave at none, attraction area east <agent> what time were you wanting to leave by or arrive by? <customer> I want to arrive by 12:15. <sep> arrive by of the train is 12:15</s>
Prev State	taxi destination pizza hut fen ditton
Dialog Context	<agent> What time do you want to leave and what time do you want to arrive by? <customer> I want to leave after 17:15.
Prompt	leave at of the taxi is</s>
Label	after 17:15

Table 7: A practical example used during inference which uses our fine-tuned sentence embedder for exemplar retrieval. To be easy to read, we separate each component, including exemplars, query sequence and prompt. Each exemplar contains previous states, dialog context, prompt and label, which corresponds to Sec. 3.5. The 0-th exemplar is truncated so that the entire sequence length can fit into the model.

Can Demographic Factors Improve Text Classification? Revisiting Demographic Adaptation in the Age of Transformers

Chia-Chien Hung^{1,5}, Anne Lauscher², Dirk Hovy³,
Simone Paolo Ponzetto¹ and Goran Glavaš⁴

¹Data and Web Science Group, University of Mannheim, Germany

²Data Science Group, University of Hamburg, Germany

³MilaNLP, Bocconi University, Italy ⁴CAIDAS, University of Würzburg, Germany

⁵NEC Laboratories Europe GmbH, Heidelberg, Germany

{chia-chien.hung, ponzetto}@uni-mannheim.de

anne.lauscher@uni-hamburg.de, dirk.hovy@unibocconi.it

goran.glavas@uni-wuerzburg.de

Abstract

Demographic factors (e.g., gender or age) shape our language. Previous work showed that incorporating demographic factors can consistently improve performance for various NLP tasks with traditional NLP models. In this work, we investigate whether these previous findings still hold with state-of-the-art pretrained Transformer-based language models (PLMs). We use three common specialization methods proven effective for incorporating external knowledge into pretrained Transformers (e.g., domain-specific or geographic knowledge). We adapt the language representations for the demographic dimensions of gender and age, using continuous language modeling and dynamic multi-task learning for adaptation, where we couple language modeling objectives with the prediction of demographic classes. Our results, when employing a multilingual PLM, show substantial gains in task performance across four languages (English, German, French, and Danish), which is consistent with the results of previous work. However, controlling for confounding factors – primarily domain and language proficiency of Transformer-based PLMs – shows that downstream performance gains from our demographic adaptation do *not* actually stem from demographic knowledge. Our results indicate that demographic specialization of PLMs, while holding promise for positive societal impact, still represents an unsolved problem for (modern) NLP.

1 Introduction

Demographic factors like social class, education, income, age, or gender, categorize people into specific groups or populations. At the same time, demographic factors both shape and are reflected in our language (e.g., Trudgill, 2000; Eckert and McConnell-Ginet, 2013). A large body of work focused on modeling demographic language vari-

ation, especially the correlations between words and demographic factors (Bamman et al., 2014; Garimella et al., 2017; Welch et al., 2020, *inter alia*). In a similar vein, Volkova et al. (2013) and Hovy (2015) demonstrated that explicitly incorporating demographic information in language representations improves performance on downstream NLP tasks, e.g., topic classification or sentiment analysis. However, these observations rely on approaches that leverage gender-specific lexica to specialize word embeddings and text encoders (e.g., recurrent networks) that have not been pretrained for (general purpose) language understanding. To date, the benefits of demographic specialization have not been tested with Transformer-based (Vaswani et al., 2017) pretrained language models (PLMs), which have been shown to excel on the vast majority of NLP tasks and even surpass human performance in some cases (Wang et al., 2018).

More recent studies focus mainly on monolingual English datasets and introduce demographic features in task-specific fine-tuning (Voigt et al., 2018; Buechel et al., 2018), which limits the benefits of demographic knowledge to tasks at hand. In this work, we investigate the (task-agnostic) demographic specialization of PLMs, aiming to impart the associations between demographic categories and linguistic phenomena into the PLMs parameters. If successful, such specialization could benefit any downstream NLP task in which demographic factors (i.e., demographically conditioned language phenomena) matter. For this, we adopt intermediate training paradigms that have been proven effective for the specialization of PLMs for other types of knowledge, e.g., in domain, language, and geographic adaptation (Glavaš et al., 2020; Hung et al., 2022a; Hofmann et al., 2022). To this effect, we perform (i) continued language modeling on text corpora produced by a demographic group and (ii)

dynamic multi-task learning (Kendall et al., 2018), wherein we combine language modeling with the prediction of demographic categories.

We evaluate the effectiveness of the demographic PLM specialization on both intrinsic (demographic category prediction) and extrinsic (sentiment classification and topic detection) evaluation tasks across four languages: English, German, French, and Danish, using a multilingual corpus of reviews (Hovy et al., 2015) annotated with demographic information. In line with earlier findings (Hovy, 2015), our initial experiments based on a multilingual PLM (mBERT; Devlin et al., 2019), render demographic specialization effective: we observe gains in most tasks and settings. Through a set of controlled experiments, where we (1) adapt with in-domain language modeling alone, without leveraging demographic information, (2) demographically specialize *monolingual* PLMs of evaluation languages, (3) carry out a meta-regression analysis over dimensions that drive the performance, and (4) analyze the topology of the representation spaces of demographically specialized PLMs, we show, however, that most of the original gains can be attributed to confounding effects of language and/or domain specialization.

Our findings indicate that specialization approaches, proven effective for other types of knowledge, fail to adequately instill demographic knowledge into PLMs, making demographic specialization of NLP models an open problem in the age of large pretrained Transformers. Our research code is publicly available at: <https://github.com/umanlp/SocioAdapt>.

2 Demographic Adaptation

Our goal is to inject demographic knowledge through intermediate PLM training in a task-agnostic manner. To achieve this goal, we train the PLM in a dynamic multi-task learning setup (Kendall et al., 2018), in which we couple masked language modeling (MLM-ing) with predicting the demographic category – gender or age group of the text author. Such multi-task learning setup is designed to force the PLM to learn associations between the language constructs and demographic groups, if these associations are salient in the training corpora.

Masked Language Modeling (MLM). Following successful work on pretraining via language modeling for domain-adaptation (Gururangan et al.,

2020; Hung et al., 2022a), we investigate the effect of running standard MLM-ing on the text corpora of a specific demographic dimension (e.g., gender-related corpora). We compute the MLM loss L_{mlm} in the common way, as negative log-likelihood of the true token probability.

Demographic Category Prediction. In the multi-task learning setup, the representation of the input text, as output by the Transformer, is additionally fed into a classification head that predicts the corresponding demographic category: *age* (below 35 and above 45¹), and *gender* (female and male). The demographic prediction loss L_{dem} is computed as the standard binary cross-entropy loss.

We experiment with two different ways of predicting the demographic category of the text: (i) from the transformed representation of the sequence start token ([CLS]) and (ii) from the contextualized representations of each masked token. We hypothesized that the former variant, in which we predict the demographic class from the [CLS] token representation, would establish links between more complex demographically conditioned linguistic phenomena (e.g., syntactic patterns or patterns of compositional semantics that a demographic group might exhibit), whereas the latter – predicting demographic class from representations of masked tokens – is more likely to establish simpler lexical links, i.e., capture the vocabulary differences between the demographic groups.

Multi-Task Learning. Since both losses can be computed from the same input instances, we opt for joint multi-task learning (MTL) and resort to dynamic MTL based on the *homoscedastic* uncertainty of the losses, wherein the loss variances are used to balance the contributions of the tasks (Kendall et al., 2018). The intuition is that more effective MTL occurs if we dynamically assign less importance to more uncertain tasks, as opposed to assigning uniform task weights throughout the whole training. Homoscedastic uncertainty weighting in MTL has been effective in different NLP settings (Lauscher et al., 2018; Hofmann et al., 2022). In our scenario, L_{mlm} and L_{dem} are measured on different scales in which the model would favor (i.e., be more confident for) one objective than the other. The confidence level of the model prediction for each task would change throughout

¹As suggested by Hovy (2015) the split for the age ranges result in roughly equally-sized data sets for each sub-group and is non-contiguous, avoiding fuzzy boundaries.

the training progress: this makes dynamic weighting desirable. We dynamically prioritize the tasks via homoscedastic uncertainties σ_t :

$$\tilde{L}_t = \frac{1}{2\sigma_t^2} L_t + \log \sigma_t, \quad (1)$$

where σ_t^2 is the variance of the task-specific loss over training instances for quantifying the uncertainty of the task $t \in \{mlm, dem\}$. In practice, we train the network to predict the log variance, $\eta_t := \log \sigma_t^2$, since it is more numerically stable than regressing the variance σ_t^2 , as the log avoids divisions by zero. The adjusted losses are then computed as:

$$\tilde{L}_t = \frac{1}{2}(e^{-\eta_t} L_t + \eta_t). \quad (2)$$

The final loss we minimize is the sum of the two uncertainty-adjusted losses: $\tilde{L}_{mlm} + \tilde{L}_{dem}$.

3 Experimental Setup

Here we describe evaluation tasks and provide details on the data used for demographic specialization and downstream evaluation.

Evaluation Tasks. We follow Hovy (2015) and measure the effects of demographic specialization of PLMs on three text-classification tasks, coupling intrinsic demographic *attribute classification* (AC) with two extrinsic text classification tasks: *sentiment analysis* (SA) and *topic detection* (TD). As an intrinsic evaluation task, AC directly tests if the intermediate demographic specialization results in a PLM that can be more effectively fine-tuned to predict the same demographic classes used in the intermediate specialization: PLMs (vanilla PLM and our demographically specialized counterpart) – are fine-tuned in a supervised fashion to predict the demographic class (gender or age) of the text author. SA is a ternary classification task in which the reviews with ratings of 1, 3, and 5 stars represent instances of *negative*, *neutral*, and *positive* class, respectively. TD classifies texts into 5 different topic categories. We report the F_1 -measure for each task following Hovy (2015).

Data. We carry out our core experimentation on the multilingual demographically labeled dataset of reviews (Hovy et al., 2015), created from the internationally popular user review website Trustpilot.² For comparison and consistency, we work with exactly the same data portions as Hovy (2015):

²<https://www.trustpilot.com/>

collections that cover (1) two most prominent demographic dimensions – *gender* and *age*, with two categories in each (gender: male or female; age: below 35 or above 45³) and (2) five countries (four languages): United States (US), Denmark, Germany, France, and United Kingdom (UK).

To avoid any information leakage, we ensure – for each country-demographic dimension collection (e.g., US, gender) – that there is zero overlap between the portions we select for intermediate demographic specialization and portions used for downstream fine-tuning and evaluation (for AC, SA, and TD). For TD, we aim to eliminate the confounding effect of demographically-conditioned label distributions (e.g., female authors wrote reviews for *clothing store* more frequently than male authors; vice-versa for *electronics & technology*). To this effect, we select, for each country, reviews from the five most frequent topics and sample the same number of reviews in each topic for both demographic groups (i.e., *male* and *female* for gender; *below 35* and *above 45* for age). For the intrinsic AC task (i.e., fine-tuning to predict either gender or age category), we report the results for two different review collections: the first is the set of reviews that have, besides the demographic classes, been annotated with sentiment labels (we refer to this as AC-SA) and the second are the reviews that have topic labels (i.e., product/service category; we refer to this portion as AC-TD). For these fine-tuning and evaluation datasets, we make sure that the two demographic classes (*male* and *female* for gender *under 35* and *above 45* for age) are equally represented in each dataset portion (train, development, and test). Table 1 displays the numbers of reviews for each country, demographic aspect, and dataset portion (specialization vs. fine-tuning).

For intermediate specialization of the multilingual model, we randomly sample 100K instances per demographic group from the *gender* specialization portion and 50K instances each from the texts reserved for *age* specialization concatenated across all 5 countries. For the specialization of monolingual PLMs, we randomly sample the same number of instances but from the specialization portions of a *single* country. Following the established procedure (e.g., Devlin et al., 2019; Liu et al., 2019), we dynamically mask 15% of the tokens in the demographic specialization portions for MLM.

³As suggested by Hovy (2015), the split for the age ranges results in roughly equally-sized data sets for each sub-group and is non-contiguous, avoiding fuzzy boundaries.

Country	Language	gender				age			
		Specialization		SA, AC-SA	TD, AC-TD	Specialization		SA, AC-SA	TD, AC-TD
		F	M	F / M		<35	>45	<35 / >45	
Denmark	Danish	1,596,816	2,022,349	250,485	120,805	833,657	494,905	75,300	44,815
France	French	489,778	614,495	67,305	55,570	40,448	36,182	6,570	6,120
Germany	German	210,718	284,399	28,920	30,580	66,342	47,308	5,865	8,040
UK	English	1,665,167	1,632,894	156,630	183,995	231,905	274,528	26,325	22,095
US	English	575,951	778,877	72,270	61,585	124,924	70,015	6,495	12,090

Table 1: Number of instances in different portions of the Trustpilot dataset (Hovy et al., 2015) used in our experiments. For each country (Denmark, France, Germany, UK, and US), we report the size of the specialization and fine-tuning portions, the latter for each of the two extrinsic tasks: Sentiment Analysis (SA) and Topic Detection (TD). Note that we use the same SA and TD reviews for the intrinsic AC tasks of predicting the demographic categories (denoted AC-SA and AC-TD, respectively). Numbers are shown separately for the two demographic dimensions: gender and age. For fine-tuning datasets (for SA/AC-SA, and for TD/AC-TD), we indicate the number of instances in each category (which is the same for both categories: F and M for gender, <35 and >45 for age). We split the fine-tuning datasets randomly into train, validation, and test portions in the 60/20/20 ratio.

Pre-trained language models. Given that we experiment with Trustpilot data in four different languages, in our core experiments, we resorted to multilingual BERT (mBERT)⁴ (Devlin et al., 2019) as the starting PLM. This allows us to merge the (fairly large) specialization portions of Trustpilot in different languages (see Table 1) and run a single multilingual demographic specialization procedure on the combined multilingual review corpus. We then fine-tune the demographically-specialized mBERT and evaluate downstream task performance separately for each of the five countries (using train, development, and test portions of the respective country). We report the results for two different variants of our dynamic multi-task demographic specialization (DS): (1) when the demographic category is predicted from representations of masked tokens (DS-Tok) and (2) when we predict the demographic category from the encoding of the whole sequence (i.e., review; this version is denoted with DS-Seq). We compare these demographic-specialized PLM variants against two baselines: vanilla PLM and PLM specialized on the same review corpora as our MTL variants but only via MLM-ing (i.e., without providing the demographic signal).

Training and Optimization. In demographic specialization training, we fix the maximum sequence length to 128 subword tokens. We train for 30 epochs in batches of 32 instances and search for the optimal learning rate among the following values: $\{5 \cdot 10^{-5}, 1 \cdot 10^{-5}, 1 \cdot 10^{-6}\}$. We apply early stopping based on the development set performance: we stop if the joint MTL loss does

⁴We load the bert-base-multilingual-cased weights from HuggingFace Transformers.

not improve for 3 epochs). For downstream fine-tuning and evaluation, we train for maximum 20 epochs in batches of 32. We search for the optimal learning rate between the following values: $\{5 \cdot 10^{-5}, 1 \cdot 10^{-5}, 5 \cdot 10^{-6}, 1 \cdot 10^{-6}\}$ and apply early stopping based on the validation set performance (patience: 5 epochs). We use AdamW (Loshchilov and Hutter, 2019) as the optimization algorithm.

4 Results and Discussion

We first discuss the results of multilingual demographic specialization with mBERT as the PLM (§4.1). We then provide a series of control experiments in which we isolate the effects that contribute to performance gains of demographically specialized PLMs (§4.2).

4.1 Multilingual Specialization Results

Table 2 shows the results of gender- and age-specialized mBERT variants – DS-Seq and DS-Tok – on gender and age classification (AC-SA and AC-TD) as intrinsic tasks together with sentiment analysis (SA) and topic detection (TD) as extrinsic evaluation tasks, for each of the five countries encompassed by the Trustpilot datasets (Hovy et al., 2015). The performance of DS-Seq and DS-Tok is compared against the PLM baselines that have not been exposed to demographic information: vanilla mBERT and mBERT with additional MLM-ing on the same Trustpilot data on which DS-Seq and DS-Tok were trained.

Our demographically specialized models generally outperform the vanilla mBERT across the board, both on intrinsic and extrinsic tasks, unsurprisingly with much more prominent gains on the former. The comparison against the domain-

Country	Model	Demographic: <i>gender</i>									Demographic: <i>age</i>								
		Gender class.		SA			TD			Age class.		SA			TD				
		AC-SA	AC-TD	F	M	X	F	M	X	AC-SA	AC-TD	<35	>45	X	<35	>45	X		
Denmark	mBERT	64.0	61.8	69.2	64.8	67.2	59.3	58.3	59.0	57.2	64.5	62.7	62.7	62.9	56.1	52.2	53.4		
	MLM	65.2	63.4	69.5	65.8	67.8	59.7	58.8	59.4	65.5	65.1	63.3	62.1	63.0	57.1	52.6	54.1		
	DS-Seq	64.9	63.5	69.9	65.7	67.7	59.7	57.8	59.1	65.2	65.2	63.1	62.9	63.0	56.9	53.3	54.5		
	DS-Tok	65.0	63.5	69.1	65.6	68.0	59.9	58.9	59.0	65.3	64.6	64.2	63.3	63.2	56.2	53.2	54.3		
Germany	mBERT	59.5	57.9	66.1	63.2	64.5	67.8	65.6	65.8	58.0	56.9	52.6	55.0	55.0	60.1	55.3	57.1		
	MLM	61.2	60.1	67.7	65.3	66.1	68.6	67.0	67.1	61.1	58.9	53.6	55.5	56.7	61.5	56.5	58.7		
	DS-Seq	60.1	60.3	66.7	64.0	65.7	67.6	65.7	66.4	56.4	58.2	53.8	55.3	55.5	60.8	57.6	59.3		
	DS-Tok	62.9	58.3	66.8	64.3	66.8	68.3	67.0	66.7	56.6	57.4	53.0	56.5	56.7	59.3	56.5	59.3		
US	mBERT	62.6	58.1	66.3	64.4	66.0	71.2	68.4	70.2	62.9	60.7	57.7	57.9	57.8	68.0	64.3	64.3		
	MLM	63.3	59.6	67.3	66.2	66.9	72.1	69.4	70.3	63.6	61.9	59.4	57.8	58.2	69.0	64.2	65.2		
	DS-Seq	63.8	59.2	67.2	66.3	67.0	72.3	69.2	70.4	60.7	61.5	59.3	57.9	58.0	69.8	64.4	65.8		
	DS-Tok	62.2	58.8	68.0	66.4	67.3	72.8	69.5	70.5	59.7	61.2	59.9	58.6	57.8	69.2	65.4	64.9		
UK	mBERT	61.9	63.1	71.0	69.0	69.7	70.4	67.9	68.9	65.1	65.2	63.8	63.9	63.7	64.7	67.1	66.3		
	MLM	63.0	65.3	72.0	70.4	71.0	70.6	67.9	69.8	65.4	65.6	62.8	62.0	63.0	65.1	67.3	67.3		
	DS-Seq	63.4	64.9	72.9	70.9	71.7	70.6	68.2	69.8	65.3	62.8	63.8	64.9	64.9	66.0	68.1	66.5		
	DS-Tok	63.5	65.6	73.0	71.0	71.9	70.8	68.2	69.9	64.0	62.8	64.6	65.2	65.1	66.4	67.3	67.6		
France	mBERT	63.9	61.2	69.3	67.0	67.8	44.6	42.4	43.1	55.7	56.6	59.6	57.4	61.5	52.0	47.1	49.0		
	MLM	64.6	62.1	69.9	67.1	68.4	45.8	43.3	44.3	56.8	57.2	59.9	59.5	61.6	52.5	47.2	50.3		
	DS-Seq	64.1	63.1	70.6	67.3	68.4	46.0	43.4	44.2	55.1	55.5	60.4	60.3	62.8	51.1	47.3	50.3		
	DS-Tok	65.0	62.9	70.1	67.5	68.8	45.5	43.9	44.4	54.4	55.9	60.9	59.8	59.7	50.2	48.0	50.8		
Average	mBERT	62.4	60.4	68.4	65.7	67.0	62.7	60.5	61.4	59.8	60.8	59.3	59.4	60.2	60.2	57.2	58.0		
	MLM	63.5	62.1	69.3	67.0	68.0	63.4	61.3	62.2	62.5	61.7	59.8	59.4	60.5	61.0	57.6	59.1		
	DS-Seq	63.3	62.2	69.5	66.8	68.1	63.2	60.9	62.0	60.5	60.6	60.1	60.3	60.8	60.9	58.1	59.3		
	DS-Tok	63.7	61.8	69.4	67.0	68.6	63.5	61.5	62.1	60.0	60.4	60.5	60.7	60.5	60.3	58.1	59.4		

Table 2: Results of gender-specialized (age-specialized) multilingual BERT (DS-Seq and DS-Tok) on gender (age) classification (AC-SA and AC-TD) as intrinsic task and sentiment analysis (SA) and topic detection (TD) as extrinsic evaluation tasks. Comparisons against the vanilla mBERT and mBERT additionally trained on the same review corpora but without demographic information, only with masked language modeling (MLM). For SA and TD, we separately report the performance on the test sets consisting of only one demographic class (gender: F and M, age: <35 and >45) as well as on the mixed test sets containing reviews from both demographic classes (X for both gender and age). Bold numbers indicate the best-performing model (between mBERT, MLM, DS-Seq and DS-Tok) for each country-task combination.

adaptation in which mBERT was intermediately trained only MLM-ed on Trustpilot reviews, but without demographic category prediction, however, reveals that much of the gains that DS-Seq and DS-Tok have over vanilla mBERT stem from domain adaptation: somewhat surprisingly, DS models fall behind MLM-based domain adaptation on the intrinsic tasks of gender/age classification (e.g., for age group classification on AC-SA, the DS variants fall short of MLM by 2 F_1 points), while exhibiting small but fairly consistent gains over MLM for extrinsic SA and TD tasks, both in gender and age intermediate specialization. Although the gains are not particularly convincing, the SA and TD still seem to favor intermediate demographic specialization, which is in line with findings from Hovy (2015), who also reported small but (mostly) consistent gains for these two tasks.

4.2 Control Experiments

To more precisely measure the contributions of demographic information that DS-* variants incorporate, we design further experiments that control for two key side-effects of demographic specialization: (i) language specialization and (ii) domain adaptation. We then carry out the meta-regression analysis to tease out the individual contributions of language, domain, and demographic knowledge on the performance difference between vanilla mBERT and respective intermediately specialized models (mBERT or monolingual BERT specialized on the data of the same or different domain with or without demographic signal). Finally, we compare the representations spaces of the PLMs – before and after demographic specialization – along the demographic dimension.

Controlling for Language Proficiency. Massively multilingual Transformers (MMTs) like mBERT or XLM-R (Conneau et al., 2020) suffer

from the *curse of multilinguality* (Conneau et al., 2020; Lauscher et al., 2020b; Pfeiffer et al., 2020): given a fixed capacity of the Transformer, the representations from an MMT for any individual (high-resource) language will be of lower quality than those of the monolingual PLM, as MMTs share their limited capacity over many languages. It is thus possible that demographic specialization of mBERT on Trustpilot data in our four languages leads to substantial gains over vanilla mBERT (pre-trained on 104 languages) primarily because of mBERT’s acquisition of additional language competencies for these four languages.

To test this, we additionally execute demographic specialization individually for each language (i.e., as opposed to a single multilingual specialization), starting from a monolingual PLM of that language⁵. Monolingual PLMs produce higher quality representations for their respective language than mBERT. Because of this, we hypothesize that subjecting them to demographic specialization on Trustpilot is unlikely to improve their “command” of the language substantially. Consequently, should we still see (downstream) gains from demographic specialization for monolingual PLMs, we can be more confident that they stem from the injected demographic information.

Table 3 shows the effects of demographic specialization on monolingual PLMs of the four languages. For brevity (full results in the Appendix), we average the demographic attribute classification (AC) results from two different test portions from Table 2 (having labels for different downstream tasks, AC-SA and AC-TD); for extrinsic tasks, SA and TD, we report only the score on demographically balanced test sets (denoted “X” in Table 2). The results show that, when we control for language proficiency (as monolingual PLMs are more proficient in their respective language than mBERT), the downstream gains of demographic specialization (on SA and TD) vanish. The DS-Seq and DS-Tok still retain marginal numeric (statistically insignificant) gains over MLM in gender-based specialization, but they lag behind in age-based specialization. Also, both DS-* variants and MLM display only marginal gains with respect to vanilla monolingual BERT models of the four languages: e.g., in gender-specialization and for SA, DS-Tok

⁵We use the following monolingual PLMs from Hugging-Face: bert-base-cased, bert-base-german-cased, dbmdz/bert-base-french-europeana-cased and Maltehb/danish-bert-botxo.

Country	Model	Gender			Age		
		AC	SA	TD	AC	SA	TD
Denmark	BERT	65.0	70.4	59.9	66.5	66.0	56.3
	MLM	65.1	70.3	60.6	67.4	67.6	57.6
	DS-Seq	65.2	70.6	60.0	67.1	67.1	56.5
	DS-Tok	65.1	70.6	60.8	67.2	67.2	56.7
Germany	BERT	59.4	64.3	67.8	58.8	57.1	58.3
	MLM	60.9	65.4	67.7	60.1	58.1	59.9
	DS-Seq	60.1	66.2	67.8	59.8	55.8	59.1
	DS-Tok	60.6	66.0	67.9	58.9	54.0	59.2
US	BERT	61.5	67.1	71.0	64.1	57.2	67.2
	MLM	61.7	67.8	71.3	64.1	60.4	66.7
	DS-Seq	61.6	68.0	71.6	65.2	59.4	67.1
	DS-Tok	62.1	67.9	71.6	64.3	59.4	66.7
UK	BERT	64.1	72.3	70.1	65.8	65.5	68.0
	MLM	64.3	72.6	70.0	66.5	66.9	70.0
	DS-Seq	64.2	72.4	70.2	65.9	67.6	69.4
	DS-Tok	64.1	72.2	70.3	66.0	67.1	69.2
France	BERT	63.6	68.6	45.1	56.5	60.3	49.6
	MLM	64.1	67.6	45.5	56.4	61.6	50.2
	DS-Seq	63.7	69.3	45.3	56.1	62.0	50.2
	DS-Tok	63.7	69.5	45.6	56.3	61.5	50.3
Average	BERT	62.7	68.5	62.8	62.3	61.2	59.9
	MLM	63.2	68.7	63.0	62.9	62.9	60.9
	DS-Seq	62.9	69.3	63.0	62.8	62.4	60.5
	DS-Tok	63.1	69.2	63.2	62.5	61.8	60.4

Table 3: Results of gender/age-specialized **monolingual** PLMs – DS-Seq and DS-Tok – on demographic attribute classification (AC), sentiment analysis (SA) and topic detection (TD). Bold numbers indicate the best-performing model (between BERT, MLM, DS-Seq and DS-Tok) for each country-task combination.

has an average advantage of 0.7 F_1 over the non-specialized vanilla monolingual BERTs; compare this to a gain of 1.6 F_1 points that mBERT-based DS-Tok has over vanilla mBERT (Table 2). These results question the downstream usefulness of demographic specialization – suggested by findings from prior work (Hovy, 2015) and our results for multilingual PLMs (Table 2) – if one starts from the most proficient PLM for the concrete language at hand, i.e., a monolingual PLM.

Controlling for Domain Knowledge. Both simple additional MLM-ing on Trustpilot data, as well as multi-task demographic specialization training (DS-* variants), inject knowledge about the domain-specific language of reviews into the PLM. As shown by previous work (Glavaš et al., 2020; Diao et al., 2021; Hung et al., 2022a), domain adaptation generally leads to better downstream performance on in-domain data for any task. We next investigate to which extent the domain specialization is responsible for performance gains. To this end, we perform demographic specialization

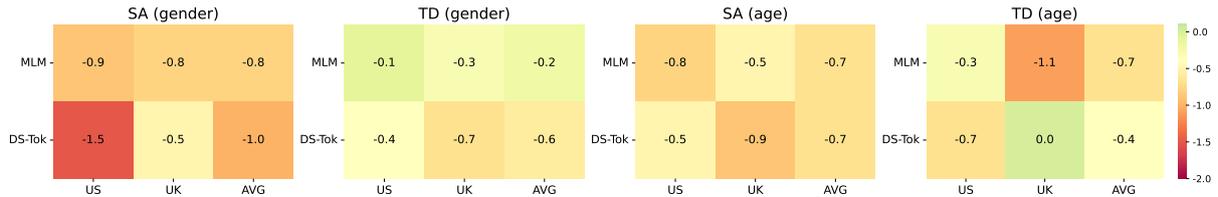


Figure 1: Evaluation results on Trustpilot for Sentiment Analysis (SA) and Topic Detection (TD) when running the intermediate specialization on out-of-domain data (RtGender (Voigt et al., 2018) for *gender* and BAC (Schler et al., 2006) for *age*). We report the delta in F_1 -score in comparison to the specialization on Trustpilot in-domain data.

Task	Selected features	all	-D	-M	-S	-C	-A
<i>gender</i>							
AC-SA	US (1.0); Denmark (0.9); MLM (0.9); DS-Tok (0.9);	0.51	-	0.56	-	0.63	0.62
AC-TD	MLM (1.0); Monoling (1.0) DS-Tok (0.9);	0.51	-	0.73	-	0.54	0.66
SA	France (1.0); DS-Tok (1.0); Denmark (0.8); MLM (0.8); In-domain (0.6)	0.92	0.94	0.95	0.94	0.97	0.98
TD	DS-Tok (0.6); MLM (0.5); In-domain (0.5)	0.33	0.36	0.35	0.34	0.35	0.40
<i>age</i>							
AC-SA	Denmark (3.0); MLM (1.5); Monoling (0.9)	1.93	-	1.98	-	2.31	2.02
AC-TD	UK (2.1); France (1.4); MLM (0.9);	0.68	-	0.69	-	1.02	0.82
SA	In-domain (1.3); DS-Tok (1.0); MLM (0.9);	0.96	1.03	0.97	0.97	0.98	1.03
TD	Denmark (1.6); <35 (0.7); DS-Seq (0.6); DS-Tok (0.6)	1.52	1.53	1.53	1.55	1.61	1.54

Table 4: Results of meta-regression analysis. We report the goodness-of-fit (RMSE) results for predicting deltas in downstream performance between specialized models and their respective vanilla PLM. Results reported for three tasks – intrinsic demographic attribute classification (AC; on datasets AC-SA and AC-TD), Sentiment Analysis (SA), and Topic Detection (TD) with both demographic factors, *gender* and *age*. We compare the results across different feature sets – for all features (**all**), and excluding individual features: domain (**-D**), mono- vs. multilingual (**-M**), fine-tuning demographic setting (e.g., F vs. M vs. X for gender; **-S**), country (**-C**), and the adaptation approach (i.e., MLM vs. DS-Tok vs. DS-Seq; **-A**). For each task, when including all features (column: **in**), we list the most important features, those with weights > 0.5 (*selected features*).

on (demographically labeled) training data from a different domain: for *gender* specialization, we use the RtGender (Voigt et al., 2018) consisting of social media posts collected from diverse sources, whereas for *age* specialization we resort to the Blog Authorship Corpus (BAC; Schler et al., 2006) containing blogposts from blogger.com.

Figure 1 displays the effects of out-of-domain specialization of mBERT on downstream SA and

TD performance (i.e., performance differences w.r.t. corresponding in-domain specialized models). Since RtGender and BAC are English-only datasets, we report the results only for US and UK (for brevity, we report the performance only on the demographically balanced test sets, i.e., setups indicated with “X” in Table 2; both DS-* variants exhibit very similar behavior, so for brevity, we only display results for DS-Tok; complete results are in the Appendix). Expectedly, the out-of-domain specialization deteriorates the downstream performance for both MLM and DS-Tok. Interestingly, MLM, which is not exposed explicitly to the demographic signal in specialization, tends to suffer less from out-of-domain specialization than the gender-informed DS-Tok. In contrast, age-informed DS-Tok seems to exhibit similar losses as MLM due to out-of-domain specialization. These results further question the hypothesis that demographic information guides downstream gains, suggested by prior work (Hovy, 2015) and our in-domain specialization results (with mBERT) from Table 2.

Meta-regression Analysis. Next, we aim to quantify, via a meta-regression analysis, the contributions of individual factors (country, in-domain vs. out-of-domain specialization, language, specialization approach, test set structure) on the task performance (AC-SA, AC-TD, SA, TD). We use the difference in performance between the specialized model and its corresponding vanilla PLM (mBERT or monolingual PLM) as the label (i.e., output, dependent) variable for the regression. We use the following input features (all one-hot encoded) as prediction variables: (i) country/language of fine-tuning/evaluation data, (ii) specialization method (MLM vs. DS-Tok vs. DS-Seq), (iii) in-domain vs. out-of-domain specialization, (iv) whether the starting/vanilla PLM is monolingual (e.g., French BERT) or multilingual (mBERT), (v) and the demographic group from which the fine-tuning/evaluation data comes from (F vs. M vs. X

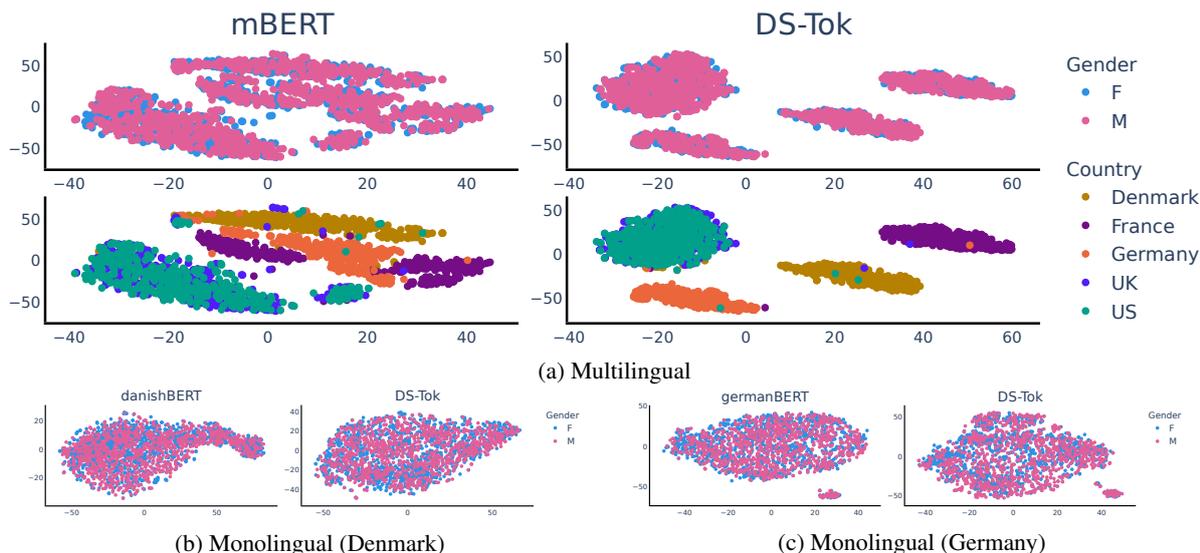


Figure 2: Results of our multilingual and monolingual qualitative analysis for *gender*. For multilingual case as plotted in (a), we show a tSNE visualization of review texts embedded with a non-specialized (mBERT) and specialized (DS-Tok) model. Colors indicate the demographic subgroup (upper figures) and countries (lower figures), respectively. For monolingual case as illustrated in (b) and (c) for Denmark and Germany, we show a tSNE visualization of texts embedded with non-specialized (danishBERT, germanBERT) and specialized (DS-Tok) monolingual PLMs. Each subfigure is plotted with 2K instances.

for gender and <35 vs. >45 vs. X for age). We then fit a linear regressor on all data points, using either the full set of features or, in ablations, excluding certain subsets; we report the goodness of fit as average root mean square error (RMSE).

We summarize the results of our meta-regression analysis in Table 4. For each task, we list the selected features (weights for **in** in parenthesis) paired with the RMSE scores. When we fit regression using all features (**all**), the country of origin of fine-tuning data (i.e., features *Denmark*, *France*, *UK*, etc.) tends to overall explain the variance of specialization effect on model performance as good as or even better than the specialization approach (demographically-informed DS-* variants and demographically-uninformed MLM). The specialization approach features (MLM, DS-Tok, and DS-Seq), however, do appear among the most important features in most settings, suggesting that knowing the specialization approach does help predict the performance of the specialized model. Note, however, that in terms of assessing whether demographic information generally improves specialization, this needs to be combined with actual task performance results from Tables 2 and 3. For example, feature DS-Tok is among the most important features for SA performance after *gender* specialization: looking at the results for DS-Tok in both Tables 2 and 3, we see that it achieves, in most cases, scores above MLM – this, in turn, suggests

that demographically-informed gender specialization does (regardless of other factors) improve the downstream SA performance. The ablation results offer a complementary view into the importance of individual features: the larger the increase in RMSE when excluding a feature (compared to using all features), the more important the feature is. The regressions in which we exclude the information on the specialization approach (**-A**) result in the highest RMSE for gender specialization on both extrinsic tasks (SA and TD). In all other setups (AC for both gender and age specialization, as well as SA and TD for age), there is another type of information, the removal of which results in a less predictable specialization effect: for instance, AC after age specialization, the **-C** setting increases the RMSE the most, representing that features indicating the demographic composition of the fine-tuning dataset – <35 , >45 or balanced (X) – jointly have the largest effect on performance.

Combining results from Tables 2 and 3 with findings from the meta-regression analysis leads to the overall conclusion that gender-based language specialization of PLMs generally leads to downstream gains, whereas age-based specialization does not.

Qualitative Analysis. Finally, we analyze the topology of the PLMs representation space before and after demographic specialization. We encode the reviews from both demographic dimensions –

(i) with the vanilla PLM (mBERT or monolingual BERT) and its DS-Tok specialized counterpart – and then compress those representations into two dimensions with t-distributed stochastic neighbor embedding (tSNE; van der Maaten and Hinton, 2008). Figure 2 depicts these representation spaces after gender-specialization (the age-specialization effects lead to similar conclusions; for brevity, we leave them for the Appendix). The tSNE plots do not show any salient gender specialization effect. In the case of mBERT, gender-specialization (corresponding DS-Tok plot) leads to the separation of representation areas according to review language and not gender of its author.⁶ In the monolingual cases (illustrated for Danish and German BERT), the space of the gender-specialized encoder visually largely resembles that of the vanilla one, indicating that the demographic specialization procedure (DS-Tok) does not impart dimensions that allow for easy separation of representation space along the specialization dimension (here: gender).

5 Related Work

Intermediate Training (Adaptation). Intermediate language modeling on texts from the same or similar distribution as the downstream data has been shown to lead to improvements on various NLP tasks (e.g., Gururangan et al., 2020). During this process, the goal is to inject additional information into the PLM and thus specialize the model for a particular domain (e.g., Aharoni and Goldberg, 2020; Hung et al., 2022a; Bombieri et al., 2023) or language (e.g., Glavaš et al., 2020) or to encode other types of knowledge such as common sense knowledge (e.g., Lauscher et al., 2020a), argumentation knowledge (e.g., Holtermann et al., 2022), or geographic knowledge (e.g., Hofmann et al., 2022).

For instance, Hung et al. (2022a) propose a computationally efficient approach by employing domain-specific adapter modules. They show that their domain adaptation approach leads to improvements in task-oriented dialog. Glavaš et al. (2020) and Hung et al. (2022b) perform language adaptation through intermediate MLM in the target languages with filtered text corpora, demonstrating substantial gains in downstream zero-shot cross-lingual transfer for abusive language detection and dialog tasks, respectively. These specialization approaches mainly rely on a single objective (e.g.,

⁶Note that the green and blue regions, indicating US and UK overlap due to shared language.

masked language modeling on “plain” text data). Instead, Hofmann et al. (2022) conduct geoadaptation by coupling MLM with a token-level geolocation prediction in a dynamic multi-task learning setup. In this work, we adopt a similar approach and perform continued language modeling on the text corpora of a specific demographic dimension.

Demographic Specialization. Language preferences vary with user demographics (Loveys et al., 2018). Accordingly, several studies have leveraged demographic information (e.g., gender, age, education) to investigate the effect of encoded sociodemographic knowledge in the representations of PLMs (Lauscher et al., 2022a) or obtain better language representations for various NLP tasks (Volkova et al., 2013; Garimella et al., 2017). Recent research studies on demographic adaptation mainly focus on (1) learning demographic-aware word embeddings and do not work with large PLMs (Hovy, 2015) or (2) leveraging demographic information with special PLM architectures specifically designed for certain downstream tasks (e.g., empathy prediction (Guda et al., 2021)). The latter, however, do not consider a task-agnostic approach to injecting demographic knowledge into language models, and also focus on a monolingual setup only. Further, what roles the different factors (i.e., domain, language, demographic aspect) in the specialization really play remains unexplored.

6 Conclusion

In this work, we thoroughly examined the effects of demographic specialization of Transformers via straightforward injection methods that have been proven effective for other types of knowledge. Initial results on intrinsic and extrinsic evaluation tasks using a multilingual PLM indicated the usefulness of our approach. However, running a series of additional experiments in which we controlled for potentially confounding factors (language and domain) and a meta-analysis indicated that the demographic aspects only have a negligible impact on the downstream performance. This observation is supported by additional qualitative analysis. Overall, our findings point to the difficulty of injecting demographic knowledge into Transformers: we hope that our in-depth analysis and findings catalyze future research on the topic of truly human-centered NLP, especially in multilingual settings.

Acknowledgements

Chia-Chien Hung and Simone Paolo Ponzetto have been supported by the JOIN-T 2 project of the Deutsche Forschungsgemeinschaft (DFG). The work of Anne Lauscher and Dirk Hovy has been funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No. 949944, INTEGRATOR). Anne Lauscher has been additionally supported by the Excellence Strategy of the German Federal Government and the Länder. Dirk Hovy is the Scientific director of the Data and Marketing Insights research unit at the Bocconi Institute for Data Science and Analysis. Goran Glavaš has been supported by the EUINACTIION grant from NORFACE Governance (462-19-010, GL950/2-1). We thank the reviewers for their feedback.

Limitations

In this paper, we concentrated on the demographic adaptation of PLMs for a few key demographic aspects (i.e., gender and age). There are other known factors, like ethnicity and education, that we cannot explore here. However, there are likely further effects, as well as intersectional effects. We conducted our experiments using only five Western countries and four Indo-European languages (Hovy et al., 2015), ignoring other world regions and language families. However, due to the scarcity of data, we can only hypothesize that the limited effects of demographic specialization also apply to resource-lean languages (i.e., the language specialization effects are likely to outweigh the ones of the demographic specialization). Another limitation is the use of pretrained language models, which are all pre-trained on general-purpose data and are freely available. We acknowledge that results may differ for models with greater capacity that have been pretrained on data from other, more specific domains. We primarily concentrate on BERT-like models, which are only a subset of language models, and we leave language model variants for future research.

Ethics Statement

Our work deals with demographic adaptation from reviews that should be considered sensitive information. We acknowledge that the limitations in data resources and annotations (Schler et al., 2006;

Hovy et al., 2015; Voigt et al., 2018) give rise to potential risks of overgeneralizing our findings and applying our methods. These risks are due to: (1) *partial language coverage*, where languages are from Indo-European subfamilies that do not represent typologically diverse languages; (2) *limited cultural coverage* (Joshi et al., 2020), where the countries, although speaking different languages, still belong a culturally relatively homogeneous part of the world, i.e., the West; (3) *simplified gender identities* (Dev et al., 2021), where gender is modeled as a binary variable, which does not reflect the wide variety of possible identities along the gender spectrum and beyond (Lauscher et al., 2022b); (4) *unfair stereotypical biases* (Blodgett et al., 2020), namely potential harms that might arise from unfair stereotypical biases in the data (despite our efforts to balance the sample across demographic groups) or pre-encoded in the model (Lauscher et al., 2021). Further, the sensitive user profile data might bias the model towards additional demographic characteristics and lead to potentially harmful predictions and applications.

In this work, however, we are interested in advancing NLP research to understand better this fine-grained aspect of the intertwined relationship between demographic adaptation and large pretrained language models in both monolingual and multilingual scenarios. While limited data resources may hinder our ability to fully consider language coverage, cultural coverage, gender identities, and stereotypical biases, it is our obligation to be transparent about these limitations and ethical concerns and to continually work towards improving data collection and methodologies to better serve the needs and perspectives of all users. We believe these insights will lead us toward fairer and more inclusive language technologies. We hope that future research builds on top of our findings and explores other demographic factors, other groups within these factors, and also other languages and countries.

References

- Roei Aharoni and Yoav Goldberg. 2020. *Unsupervised domain clusters in pretrained language models*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.
- David Bamman, Chris Dyer, and Noah A. Smith. 2014. *Distributed representations of geographically situated language*. In *Proceedings of the 52nd Annual*

- Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 828–834, Baltimore, Maryland. Association for Computational Linguistics.
- Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. [Language \(technology\) is power: A critical survey of “bias” in NLP](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.
- Marco Bombieri, Marco Rospocher, Simone Paolo Ponzetto, and Paolo Fiorini. 2023. [Machine understanding surgical actions from intervention procedure textbooks](#). *Comput. Biol. Medicine*, 152:106415.
- Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and João Sedoc. 2018. [Modeling empathy and distress in reaction to news stories](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4758–4765, Brussels, Belgium. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Sunipa Dev, Masoud Monajatipoor, Anaelia Ovalle, Arjun Subramonian, Jeff Phillips, and Kai-Wei Chang. 2021. [Harms of gender exclusivity and challenges in non-binary representation in language technologies](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1994, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Shizhe Diao, Ruijia Xu, Hongjin Su, Yilei Jiang, Yan Song, and Tong Zhang. 2021. [Taming pre-trained language models with n-gram representations for low-resource domain adaptation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3336–3349, Online. Association for Computational Linguistics.
- Penelope Eckert and Sally McConnell-Ginet. 2013. *Language and gender*. Cambridge University Press.
- Aparna Garimella, Carmen Banea, and Rada Mihalcea. 2017. [Demographic-aware word associations](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2285–2295, Copenhagen, Denmark. Association for Computational Linguistics.
- Goran Glavaš, Mladen Karan, and Ivan Vulić. 2020. [XHate-999: Analyzing and detecting abusive language across domains and languages](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6350–6365, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Bhanu Prakash Reddy Guda, Aparna Garimella, and Niyati Chhaya. 2021. [EmpathBERT: A BERT-based framework for demographic-aware empathy prediction](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3072–3079, Online. Association for Computational Linguistics.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don’t stop pretraining: Adapt language models to domains and tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Valentin Hofmann, Goran Glavaš, Nikola Ljubešić, Janet B Pierrehumbert, and Hinrich Schütze. 2022. [Geographic adaptation of pretrained language models](#). *arXiv preprint arXiv:2203.08565*.
- Carolin Holtermann, Anne Lauscher, and Simone Ponzetto. 2022. [Fair and argumentative language modeling for computational argumentation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7841–7861, Dublin, Ireland. Association for Computational Linguistics.
- Dirk Hovy. 2015. [Demographic factors improve classification performance](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 752–762, Beijing, China. Association for Computational Linguistics.
- Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. [User review sites as a resource for large-scale sociolinguistic studies](#). In *Proceedings of the 24th international conference on World Wide Web*, pages 452–461.
- Chia-Chien Hung, Anne Lauscher, Simone Paolo Ponzetto, and Goran Glavaš. 2022a. [DS-TOD: Efficient domain specialization for task-oriented dialog](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 891–904, Dublin, Ireland. Association for Computational Linguistics.

- Chia-Chien Hung, Anne Lauscher, Ivan Vulić, Simone Ponzetto, and Goran Glavaš. 2022b. [Multi2WOZ: A robust multilingual dataset and conversational pre-training for task-oriented dialog](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3687–3703, Seattle, United States. Association for Computational Linguistics.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. [The state and fate of linguistic diversity and inclusion in the NLP world](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.
- Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. [Multi-task learning using uncertainty to weigh losses for scene geometry and semantics](#). In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.
- Anne Lauscher, Federico Bianchi, Samuel R. Bowman, and Dirk Hovy. 2022a. [SocioProbe: What, when, and where language models learn about sociodemographics](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7901–7918, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Anne Lauscher, Archie Crowley, and Dirk Hovy. 2022b. [Welcome to the modern world of pronouns: Identity-inclusive natural language processing beyond gender](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1221–1232, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Anne Lauscher, Goran Glavaš, Simone Paolo Ponzetto, and Kai Eckert. 2018. [Investigating the role of argumentation in the rhetorical analysis of scientific publications with neural multi-task learning models](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3326–3338, Brussels, Belgium. Association for Computational Linguistics.
- Anne Lauscher, Tobias Lueken, and Goran Glavaš. 2021. [Sustainable modular debiasing of language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4782–4797, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. 2020a. [Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers](#). In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 43–49, Online. Association for Computational Linguistics.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020b. [From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Kate Loveys, Jonathan Torrez, Alex Fine, Glen Moriarty, and Glen Coppersmith. 2018. [Cross-cultural differences in language markers of depression online](#). In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, pages 78–87, New Orleans, LA. Association for Computational Linguistics.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. [MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.
- Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. [Effects of age and gender on blogging](#). In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*.
- Peter Trudgill. 2000. *Sociolinguistics: An introduction to language and society*. Penguin UK.
- Laurens van der Maaten and Geoffrey Hinton. 2008. [Visualizing data using t-sne](#). *Journal of Machine Learning Research*, 9(86):2579–2605.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Rob Voigt, David Jurgens, Vinodkumar Prabhakaran, Dan Jurafsky, and Yulia Tsvetkov. 2018. [RtGender: A corpus for studying differential responses to gender](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. [Exploring demographic language variations to improve multilingual sentiment analysis in social media](#). In *Proceedings of the 2013 Conference on*

Empirical Methods in Natural Language Processing, pages 1815–1827, Seattle, Washington, USA. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Charles Welch, Jonathan K. Kummerfeld, Verónica Pérez-Rosas, and Rada Mihalcea. 2020. [Compositional demographic word embeddings](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4076–4089, Online. Association for Computational Linguistics.

A Additional Experiments

Country	Model	Gender class.				SA						TD					
		AC-SA		AC-TD		F	M	X	F	M	X	F	M	X	F	M	X
		Mono	Multi	Mono	Multi	Mono			Multi			Mono			Multi		
Denmark	BERT	66.1	64.0	63.8	61.8	72.3	67.9	70.4	69.2	64.8	67.2	60.7	59.8	59.9	59.3	58.3	59.0
	MLM	66.0	65.2	64.2	63.4	72.5	68.3	70.3	69.5	65.8	67.8	60.6	60.6	60.6	59.7	58.8	59.4
	DS-Seq	66.2	64.9	64.1	63.5	72.6	68.6	70.6	69.9	65.7	67.7	61.3	60.5	60.0	59.7	57.8	59.1
	DS-Tok	66.0	65.0	64.1	63.5	72.4	68.4	70.6	69.1	65.6	68.0	61.1	60.2	60.8	59.9	58.9	59.0
Germany	BERT	59.8	59.5	58.9	57.9	66.5	63.7	64.3	66.1	63.2	64.5	67.9	66.1	67.8	67.8	65.6	65.8
	MLM	62.0	61.2	59.7	60.1	68.1	65.8	65.4	67.7	65.3	66.1	68.5	66.7	67.7	68.6	67.0	67.1
	DS-Seq	61.1	60.1	59.0	60.3	68.8	64.4	66.2	66.7	64.0	65.7	68.9	66.4	67.8	67.6	65.7	66.4
	DS-Tok	60.9	62.9	60.3	58.3	67.9	65.6	66.0	66.8	64.3	66.8	68.6	66.8	67.9	68.3	67.0	66.7
US	BERT	64.3	62.6	58.7	58.1	68.6	67.0	67.1	66.3	64.4	66.0	72.5	69.7	71.0	71.2	68.4	70.2
	MLM	64.6	63.3	58.7	59.6	68.4	67.6	67.8	67.3	66.2	66.9	73.1	70.1	71.3	72.1	69.4	70.3
	DS-Seq	64.3	63.8	58.8	59.2	68.6	68.0	68.0	67.2	66.3	67.0	73.1	70.3	71.6	72.3	69.2	70.4
	DS-Tok	64.7	62.2	59.4	58.8	68.9	67.5	67.9	68.0	66.4	67.3	73.3	69.9	71.6	72.8	69.5	70.5
UK	BERT	63.2	61.9	65.0	63.1	73.4	71.0	72.3	71.0	69.0	69.7	71.2	69.1	70.1	70.4	67.9	68.9
	MLM	63.7	63.0	64.8	65.3	73.9	71.0	72.6	72.0	70.4	71.0	71.2	69.4	70.0	70.6	67.9	69.8
	DS-Seq	63.2	63.4	65.2	64.9	73.6	72.2	72.4	72.9	70.9	71.7	71.5	69.3	70.2	70.6	68.2	69.8
	DS-Tok	63.3	63.5	64.8	65.6	73.7	72.0	72.2	73.0	71.0	71.9	71.4	69.1	70.3	70.8	68.2	69.9
France	BERT	64.1	63.9	63.1	61.2	70.5	67.3	68.6	69.3	67.0	67.8	46.0	44.5	45.1	44.6	42.4	43.1
	MLM	64.9	64.6	63.2	62.1	71.0	67.7	67.6	69.9	67.1	68.4	46.2	44.3	45.5	45.8	43.3	44.3
	DS-Seq	64.2	64.1	63.1	63.1	70.5	67.5	69.3	70.6	67.3	68.4	47.1	44.2	45.3	46.0	43.4	44.2
	DS-Tok	64.4	65.0	62.9	62.9	71.7	68.3	69.5	70.1	67.5	68.8	46.9	44.3	45.6	45.5	43.9	44.4
Average	BERT	63.5	62.4	61.9	60.4	70.3	67.4	68.5	68.4	65.7	67.0	63.7	61.8	62.8	62.7	60.5	61.4
	MLM	64.2	63.5	62.1	62.1	70.8	68.1	68.7	69.3	67.0	68.0	63.9	62.2	63.0	63.4	61.3	62.2
	DS-Seq	63.8	63.3	62.0	62.2	70.8	68.1	69.3	69.5	66.8	68.1	64.4	62.1	63.0	63.2	60.9	62.0
	DS-Tok	63.9	63.7	62.3	61.8	70.9	68.4	69.2	69.4	67.0	68.6	64.3	62.1	63.2	63.5	61.5	62.1

Table 5: Evaluation results compared with monolingual BERT and multilingual BERT (mBERT) on five countries with *gender* data for intrinsic attribute classification tasks (AC-SA, AC-TD) and extrinsic evaluation tasks: sentiment analysis (SA) and topic detection (TD).

Country	Model	Age class.				SA						TD					
		AC-SA		AC-TD		<35	>45	X									
		Mono	Multi	Mono	Multi	Mono			Multi			Mono			Multi		
Denmark	BERT	67.7	57.2	65.3	64.5	67.3	66.2	66.0	62.7	62.9	62.9	58.4	54.4	56.3	56.1	52.2	53.4
	MLM	67.4	65.5	67.4	65.1	67.7	67.3	67.6	63.3	62.1	62.9	59.3	55.3	57.6	57.1	52.6	54.1
	DS-Seq	67.4	65.2	66.8	65.2	67.4	66.2	67.1	63.1	62.9	63.0	58.7	55.0	56.5	56.9	53.3	54.5
	DS-Tok	67.8	65.3	66.6	64.6	67.6	66.1	67.2	64.2	63.3	63.2	59.0	55.4	56.7	56.2	53.2	54.3
Germany	BERT	57.9	58.0	59.6	56.9	53.6	57.9	57.1	52.6	55.0	55.0	61.6	57.4	58.3	60.1	55.3	57.1
	MLM	58.1	61.1	62.0	58.9	58.1	58.2	58.1	53.6	55.5	56.7	62.2	57.6	59.9	61.5	56.5	58.7
	DS-Seq	58.2	56.4	61.3	58.2	56.3	57.3	55.8	53.8	55.3	55.5	63.5	57.9	59.1	60.8	57.6	59.3
	DS-Tok	57.2	56.6	60.6	57.4	57.9	58.1	54.0	53.0	56.5	56.7	63.5	58.2	59.2	59.3	56.5	59.3
US	BERT	65.2	62.9	63.0	60.7	60.5	58.7	57.2	57.7	57.9	57.8	68.8	64.9	67.2	68.0	64.3	64.3
	MLM	65.3	63.6	62.9	61.9	60.5	59.5	60.4	59.4	57.8	58.2	71.2	65.7	66.7	69.0	64.2	65.2
	DS-Seq	66.2	60.7	64.1	61.5	61.6	58.3	59.4	59.3	57.9	58.0	72.5	65.5	67.1	69.8	64.4	65.8
	DS-Tok	65.7	59.7	62.9	61.2	61.1	58.7	59.4	59.9	58.6	57.8	69.4	65.7	66.7	69.2	65.4	64.9
UK	BERT	65.7	65.1	65.8	65.2	65.2	66.3	65.5	63.8	63.9	63.7	68.1	68.1	68.0	64.7	67.1	66.3
	MLM	66.9	65.4	66.1	65.6	68.2	67.2	66.9	62.8	62.0	63.0	68.8	70.1	70.0	65.1	67.3	67.3
	DS-Seq	67.0	65.3	64.7	62.8	67.8	66.4	67.6	63.8	64.9	64.9	67.8	68.9	69.4	66.0	68.1	66.5
	DS-Tok	66.8	64.0	65.2	62.8	67.6	66.5	67.1	64.6	65.2	65.1	68.2	69.6	69.2	66.4	67.3	67.6
France	BERT	56.0	55.7	57.0	56.6	59.7	57.5	60.3	59.6	57.4	61.5	51.9	49.1	49.6	52.0	47.1	49.0
	MLM	55.9	56.8	56.9	57.2	60.7	59.4	61.6	59.9	59.5	61.6	53.8	48.5	50.2	52.5	47.2	50.3
	DS-Seq	55.5	55.1	56.7	55.5	61.3	58.7	62.0	60.4	60.3	62.8	53.8	49.0	50.2	51.1	47.3	50.3
	DS-Tok	55.8	54.4	56.7	55.9	60.2	60.7	61.5	60.9	59.8	59.7	54.6	51.4	50.3	50.2	48.0	50.8
Average	BERT	62.5	59.8	62.1	60.8	61.3	61.3	61.2	59.3	59.4	60.2	61.8	58.8	59.9	60.2	57.2	58.0
	MLM	62.7	62.5	63.1	61.7	62.9	62.3	62.9	59.8	59.4	60.5	63.1	59.4	60.9	61.0	57.6	59.1
	DS-Seq	62.9	60.5	62.7	60.6	62.9	61.4	62.4	60.1	60.3	60.8	63.3	59.3	60.5	60.9	58.1	59.3
	DS-Tok	62.7	60.0	62.4	60.4	62.9	62.0	61.8	60.5	60.7	60.5	62.9	60.1	60.4	60.3	58.1	59.4

Table 6: Evaluation results compared with monolingual BERT and multilingual BERT (mBERT) on five countries with *age* data for intrinsic attribute classification tasks (AC-SA, AC-TD) and extrinsic evaluation tasks: sentiment analysis (SA) and topic detection (TD).

gender	Country	Model	SA						TD					
			F	M	X	F	M	X	F	M	X	F	M	X
US	MLM		68.3	67.3	66.9	68.4	67.6	67.8	72.7	69.9	71.1	73.1	70.1	71.3
		DS-Seq	68.1	67.4	66.9	68.6	68.0	68.0	72.7	69.3	71.2	73.1	70.3	71.6
		DS-Tok	68.6	67.2	66.4	68.9	67.5	67.9	72.4	69.6	71.2	73.3	69.9	71.6
		MLM	73.3	71.0	71.7	73.9	71.0	72.6	71.1	69.3	69.8	71.2	69.4	70.0
UK	DS-Seq		73.3	71.1	71.9	73.6	72.2	72.4	71.2	69.0	69.5	71.5	69.3	70.2
		DS-Tok	73.4	71.1	71.6	73.7	72.0	72.2	71.3	69.2	69.6	71.4	69.1	70.3
		MLM	70.8	69.2	69.3	71.2	69.3	70.2	71.9	69.6	70.5	72.2	69.8	70.7
		DS-Seq	70.7	69.3	69.4	71.1	70.1	70.2	72.0	69.2	70.4	72.3	69.8	70.9
Average	DS-Tok		71.0	69.2	69.0	71.3	69.8	70.1	71.9	69.4	70.4	72.4	69.5	71.0
		MLM	62.8	62.6	62.7	64.0	63.4	63.7	68.1	66.7	67.9	70.0	67.9	68.4
		DS-Seq	62.3	62.0	62.4	64.7	62.4	63.5	68.2	66.6	68.0	70.2	67.2	68.3
		DS-Tok	62.6	62.3	62.6	64.4	62.6	63.3	68.7	66.9	68.2	68.8	67.7	68.0

Table 7: Evaluation results on Trustpilot classification tasks (SA, TD) compared by specializing on out-domain data (RtGender (Voigt et al., 2018) for *gender* and BAC (Schler et al., 2006) for *age*) and in-domain data (Trustpilot).

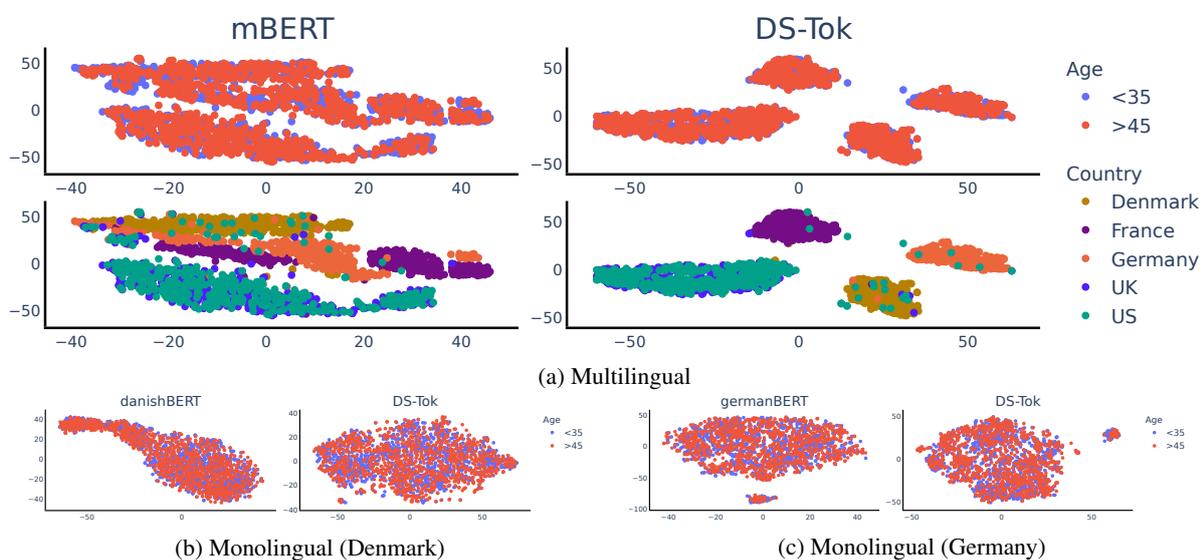


Figure 3: Results of our multilingual and monolingual qualitative analysis for *age*. For multilingual case as plotted in (a), we show a tSNE visualization of review texts embedded with a non-specialized (mBERT) and specialized (DS-Tok) model. Colors indicate the demographic subgroup (upper figures) and countries (lower figures), respectively. For monolingual case as illustrated in (b) and (c) for Denmark and Germany, we show a tSNE visualization of texts embedded with non-specialized (danishBERT, germanBERT) and specialized (DS-Tok) monolingual PLMs. Each subfigure is plotted with 2K instances.

JBLiMP: Japanese Benchmark of Linguistic Minimal Pairs

Taiga Someya and Yohei Oseki

The University of Tokyo

{taiga98-0809,oseki}@g.ecc.u-tokyo.ac.jp

Abstract

In this paper, we introduce **JBLiMP** (Japanese Benchmark of Linguistic Minimal Pairs), a novel dataset for targeted syntactic evaluations of language models in Japanese. JBLiMP consists of 331 minimal pairs, which are created based on acceptability judgments extracted from journal articles in theoretical linguistics. These minimal pairs are grouped into 11 categories, each covering a different linguistic phenomenon. JBLiMP is unique in that it combines two important features independently observed in existing datasets: (i) coverage of complex linguistic phenomena (cf. CoLA) and (ii) presentation of sentences as minimal pairs (cf. BLiMP). In addition, JBLiMP is the first dataset for targeted syntactic evaluations of language models in Japanese, thus allowing the comparison of syntactic knowledge of language models across different languages. We then evaluate the syntactic knowledge of several language models on JBLiMP: GPT-2, LSTM, and n -gram language models. The results demonstrated that all the architectures achieved comparable overall accuracies around 75%. Error analyses by linguistic phenomenon further revealed that these language models successfully captured local dependencies like nominal structures, but not long-distance dependencies such as verbal agreement and binding.

1 Introduction

The past few years have seen a remarkable success of neural language models, and some language models based on Transformer (Vaswani et al., 2017) have achieved the state-of-the-art performance in various natural language processing (NLP) tasks (Wang et al., 2018, 2019). In fact, recent neural language models are extremely successful in solving a variety of downstream tasks, but it remains to be understood how well these neural language models understand the syntax of natural languages. In order to address this question, some studies investigated the syntactic knowledge of language models

with a specially designed dataset for targeted syntactic evaluations (e.g., Linzen et al., 2016; Marvin and Linzen, 2018; Wilcox et al., 2018; Gulordava et al., 2018; Futrell et al., 2019; Chaves, 2020). However, most of these studies have focused on English and other European languages, and only few studies extended this investigation to non-European languages (Gulordava et al., 2018; Ravfogel et al., 2018). Importantly for the purpose here, even fewer studies have dealt with a wide variety of linguistic phenomena in non-English languages (Xiang et al., 2021; Trotta et al., 2021).

In this paper, we introduce **JBLiMP** (Japanese Benchmark of Linguistic Minimal Pairs), a novel dataset for targeted syntactic evaluations of language models in Japanese.¹ JBLiMP consists of 331 minimal pairs, which are created based on acceptability judgments extracted from journal articles in theoretical linguistics. These minimal pairs are grouped into 11 categories, each covering a different linguistic phenomenon. JBLiMP is unique in that it successfully combines two important features independently observed in existing datasets: (i) coverage of complex linguistic phenomena (cf. CoLA; Warstadt et al., 2019) and (ii) presentation of sentences as minimal pairs (cf. BLiMP; Warstadt et al., 2020). We evaluate the syntactic knowledge of several language models on JBLiMP: GPT-2 (Radford et al., 2019), LSTM (Hochreiter and Schmidhuber, 1997) and n -gram language models. The results demonstrated that all the architectures achieved comparable overall accuracies around 75%. Error analyses by linguistic phenomenon further revealed that these language models successfully captured local dependencies like nominal structures, but not long-distance dependencies such as verbal agreement and binding.

¹JBLiMP is available at <https://github.com/osekilab/JBLiMP>.

Language	Linguistic Phenomenon			
	Subject-verb agreement	Filler-gap	Anaphor/binding	Argument structure
English	Linzen et al. (2016); Gulordava et al. (2018); Marvin and Linzen (2018); Warstadt et al. (2019)	Wilcox et al. (2018); Futrell et al. (2019); Chaves (2020); Da Costa and Chaves (2020); Warstadt et al. (2019)	Marvin and Linzen (2018); Warstadt et al. (2019); Futrell et al. (2019)	Warstadt et al. (2019); Kann et al. (2019); Chowdhury and Zamparelli (2019)
French	Gulordava et al. (2018); Mueller et al. (2020); An et al. (2019)			
Italian	Gulordava et al. (2018); Mueller et al. (2020); Trotta et al. (2021)	Trotta et al. (2021)	Trotta et al. (2021)	
Russian	Gulordava et al. (2018); Mueller et al. (2020)			
German	Mueller et al. (2020)			
Basque	Ravfogel et al. (2018)			
Hebrew	Gulordava et al. (2018); Mueller et al. (2020)			
Chinese	Xiang et al. (2021)	Xiang et al. (2021)	Xiang et al. (2021)	Xiang et al. (2021)
Japanese			This work	

Table 1: Related work organized by language and linguistic phenomenon

2 Related Work

Evaluation of language models has been mainly performed by computing metrics such as perplexity. This gives us an objective standard of the performance of language models, but doesn’t provide insight into their performance on specific downstream tasks. While recent large-scale benchmarks like GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) are informative in this respect, many recent studies have sought to provide evidence that language models have learned the syntax of natural languages. In a pioneering work by Linzen et al. (2016), minimal pairs were employed to investigate whether language models are sensitive to subject-verb agreement in English. For instance, they tested whether language models assign a higher probability to *are* than *is* in (1).

- (1) a. The keys to the cabinet are on the table.
b. *The keys to the cabinet is on the table.

Their results suggested that LSTM language models are fairly sensitive to English subject-verb agreement. However, this and related studies (e.g., Marvin and Linzen, 2018; Futrell et al., 2019) only covered a limited range of linguistic phenomena like subject-verb agreement.

In order to tackle this problem, more recent studies have introduced large-scale datasets for comprehensive syntactic evaluations (Warstadt et al., 2019, 2020). One such dataset is CoLA (Corpus of Linguistic Acceptability; Warstadt et al., 2019), which

consists of 10,000 sentences with binary acceptability labels extracted from linguistics journals and textbooks. CoLA is incorporated into GLUE benchmark (Wang et al., 2018) and has been used to evaluate the sensitivity of language models to the syntax of natural languages. While CoLA has enabled the comprehensive syntactic evaluations of language models, this dataset is not without its limitation, as noted by Warstadt et al. (2019) themselves. The limitation lies in the need to train a supervised classifier on CoLA for evaluation. In short, CoLA is designed for binary classification of acceptability judgements, but there is no clear way to map the probability of the sentence estimated by language models to binary acceptability judgements. Unfortunately, “the use of supervision prevents making strong conclusions about the sentence encoding component, since it is not possible to distinguish what the encoder knows from what is learned through supervised training on acceptability data” (Warstadt et al., 2019).

Dataset	Linguistics Journal	Minimal Pairs
CoLA (Warstadt et al., 2019)	✓	
ItaCoLA (Trotta et al., 2021)	✓	
BLiMP (Warstadt et al., 2020)		✓
CLiMP (Xiang et al., 2021)		✓
JBLiMP	✓	✓

Table 2: Comparison of JBLiMP and other existing datasets

With this limitation in mind, BLiMP (Benchmark of Linguistic Minimal Pairs; Warstadt et al.,

2020) is developed, which includes 67 datasets automatically generated from grammar templates created by linguists. These 67 datasets are grouped into 12 categories based on linguistic phenomenon, each containing 1,000 minimal pairs. Note that each pair has one acceptable sentence and one unacceptable sentence. Importantly, this dataset has overcome an aforementioned problem, because sentences are not presented as binary classification problems, but as minimal pairs: the evaluation can be readily performed by comparing the probabilities of an acceptable sentence and an unacceptable sentence. Nevertheless, BLiMP also has its limitation to overcome. Namely, since minimal pairs are automatically generated with template grammars and vocabularies, BLiMP doesn't necessarily cover complex and important linguistic phenomena (cf. Class III judgement, see [Marantz 2005](#); [Linzen and Oseki 2018](#)), compared to those datasets which are created by extracting sentences from linguistics journals.

There is also a general problem with the datasets for targeted syntactic evaluations of language models as a whole: imbalance in target languages and linguistic phenomena (cf. Table 1). In fact, most of the existing datasets have focused on English. Although some studies have extended the scope of their research to other languages ([Gulordava et al., 2018](#); [An et al., 2019](#); [Ravfogel et al., 2018](#); [Mueller et al., 2020](#)), only few studies have covered a wide range of syntactic phenomena and focused on languages other than English ([Xiang et al., 2021](#); [Trotta et al., 2021](#)).

3 JBLiMP

In order to overcome all the limitations mentioned above, we introduce JBLiMP (Japanese Benchmark of Linguistic Minimal Pairs), a novel dataset for targeted syntactic evaluations of language models in Japanese. JBLiMP is unique in that it successfully combines two important features independently observed in existing datasets (Table 2): (i) coverage of complex linguistic phenomena (cf. CoLA; [Warstadt et al., 2019](#)) and (ii) presentation of sentences as minimal pairs (cf. BLiMP; [Warstadt et al., 2020](#)). In addition, JBLiMP is the first dataset for targeted syntactic evaluations of language models in Japanese, thus alleviating the imbalance in target languages and allowing the comparison of syntactic knowledge of language models across different languages.

3.1 Data Collection

JBLiMP consists of acceptability judgments from journal articles on Japanese syntax published in JEAL (Journal of East Asian Linguistics): one of the prestigious journals in theoretical linguistics. Specifically, we examined all the articles published in JEAL between 2006 and 2015 (133 papers in total), and extracted 2,323 acceptability judgments from 28 papers on Japanese syntax (cf. Table 3). Acceptability judgments include sentences in appendices and footnotes, but not sentences presented for analyses of syntactic structures (e.g. sentences with brackets to show their syntactic structures).

3.2 Categorization by linguistic phenomenon

We categorized the extracted sentences into different groups to enable detailed analyses of results by linguistic phenomenon. The categorization mostly followed that of BLiMP ([Warstadt et al., 2020](#)) and was conducted at three levels of granularity: type, phenomenon and paradigm.

3.2.1 Type

First, the extracted sentences were categorized based on the type of acceptability judgements and how those sentences were presented in the articles. This level of categorization has 8 different types. These categories are mutually exclusive, meaning that no further typing is done for sentences in footnotes or appendices.

Acceptability: acceptability judgements that do not depend on a specific context or interpretation.

Interpretation: acceptability judgements that depend on a specific context or interpretation.

Coreference: acceptability judgements that depend on a specific interpretation of coreference.

Lexical: acceptability judgements that depend on a specific lexical item.

Footnote: acceptability judgements presented in footnotes.

Appendix: acceptability judgements presented in appendices.

Repeat: acceptability judgements repeated by the authors.

Variation: acceptability judgements that only differ in unimportant elements for theory construction. For example, (2b) below is categorized into

variation because the difference between *da* ‘is’ and *desu*, a polite form of ‘is’, is not relevant for theory construction.

- (2) a. Taro-ga atta no-wa Hanako-ni da
Taro-Nom saw that-Top Hanako-Dat is
‘It was Hanako that Taroo saw.’
- b. Taro-ga atta no-wa Hanako-ni desu
Taro-Nom saw that-Top Hanako-Dat is
‘It was Hanako that Taroo saw.’

Source	# Sentences
Takahashi (2006)	60
Oshima (2006)	34
Tenny (2006)	70
Bobaljik and Wurmbrand (2007)	18
Ivana and Sakai (2007)	51
Kishimoto (2008)	254
Saito et al. (2008)	46
Takita (2009)	13
Hayashishita (2009)	73
Miyamoto (2009)	36
Tomioka (2009)	27
Asano and Ura (2010)	144
Watanabe (2010)	40
Grosu (2010)	43
Takahashi (2010)	77
Tsujioka (2011)	226
Abe (2011)	53
Takano (2011)	81
Kishimoto (2012)	120
Grosu and Landman (2012)	28
Kishida and Sato (2012)	98
Yoon (2013)	55
Sawada (2013)	81
Watanabe (2013)	118
Nishigauchi (2014)	115
Shimoyama (2014)	63
Sudo (2015)	184
Shibata (2015)	115
Total	2,323

Table 3: Number of extracted sentences by source

3.2.2 Phenomenon

Second, the extracted sentences were further categorized based on linguistic phenomena. Phenomenon basically corresponds to that in BLiMP, but some modifications were applied to make the categorization more suitable for Japanese.

Argument Structure: acceptability judgements based on the order of arguments and case marking.

- (3) a. Taroo-ga Hanako-**ni** au.
Taroo-Nom Hanako-Dat see.
‘Taroo sees Hanako.’

- b. *Taroo-ga Hanako-**o** au.
Taroo-Nom Hanako-Acc see.
‘Taroo sees Hanako.’

Binding: acceptability judgements based on the binding of noun phrases. For instance, this includes the coreference resolution of anaphors.

- (4) a. Hazimete **soitu-ni** atta
for-the-first-time him-Dat saw
hito-ga **Taroo-o** kenasita
person-Nom Taroo-Acc criticized
‘The person who saw him for the first time criticized Taroo.’
- b. *Hazimete **soitu-ni** atta
for-the-first-time him-Dat saw
hito-ga **daremo-o** kenasita
person-Nom everyone-Acc criticized
‘The person who saw him for the first time criticized everyone.’

Control/Raising: acceptability judgements based on predicates that are categorized as control or raising.

- (5) a. **Taroo-ga** korobi sokoneta.
tumbler.doll-Nom tumble failed.
‘Taroo failed to tumble.’
- b. ***Daruma-ga** korobi sokoneta.
tumbler.doll-Nom tumble failed.
‘Tumbler doll failed to tumble.’

Ellipsis: acceptability judgements based on the possibility of omitting elements in the sentences. For instance, this includes nominal and verbal ellipsis.

- (6) a. Hare-no-hi-ha yoi ga
clear-NO-day-Top good though
ame-no-hi-ha otikomu.
rain-NO-day-Top feel.depressed.
‘Clear days are OK, but I feel depressed on rainy days.’
- b. *Hare-no-hi-ha yoi ga
clear-NO-day-Top good though
ame-no-ha otikomu.
rain-NO-Top feel.depressed.
‘Clear days are OK, but I feel depressed on rainy days.’

Filler-gap: acceptability judgements based on the dependency between the moved element and the gap. For instance, this includes wh-movements and cleft sentences.

- (7) a. **Nani-o daremo** yom-ana-katta-no.
What-Acc anyone read-neg-past-Q.
'What did no one read?'
- b. ***Daremo nani-o** yom-ana-katta-no.
anyone What-Acc read-neg-past-Q.
'What did no one read?'

Island effects: acceptability judgements based on the restrictions on filler-gap dependencies such as wh-movements.

- (8) a. Taroo-ha Hanako-ga naze kare-no
Taroo-Top Hanako-Nom why he-Gen
tegami-o suteta **to** omotteiru no.
letter-Acc discarded C think Q
'Why is Taro angry because Hanako
discarded his letters?'
- b. *Taroo-ha Hanako-ga naze kare-no
Taroo-Top Hanako-Nom why he-Gen
tegami-o suteta **kara** okotteiru
letter-Acc discarded because be.angry
no.
Q
'Why is Taro angry because Hanako
discarded his letters?'

Morphology: acceptability judgements based on the morphology. BLiMP has irregular forms category for the conjugation of past tenses, but we adopted this category instead to incorporate minimal pairs on morphology in general.

- (9) a. sore-wa keesoku
that-Top measurement
kanoo-**na** ryuusi-da
possibility-Cop.Adnom particle-Cop
'That is a measurable particle'
- b. *sore-wa keesoku
that-Top measurement
kanoo-**da** ryuusi-da
possibility-Cop.Fin particle-Cop
'That is a measurable particle'

Nominal Structure: acceptability judgements based on the internal structure of noun phrases. BLiMP has determiner-noun agreement category, but we adopted this category instead, because Japanese doesn't have explicit determiner-noun agreements.

- (10) a. Watashi-ga kinoo **mita hito-wa**
I-Nom yesterday saw person-Top
suteki datta
beautiful was
'The person I saw yesterday was beautiful'
- b. *Watashi-ga kinoo **mita no**
I-Nom yesterday saw *no*
hito-wa suteki datta
person-Top beautiful was
'The person I saw yesterday was beautiful'

NPI Licensing: acceptability judgements based on the restrictions on where negative polarity items (NPIs) can appear. For instance, NPIs include *nani-mo*, a Japanese counterpart of 'any'.

- (11) a. *John-ga moshi **nani-ka**
John-Nom if something
nusun-dara, taihos-areru daroo.
steal-COND arrest-PASS be.will
'If John steals anything, he will be arrested.'
- b. *John-ga moshi **nani-mo**
John-Nom if what-MO
nusun-dara, taihos-areru daroo.
steal-COND arrest-PASS be.will
'If John steals anything, he will be arrested.'

Quantifiers: acceptability judgements based on the distribution of quantifiers such as floating quantifiers.

- (12) a. Taroo-ga tomodati-ni **huta-ri** CD-o
Taroo-Nom friend-Dat 2-CL CD-Acc
okutta.
sent.
'Taro sent two friends a package.'
- b. *Taroo-ga CD-o tomodati-ni **huta-ri**
Taroo-Nom CD-Acc friend-Dat 2-CL
okutta.
sent.
'Taro sent two friends a package.'

Verbal Agreement: acceptability judgements based on the dependency between subjects and verbs. Japanese doesn't have the same kind of subject-verb agreement as in English. Instead, this includes the linguistic phenomena such as subject honorification where the social status of subjects are reflected in the morphology of verbs.

- (13) a. **Ito-sensei-ga** Mary-o
 Ito-teacher-Nom Mary-Acc
o-home-ni-nat-ta
 Hon-praise-Lv-Past
 'Prof. Ito praised Mary.'
- b. ***Watashi-ga** Mary-o
 I-Nom Mary-Acc
o-home-ni-nat-ta
 Hon-praise-Lv-Past
 'I praised Mary.'

3.2.3 Paradigm

Finally, the extracted sentences are further categorized into 39 more fine-grained types named paradigm. Paradigm also corresponds to that in BLiMP and is basically sub-categorization of phenomenon.

3.3 Minimal pairs

For direct evaluation of language models through the probabilities assigned by these language models, we created minimal pairs using the sentences categorized above. First, we selected all the sentences that satisfy the following conditions:

- The sentences are presented as unacceptable examples (marked with '?' or '*'), for example). Exceptions are those sentences that are presented as acceptable examples, but marked with '?' or '%'.
- Type is not one of variation, repeat, footnote or appendix.
- The sentences are grouped into one of the 11 phenomena.

We deduplicated the selected unacceptable examples and removed those unacceptable examples whose (un)acceptability depends on the context. Second, since we are concerned with sentence-level acceptability judgements, we augmented incomplete sentences, replacing, for example, (14a) with (14b).

Phenomenon	# Minimal pairs
ARGUMENT STRUCTURE	140
VERBAL AGREEMENT	61
MORPHOLOGY	35
NOMINAL STRUCTURE	23
ELLIPSIS	19
QUANTIFIERS	14
BINDING	13
ISLAND EFFECTS	11
FILLER-GAP	9
NPI LICENSING	4
CONTROL/RAISING	2
Total	331

Table 4: Number of minimal pairs by phenomenon

- (14) a. *Sono futari gakusei
 that two-CL student
 'those two students'
- b. *Taroo-ha sono futari gakusei-ni atta
 Taroo-Top that two-CL student-Dat saw
 'Taroo saw those two students.'

Finally, we created minimal pairs based on the selected unacceptable sentences, on the assumption that all the unacceptable sentences for theory construction generally have their acceptable counterparts to demonstrate the contrasts in acceptability (Sprouse et al., 2013). Specifically, for each unacceptable example, we either found an appropriate acceptable example from the extracted sentences, or created a corresponding acceptable example. When creating acceptable sentences, we read the relevant papers to understand the authors' intent to present the corresponding unacceptable sentences.

3.4 Data Validation

In order to validate the quality of minimal pairs in JBLiMP, we conducted an acceptability judgement experiment with Lancers, a Japanese crowdsourcing platform.² For each minimal pair, 15 native speakers of Japanese completed a forced-choice task which reflects the evaluation procedure of language models. Specifically, annotators are asked to select the more grammatical of the two sentences, following the experimental design in Sprouse et al. (2013). To minimize the burden on annotators, we split 367 minimal pairs into 16 different groups:

²<https://www.lancers.jp>

15 groups of 23 minimal pairs and 1 group of 22 minimal pairs. Each annotator completes 22 or 23 acceptability judgements and is compensated 150 yen (\simeq \$ 1.2). The order of minimal pairs and the vertical order of acceptable and unacceptable examples within a minimal pair was randomized. Majority vote is taken to determine human-annotated acceptable sentences. For each minimal pair, if the annotation of JBLiMP and the majority vote of human annotations do not match, that minimal pair is removed from JBLiMP. In this way, 36 minimal pairs were removed, resulting in 331 minimal pairs in total (Table 4). In addition, we calculated human baseline accuracy, dividing the number of human annotations that match JBLiMP’s judgements by the total number of annotations. As a result, the human baseline accuracy was 90.90% as reported in Table 5.

4 Experiment

4.1 Models

In this paper, we evaluate language models trained by Kuribayashi et al. (2021) with JBLiMP.

GPT-2 GPT-2 (Radford et al., 2019) is one of the large-scale language models based on Transformer architectures (Vaswani et al., 2017). We evaluate two different sizes of GPT-2 models (Trans-LG, Trans-SM). Trans-LG has 24 layers, 16 attention heads, and 1024 embedding dimensions. Trans-SM has 8 layers, 6 attention heads, and 384 embedding dimensions.

LSTM LSTM (Hochreiter and Schmidhuber, 1997) is a language model based on RNN architectures (Elman, 1990), which is known to achieve a better language modeling performance than vanilla RNN language models (Sundermeyer et al., 2012). We evaluate a 2-layer LSTM language model with 1024 hidden layer dimensions and 400 embedding dimensions.

***n*-gram** We also evaluate a 5-gram language model as a baseline. This model is implemented by KenLM (Heafield et al., 2013).

Training settings (Kuribayashi et al., 2021) Training data was approximately 5M sentences extracted from news and Japanese Wikipedia. Each sentence in training data was first segmented by MeCab and then segmented into subwords by

BPE (Byte-Pair Encoding).³ All the neural language models (Trans-LG, Trans-SM and LSTM) were trained with the data of three different sizes: LG (full training data), MD (1/10 training data), SM (1/100 training data). These language models were trained with three different random seeds, and saved at four different points in the training: 100, 1,000, 10,000, 100,000 training steps.

4.2 Evaluation metrics

The probability assigned to a sentence can be mapped into acceptability judgements in multiple ways (Lau et al., 2017). In this work, we employ SLOR (Lau et al., 2017) as a mapping function, which mitigates the confounding effects of sentence lengths and lexical frequencies. SLOR score for a sentence X is defined as follows:

$$SLOR(X) = \frac{\log p_m(X) - \log p_u(X)}{|X|}$$

where $p_m(X)$ is the probability of a sentence given by a language model, and $p_u(X) = \prod_{w \in X} p_u(w)$ is the unigram probability of a sentence. Unigram probabilities are estimated via maximum likelihood estimation for each subword in the training corpus. For each minimal pair, we examine whether language models assign a higher probability/acceptability to an acceptable sentence than an unacceptable one.

5 Results and Discussion

5.1 Overall accuracy

Overall accuracy of each language model on JBLiMP is reported in Table 5. While Trans-LG achieves the best accuracy of 77.95%, all the models notably achieve the comparable accuracy and fall short of human accuracy by a wide margin, which may suggest that language models can’t necessarily recognize complex linguistic phenomena.

5.2 Accuracy by linguistic phenomenon

For each language model, we calculate accuracy by linguistic phenomenon on JBLiMP, as reported in Table 5. Analysis by linguistic phenomenon reveals that the performance of language models drastically differs depending on linguistic phenomenon. Language models achieve a relatively high accuracy on

³Vocabulary size was set to 100,000 and character coverage to 0.9995. Implementation by SentencePiece (Kudo and Richardson, 2018) was employed.

Model	Overall	Argument Structure	Verbal Agr.	Morph.	Nominal Structure	Ellipsis	Quant.	Binding	Island Effects	Filer Gap	NPI Licensing	Control Raising
Trans-LG	77.95	89.05	53.55	82.86	95.65	85.96	73.81	58.97	75.76	55.56	50.00	<u>16.67</u>
Trans-SM	76.54	89.05	44.26	82.86	97.10	89.47	71.43	46.15	84.85	55.56	75.00	<u>0.00</u>
LSTM	75.73	86.67	46.99	83.81	95.65	91.23	66.67	<u>41.03</u>	87.88	44.44	66.67	50.00
5-gram	74.02	78.57	57.38	82.86	86.96	89.47	78.57	53.85	72.73	66.67	<u>50.00</u>	0.00
Human	90.90	92.19	89.62	94.86	97.68	87.37	85.71	82.05	92.12	78.52	90.00	<u>70.00</u>
Model Ave.	76.06	85.76	50.55	83.10	93.84	89.03	72.62	50.00	80.31	55.56	60.42	<u>16.67</u>

Table 5: Accuracy of each language model and human by phenomenon. Accuracy is averaged over 3 different random seeds except 5-gram and human. All the language models are trained for 100,000 steps on full training corpus (LG). The number in bold indicates the best score within a model, while the number with underscore indicates the worst score.

phenomena like nominal structure. This phenomenon includes minimal pairs with relatively local dependencies, as exemplified in (15).

(15) Nominal structure

- a. **Watashi-ga** kinoo **mita hito-wa**
 I-Nom yesterday saw person-Top
 suteki datta
 beautiful was
 ‘The person I saw yesterday was beautiful’
- b. ***Watashi-ga** kinoo **mita no**
 I-Nom yesterday saw *no*
hito-wa suteki datta
 person-Top beautiful was
 ‘The person I saw yesterday was beautiful’

In sharp contrast, language models suffer a sharp drop in accuracy on linguistic phenomena such as verbal agreement and binding. (Here, control/raising is taken out of consideration because its data size is small compared to the other phenomena.) These phenomena generally involve relatively long dependencies: verbal agreement involves dependency between the subject and the verb of the sentence as exemplified in (16), while binding involves dependency between anaphors and their antecedents as illustrated in (17).

(16) Verbal agreement

- a. **Ito-sensei-ga** Mary-o
 Ito-teacher-Nom Mary-Acc
o-home-ni-nat-ta
 Hon-praise-Lv-Past
 ‘Prof. Ito praised Mary.’
- b. ***Watashi-ga** Mary-o
 I-Nom Mary-Acc
o-home-ni-nat-ta
 Hon-praise-Lv-Past
 ‘I praised Mary.’

(17) Binding

- a. Hazimete **soitu-ni** atta
 for-the-first-time him-Dat saw
 hito-ga **Taroo-o** kenasita
 person-Nom Taroo-Acc criticized
 ‘The person who saw him for the first time criticized Taroo.’
- b. *Hazimete **soitu-ni** atta
 for-the-first-time him-Dat saw
 hito-ga **daremo-o** kenasita
 person-Nom everyone-Acc criticized
 ‘The person who saw him for the first time criticized everyone.’

Lower accuracy in these kinds of minimal pairs suggests that language models are less sensitive to long-distance dependencies. These results are compatible with the previous results that RNN-based language models cannot capture long-distance dependencies without explicit supervision (Linzen et al., 2016), but are not necessarily consistent with the results that Transformer-based language models can successfully capture long-distance dependencies (Goldberg, 2019).

5.3 Human confidence and model confidence

Figure 1 shows the relationship between model confidence and human confidence. Each model’s confidence on a minimal pair is defined as the difference of the SLOR scores between the acceptable and unacceptable sentence: $SLOR(X_{pos}) - SLOR(X_{neg})$ where X_{pos} is an acceptable sentence and X_{neg} is an unacceptable sentence. Human confidence on a minimal pair is defined as the number of annotators who had the same annotation as the JBLiMP. While the language models are able to make predictions with relatively high confidence for sentences with high human confidence, the confidence of the language models is low for sentences with low human confidence, i.e., for which there

are fluctuations in acceptability judgments among humans. Furthermore, many of the language models have negative confidence for the sentences with low human confidence. These results may suggest that language models have successfully captured the gradience in human acceptability judgements, whose existence was suggested in Lau et al. (2017).

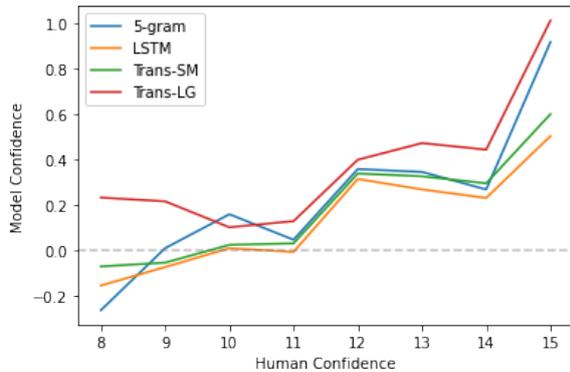


Figure 1: The relationship between model confidence and human confidence. All the neural language models are trained for 100,000 steps on full training corpus (LG).

5.4 Effects of perplexity on accuracy

We investigate the relationship between the perplexity, which is widely used as an evaluation metric of language models’ performance, and the accuracy on JBLiMP for each language model. The perplexity is calculated on the validation data in Kuribayashi et al. (2021). Figure 2 shows the language models’ accuracy on JBLiMP as a function of perplexity. In contrast to the results in Kuribayashi et al. (2021) that lower perplexity does not necessarily ensure better psychometric predictive power of language models, our results suggest that language models with lower perplexity will generally achieve better syntactic performance. Note incidentally that language models with particularly high perplexity ($> 3 \times 10^4$), represented as the points to the right of the black dashed line in Figure 2, are trained for more than 10,000 steps with relatively small data (SD or MD). These language models seem to be overfitted to the training data, and thus were taken out of consideration in this discussion.

6 Conclusion

In this paper, we introduced JBLiMP (Japanese Benchmark of Linguistic Minimal Pairs), a novel dataset for targeted syntactic evaluations of language models in Japanese. JBLiMP consists of

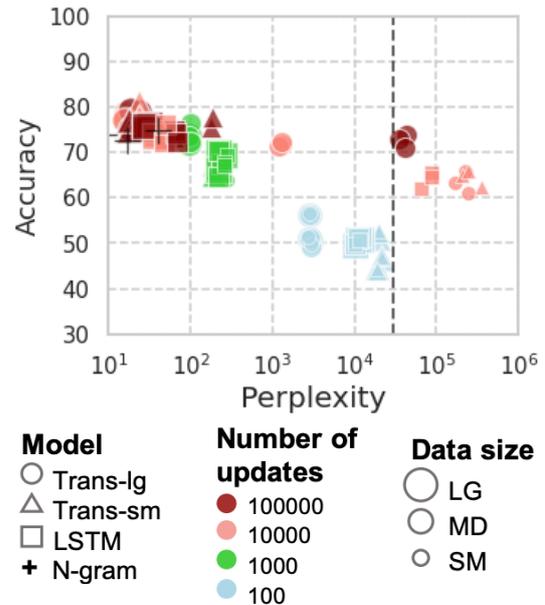


Figure 2: Models’ accuracy on JBLiMP as a function of perplexity. The perplexity is calculated on the validation data in Kuribayashi et al. (2021). The vertical dashed line in black indicates the perplexity of 3×10^4 .

331 minimal pairs, which are created based on acceptability judgments extracted from journal articles in theoretical linguistics. These minimal pairs are grouped into 11 categories, each covering a different linguistic phenomenon. JBLiMP is unique in that it successfully combines two important features independently observed in existing datasets: (i) coverage of complex linguistic phenomena (cf. CoLA) and (ii) presentation of sentences as minimal pairs (cf. BLiMP). In addition, JBLiMP is the first dataset for targeted syntactic evaluations of language models in Japanese, thus allowing the comparison of syntactic knowledge of language models across different languages. We then evaluated the syntactic knowledge of several language models: GPT-2, LSTM and n -gram language models. The results demonstrated that all the architectures achieved comparable overall accuracies around 75%. Error analyses by linguistic phenomenon further revealed that these language models successfully captured local dependencies like nominal structures, but not long-distance dependencies such as verbal agreement and binding. Finally, these detailed analyses of language models’ knowledge on complex linguistic phenomena using minimal pairs are only possible with the unique design of JBLiMP. This paper will hopefully encourage the development of the datasets with JBLiMP’s two important features in other languages.

Limitations

All the example sentences in JBLiMP were manually transcribed from linguistic journals. While this method of data collection has enabled it to cover complex linguistic phenomena, it also made it difficult to increase the size of the dataset. Additionally, the quantity of minimal pairs on a specific linguistic phenomenon is directly influenced by how often that phenomenon is discussed in linguistic journals, hence the imbalanced distribution of minimal pairs across different linguistic phenomena in JBLiMP. These problems could be overcome by collecting additional examples from linguists (if possible, the authors of the source linguistic journals in JBLiMP).

Acknowledgements

This work was supported by JST PRESTO Grant Number JPMJPR21C2, Japan. We are also grateful for the anonymous reviewers and area chairs for their detailed and helpful feedback.

References

- Jun Abe. 2011. Real parasitic gaps in Japanese. *J. East Asian Ling.*, 20(3):195–218.
- Aixiu An, Peng Qian, Ethan Wilcox, and Roger Levy. 2019. Representation of constituents in neural language models: Coordination phrase as a case study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2888–2899, Hong Kong, China. Association for Computational Linguistics.
- Shin’ya Asano and Hiroyuki Ura. 2010. Mood and case: with special reference to genitive case conversion in Kansai Japanese. *J. East Asian Ling.*, 19(1):37–59.
- Jonathan D Bobaljik and Susi Wurmbrand. 2007. Complex predicates, aspect, and anti-reconstruction. *J. East Asian Ling.*, 16(1):27–42.
- Rui P Chaves. 2020. What don’t RNN language models learn about Filler-Gap dependencies? *Proceedings of the Society for Computation in Linguistics*, 3(1):20–30.
- Shammur Absar Chowdhury and Roberto Zamparelli. 2019. An LSTM adaptation study of (un)grammaticality. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 204–212, Florence, Italy. Association for Computational Linguistics.
- Jillian Da Costa and Rui Chaves. 2020. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 12–21, New York, New York. Association for Computational Linguistics.
- Jeffrey L Elman. 1990. Finding structure in time. *Cogn. Sci.*, 14(2):179–211.
- Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yoav Goldberg. 2019. [Assessing BERT’s syntactic abilities.](#)
- Alexander Grosu. 2010. The status of the internally-headed relatives of Japanese/Korean within the typology of “definite” relatives. *J. East Asian Ling.*, 19(3):231–274.
- Alexander Grosu and Fred Landman. 2012. A quantificational disclosure approach to Japanese and Korean internally headed relatives. *J. East Asian Ling.*, 21(2):159–196.
- Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.
- J-R Hayashishita. 2009. Yori-Comparatives: A reply to Beck et al. (2004). *J. East Asian Ling.*, 18(2):65–100.
- Kenneth Heafield, Ivan Pouzyrevsky, J. Clark, and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In *ACL*.
- S Hochreiter and J Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.
- Adrian Ivana and Hiromu Sakai. 2007. Honorification and light verbs in Japanese. *J. East Asian Ling.*, 16(3):171–191.
- Katharina Kann, Alex Warstadt, Adina Williams, and Samuel R Bowman. 2019. Verb argument structure alternations in word and sentence embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 287–297.
- Maki Kishida and Yosuke Sato. 2012. On the argument structure of *zi*-verbs in Japanese: reply to Tsujimura and Aikawa (1999). *J. East Asian Ling.*, 21(2):197–218.

- Hideki Kishimoto. 2008. Ditransitive idioms and argument structure. *J. East Asian Ling.*, 17(2):141–179.
- Hideki Kishimoto. 2012. Subject honorification and the position of subjects in Japanese. *J. East Asian Ling.*, 21(1):1–41.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, and Kentaro Inui. 2021. Lower perplexity is not always Human-Like. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5203–5217, Online. Association for Computational Linguistics.
- Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cogn. Sci.*, 41(5):1202–1241.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn Syntax-Sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.
- Tal Linzen and Yohei Oseki. 2018. The reliability of acceptability judgments across languages. *Glossa: a journal of general linguistics*, 3(1).
- A Marantz. 2005. Generative linguistics within the cognitive neuroscience of language. *The Linguistic Review*.
- Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.
- Yoichi Miyamoto. 2009. On the Nominal-Internal distributive interpretation in Japanese. *J. East Asian Ling.*, 18(3):233–251.
- Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Taisuke Nishigauchi. 2014. Reflexive binding: awareness and empathy from a syntactic point of view. *J. East Asian Ling.*, 23(2):157–206.
- David Y Oshima. 2006. Adversity and Korean/Japanese passives: Constructional analogy. *J. East Asian Ling.*, 15(2):137–166.
- A Radford, J Wu, R Child, D Luan, D Amodei, and others. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.
- Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. 2018. Can LSTM learn to capture agreement? the case of Basque. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 98–107, Brussels, Belgium. Association for Computational Linguistics.
- Mamoru Saito, T-H Jonah Lin, and Keiko Murasugi. 2008. N'-Ellipsis and the structure of noun phrases in Chinese and Japanese. *J. East Asian Ling.*, 17(3):247–271.
- Osamu Sawada. 2013. The comparative morpheme in modern Japanese: looking at the core from 'outside'. *J. East Asian Ling.*, 22(3):217–260.
- Yoshiyuki Shibata. 2015. Negative structure and object movement in Japanese. *J. East Asian Ling.*, 24(3):217–269.
- Junko Shimoyama. 2014. The size of noun modifiers and degree quantifier movement. *J. East Asian Ling.*, 23(3):307–331.
- J Sprouse, C T Schütze, and D Almeida. 2013. A comparison of informal and formal acceptability judgments using a random sample from linguistic inquiry 2001-2010. *Lingua*, 134:219–248.
- Yasutada Sudo. 2015. Hidden nominal structures in Japanese clausal comparatives. *J. East Asian Ling.*, 24(1):1–51.
- Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.
- Daiko Takahashi. 2006. Apparent parasitic gaps and null arguments in Japanese. *J. East Asian Ling.*, 15(1):1–35.
- Masahiko Takahashi. 2010. Case, phases, and Nominative/Accusative conversion in Japanese. *J. East Asian Ling.*, 19(4):319–355.
- Yuji Takano. 2011. Double complement unaccusatives in Japanese: puzzles and implications. *J. East Asian Ling.*, 20(3):229–254.
- Kensuke Takita. 2009. If Chinese is Head-Initial, Japanese cannot be. *J. East Asian Ling.*, 18(1):41–61.
- Carol L Tenny. 2006. Evidentiality, experiencers, and the syntax of sentence in Japanese. *J. East Asian Ling.*, 15(3):245–288.

- S Tomioka. 2009. Why questions, presuppositions, and intervention effects. *J. East Asian Ling.*, 18(4):253–271.
- Daniela Trotta, Raffaele Guarasci, Elisa Leonardelli, and Sara Tonelli. 2021. [Monolingual and cross-lingual acceptability judgments with the Italian CoLA corpus](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2929–2940, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Takae Tsujioka. 2011. Idioms, mixed marking in nominalization, and the basegeneration hypothesis for ditransitives in Japanese. *J. East Asian Ling.*, 20(2):117–143.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. [Superglue: A stickier benchmark for general-purpose language understanding systems](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.
- Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.
- Akira Watanabe. 2010. Notes on nominal ellipsis and the nature of no and classifiers in Japanese. *J. East Asian Ling.*, 19(1):61–74.
- Akira Watanabe. 2013. Non-neutral interpretation of adjectives under measure phrase modification. *J. East Asian Ling.*, 22(3):261–301.
- Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about Filler–Gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.
- Beilei Xiang, Changbing Yang, Yu Li, Alex Warstadt, and Katharina Kann. 2021. CLiMP: A benchmark for Chinese language model evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2784–2790, Online. Association for Computational Linguistics.
- Suwon Yoon. 2013. Parametric variation in subordinate evaluative negation: Korean/Japanese versus others. *J. East Asian Ling.*, 22(2):133–166.

A Examples of minimal pairs in JBLiMP

Phenomenon	Paradigm	Unacceptable	Acceptable
QUANTIFIERS	floating quantifiers	学生が 4人 家を 買った。 student four-CL house-Acc buy-Past	学生が 4人 家を 買った。 student four-CL house-Acc buy-Past
	universal quantifiers	みんながみんな 大学へ 行かない。 everyone-Nom-everyone university-to go-Neg-Pres	みんながみんな 大学へ 行く 訳では ない。 everyone-Nom-everyone university-to go-Pres reason-Cop-Top Neg-Pres
ISLAND EFFECTS	classifier	太郎は 3本ずつの 鉛筆を 買った。 Taroo-Top three-CL-Dist-Gen that pencil-Acc buy-Past	太郎は 3本ずつの 鉛筆を 買った。 Taroo-Top that three-CL-Dist-Gen pencil-Acc buy-Past
	negation	ジョンは メアリーが 賢い 以上に 賢くない。 John-Top Mary-Nom smart more smart-Neg	ジョンは メアリーが 賢い 以上に 賢い。 John-Top Mary-Nom smart more smart
FILLER-GAP	complex-NP island	太郎が 昨日 買った 人を 探している のは 花子に だ。 Taroo-Nom yesterday saw person-Acc looks-for that-Top Hanako-Dat is	太郎が 昨日 花子に 買った 人を 探している のだ。 Taroo-Nom yesterday Hanako-Dat saw person-Acc looks-for that-is
	adjunct island	太郎が 読んだ から 怒った のは その 本を だ。 Taroo-Nom read because Hanako-Nom got-angry that-Top that book-Acc is	太郎が 読んだ から 花子が 怒った のだ。 Taroo-Nom that book-Acc read because Hanako-Nom got-angry that-is
	specificity island	ジョンは その メアリーより 高い 指輪を 買った。 John-Top that Mary-than expensive ring-Acc bought	ジョンは メアリーより 高い 指輪を 買った。 John-Top Mary-than expensive ring-Acc bought
	negative island	ジョンは メアリーが 雇わなかったより 賢い 人を 雇った。 John-Top Mary-Nom hire-Neg-Past-than smart person-Acc found	ジョンは メアリーが 雇ったより 賢い 人を 雇った。 John-Top Mary-Nom hire-Past-than smart person-Acc found
	factive island	メアリーが ジョンが 自分の 学生が 新しい 仮説を 提案した Mary-Nom John-Nom self-Gen student-Nom new hypothesis-Acc proposed と 知っていたのの 欠陥を 指摘した。 Czer know-had-no-Gen defect-Acc pointed-out	メアリーが ジョンが 自分の 学生が 新しい 仮説を 提案した Mary-Nom John-Nom self-Gen student-Nom new hypothesis-Acc proposed と 知っていたのの 欠陥を 指摘した。 Czer say-had-no-Gen defect-Acc pointed-out
NPI LICENSING	intervention effects	誰も 何を 読まなかったの？ anyone what-Acc read-Neg-Past-Q	何を 誰も 読まなかったの？ what-Acc anyone read-Neg-Past-Q
	relative clause	山田先生は この本を 買った ことは お読み だ。 Yamada-teacher-Top this-book-Ac become-Past fact-Top Hon-read-Ren Cop	山田先生は この本を お読みに なった。 Yamada-teacher-Top this-book-Ac Hon-read-Ren-Obl become-Past
	cleft	山田先生が 買った この本の お読み だ。 Yamada-teacher-Nom become-Pastの this-book-Gen Hon-read-Ren Cop	山田先生が この本を お読みに なった。 Yamada-teacher-Nom this-book-Acc Hon-read-Ren-Obl become-Past
	resumptive pronoun	トムが それらを 食べた ことが 明らか 芋は 大きかった。 Tom-Nom these-Acc ate fact-Nom clear potato-Top big-Past	トムが 食べた ことが 明らか 芋は 大きかった。 Tom-Nom ate fact-Nom clear potato-Top big-Past
	NPI	今回は 誰が 寄付を 呼びかけも しなかった this-time-Top anyone-Nom donation-Acc call.for-Q do-Neg-Past	今回は 誰から 寄付を 呼びかけも しなかった。 this-time-Top anyone-from donation-Acc call.for-Q do-Neg-Past
NOMINAL STRUCTURE	NCI	ジョンが もし 何も 盗んだら、 逮捕される だろう。 John-Nom if what-MO steal-Cond arrest-Pass be.will	ジョンが もし 何か 盗んだら、 逮捕される だろう。 John-Nom if what-Q steal-Cond arrest-Pass be.will
	modifier	私が 昨日 見たのは 素敵だった。 I-Nom yesterday saw-NO-person-Top beautiful-Past	私が 昨日 見た人は 素敵だった。 I-Nom yesterday saw-person-Top beautiful-Past
CONTROL/RAISING	measure phrase	このビルは 高さ 20メートル ある。 this-building-Top shortness 20-meter is	このビルは 高さ 20メートル ある。 this-building-Top height 20-meter is
	subject control	だるまが 転び倒れた。 Dharma-Nom tumble failed	太郎が 転び倒れた。 Taroo-Nom tumble failed

SMATCH++: Standardized and Extended Evaluation of Semantic Graphs

Juri Opitz

Heidelberg University

opitz.sci@gmail.com

Abstract

The SMATCH metric is a popular method for evaluating graph distances, as is necessary, for instance, to assess the performance of semantic graph parsing systems. However, we observe some issues in the metric that jeopardize meaningful evaluation. E.g., opaque pre-processing choices can affect results, and current graph-alignment solvers do not provide us with upper-bounds. Without upper-bounds, however, fair evaluation is not guaranteed. Furthermore, adaptations of SMATCH for extended tasks (e.g., fine-grained semantic similarity) are spread out, and lack a unifying framework.

For better inspection, we divide the metric into three modules: pre-processing, alignment, and scoring. Examining each module, we specify its goals and diagnose potential issues, for which we discuss and test mitigation strategies. For pre-processing, we show how to fully conform to annotation guidelines that allow structurally deviating but valid graphs. For safer and enhanced alignment, we show the feasibility of optimal alignment in a standard evaluation setup, and develop a lossless graph compression method that shrinks the search space and significantly increases efficiency. For improved scoring, we propose standardized and extended metric calculation of fine-grained sub-graph meaning aspects. Our code is available at <https://github.com/flipz357/smatchpp>

1 Introduction

Semantic graphs such as meaning representations (MRs) aim at capturing the meaning of a text. Typically, these graphs are rooted, directed, acyclic, and labeled. Vertices denote semantic entities, and edges represent semantic relations (e.g., *instrument*, *cause*, etc.). A prominent MR framework is *Abstract Meaning Representation (AMR)*, proposed by Banarescu et al. (2013), which anchors in a propositional knowledge base (Palmer et al., 2005).

Using a metric such as SMATCH (Cai and Knight, 2013), we can measure a distance (or similarity) between graphs, by aligning nodes, and counting matching graph triples. In fact, SMATCH measurement has various applications. It is used for selecting parsing systems that project AMR structures (Flanigan et al., 2014; May and Priyadarshi, 2017; Xu et al., 2020; Hoang et al., 2021a; Bevilacqua et al., 2021) and various other semantic graphs (van Noord et al., 2018; Zhang et al., 2018; Oepen et al., 2020; Stengel-Eskin et al., 2020; Martínez Lorenzo et al., 2022; Lin et al., 2022), for MR-based evaluation and diagnostics of text generation systems (Opitz and Frank, 2021; Manning and Schneider, 2021; Ribeiro et al., 2021; Hoyle et al., 2021), as backbone in an ensemble parsing algorithm (Hoang et al., 2021b), and for studying cross-lingual phenomena (Uhrig et al., 2021; Wein et al., 2022). Through SMATCH measured on sub-graphs, we can assess similarity of linguistic phenomena such as semantic roles, negation, or coreference (Damonte et al., 2017), a property that can be leveraged in neural text embeddings (Opitz and Frank, 2022b).

However, SMATCH measurement is non-trivial and lacks specification. For instance, SMATCH involves an NP-hard *optimization problem* of structural *graph alignment*, which distinguishes it from most metrics used in other evaluation tasks. In practice, a solution of this problem is found by employing a hill-climber. However, a hill-climber terminates at local optima, and it cannot inform us about a score upper-bound. In the end, this means that we lack information about the quality of the returned solution, potentially lowering our trust in the final evaluation. To mitigate this issue, we would like to study the possibility of optimal solution, or solution with a tight upper-bound. There are also other issues, on which we lack understanding. E.g., we do not know to what extent different pre-processing choices may affect the evaluation results, and we miss specification of SMATCH’s

popular fine grained sub-graph metrics (Damonte et al., 2017), where it is unclear how sub-graphs should be best extracted and compared.

Paper structure and contributions First, we describe and generalize the SMATCH metric (§3), and summarize recent SMATCH variants in one framework. Then we break the metric down into three modules (§4), which lets us better distribute our attention over its key components. For each module, we discuss specification of goals and mitigation of issues. In the pre-processing module (§5), we motivate graph standardization to allow safer matching of equivalent MR graphs with different structural choices. In the optimization module (§6), we test strategies for solving the alignment problem with optimality guarantees. In the scoring module (§7), we discuss standardized and extended scoring of fine-grained semantic aspects, such as causality, tense, and location.

2 Related work

Metric standardization An inspiration for us is the work of Post (2018), who propose the popular SACREBLEU framework for fairer comparison of machine translation systems with a standardized BLEU metric (Papineni et al., 2002). Specifically, SACREBLEU ships BLEU *together with* a specified tokenizer – prior to this, BLEU differences between systems could depend on different tokenization protocols. Facing the challenging problem of graph evaluation, a main contribution of our work is that we i) analyze weak spots in the current evaluation setup and ii) discuss ways of mitigating these issues, aiming at best evaluation practices.

MR metrics Cai and Lam (2019) introduce a variant of SMATCH (Cai and Knight, 2013) that penalizes dissimilar structures if they are situated in proximity of the graph root, motivated by their assumption that ‘core-semantics’ are located near the root of MR graphs. Furthermore, Opitz et al. (2020) introduce a SMATCH variant that performs a graded match of semantic concepts (e.g., *cat* vs. *kitten*), aiming at extended use-cases beyond parsing evaluation, where MRs of different sentences need to be compared. Similarly, Wein and Schneider (2022) adapt an embedding-based variant of SMATCH for cross-lingual MR comparison. We show that the different SMATCH adaptations can be viewed through the same lens with a generalized notion of triple match. Furthermore, Damonte et al.

(2017) propose fine-grained SMATCH that measure MR agreement in different aspects, such as *semantic roles*, *coreference* or *polarity*. We diagnose and mitigate issues in the aspectual assessment, and show how to extend the measured aspects.

Conceptually different MR metrics have been proposed by Anchieta et al. (2019) and Song and Gildea (2019) who aim at increased efficiency using structure extraction via breadth-first traversals, or Opitz et al. (2021) who compare MRs of different sentences with Wasserstein Weisfeiler-Leman kernels (Weisfeiler and Leman, 1968; Togninalli et al., 2019). Since significant parts of this paper are independent from SMATCH-specific scoring¹, other MR metrics can profit from our work.

3 SMATCH: Overview and generalization

We introduce SMATCH and define a generalized SMATCH, so that we can summarize recent SMATCH variants in one framework.

Preliminary I: MR graph If not mentioned otherwise, we view an MR graph a as a set of triples, where a triple has one of two types. Unary triples have the structure $\langle x, :rel, c \rangle$, where the source x is a variable and the target c is a descriptive label that shows the type or an attribute of x , depending on the edge label $:rel$.² Using variables such as x we can (co-)refer to different events and entities and capture complex events. Binary triples have the structure $\langle x, :rel, y \rangle$, where both the source x and the target y are variables.³

Preliminary II: SMATCH The idea of SMATCH is to measure structural similarity of graphs via the amount of triples that are shared by a and b . To obtain a meaningful score, we must know an alignment $map: vars(a) \leftrightarrow vars(b)$ that tells us how to map a variable in the first MR to a variable in the second MR. In this alignment, every variable from a can have at maximum one partner in b (and vice versa). Let an application of a map to a graph a be denoted as $a^{map} := \{t^{map} ; t \in a\}$, where t^{map} of a triple $t = \langle x, :rel, y \rangle$ is set to $t^{map} = \langle map(x), :rel, map(y) \rangle$ for binary triples, and $t^{map} = \langle map(x), :rel, c \rangle$

¹E.g., input standardization (§5) and sub-graph extraction for fine-grained aspectual matching (§7.3).

²E.g., $\langle x, :instance, cat \rangle$ would indicate that ‘ x is a cat’, while $\langle y, :polarity, - \rangle$ means y is negated.

³E.g., $\langle x, :location, y \rangle$, which means that x is located at y , or $\langle x, :arg0, y \rangle$ which usually indicates that y participates as the agent in the event referred to by x .

for unary triples.

Under any alignment map , we can calculate an overlap score f . In original SMATCH, f is the size of the triple overlap of a and b :

$$f(a, b, map) = |a^{map} \cap b|. \quad (1)$$

Ultimately we are interested in

$$F = \max_{map} f(a, b, map), \quad (2)$$

Finding a maximizer map^* lies at the heart of SMATCH, and we will dedicate ourselves to it later in §6. For now, we assume that we have map^* at our disposal. Therefore, we can calculate *precision* (P) and *recall* (R):

$$P = |a|^{-1}F, \quad R = |b|^{-1}F, \quad (3)$$

to obtain a final F1 evaluation score: $2PR/(P + R)$. With such a score, we can assess the similarity of MRs, and compare and select parsing systems.

Generalizing SMATCH In SMATCH, two triples are said to match if they are identical under a mapping. I.e., we match with $match(t, t') := I[t = t']$ that returns 1 if two triples t and t' are the same, and zero else (we omit the map for simplicity). Recently, SMATCH has been adapted and tailored to different use-cases. E.g., SMATCH has been extended to incorporate word embeddings (Opitz et al., 2020; Wein and Schneider, 2022) to match $\langle x, :instance, c \rangle$ triples for studying cross-lingual MRs or MRs of different sentences.⁴ On the other hand, Cai and Lam (2019) propose a root-distance bias, based on the assumption that ‘core-semantics’ lie in the proximity of an MR’s root.

We find that we can summarize such variants in one framework. We achieve this by introducing a *scaled triple matching* function:

$$match(t, t') = w_t^{t'} \cdot \begin{cases} I[t = t'], & \text{if } t_2, t'_2 \neq :inst. \\ I[t_1 = t'_1] \cdot sim(t_3, t'_3) & \text{else} \end{cases}$$

For matching concepts with embeddings, we can use an embedding similarity on the descriptive concept labels with $sim(c, c')$ and the importance

⁴Consider $\langle x, :instance, cat \rangle$ extracted from one sentence vs. $\langle y, :instance, kitten \rangle$ extracted from another sentence. A graded match is required to properly assess the similarity of the concepts.

“(d / dog :location (h / house))”

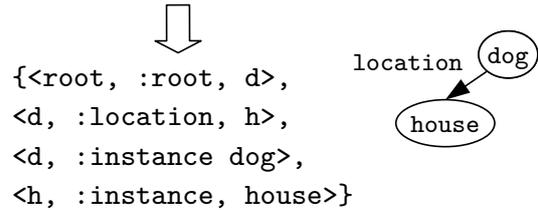


Figure 1: A serialized MR string is read into a graph.

weight $w_t^{t'} = 1 \forall t, t'$.⁵ For Root-distance biased SMATCH as proposed by Cai and Lam (2019) we set $w_t^{t'}$ such that we discount triple matches that are distant to the root.⁶

Our generalization does not change or constrain the original SMATCH. Instead, our goal was to define a more general framework of SMATCH-type metrics that unifies recently proposed SMATCH variants and show possibilities for further extension. For the following studies, we set SMATCH++ to basic SMATCH, which is recovered by setting $\forall t, t' : w_t^{t'} = 1$ and $sim(c, c') := I[c = c']$.

4 A modular view on SMATCH

To set the stage for inspection, we break SMATCH down into three modules. i) *Preprocessing*, ii) *Alignment*, and iii) *Scoring*. In particular, i) *Preprocessing* discusses any graph reading and processing in advance of the alignment. ii) *Alignment* revolves around the search mechanism used for finding an optimal mapping map^* . iii) *Scoring* involves calculating final scores and statistics that are returned to a user. For each module, we will specify its goals, assess potential weak spots and discuss mitigation.

5 Module I: Pre-processing

5.1 Module goal and current implementation

MRs are typically stored and distributed in a ‘Penman’ string format, which can serialize any rooted and directed graph into a string. The goal of this module is to project two serialized textual MRs onto two sets of triples, as outlined in Figure 1.

The target domain of this projection should be a *standardized* MR graph space, where format divergences that do not impact graph seman-

⁵That is, if the triples are not instance triples, we check whether the triples are equivalent (as in standard SMATCH), but if both triples are instance relation triples and the variables t_1, t'_1 are set to equal each other, we calculate the similarity between their descriptive concept labels.

⁶For a properly normalized final score if $\exists (t, t'), w_t^{t'} \neq 1$, we may have to change denominators in Eq. 3

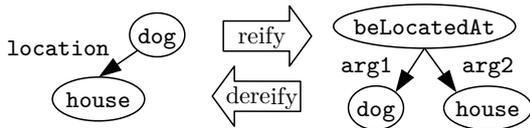


Figure 2: Outline of *location*-reification.

tics are eliminated. Original SMATCH performs pre-processing as follows: i) lower-case strings, ii) de-invert edges (e.g., $\langle x, :relation-of, y \rangle \rightarrow \langle y, :relation, x \rangle$). However, while these steps seem sensible, more steps can be undertaken to enhance evaluation.

5.2 Two structures, one meaning: reification

Some MR guidelines, including the AMR guideline, allow meaning-preserving structural graph translations (Banarescu et al., 2019; Goodman, 2019) with so called *reifications* (or *de-reification* as an inverse mechanism). A subset of relations is selected to constitute a semantic relation core set (e.g., $:arg0, :arg1, \dots, :op1, :op2, \dots$) and for all other remaining relations (e.g., $:location, :time$), we use rules to map the relation to a sub-graph, where the rule-triggering relation label is projected onto a node, and the former source and target of the relation are attached with outgoing core relations. E.g., consider Figure 2, where a reification is applied to a $\langle x, :location, y \rangle$ relation. In this case, the rule is:

- location (de)reification:

$$\begin{aligned} &\langle x, :location, y \rangle \\ &\iff \\ &\langle z, :instance, beLocatedAt \rangle \\ &\wedge \langle z, :arg1, x \rangle \\ &\wedge \langle z, :arg2, y \rangle, \end{aligned}$$

where $:arg1$ indicates the thing that is found at a location $:arg2$.

The question whether an annotator should use either means of representation, is answered in the guidelines as follows: *whenever they feel like it* (Banarescu et al., 2019). Therefore, a parser should not be penalized or rewarded for projecting reified (or non-reified) structures.

Empirical assessment of effect To understand the effect that reification can have on the final SMATCH score, it is interesting to study an edge-case: evaluating graphs that are fully reified against graphs that are fully de-reified. As a data set we take LDC2017T10, a standard AMR benchmark.

	Data setup		SMATCH	
	X	Y	Orig	rlyStd
i)	gold dereify	gold reify	73.8	100.0
ii)	gold standard	gold reify	73.9	100.0
iii)	gold standard	gold dereify	100.0	100.0
iv)	parser dereify	gold reify	60.9	82.8
v)	parser reify	gold reify	82.8	82.8
vi)	parser dereify	gold dereify	81.4	82.8
vii)	parser standard	gold standard	81.4	82.8
viii)	parser standard	gold reify	60.9	82.8
ix)	parser standard	gold dereify	81.4	82.8

Table 1: Results of meaning-preserving translations. rlyStd: score when we project X and Y into standardized reified space.

Additionally, we gather automatic parses by applying an AMR parser (Xu et al., 2020).

The results of this experiment are shown in Table 1. In the first three lines (i-iii) we compare *equivalent* translated versions of the test partition (gold vs. gold). We find that two equivalent gold standards can be judged to be very different (73.9 points, -26.1 points). A similar phenomenon can be observed when looking at the parses. The best parser score is achieved when comparing parses and references in the domain of reified graphs (82.8 points). On the other hand, if only the reference is reified, the parser score drops by 20 points (viii).

However, we also see that the results of a basic evaluation (vii) is practically the same as the result when evaluating with de-reified graphs (vi), indicating that both parser and gold annotation abstain from reification, where possible.

Discussion Having established that rule-based graph translations can enhance evaluation fairness, we pose the question: *should we prefer reification or de-reification for space standardization?*

The answer should be *reification*, since it can be seen as a form of generalization. More precisely, we note that reification of non-core relations is *always* possible. In fact, an interesting effect of reified structures is that they equip us with the means to attach further structure, or features, to semantic relations. On the other hand, however, de-reification is not always possible. It is only well-defined if there is no incoming edge into the node that corresponds to the non-core relation⁷, and if

⁷It is not clear to which node the incoming edge (that now does not have a target) should be re-attached: the $arg0$ or $arg1$ of the outgoing edges of the former node? Either choice would likely come with a change in meaning.

there are not more than two outgoing edges⁸.

However, there are also (practical) arguments against reification. Consider that de/non-reified MRs are smaller and have more edge label differentiation. This i) may facilitate more intuitive display for humans and ii) shrinks the alignment search space. Indeed, a large solution space may have ramifications for evaluation optimality and efficiency (in §6, we empirically study this issue). Therefore, when taking into account that the empirical effect size appears neglectable in the average case, these trade-offs may not always be justified, and we may instead use de-reification, where possible.

5.3 Triple removals

Duplicate triples are triples that occur more than once. We find that they are sometimes produced by some parsers. Additionally, some parsers introduce a node more than once, which results in two triples $\langle x, :instance, a \rangle$ and $\langle x, :instance, b \rangle$. Currently, SMATCH removes all such introductions of a second concept, but does not remove duplicate triples. By contrast, we propose to remove all duplicate triples, since they have no clear semantics, and stay agnostic to second introductions of a concept (in some MRs, it may be acceptable that an entity is the instance of two concepts), keeping all such triples (if they are not identical).⁹

6 Module II: Alignment

The goal of this module is solving Eq. 2, finding a map^* for optimal matching score.

SMATCH uses a hill-climber for solving Eq. 2. An issue with this is that such a heuristic terminates at local optima and cannot provide us with any *upper-bounds*. Upper-bounds, however, can inform users about the *quality* of the outputted solution and thus increase the trustworthiness of the final score (and any parser comparison that is based thereupon). Therefore, we can conclude that using a hill-climber seems **practical but may not be optimal**, especially when considering cases where fair comparison needs to be *guaranteed*. Instead, we would like to use an Integer Linear Program (ILP) to obtain the (optimal) solution. Alternatively, at least, we would like to know a tight upper-bound to inform ourselves about the trustworthiness of

⁸I.e., since reification can potentially be used to model n -ary relations, only in the case where $n = 2$ we can model the structure with a single (labelled) edge

⁹Due to rare occurrence of such phenomena in our parsed data, we find the effects of either choice to be negligible.

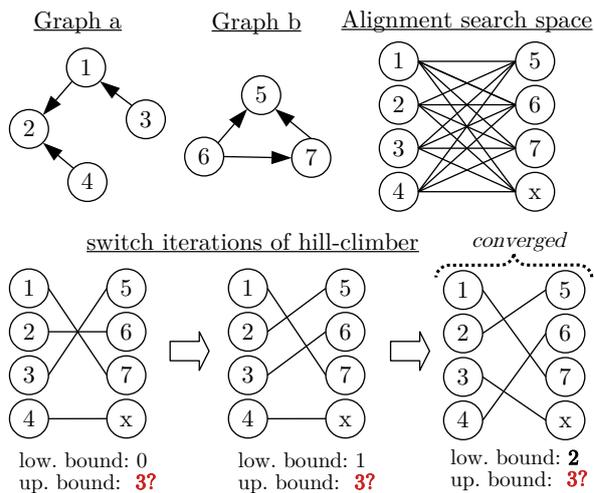


Figure 3: Sketch of search space (top) and hill-climber run (bottom). Every hill-climber step constitutes an improved lower bound, but we cannot obtain a tight upper-bound (an accessible trivial upper-bound is the amount of triples in the smaller of two graphs: 3).

the final score. But ILP is NP hard, and therefore it seems **optimal but possibly not practical**, a conception that might favor the usage of a hill-climber.

Triggered by these considerations, we review the hill-climber and the ILP and assess their effects on MR evaluation, with two desiderata in mind: evaluation quality and efficiency. Additionally, we propose a strategy for loss-less MR compression that can improve efficiency of any solver.

6.1 Practical but not optimal: hill-climber

SMATCH hill-climbing uses two operations, which we denote as *switch*, and *assign*. The *assign*-operation assigns a variable from $vars(a)$ to an unaligned variable from $vars(b)$: $(i, \emptyset) \rightarrow (i, j = map'(i))$, where map' is a candidate map. The *switch* operation does an alignment cross-over with respect to two alignment pairs, i.e.: $(i, j = map(i)) \wedge (k, l = map(k)) \rightarrow (i, l = map'(i)) \wedge (k, j = map'(k))$, where map is the current alignment and map' the candidate alignment. In each iteration, we examine all possible *switch*- and *assign* options, and greedily choose the best one.¹⁰ An example alignment procedure is shown in Figure 3.

In practice, we can resort to multiple random restarts, to find better optima. However, this hardly

¹⁰*Assign* is just a special instance of the more general *switch* so we can ablate the *assign* step. Then, *assign* becomes $(i, \emptyset = map(i)) \wedge (k, j = map(k)) \rightarrow (i, j = map'(i)) \wedge (k, \emptyset = map'(k))$, which is a *switch*.

addresses the underlying issue: we lack any information on upper-bounds, which may decrease trustworthiness of results, especially when facing larger graphs with lots of local optima.

6.2 ILP: Optimal, but less practical?

We would like to use Integer Linear Programming (ILP) for optimal solution of the graph alignment.

Problem statement Assume two graphs g, g' with node sets V, V' . Let $u(i, j)$ denote the amount of unary triple matches, given we align i from V to j from V' , counting matches of triples that involve one MR variable. On the other hand, $b(i, j, k, l)$ will denote the amount of structural binary triple matches, given we align i from V to j from V' and k from V to l from V' . Here, we count matching binary triples that involve two MR variables. Usually, these data are pre-computed. Let x indicate our current *map*, i.e., if $x_{ij} = 1$ then we align i from V to j from V' . We find our solution at

$$\begin{aligned} \max \quad & \sum_{\substack{(i,j) \in \\ V \times V}} u(i, j)x_{ij} + \sum_{\substack{(i,j,k,l) \in \\ (V \times V)^2}} b(i, j, k, l)x_{ij}x_{kl} \\ \text{st} \quad & \sum_j x_{ij} \leq 1; \quad \sum_i x_{ij} \leq 1 \\ & x_{ij} \in \{0, 1\} \quad \forall (i, j) \in V \times V' \end{aligned}$$

The constraint ensures that every node from one graph is aligned, at maximum, to one node from the other graph. By linearization, and introducing structural variables y , we obtain the equivalent ILP:

$$\begin{aligned} \max \quad & \sum_{\substack{(i,j) \in \\ V \times V'}} u(i, j)x_{ij} + \sum_{\substack{(i,j,k,l) \in \\ (V \times V')^2}} b(i, j, k, l)y_{ijkl} \\ \text{st} \quad & \sum_j x_{ij} \leq 1; \quad \sum_i x_{ij} \leq 1 \\ & y_{ijkl} \leq x_{ij}, \quad \forall (i, j, k, l) \in (V \times V')^2 \\ & y_{ijkl} \leq x_{kl}, \quad \forall (i, j, k, l) \in (V \times V')^2 \\ & x_{ij} \in \{0, 1\} \quad \forall (i, j) \in V \times V' \\ & y_{ijkl} \in \{0, 1\} \quad \forall (i, j, k, l) \in (V \times V')^2, \end{aligned}$$

where the structural variables, if active, show us countable binary triple matches. This is an NP complete problem, imposing limits on its capability to provide us with optimal solutions for larger graphs (note, however, that we can retrieve intermediate solutions and upper-bounds).

6.3 Reduced search space with lossless graph compression

We observe that in an MR a , every variable $x \in vars(a)$ is related to a concept c , e.g., $\langle x, :instance, cat \rangle$. This means that a concept c does *identify* a variable $x \in vars(a)$ iff $\forall y \in vars(a) : \langle y, :instance, c \rangle \Rightarrow y = x$. Therefore, if x denotes a *cat*, and there is no other entity in the MR that also denotes a *cat*, then x may be referred to simply by *cat*. This carries over to pairs of MRs: which are the focus of the paper – instead of considering $vars(a)$, we simply consider $vars(a) \cup vars(b)$. Therefore, we can replace all n variables from $vars(a) \cup vars(b)$ that are *identified* by concepts, with the corresponding concepts (see Appendix A.1 for a full example). This shrinks the search space by reducing the amount of variables that the optimizer has to consider. Note that such a compression is *lossless*, in the sense that the possibility of full reconstruction of the original MR is ensured. This implies that if two compressed MRs are assessed as (non-)isomorphic, then the uncompressed MRs are also (non-)isomorphic.

6.4 Solver experiments

Two questions are of main interest: 1. *RQ1, solution quality*: (How) do the final SMATCH results depend on the solver? 2. *RQ2, solution efficiency*: How does the evaluation time depend on the solver? In addition, we would like to assess how our answers to RQ1 and RQ2 might be affected by reification (resulting in a bigger search space) and MR compression (resulting in a smaller search space).

Setup We simulate a standard AMR parsing evaluation setting. We parse the LDC2017T10 testing data with six parsers: \mathcal{P}_1 (Xu et al., 2020), \mathcal{P}_2 (Cai and Lam, 2020), \mathcal{P}_3 (Lindemann et al., 2020), \mathcal{P}_4 (Zhang et al., 2019), \mathcal{P}_5 (Lyu and Titov, 2018), \mathcal{P}_6 (Cai and Lam, 2019). We evaluate the parsers using ILP or hill-climber (denoted by \triangle). As is standard, we show F1 micro corpus scores. For reference, we also run evaluation with the standard SMATCH hill-climbing script (denoted as *previous*). We observe that we successfully reproduce the scores from the standard SMATCH script with our \triangle implementation (first two lines of Table 2).¹¹

		parser scores (ranked)							time	# vars	quality
		\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{P}_5	\mathcal{P}_6	secs	(tot., avg., max.)	(yield, bound)	
all vars	basic	prev.	81.4 ₍₁₎	80.3 ₍₂₎	77.0 ₍₃₎	76.3 ₍₄₎	74.5 ₍₅₎	73.1 ₍₆₎	50.4	(20346, 15, 129)	(217702, +?)
	basic	\triangle_4	81.4 ₍₁₎	80.3 ₍₂₎	77.0 ₍₃₎	76.2 ₍₄₎	75.1 ₍₅₎	73.1 ₍₆₎	49.9	see above	(217716, +?)
	basic	ILP	81.5 ₍₁₎	80.4 ₍₂₎	77.1 ₍₃₎	76.5 ₍₄₎	75.2 ₍₅₎	73.3 ₍₆₎	98.0	see above	(218072, +0)
	reify	\triangle_4	82.8 ₍₁₎	81.3 ₍₂₎	78.3 ₍₃₎	77.7 ₍₄₎	76.9 ₍₅₎	74.7 ₍₆₎	134.5	(27812, 20, 174)	(288597, +?)
	reify	ILP	83.5 ₍₁₎	82.1 ₍₂₎	79.3 ₍₃₎	78.7 ₍₄₎	77.7 ₍₅₎	75.8 ₍₆₎	300.5	see above	(291370, +13)
	<hr/>										
compress	basic	\triangle_1	72.9 ₍₁₎	70.9 ₍₂₎	67.1 ₍₃₎	66.2 ₍₄₎	64.6 ₍₅₎	61.5 ₍₆₎	7.3	(5568, 4, 62)	(74163, +?)
	basic	ILP	73.3 ₍₁₎	71.3 ₍₂₎	67.5 ₍₃₎	66.3 ₍₄₎	65.0 ₍₅₎	62.1 ₍₆₎	11.7	see above	(75036, +0)
	reify	\triangle_1	74.9 ₍₁₎	72.8 ₍₂₎	69.5 ₍₃₎	68.7 ₍₄₎	67.7 ₍₅₎	64.6 ₍₆₎	31.4	(10704, 8, 106)	(124323, +?)
	reify	ILP	76.7 ₍₁₎	74.1 ₍₂₎	71.3 ₍₃₎	70.6 ₍₄₎	69.5 ₍₅₎	66.4 ₍₆₎	27.3	see above	(129019, +0)

Table 2: Parser evaluation. *time* refers to the approximate total time needed to evaluate a single parser (i.e., processing 1371 graph pairs). $\triangle-N$ indicates hill-climber optimizer with N restarts. *quality*: solution quality of solver – first number is the amount of matching triples summed over all six parser evaluations (yield); second number indicates the tightest found upper-bound (which is only known by ILP).

6.4.1 RQ1: solution quality

Insight: Better alignment → safer evaluation

Importantly, we see that the ILP yields score increments for all parsers, which signals the occurrence of alignment problems, where the \triangle (despite multiple restarts) did not find the optimal solution. The effect-size is larger for reified graphs. We find differences of up to 1 point F1 score (Table 2: reify \triangle_4 vs. reify ILP). This can be explained by the growth of the alignment search space – reification makes graphs larger and introduces more MR variables. This explanation is further supported by contrasting the amount of unique final objective values against the size of the alignment space with different random initializations of the hill-climber (Appendix A.2, Figure 6). We see that i) for many graph pairs there are multiple local optima, and ii) the likelihood of finding a non-global optimum with the \triangle increases for larger/reified graphs.

We further study upper-bounds and solution quality (right column of Table 2). The ILP found the optimal solution in all cases, yielding 218072 matching triples. The \triangle_4 finds 217,700 matching triples (99.83%), which misses the mark by 350 triples. When evaluating reified graphs, the ILP returns 291370 matches and thus misses its temporary tightest upper-bound by 13 triples, indicating that in a few cases, a sub-optimal solution might have been found.¹² The \triangle_4 , however, yields only

288,597 matches (99.04%) and misses the temporary ILP upper-bound by 2,786. The growing gap underlines the degrading quality of the hill-climber when facing larger graphs.

Finally, the (slight) *differences* in increments among parsers when we evaluate them on reified graphs indicate that different parsers do make different decisions on when to reify an edge. For instance the score difference Δ for reified graphs vs. non-reified graphs (using ILP) of $\mathcal{P}_5, \mathcal{P}_6$ is 2.5 points, for \mathcal{P}_1 2 points and for \mathcal{P}_2 1.7 points. This supports our theoretical insights from §5.2 – reification can make parser comparison fairer.

6.4.2 RQ2: Solution efficiency

Insight I: ILP isn’t that impractical It seems to be commonly presumed that original SMATCH uses a hill-climber to make evaluation more practical and fast. However, our results qualify this presumption. For evaluating a full corpus (1371 graph pairs), SMATCH with ILP needs only about 48 seconds longer than original SMATCH with hill-climber (50s vs 98s). When the search space grows (due to reification) the time gap widens to a difference of 165 seconds. However, the consistent improvement of scores due to ILP (signaling sub-optimal hill-climber solutions) can make the time increase acceptable for evaluations where fairness is critical.

Insight II: MR compression increases evaluation speed

Viewing the last four rows of Table 2, we see that the MR compression i) did not lead to switched system ranks and ii) increased the evaluation speed by a large factor. Using MR compression, the ILP runs a full system evaluation in 11.7

¹¹An improvement is obtained for \mathcal{P}_5 . We find that we can mostly attribute this to a bug in the original script that prevents proper graph reading of some parses of \mathcal{P}_5 .

¹²Indeed, we find one graph by \mathcal{P}_2 , and one graph by \mathcal{P}_6 , where the ILP terminates after a 240s timeout that we set, and returns a temporary solution.

avg.	parser scores					
	\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{P}_5	\mathcal{P}_6
mic.	81.5 ^{81.1} _{80.7}	80.4 ^{81.2} _{79.6}	77.1 ^{77.8} _{76.2}	76.5 ^{77.2} _{75.6}	75.2 ^{75.8} _{74.5}	73.3 ^{74.1} _{72.4}
mac.	82.6 ^{81.8} _{83.3}	81.4 ^{82.1} _{80.7}	79.0 ^{79.8} _{78.2}	78.3 ^{79.1} _{77.5}	76.2 ^{77.0} _{75.4}	75.9 ^{75.0} _{76.6}

Table 3: Evaluation with additional macro statistics and confidence intervals. Solver: ILP.

seconds for standard graphs and 27.3 seconds for the reified graphs. Given that the MR compression is lossless (c.f. §6.2), it provides us with an option for more efficient evaluation that is also safe (i.e., optimal).

7 Module III: scoring

7.1 Main scores: Precision, Recall and F1

The goal of this module is to provide the user with a final result. As discussed in §3, the main scores (Precision, Recall, and F1) follow directly from the map^* . The final score is typically micro averaged, summing matching statistics across all graph pairs before they are normalized. SMATCH++ makes two additions, macro-scoring and confidence intervals. Macro-averaging scores over graph pairs can be a useful complementary signal, specifically when comparing high-performance parsers (Opitz and Frank, 2022a). Additionally, we adopt the bootstrap assumption (Efron, 1992) for calculating confidence intervals. To make calculation feasible, bootstrapping is performed after the alignment stage. Table 3 shows results of the additional statistics. Confidence intervals range between $\pm[0.5, 1]$ points for all parsers. Macro score shows an outlier, where \mathcal{P}_6 (+2.6 points) is more positively affected than other parsers ($\pm[1.0, 1.9]$ points).¹³

7.2 Measuring aspectual semantic similarity

We observe considerable interest in applying fine-grained aspectual MR metrics (Damonte et al., 2017) for inspecting linguistic aspects captured by MRs (e.g., semantic roles, negation, etc.). Applications range from parser diagnostics (Lyu and Titov, 2018; Xu et al., 2020; Bevilacqua et al., 2021; Martínez Lorenzo et al., 2022), to NLG system diagnostics and sentence similarity (Opitz and Frank, 2021, 2022b). Formally, given an aspect of interest asp and an MR g , we apply a sub-graph-extraction

¹³We find a potential explanation in a motivation of \mathcal{P}_6 's creators to focus on semantics in proximity of an MR's top node (the proportion of such semantics increases when the graph is smaller, and smaller graphs have more influence on macro average than on micro average).

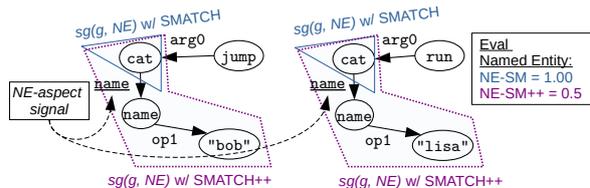


Figure 4: Named Entity (NE) sub-graph extraction with SMATCH vs. SMATCH++

function $sg(g, asp)$ to build an aspect-focused sub-graph, and compute a matching score (e.g., F1).

Review of previous implementation We study the description in Damonte et al. (2017) and the most frequently used implementation (Lyu, 2018). The treated aspects¹⁴ are divided in two broad groups: **i) alignment-based matching:** For some aspects, we extract aspect-related genuine sub-graphs, on which we calculate an optimal alignment. **ii) bag-of-label matching:** for other aspects, we detect aspect-related variables and gather associated node labels¹⁵ in a bag/list, to compute an overlap score based on simple set intersection.

E.g., *SRL*-aspect belongs to the first category **(i)**: we extract $\langle x, :arg_n, y \rangle$ relations, and their corresponding *instance* triples (here: $\langle x, :instance, c \rangle$, and $\langle y, :instance, c' \rangle$). Then we calculate SMATCH on such SRL-subgraphs. The *Negation*, *Named Entity* (NEs) and *Frames* aspect is put into the second group **(ii)**. We look for a relation/node-label that signals a particular aspect, e.g., $\langle x, :polarity, - \rangle$ (for negation) or $\langle x, :name, y \rangle$ (for NEs), we extract x , and replace x with the descriptive label c from $\langle x, :instance, c \rangle$. For *Frames*, we search for $\langle x, :instance, c \rangle$ where c is a PropBank predicate, and collect c . Finally, we can evaluate without an alignment, using set intersection.

Open questions We pose two questions:

1. Can the sub-graph extraction be improved?
2. Are there other aspects that we can measure?

7.3 Improving sub-graph extraction

Sensible range of extraction For some phenomena, the current extraction range is clearly too limited. For instance, let us consider named entities, which can be captured in more complex and nested

¹⁴See Appendix A.3 for a full overview.

¹⁵I.e., from $\langle x, :instance, label \rangle$ triples

MR structure. E.g., in AMR, one node typically indicates the type of the named entity (NE), and another multi-node structure represents its name and other attributes. Consider two AMRs a and b , from which we want to extract NE structures to measure the agreement of the graphs w.r.t. NE similarity. As shown in Figure 4, assume that one graph is about *a cat named Bob*¹⁶, while the other graph is about *a cat named Lisa*¹⁷. Obviously, the MRs have similarities in their NE structure (since there are named cats), but also differences (since the cats have different names). However, NE-focused SMATCH only extracts *cat* and *cat*, and returns maximum score.

Hence, for all finer-grained aspects that are captured by non-atomic MR structures (e.g., Named Entities), we propose to gather the full sub-graph starting at the aspect-indicating relation or node label. In the NE example, as shown in Figure 4, we would be provided a score of 0.5, better reflecting the similarity of the two NE structures.

Sub-graph compression, align and match We find a middle-ground in the advantages of the coarse matching (concreteness, efficiency) and graph alignment (safe matching) by using alignment with lossless MR compression. This is optimal and efficient, and alleviates the need to switch among fine and coarse extraction methods.

7.4 Extending fine-grained scores

Beyond negation and named entities – other semantic aspects We find that the fine-grained SMATCH metrics by Damonte et al. (2017) miss some interesting features captured by MRs. For instance, four interesting AMR aspects that are currently not captured are *cause*, *location*, *quantification*, and *tense*. SMATCH++ allows their integration in a straightforward way. An example for tense extraction is displayed in Figure 5, where our SMATCH++ sub-graph extraction extracts the complete temporal sub-graph, triggered by the edge label `:time` (if we would resort to the style of fine-grained SMATCH, we would miss larger parts of the temporal structure, only extracting the node label *end*).

Results of fine-grained parser diagnostics for *cause*, *location*, *quantification*, and *tense* are shown

¹⁶Triples: `<x, :instance, cat>`, `<y, :instance, name>`, `<x, :name, y>`, `<y, :opl, "bob">`.

¹⁷Triples: `<x, :instance, cat>`, `<y, :instance, name>`, `<x, :name, y>`, `<y, :opl, "lisa">`.

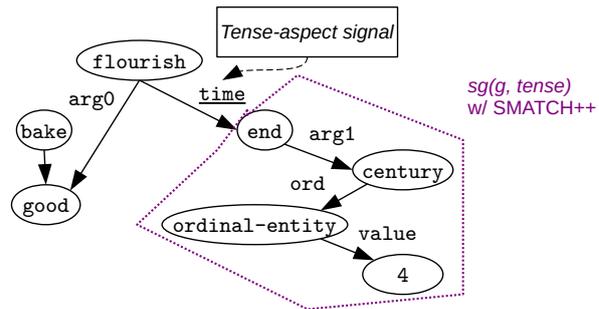


Figure 5: Temporal sub-graph extraction with SMATCH++ for an MR capturing “Baked goods flourished at the end of the fourth century”.

aspect	parser scores					
	\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{P}_5	\mathcal{P}_6
cause	47.8	47.4	44.4	35.7	31.4	31.2
location	61.8	53.2↓	54.7↑	49.2↓	51.7↑	40.0
quant	69.4	67.4	58.4	56.8	56.5	55.8
tense	67.7	62.3	58.5	56.5	50.3	48.4

Table 4: Evaluation for *causal* and *temporal* structures. ↓↑ indicate switched ranks. Solver: ILP.

in Table 4.

Interestingly, we see that projecting *causality* seems hard: all parsers tend to struggle when assessing causal structures (31.2 up to 47.8 F1 points), showing much room for improvement. The temporal structures, on the other hand, can be assessed with somewhat higher accuracy (48.4 up to 67.7 points). We also see some switched ranks, indicating different parser strengths. Overall, parser score differences seem notably more pronounced than when calculating SMATCH (++) on the full graphs, showing the difficulty of capturing finer phenomena, and highlighting strengths of more recent parsers.

8 Conclusion

SMATCH++ is the first specification of a standardized, extended, and extensible SMATCH metric. We aim at i) standardized and transparent comparison of graph parsing systems, and ii) improved extensibility for custom applications.¹⁸ The applications can include finer parser diagnostics and measuring semantic sub-graph similarities such as *quantification*, *cause*, or *tense* with our fine-grained metrics.

Acknowledgments

We thank our reviewers for their helpful feedback.

¹⁸See Appendix A.4 for a summary of the default setup.

Limitations

We have to leave some questions open. First, we would have liked to shed more light on the solvers' behaviors when facing large graphs, in isolation. On one hand, our benchmark corpus indeed contains some large MRs with many variables, including reified MRs and MRs that represent multiple sentences (up to 174 variables, cf. Table 2). We have shown that ILP could cope with these harder problems, providing optimal solutions in reasonable time. When facing bigger graphs, however, we can expect that the solution quality of the hill-climber quickly degrades, while the ILP will struggle to find optimal solutions. While our graph compression strategy can help mitigate this issue by reducing the alignment search space, it would be interesting to study the quality of temporary solutions, or of solutions of LP relaxation. There are also relaxed ILP solvers (Klau, 2009) that iteratively tighten the lower and the upper-bound. They could prove useful for aligning larger MR graphs, or, at least, to find useful upper-bounds.

Second, in this paper we studied SMATCH (++) that measures *structural overlap* and assigns each triple the *same weight*. But structural differences of similar degree can have a different impact on overall meaning similarity as perceived by humans, which can have ramifications for measuring sentence similarity (Opitz et al., 2021) and meaningful evaluation of strong AMR parsers (Opitz and Frank, 2022a). Therefore, for a deeper assessment of MR similarity we may have to use conceptually different metrics, or explore SMATCH++-based strategies and (sensibly) weigh triples depending on label importance or compose an overall score by weighting measured sub-aspect similarities.

References

- Rafael Torres Anchieta, Marco Antonio Sobrevilla Cabezudo, and Thiago Alexandre Salgueiro Pardo. 2019. Sema: an extended semantic evaluation for amr. In *(To appear) Proceedings of the 20th Computational Linguistics and Intelligent Text Processing*. Springer International Publishg.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. [Abstract meaning representation for sembanking](#). In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2019. [Amr guidelines](https://github.com/amrisi/amr-guidelines/blob/master/amr.md). <https://github.com/amrisi/amr-guidelines/blob/master/amr.md>.
- Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*.
- Deng Cai and Wai Lam. 2019. [Core semantic first: A top-down approach for AMR parsing](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3799–3809, Hong Kong, China. Association for Computational Linguistics.
- Deng Cai and Wai Lam. 2020. [AMR parsing via graph-sequence iterative inference](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online. Association for Computational Linguistics.
- Shu Cai and Kevin Knight. 2013. [Smatch: an evaluation metric for semantic feature structures](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.
- Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. [An incremental parser for Abstract Meaning Representation](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.
- Bradley Efron. 1992. *Bootstrap methods: another look at the jackknife*. Springer.
- Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. [A discriminative graph-based parser for the Abstract Meaning Representation](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.
- Michael Wayne Goodman. 2019. [AMR normalization for fairer evaluation](#). *CoRR*, abs/1909.01568.
- Thanh Lam Hoang, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam Nguyen, Dzung Phan, Vanessa López, and Ramon Fernandez Astudillo. 2021a. Ensembling graph predictions for amr parsing. *Advances in Neural Information Processing Systems*, 34.
- Thanh Lam Hoang, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam Nguyen, Dzung Phan, Vanessa Lopez, and Ramon Fernandez Astudillo. 2021b. [Ensembling](#)

- graph predictions for amr parsing. In *Advances in Neural Information Processing Systems*, volume 34, pages 8495–8505. Curran Associates, Inc.
- Alexander Miserlis Hoyle, Ana Marasović, and Noah A. Smith. 2021. Promoting graph awareness in linearized graph-to-text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 944–956, Online. Association for Computational Linguistics.
- Gunnar W Klau. 2009. A new graph-based method for pairwise global network alignment. *BMC bioinformatics*, 10(1):1–9.
- Zi Lin, Jeremiah Liu, and Jingbo Shang. 2022. Neural-symbolic inference for robust autoregressive graph parsing via compositional uncertainty quantification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4759–4776, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2020. Fast semantic parsing with well-typedness guarantees. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3929–3951, Online. Association for Computational Linguistics.
- Chunchuan Lyu. 2018. Fine-grained smatch implementation. <https://github.com/ChunchuanLv/amr-evaluation-tool-enhanced>.
- Chunchuan Lyu and Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 397–407, Melbourne, Australia. Association for Computational Linguistics.
- Emma Manning and Nathan Schneider. 2021. Referenceless parsing-based evaluation of AMR-to-English generation. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 114–122, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Abelardo Carlos Martínez Lorenzo, Marco Maru, and Roberto Navigli. 2022. Fully-Semantic Parsing and Generation: the BabelNet Meaning Representation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1727–1741, Dublin, Ireland. Association for Computational Linguistics.
- Jonathan May and Jay Priyadarshi. 2017. Semeval-2017 task 9: Abstract meaning representation parsing and generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 536–545.
- Stephan Open, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O’Gorman, Nianwen Xue, and Daniel Zeman. 2020. Mrp 2020: The second shared task on cross-framework and cross-lingual meaning representation parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22.
- Juri Opitz, Angel Daza, and Anette Frank. 2021. Weisfeiler-leman in the bamboo: Novel AMR graph metrics and a benchmark for AMR graph similarity. *Transactions of the Association for Computational Linguistics*, 9:1425–1441.
- Juri Opitz and Anette Frank. 2021. Towards a decomposable metric for explainable evaluation of text generation from AMR. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1504–1518, Online. Association for Computational Linguistics.
- Juri Opitz and Anette Frank. 2022a. Better Smatch = better parser? AMR evaluation is not so simple anymore. In *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*, pages 32–43, Online. Association for Computational Linguistics.
- Juri Opitz and Anette Frank. 2022b. SBERT studies meaning representations: Decomposing sentence embeddings into explainable semantic features. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 625–638, Online only. Association for Computational Linguistics.
- Juri Opitz, Letitia Parcalabescu, and Anette Frank. 2020. AMR Similarity Metrics from Principles. *Transactions of the Association for Computational Linguistics*, 8:522–538.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Leonardo F. R. Ribeiro, Yue Zhang, and Iryna Gurevych. 2021. Structural adapters in pretrained language models for AMR-to-Text generation. In *Proceedings of the 2021 Conference on Empirical Methods*

- in *Natural Language Processing*, pages 4269–4282, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Linfeng Song and Daniel Gildea. 2019. **SemBleu: A robust metric for AMR parsing evaluation**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4547–4552, Florence, Italy. Association for Computational Linguistics.
- Elias Stengel-Eskin, Aaron Steven White, Sheng Zhang, and Benjamin Van Durme. 2020. **Universal decompositional semantic parsing**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8427–8439, Online. Association for Computational Linguistics.
- Matteo Togninalli, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt. 2019. **Wasserstein weisfeiler-lehman graph kernels**. In *Advances in Neural Information Processing Systems*, volume 32, pages 6436–6446. Curran Associates, Inc.
- Sarah Uhrig, Yoalli Garcia, Juri Opitz, and Anette Frank. 2021. **Translate, then parse! a strong baseline for cross-lingual AMR parsing**. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 58–64, Online. Association for Computational Linguistics.
- Rik van Noord, Lasha Abzianidze, Hessel Haagsma, and Johan Bos. 2018. **Evaluating scoped meaning representations**. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).
- Shira Wein, Wai Ching Leung, Yifu Mu, and Nathan Schneider. 2022. **Effect of source language on AMR structure**. In *Proceedings of the 16th Linguistic Annotation Workshop (LAW-XVI) within LREC2022*, pages 97–102, Marseille, France. European Language Resources Association.
- Shira Wein and Nathan Schneider. 2022. **Accounting for language effect in the evaluation of cross-lingual AMR parsers**. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3824–3834, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Boris Weisfeiler and Andrei Leman. 1968. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16.
- Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2020. **Improving AMR parsing with sequence-to-sequence pre-training**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2501–2511, Online. Association for Computational Linguistics.
- Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. **AMR parsing as sequence-to-graph transduction**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy. Association for Computational Linguistics.
- Sheng Zhang, Xutai Ma, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2018. **Cross-lingual decompositional semantic parsing**. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1664–1675, Brussels, Belgium. Association for Computational Linguistics.

A Appendix

A.1 Lossless graph pair reduction example

Consider *the cat scratches another cat*:

$$a = \{ \langle s, :instance, scratch \rangle, \langle c, :instance, cat \rangle, \langle d, :instance, cat \rangle, \langle s, :arg0, c \rangle, \langle s, :arg1, d \rangle \}$$

and *the gray cat scratches the small plant*:

$$b = \{ \langle x, :instance, scratch \rangle, \langle y, :instance, cat \rangle, \langle z, :instance, plant \rangle, \langle w, :instance, small \rangle, \langle v, :instance, gray \rangle, \langle x, :arg0, y \rangle, \langle x, :arg1, z \rangle, \langle y, :mod, v \rangle, \langle z, :mod, z \rangle \}$$

The lossless compression is $a' = \{ \langle c, :instance, cat \rangle, \langle d, :instance, cat \rangle, \langle scratch, :arg0, c \rangle, \langle scratch, :arg1, d \rangle \}$ and $b' = \{ \langle y, :instance, cat \rangle, \langle scratch, :arg0, y \rangle, \langle scratch, :arg1, plant \rangle, \langle y, :mod, gray \rangle, \langle plant, :mod, small \rangle \}$.

The alignment search space is reduced from a size of more than 100 candidates to 2 candidate options ($y = c$, or $y = d$).

A.2 Assessing solution quality variability in dependence of variables

We use the parses of an example parser (\mathcal{P}_5)¹⁹. For every evaluation pair, we re-start the hillclimber 20 times, and collect the scores related to the found local optima. The Y-axis in Figure 6 shows the amount of unique scores found among the 20 tries (note that there could be more unique alignments

¹⁹We ran the experiment also with parses from other systems but always ended up with essentially the same results

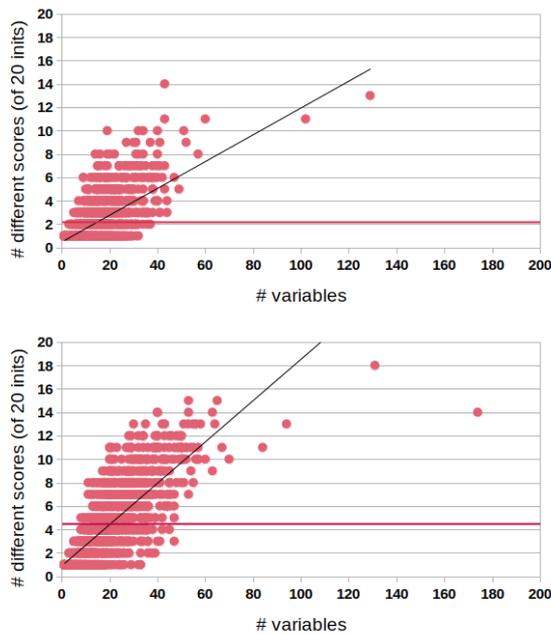


Figure 6: Assessing solution quality variability. Top: basic graphs, bottom: reified graphs. Diagonal line: linear trend. Horizontal line: arithmetic mean. See text in §A.2 for more description and §6.4.1 for discussion.

that would result in the same score – this is not captured in this Figure). The X-axis shows the amount of alignment variables. In different terms, a higher point in this Figure is equivalent to a larger pool of local optima of different quality, and thus we can conjecture a greater likelihood that the optimal solution is not returned by the hill-climber.

A.3 Aspect overview

Previously measured aspects For all aspects we retrieve F1, Precision, and Recall.

1. Measured under alignment

- (a) SRL: extract $\langle x, \text{arg}_n, y \rangle$ triples and corresponding instance triples.
- (b) Coreference/Re-entrancies: extract $\langle x, \text{rel}, y \rangle$ triples for which there is another triple $\langle z, \text{:rel}', y \rangle$ (meaning y is a re-entrant node) and also extract corresponding instance triples.

2. Measured via bag-of-structure extraction and set operations

- (a) Concepts: collect all node labels.
- (b) Frames: collect all node labels where the label is a PropBank predicate frame.
- (c) NonSenseFrames: see above, but with sense label removed

- (d) NE: Named entities, collect all node labels that have an outgoing `:name` relation.
- (e) Negation: collect all node labels that have an outgoing `:polarity` relation.
- (f) Wikification: collect all node labels that have an incoming `:wiki` relation.
- (g) IgnoreVars: replace all variables in triples with concepts, collect triples.

SRL, Named Entities, coreference (re-entrant nodes)

Additional aspects measured by us: *Cause*, *Tense*, *Location*, *Quantifier*.

We change: Add default option for extracting aspect sub-graphs, measure all aspects under alignment.

Aspects we added:

- *Cause*: Cause is modeled via `cause-01`. We extract label of `:arg1` (what is caused?) and subgraph of `:arg2`, the cause itself.
- *tense*: Tense is modeled via $\langle x, \text{:time}, y \rangle$ edge. We extract label of the thing that happens and subgraph of y , the temporal description where it happens.
- *location*: Similar to above but with `:location` edge.
- *quantifier*: Similar to above but with `:quant` edge.

A.4 Best practice

To provide a balance between efficiency, safety and meaningfulness of scores, default procedure of SMATCH++ is currently set to:

1. Pre-processing: lower-casing, duplicate-removal, de-reify where applicable.
2. Alignment: Solver: ILP. Triple-match: $w_t^t = 1 \forall t, t'; \text{sim}(c, c') := I[c = c']$
3. Scoring: Precision, Recall, F1, Bootstrap confidence intervals

An option to increase efficiency without incurring a loss in safety and meaningfulness is achieved by adding graph compression to the pre-processing. It is set as the default for fine semantic aspect scores. Also, to ensure utmost safety, we have to consider applying reification standardization (incurring a significantly longer evaluation time).

An Extended Sequence Tagging Vocabulary for Grammatical Error Correction

Stuart Mesham[♣] Christopher Bryant[◇] Marek Rei^{♡◇} Zheng Yuan^{♣◇}

[♣]Department of Computer Science, University of Cambridge

[◇]The ALTA Institute, Department of Computer Science, University of Cambridge

[♡]Department of Computing, Imperial College London

[♣]Department of Informatics, King's College London

sm2613@cantab.ac.uk, christopher.bryant@cl.cam.ac.uk

marek.rei@imperial.ac.uk, zheng.yuan@kcl.ac.uk

Abstract

We extend a current sequence-tagging approach to Grammatical Error Correction (GEC) by introducing specialised tags for spelling correction and morphological inflection using the SymSpell and LemmInflect algorithms. Our approach improves generalisation: the proposed new tagset allows a smaller number of tags to correct a larger range of errors. Our results show a performance improvement both overall and in the targeted error categories. We further show that ensembles trained with our new tagset outperform those trained with the baseline tagset on the public BEA benchmark.

1 Introduction

Current approaches to Grammatical Error Correction (GEC) fall under two broad categories: sequence-to-sequence and sequence-tagging. The former treats GEC as a machine-translation problem, "translating" from error-containing to error-free language (Yuan and Briscoe, 2016; Schmaltz et al., 2017; Junczys-Dowmunt et al., 2018; Grundkiewicz et al., 2019; Yuan et al., 2019; Rothe et al., 2021). By contrast, sequence-tagging approaches tag each input word with an edit operation such that applying the operations produces the corrected output (Yannakoudakis et al., 2017; Awasthi et al., 2019; Omelianchuk et al., 2020; Tarnavskyi et al., 2022). The basic operations include keeping a word unchanged, deleting a word, and inserting new words (Awasthi et al., 2019; Malmi et al., 2019).

One advantage of sequence-tagging over sequence-to-sequence approaches is computational efficiency: the former do not require expensive auto-regressive decoding,¹ and currently achieve competitive performance using smaller models (Tarnavskyi et al., 2022; Rothe et al., 2021).

¹Malmi et al. (2019) show that sequence-taggers can be orders of magnitude faster than comparable seq-to-seq models at inference time.

the **serendipitis** discovery of penicillin
\$KEEP \$SPELL \$KEEP \$KEEP \$KEEP

↓ SymSpell
serendipitous

It was **easy** than taming a dragon
\$KEEP \$KEEP \$INFLECT_JJR \$KEEP \$KEEP \$KEEP \$KEEP

↓ LemmInflect
easier

Figure 1: Our model applied to two inputs. Beneath each word is the tagger's output. Arrows denote transformations by SymSpell and LemmInflect respectively.

However, current sequence-tagging approaches require manual linguistic efforts to curate language-specific edit tags (Yuan et al., 2021). For example, Awasthi et al. (2019) introduce rule-based morphological inflection tags, like replacing the "-ing" suffix with "-ion" (e.g. completing → completion). Omelianchuk et al. (2020) introduce a wider range of operations including verb-form and noun-number changes. For verb-form inflections, they use a dictionary to map between verb forms.²

In this paper, we focus on a sequence-tagging approach. We extend the approach of Omelianchuk et al. (2020) by introducing more general transformation tags (Figure 1). Specifically, we introduce:

- A tag for correcting spelling errors.
- Inflection tags capable of a broader range of inflections than the tags introduced by Omelianchuk et al. (2020).

These modifications allow a broader range of errors to be handled by a smaller number of transformation tags, which simplifies the sequence tagging

²https://github.com/gutfeeling/word_forms/blob/master/word_forms/en-verbs.txt

problem, as well as improves the generalisation of the GEC system. Our results show that our modifications improve the system’s performance on the BEA-2019 development and test sets. Our code and model weights are publicly available.³

2 Methods

We extend the system described by Omelianchuk et al. (2020) by adding new tags to the model’s output vocabulary and modifying the inference and dataset preprocessing code to support our new tags. Our new tags perform spelling correction and morphological inflection and are described in Sections 2.3 and 2.4 below. We evaluate our tagset using the RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021b), DeBERTaV3 (He et al., 2021a), ELECTRA (Clark et al., 2020) and XLNet (Yang et al., 2019) encoders, as well as an ensemble of three encoders (see Section 2.6).

2.1 Model and Training Procedure

Our work builds on GECToR from Omelianchuk et al. (2020), which follows the sequence tagging approach to GEC. We use the same sequence tagger architecture: a pre-trained transformer encoder with two separate "tagging" and "detection" heads. We also follow the same multi-phase training procedure using the synthetic PIE Corpus (Awasthi et al., 2019), NUCLE (Dahlmeier et al., 2013), FCE (Yan-nakoudakis et al., 2011), Lang-8 (Mizumoto et al., 2011; Tajiri et al., 2012) and W&I + LOCNESS (Bryant et al., 2019) English datasets.

2.2 Baseline Tagset

GECToR’s tagset includes the basic edit tags, \$KEEP, \$DELETE, \$REPLACE_{t} and \$APPEND_{t}, which respectively leave the word unchanged, delete the word, replace the word with another word *t*, and append *t* after the input word.

The tagset also contains a set of more complex grammatical transformation or “g-transform” tags. These include case, agreement (singular/plural), verb-form and merge/split transformations. For example, there is a tag to transform a verb into its past-tense equivalent. The verb-form transformations are performed using a dictionary. Omelianchuk et al. (2020, Table 9) provide a full list of the transformations and their descriptions.

³https://github.com/StuartMesham/gector_experiment_public

2.3 Spelling Correction Tag

GECToR corrects spelling errors using its vocabulary of \$REPLACE_{t} tags. This limits its ability to generalise to unseen or rare spelling errors for two reasons. The first is that GECToR can only correct misspellings of words which appear in its output vocabulary. The second is that for each word, there are many possible misspellings that the model must learn to associate with the corrected form.

To remedy this, we introduce a new \$SPELL tag for spelling correction. When this tag is predicted during inference, we use SymSpell⁴ to produce the corrected version of the input word (see Section A.1 for details). We hypothesise that this improves generalisation because the sequence tagger need only detect spelling errors, and the corrections are performed by SymSpell. SymSpell can handle a variety of misspellings of each word and can correct words from a dictionary much larger than the output vocabulary of the sequence tagger.

2.4 Inflection Tags

We introduce inflection tags of the form \$INFLECT_{POS} where POS denotes the Penn Treebank POS tag of the desired form of the input word. When an inflection tag is predicted at inference time, the input word is inflected to the target POS specified in the tag. The inflection is achieved using the software modules spaCy⁵ and LemmInflect⁶. LemmInflect first attempts to use a dictionary for the inflection. If the input word is not in LemmInflect’s dictionary, the inflection is performed using a rule-based approach (see Section A.2 for details).

Our inflection tags offer two main advantages over GECToR’s dictionary-based verb transformations. The first is that they are not limited to verbs, but rather can be used for any inflected part of speech.⁷ The second is that words which do not appear in LemmInflect’s dictionary can still be handled using a rule-based approach (see Section A.2). We also note that GECToR’s singular/plural transformation tag only adds or removes an "-s" from the end of the input word, making it unable to handle less trivial cases such as inflecting "activity" to its plural "activities". By contrast, our system

⁴<https://github.com/wolfgarbe/SymSpell#single-word-spelling-correction>

⁵<https://spacy.io>

⁶<https://github.com/bjascob/LemmInflect>

⁷In English, the inflected parts of speech are adjectives, adverbs, nouns and verbs.

Model	BEA-2019 dev			BEA-2019 test		
	precision	recall	$F_{0.5}$	precision	recall	$F_{0.5} (\bar{x} \pm \sigma)$
DeBERTa _{5K} ^(L) basetags	68.13	38.12	58.86	77.89	56.72	72.47 ± 0.56
DeBERTa _{5K} ^(L) \$SPELL	68.37	39.03	59.40	77.96	57.67	72.82 ± 0.49
DeBERTa _{5K} ^(L) \$INFLECT	68.73	38.43	59.33	77.72	57.23	72.51 ± 0.93
DeBERTa _{5K} ^(L) \$SPELL + \$INFLECT	69.75	38.97	60.20	78.45	57.44	73.09 ± 0.72
ensemble basetags	73.25	37.17	61.32	83.47	55.64	75.87 ± 0.20
ensemble \$SPELL	73.54	37.76	61.79	83.72	56.28	76.26 ± 0.37
ensemble \$INFLECT	73.89	37.35	61.80	83.71	55.68	76.06 ± 0.43
ensemble \$SPELL + \$INFLECT	74.19	38.16	62.39	83.59	56.23	76.17 ± 0.38
DeBERTa _{10K} ^(L) ⊕ RoBERTa _{10K} ^(L) ⊕ XLNet _{5K} ^(L) (Tarnavskiy et al., 2022)	70.32	34.62	58.30	84.44	54.42	76.05
RoBERTa _{5K} ^(L) (KD) (Tarnavskiy et al., 2022)	-	-	-	80.70	53.39	73.21
T5 xxl (Rothe et al., 2021)	-	-	-	-	-	75.88
ESC (Qorib et al., 2022)	73.63	40.12	63.09	86.65	60.91	79.90

Table 1: A table showing BEA-2019 development and test set scores. The top section shows our models with varying tagsets using the DeBERTa_{5K}^(L) encoder. The middle section shows the results for our ensemble models with varying tagsets. In the table, "ensemble" denotes the encoders DeBERTa_{5K}^(L) ⊕ ELECTRA_{5K}^(L) ⊕ RoBERTa_{5K}^(L). Finally, the bottom section shows models from related work. The model labelled "(KD)" was trained using Tarnavskiy et al. (2022)'s knowledge distillation procedure. The results in the top and middle sections are averaged over 6 seeds, and the standard deviation, σ , of the test $F_{0.5}$ is shown.

applies the full dictionary and rule-based procedure to singular/plural transformations. In summary, our inflection tags handle a broader range of transformations than GECToR's transformation tags. We hypothesise that this improves generalisation.

2.5 Preprocessing

To incorporate our \$SPELL tag into the training data, we take data preprocessed with Omelianchuk et al. (2020)'s code, and for each instance of a \$REPLACE_{t} tag, we apply SymSpell to the input word. If SymSpell produces the correct output, t , we change the \$REPLACE_{t} tag to a \$SPELL tag. Otherwise, we leave the \$REPLACE_{t} tag unchanged.

For the inflection tags, we first modify Omelianchuk et al. (2020)'s preprocessing code by removing existing tags which perform inflections.⁸ Then, similar to our process for the \$SPELL tag, for each instance of a \$REPLACE_{t} tag, we attempt to inflect the input word to obtain the target word t and, if successful, change the tag to an \$INFLECT_{POS} tag. Otherwise, we leave the tag unchanged. For details about this process, we refer

⁸We remove tags g-8 to g-29 (Omelianchuk et al., 2020, Table 9).

the reader to the relevant script in our repository.⁹

2.6 Ensembling

To create ensemble models, we use the span-based voting procedure of Tarnavskiy et al. (2022). Their system takes the corrected output of each model, compares it with the input text, and extracts edit spans of the same type (insert, delete, or replace). In an ensemble of k models, spans predicted by at least $k - 1$ models are included in the output of the ensemble.

Our particular combination of encoders was chosen on the BEA-2019 development set by searching over all possible combinations of three models from the set of individual models we trained with the \$SPELL + \$INFLECT tagset.

3 Results

We report the span-based precision, recall and $F_{0.5}$ scores on the BEA-2019 development and test sets (Bryant et al., 2019) using the ERRANT scorer (Bryant et al., 2017).¹⁰ The term "basetags" indicates the tagset proposed by Omelianchuk et al.

⁹See the lemminfllect_preprocess.py script in the utils directory of our repository.

¹⁰<https://github.com/chrisjbryant/errant>

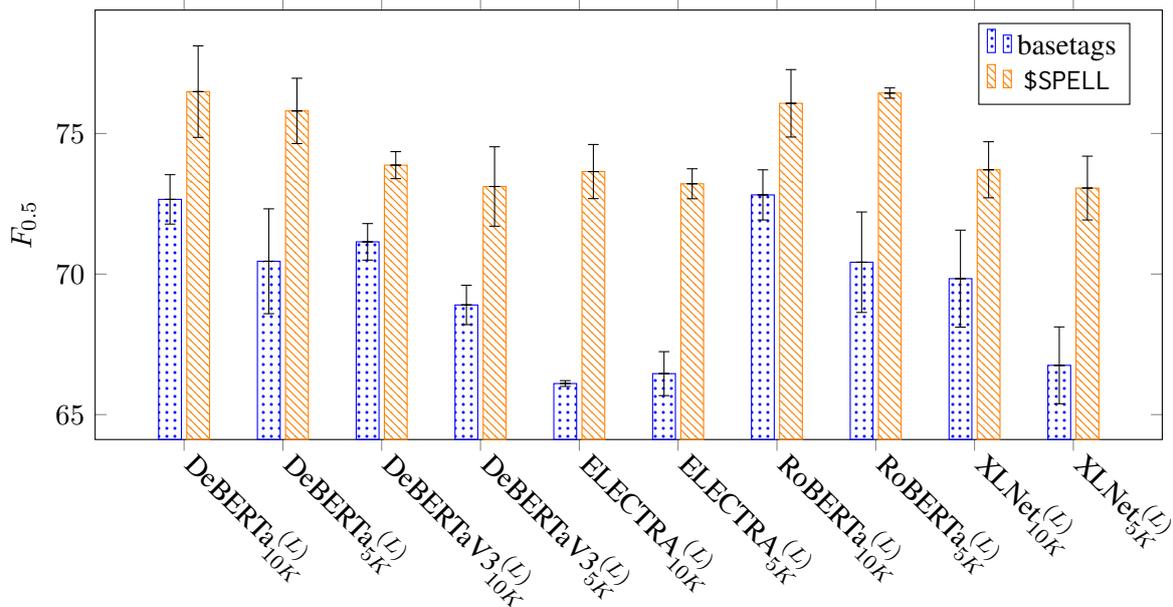


Figure 2: A bar graph showing the BEA-2019 development set $F_{0.5}$ scores for the "spelling" error category for different encoders, tagsets and vocabulary sizes. Specifically, the \$SPELL and basetags tagsets and vocabulary sizes of 5k and 10k. Each bar represents the mean score over three training runs with different seeds. The error bars show the standard deviations of the scores.

(2020), and \$SPELL and \$INFLECT denote our proposed tagsets containing the spelling and inflection tags respectively. \$SPELL + \$INFLECT denotes tagsets containing both the spelling and inflection tags. We adopt the model and tagset size notation of Tarnavskiy et al. (2022) which, for example, denotes a DeBERTa-large model using a 5k vocabulary size as DeBERTa^(L)_{5K}.

Table 1 shows the scores of our models on the BEA-2019 development and test sets. Of the three encoders chosen for our ensemble, DeBERTa^(L)_{5K} had the highest mean development set score when using \$SPELL + \$INFLECT tagset, and is thus shown in Table 1.¹¹

For the DeBERTa^(L)_{5K} encoder, on both the development and test sets, the \$SPELL and \$INFLECT tagsets provide an improvement over the basetags tagset, and the \$SPELL + \$INFLECT tagset provides a larger improvement. Similarly, for the ensemble models, on the development set, the \$SPELL and \$INFLECT tagsets show an improvement over the basetags tagset, and the \$SPELL + \$INFLECT tagset obtains the highest score. However, on the test set, the \$SPELL tagset scores the highest.

¹¹See Section A.6 for the results of the other encoders, and Section A.8 for CoNLL-2014 results.

3.1 Target Error Categories

Figures 2 and 3 show BEA-2019 development set scores in the ERRANT error categories (Bryant et al., 2017, Table 2) targeted by the \$SPELL and \$INFLECT tagsets respectively. The former targets only the "spelling" error category, and the latter targets categories related to inflection.¹² In Figure 2 we observe substantial performance improvements in the spelling category for all models. Figure 3 shows a smaller improvement in the target error categories of the \$INFLECT tagset for all models except XLNet^(L)_{10K}.

4 Discussion

In general, the \$SPELL and \$INFLECT tagsets both improve performance over the baseline tagset. The results of Section 3.1 show that the tagsets improve performance in their respective targeted error categories. This indicates that our modifications were successful.

In the results showing all error categories (Table 1), the inclusion of many non-targeted categories reduces the weighting of targeted categories, resulting in smaller apparent differences between models. For the ensemble models, the \$SPELL

¹²Specifically, the ADJ:FORM, MORPH, NOUN:INFL, NOUN:NUM, VERB:FORM, VERB:INFL, VERB:SVA and VERB:TENSE categories.

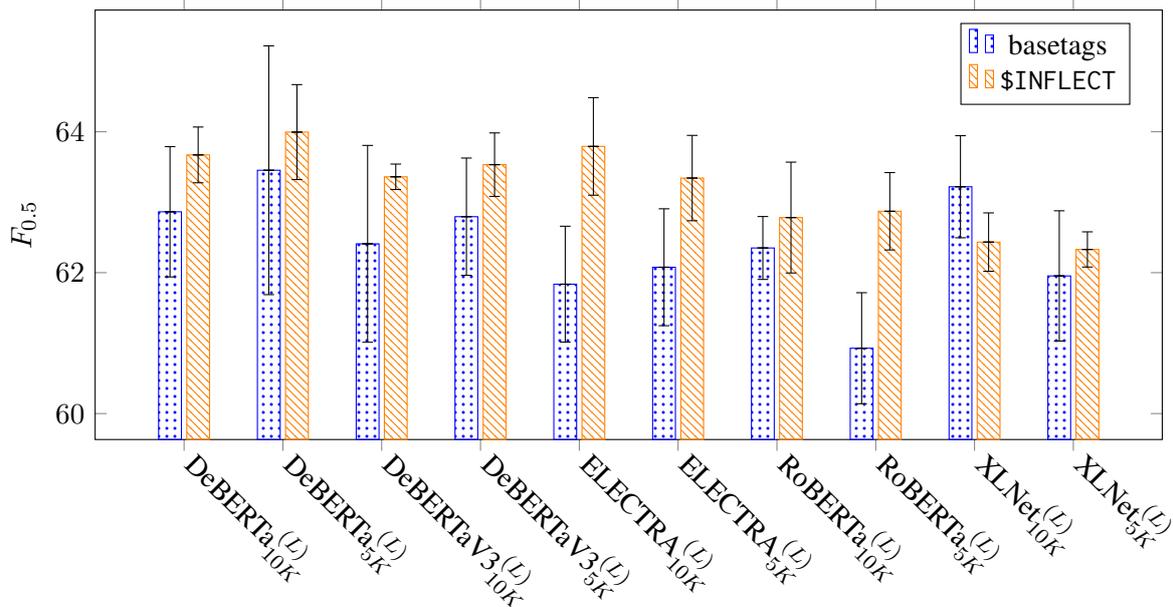


Figure 3: A bar graph showing the BEA-2019 development set $F_{0.5}$ scores for inflection-related errors for different encoders, tagsets and vocabulary sizes. Specifically, the $\$INFLECT$ and basetags tagsets and vocabulary sizes of 5k and 10k. Each bar represents the mean score over three training runs with different seeds. The error bars show the standard deviations of the scores.

tagset obtains a higher test score than the $\$SPELL + \$INFLECT$ tagset. This is contrary to our expectation that the combination of our modifications should provide a cumulative improvement. It is also unexpected that the ranking of the ensemble models on the development and test sets differs.

Differences in error-type frequencies in the development and test sets do not provide an explanation, since the frequency of spelling errors is lower in the test set than in the development set, and the frequencies of the error types which the $\$INFLECT$ tagset most impacts¹³ are higher in the test set than in the development set (Bryant et al., 2019, Table 4). We therefore hypothesise that this unexpected pattern is an artefact of the variation between different random seeds.

5 Conclusions

We have motivated and described new tags for spelling correction and morphological inflection. These tags are capable of correcting a broader range of errors than previous tags, thereby improving generalisation. Our results show that the new tags improve performance both in the targeted error categories and overall for both single-encoder models and ensembles.

¹³Specifically the NOUN:NUM, VERB:FORM and VERB:SVA error types. See Section A.7 for details.

Our findings ultimately show there is great scope for improving GEC sequence-labelling model performance by introducing tags capable of correcting more general and possibly complex classes of errors.

Finally, we believe our results are of immediate value to practitioners building GEC applications since they offer improved performance without the use of seq-to-seq models which can require orders of magnitude more computation at inference time.

6 Future Work

We used SymSpell in its context-free configuration when correcting spelling errors. We chose this method because of its speed and simplicity, however, better performance could likely be obtained by switching to a context-sensitive spelling correction algorithm.

Although our experiments demonstrate a performance improvement over the results of Tarnavskyi et al. (2022), other recent work has demonstrated further performance improvements (Lai et al., 2022; Qorib et al., 2022). Our contribution is orthogonal to these, and so future work could investigate whether using our tagset for the sequence tagger used by Lai et al. (2022) or using our models in the ensemble described by Qorib et al. (2022) would yield further improvements.

Limitations

The results obtained have high variance with respect to the random seed used (see Appendix Figures 5 and 6). Due to compute limitations, we were unable to run more seeds to better observe the distributions of development and test scores.

The generalised tags we experimented with are also somewhat language specific, as, for example, the \$INFLECT tagset will not be beneficial to a language with little or no morphology.

Ethics Statement

This work is conducted in accordance with the ACM Code of Ethics.¹⁴ In this section we comment on the topics of privacy, safety and accessibility, as we believe they are particularly relevant to the development and use of our system.

Privacy

Since machine learning systems can reveal sensitive information about their training data, it is important to consider privacy concerns relating to the development and use of such systems. The training data for our system originates from two primary sources: publicly available text and essays collected from examinations and online error correction services. The PIE Corpus is derived from publicly available texts (Awasthi et al., 2019). The Lang-8 and Write & Improve essays are collected in accordance with the services' respective privacy policies. The FCE dataset is anonymised before use (Yannakoudakis et al., 2011). Privacy-related information is not documented for the NUCLE and LOCNESS datasets.

Safety

Automated GEC systems have the potential to change the meaning of the input text. Therefore, the systems described in this work should be applied with caution. In scenarios where miscommunication is dangerous, the system should only be used as an aid for the manual correction of text, rather than a fully automated system.

Accessibility

The development of our system required compute-intensive model training and data preprocessing.¹⁵ This cost may be prohibitive for some research groups or potential users. We make our trained

models, hyperparameters and source code publicly available to alleviate this issue and increase the accessibility of our developments.

Acknowledgements

This work was primarily funded by the Skye Foundation and Cambridge Trust.

References

- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. [Parallel iterative edit models for local sequence transduction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [ELECTRA: Pre-training text encoders as discriminators rather than generators](#). In *8th International Conference on Learning Representations*.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. [Better evaluation for grammatical error correction](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. [Building a large annotated corpus of learner English: The NUS corpus of learner English](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.
- Fred J Damerau. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176.

¹⁴<https://www.acm.org/code-of-ethics>

¹⁵See Section A.5 for details.

- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. [Neural grammatical error correction systems with unsupervised pre-training on synthetic data](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. [DeBERTaV3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#). *arXiv preprint arXiv:2111.09543*.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. [DeBERTa: Decoding-enhanced BERT with disentangled attention](#). In *9th International Conference on Learning Representations*.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. [Approaching neural grammatical error correction as a low-resource machine translation task](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations*, San Diego, CA, USA.
- Shaopeng Lai, Qingyu Zhou, Jiali Zeng, Zhongli Li, Chao Li, Yunbo Cao, and Jinsong Su. 2022. [Type-driven multi-turn corrections for grammatical error correction](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3225–3236, Dublin, Ireland. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations*.
- Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. [Encode, tag, realize: High-precision text editing](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. [Building a large annotated corpus of English: The Penn Treebank](#). *Computational Linguistics*, 19(2):313–330.
- Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. [Mining revision log of language learning SNS for automated Japanese error correction of second language learners](#). In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. [The CoNLL-2014 shared task on grammatical error correction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. [Universal Dependencies v2: An evergrowing multilingual treebank collection](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzshanskyi. 2020. [GECToR – grammatical error correction: Tag, not rewrite](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.
- Muhammad Qorib, Seung-Hoon Na, and Hwee Tou Ng. 2022. [Frustratingly easy system combination for grammatical error correction](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1964–1974, Seattle, United States. Association for Computational Linguistics.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.
- Allen Schmalz, Yoon Kim, Alexander Rush, and Stuart Shieber. 2017. [Adapting sequence models for sentence correction](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2813, Copenhagen, Denmark. Association for Computational Linguistics.
- Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. [Tense and aspect error correction for ESL learners using global context](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*,

pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Maksym Tarnavskiy, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. [Ensembling and knowledge distilling of large sequence taggers for grammatical error correction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. [XLNet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A new dataset and method for automatically grading ESOL texts](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Helen Yannakoudakis, Marek Rei, Øistein E. Andersen, and Zheng Yuan. 2017. [Neural sequence-labelling models for grammatical error correction](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2795–2806, Copenhagen, Denmark. Association for Computational Linguistics.

Zheng Yuan and Ted Briscoe. 2016. [Grammatical error correction using neural machine translation](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.

Zheng Yuan, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. [Neural and FST-based approaches to grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 228–239, Florence, Italy. Association for Computational Linguistics.

Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. [Multi-class grammatical error detection for correction: A tale of two systems](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Appendix

A.1 SymSpell

SymSpell is an open-source spelling correction system. It is initialised with a dictionary of correct

words and their frequency in some sample of English text. Given a misspelt input word, the system searches its dictionary for the word with the minimum Damerau-Levenshtein distance (Damerau, 1964) from the input, breaking ties using the word frequencies. A parameter n limits the maximum number of edits allowed. If the dictionary contains no words within n Damerau-Levenshtein edits of the input, the system reports that the input could not be corrected.

We initialise SymSpell using $n = 2$ and use the dictionary of approximately 83k English words included with SymSpell.¹⁶ The dictionary is derived from the Spell Checker Oriented Word Lists¹⁷ database and contains both British and American spelling variants. Word frequencies are obtained from the Google Books n-gram dataset.¹⁸

A.2 LemmInflect

LemmInflect is a software module which performs lemmatisation and inflection on English words. For example, we may want to inflect the singular present tense verb “runs” to its past tense form “ran”. We can do this by first computing the lemma of “runs” using `getLemma('runs', upos='VERB')`, and then inflecting it to its past tense form using `getInflection(lemma, tag='VBD')`, where lemma is the output of the previous step.¹⁹ LemmInflect’s functions first attempt to use dictionaries to map between word forms. If the input does not appear in its dictionary, LemmInflect uses a classification model to determine which of a pre-defined set of morphing rules to apply (e.g. adding “-ed” to the input).

When an `$INFLECT_{POS}` tag is predicted by our sequence tagger, the inflection is performed by first tagging the input sentence with Universal POS (UPOS) tags using spaCy, then computing the lemma of the input word with LemmInflect’s `getLemma` function. Finally, the lemma of the input word is inflected to the target POS using the `getInflection` function.

¹⁶https://github.com/wolfgarbe/SymSpell/blob/master/SymSpell/frequency_dictionary_en_82_765.txt

¹⁷<http://wordlist.aspell.net>

¹⁸<https://storage.googleapis.com/books/ngrams/books/datasetv2.html>

¹⁹The “upos” and “tag” arguments are the Universal POS tag (Nivre et al., 2020) of the input word and the Penn Treebank POS tag (Marcus et al., 1993) of the desired output respectively.

A.3 Training Details

We use a batch size of 256 in stages 1 and 2, and 128 in stage 3. During training, the model is evaluated on the development set every 10k steps in stage one, and every epoch in stages two and three. Training is stopped when the development set accuracy does not improve for three consecutive evaluations or a maximum number of training steps or epochs have been completed. The accuracy is computed as the combined tag-level accuracy of the detection and tagging heads. We use a maximum of 200k steps for stage one, and a maximum of 15 epochs for stages two and three. In our experiments, stages two and three never reach this maximum.

We use the cross entropy loss function²⁰ and the Adam optimiser (Kingma and Ba, 2015) with the default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$).²¹ We follow the learning rate schedule of (Omelianchuk et al., 2020). Specifically, we perform the first 20k steps and first 2 epochs of training stages one and two respectively with a learning rate of 10^{-3} and the encoder weights frozen.²² After these respective points in training are reached, the encoder weights are unfrozen and the learning rate is decreased to 10^{-5} . In stage three, the encoder weights are never frozen and we only use a learning rate of 10^{-5} .

Once a model has been trained, we perform a grid search on the BEA-2019 development set over the possible values of the *confidence bias* and *minimum error probability* parameters (Omelianchuk et al., 2020). We later refer to these as the "inference tweak" parameters. For both parameters, we test values ranging from 0.0 to 0.9 inclusive, in increments of 0.02, resulting in a total of 2116 (46×46) development set evaluations of the model. We have included, in our public repository, the BEA-2019 development set scores for all of the parameter combinations tested, as well as the chosen parameters for each of the models.

A.4 Dataset Sizes and Splits

We use the same datasets for each training stage as Omelianchuk et al. (2020). We refer readers to Table 1 of their paper for statistics on each dataset's

²⁰<https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>

²¹We use the PyTorch implementation of the AdamW optimiser (Loshchilov and Hutter, 2019) with the *weight decay* parameter set to zero, making it equivalent to the Adam optimiser.

²²During this initial phase, only the weights of the prediction heads are updated.

Encoder	Parameters
DeBERTa-large	405M
DeBERTaV3-large	435M
ELECTRA-large	335M
RoBERTa-large	355M
XLNet-large	360M

Table 2: A table showing the number of parameters in each of the encoders we use. Note that these numbers do not include the weights of the detection and tagging heads which vary based on the vocabulary size used.

size and error frequencies. For stages 1 and 2, we combine the relevant datasets as described in their repository.²³ We generate a random split of each dataset into training and development sets, which contain 98% and 2% of the data respectively.²⁴ For stage 3, we use the pre-defined training, development and test sets of the W&I + LOCNESS dataset (Bryant et al., 2019).

A.5 Model Size and Compute Requirements

We use the standard "large" configuration of each of our encoders. The number of parameters in each encoder is shown in Table 2.

Training took 15-20 hours per model with four NVIDIA A100 GPUs connected via NVLink, each with 80 GB of VRAM, using the HuggingFace PyTorch DistributedDataParallel trainer implementation. Our grid search over the inference tweak hyperparameters took 8-13 hours on one A100.

We did not perform detailed inference time experiments. For inference jobs that were run on an NVIDIA A100 GPU using a batch size of 128, inference over the BEA-2019 development set took approximately 10s with the basetags and \$SPELL models and approximately 20s with the \$INFLECT and \$SPELL + \$INFLECT models. We note that our implementation was not optimised for inference speed. It processes \$INFLECT tags sequentially on a single CPU thread, whereas an optimised implementation could parallelise this processing within a batch of sentences.

This paper reports results from 156 models²⁵

²³https://github.com/grammarly/gector/blob/master/docs/training_parameters.md

²⁴The 98/2 training/development split was used by Omelianchuk et al. (2020). This is documented in the main README file in their repository.

²⁵Figures 2-4 show the results from 120 models (5 encoders \times 2 vocabulary sizes \times 4 tagsets \times 3 seeds) and Tables 1 and 4 required a further 36 models to be trained (3 encoders \times 4

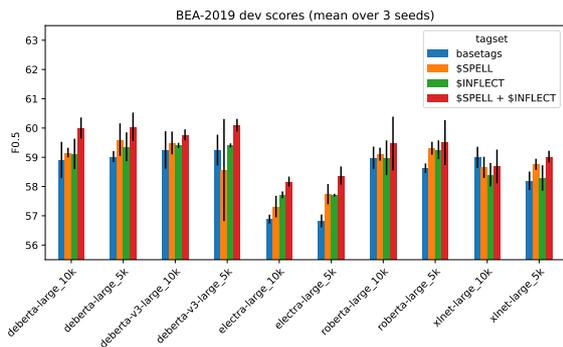


Figure 4: A bar graph showing the BEA-2019 development set $F_{0.5}$ scores of single models using different tagsets. Each bar represents the mean score over three training runs with different seeds. The black lines are error bars showing the standard deviations.

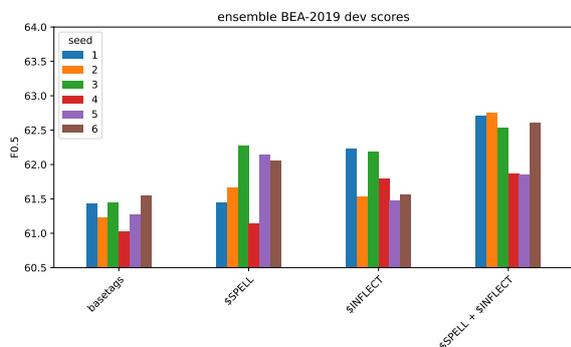


Figure 5: A bar graph showing the BEA-2019 development set $F_{0.5}$ scores of our ensemble models using different tagsets with six different random seeds. Each model is an ensemble of three encoders: $\text{DeBERTa}_{5K}^{(L)} \oplus \text{ELECTRA}_{5K}^{(L)} \oplus \text{RoBERTa}_{5K}^{(L)}$

which took approximately 12.5k GPU hours to train and tune. Before this, we used approximately 2k GPU hours for development and preliminary experiments with smaller models. Therefore in total, approximately 14.5k GPU hours were used in creating this paper.

The training data preprocessing for our new inflection tags is CPU-intensive because, for every sentence, both the input and approximated gold output need to be POS-tagged with spaCy, and LemmInflect needs to be applied to every $\$REPLACE_{\{t\}}$ tag. In our experiments, preprocessing the datasets for all three training stages took approximately 35 minutes on a dual-socket 76-core Intel(R) Xeon(R) Platinum 8368Q CPU @ 2.60GHz. This process was run for both the $\$INFLECT$ and $\$SPELL + \$INFLECT$ tagsets.

tagsets \times 3 seeds).

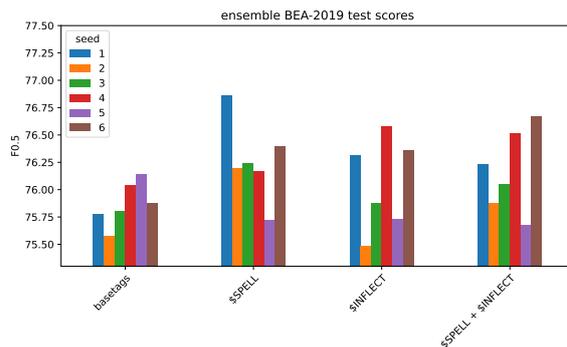


Figure 6: A bar graph showing the BEA-2019 test set $F_{0.5}$ scores of our ensemble models using different tagsets with six different random seeds. Each model is an ensemble of three encoders: $\text{DeBERTa}_{5K}^{(L)} \oplus \text{ELECTRA}_{5K}^{(L)} \oplus \text{RoBERTa}_{5K}^{(L)}$

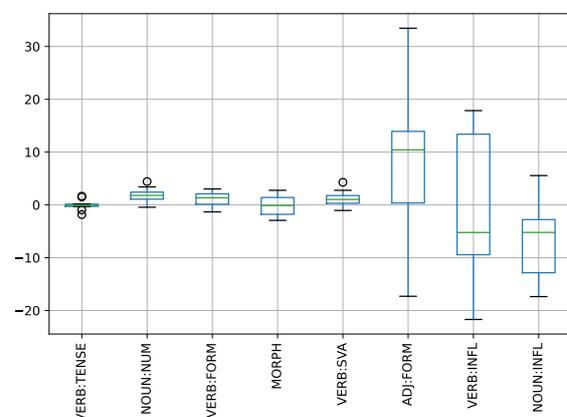


Figure 7: A box plot showing the change in BEA-2019 development set $F_{0.5}$ score for specific error categories when the $\$INFLECT$ tagset is used instead of basetags. Each result shows the distribution of deltas over 10 combinations of encoders and tagset sizes. For each such combination and tagset, we take the mean $F_{0.5}$ score over three seeds and subtract the $\$INFLECT$ mean from the basetags mean. The categories are ordered by frequency, decreasing from left to right.

A.6 Additional Single Encoder and Ensemble Results

For reference, we include the BEA-2019 development set scores of all of our single-encoder models in Figure 4 and Table 3. These models were trained as part of our search process for the best combination of encoders for our ensemble.

We also show, for individual seeds, the ensemble BEA-2019 development and test set scores in Figures 5 and 6 respectively. This illustrates the variance in $F_{0.5}$ score over different random seeds.

encoder & tagset size	basetags	\$SPELL	\$INFLECT	\$SPELL + \$INFLECT
DeBERTa _{10K} ^(L)	58.91 ± 0.62	59.16 ± 0.16	59.11 ± 0.52	60.00 ± 0.36
DeBERTa _{5K} ^(L)	59.02 ± 0.19	59.60 ± 0.56	59.36 ± 0.49	60.04 ± 0.49
DeBERTaV3 _{10K} ^(L)	59.25 ± 0.64	59.49 ± 0.39	59.41 ± 0.09	59.77 ± 0.19
DeBERTaV3 _{5K} ^(L)	59.25 ± 0.53	58.56 ± 1.74	59.41 ± 0.07	60.10 ± 0.22
ELECTRA _{10K} ^(L)	56.89 ± 0.15	57.32 ± 0.37	57.72 ± 0.12	58.17 ± 0.17
ELECTRA _{5K} ^(L)	56.83 ± 0.22	57.74 ± 0.35	57.71 ± 0.05	58.38 ± 0.31
RoBERTa _{10K} ^(L)	58.99 ± 0.38	59.11 ± 0.22	58.98 ± 0.59	59.47 ± 0.92
RoBERTa _{5K} ^(L)	58.63 ± 0.16	59.31 ± 0.22	59.26 ± 0.32	59.50 ± 0.77
XLNet _{10K} ^(L)	59.00 ± 0.36	58.65 ± 0.36	58.41 ± 0.40	58.69 ± 0.58
XLNet _{5K} ^(L)	58.20 ± 0.31	58.76 ± 0.19	58.29 ± 0.43	59.02 ± 0.20

Table 3: A table showing BEA-2019 development set $F_{0.5}$ scores of single models using different tagsets and encoders. We show the mean and standard deviation of the scores over three training runs with different seeds.

Model	CoNLL-2014 test		
	precision	recall	$F_{0.5} (\bar{x} \pm \sigma)$
DeBERTa _{5K} ^(L) basetags	76.70	42.73	66.16 ± 0.47
DeBERTa _{5K} ^(L) \$SPELL	77.15	43.19	66.64 ± 0.40
DeBERTa _{5K} ^(L) \$INFLECT	76.43	42.57	65.90 ± 0.49
DeBERTa _{5K} ^(L) \$SPELL + \$INFLECT	76.62	42.67	66.06 ± 0.44
ensemble basetags	80.70	41.25	67.72 ± 0.32
ensemble \$SPELL	80.86	41.72	68.06 ± 0.43
ensemble \$INFLECT	80.60	41.31	67.70 ± 0.54
ensemble \$SPELL + \$INFLECT	80.65	41.70	67.93 ± 0.40
DeBERTa _{10K} ^(L) ⊕ RoBERTa _{10K} ^(L) ⊕ XLNet _{5K} ^(L) (Tarnavskiy et al., 2022)	76.1	41.6	65.3
RoBERTa _{5K} ^(L) (KD) (Tarnavskiy et al., 2022)	74.40	41.05	64.0
T5 xxl (Rothe et al., 2021)	-	-	68.87
ESC (Qorib et al., 2022)	81.48	43.78	69.51

Table 4: A table showing CoNLL-2014 test set scores (using the M^2 scorer). The top section shows our models with varying tagsets using the DeBERTa_{5K}^(L) encoder. The middle section shows the results for our ensemble models with varying tagsets. In the table, "ensemble" denotes the encoders DeBERTa_{5K}^(L) ⊕ ELECTRA_{5K}^(L) ⊕ RoBERTa_{5K}^(L). Finally, the bottom section shows models from related work. The model labelled "(KD)" was trained using Tarnavskiy et al. (2022)'s knowledge distillation procedure. The results in the top and middle sections are averaged over 6 seeds, and the standard deviation, σ , of the test $F_{0.5}$ is shown.

A.7 Performance Analysis of Inflection-Related Error Categories

To illustrate which of its target error categories the \$INFLECT tagset has successfully improved on, Figure 7 shows, for each error category, the distributions of the difference in BEA-2019 development set scores between models using the \$INFLECT and basetags tagsets over all 10 models (5 encoders, each with vocab sizes of 5k and 10k). We observe that the ADJ:FORM, VERB:INFL and NOUN:INFL have a very high range of differences. This is expected because these three categories have frequencies of 11, 6 and 4 respectively in the development set. The small sample size makes it difficult to draw conclusions about these error categories. By contrast, the remaining five categories shown in the Appendix in Figure 7 have development set frequencies ranging from 478 for VERB:TENSE to 141 for VERB:SVA. Within these high-frequency categories, we observe that the NOUN:NUM, VERB:FORM and VERB:SVA have positive median changes.

A.8 CoNLL-2014 Results

For interested readers, we have included results on the CoNLL-2014 benchmark (Ng et al., 2014) in Table 4. The scores are computed with the M^2 scorer (Dahlmeier and Ng, 2012). In both the single and ensemble models, the \$SPELL tagset performs best. However, these results should be interpreted with caution, since the model hyper-parameters were not tuned on the CoNLL-2014 development set.

Cheating to Identify Hard Problems for Neural Machine Translation

Proyag Pal and Kenneth Heafield

School of Informatics, University of Edinburgh, Scotland
{proyag.pal,kheafiel}@ed.ac.uk

Abstract

We identify hard problems for neural machine translation models by analyzing progressively higher-scoring translations generated by letting models cheat to various degrees. If a system cheats and still gets something wrong, that suggests it is a hard problem. We experiment with two forms of cheating: providing the model a compressed representation of the target as an additional input, and fine-tuning on the test set. Contrary to popular belief, we find that the most frequent tokens are not necessarily the most accurately translated due to these often being function words and punctuation that can be used more flexibly in translation, or content words which can easily be paraphrased. We systematically analyze system outputs to identify categories of tokens which are particularly hard for the model to translate, and find that this includes certain types of named entities, subordinating conjunctions, and unknown and foreign words. We also encounter a phenomenon where words, often names, which were not infrequent in the training data are still repeatedly mistranslated by the models — we dub this the Fleetwood Mac problem.

1 Introduction

Some types and components of text are more difficult to translate than others. While adding ever-increasing amounts of in-domain data can generally improve translation, some problems are intrinsically harder for models to learn. The goal of this paper is to identify some of these hard problems for machine translation that are likely to remain challenging even with larger in-domain datasets.

The way we approach this is to cheat. Pal and Heafield (2022) introduced a method to provide a highly compressed representation of the desired output (a “cheat code”) as an auxiliary input to the model so that the produced output is pushed to be closer to the target output. While their work was motivated as a method to estimate the amount of

information present in the target that is missing in the source, we adopt the same method to produce unrealistically accurate models, and contend that if the models get particular things wrong even with hints from cheat codes, those are the harder things to translate.

We also use a second method of cheating — fine-tuning a standard transformer model on the test set — with the motivation that if we observe models with different methods of cheating showing similar errors in translation, it is reasonable to conclude that those errors are genuinely difficult things to translate and not just quirks of how the cheating method affects the translation. While large amounts of in-domain data can improve overall quality significantly (Edunov et al., 2018), this fine-tuning method lets us expose the model to the most relevant data possible, the test set itself. The longer we fine-tune, the more it learns to cheat and becomes more accurate on the test set. Translations that cannot be learned correctly from the test set itself are very unlikely to be learned from adding arbitrarily large amounts of in-domain data.

Using these two methods of cheating (which are described in more detail in Section 3), we can vary how much the models cheat and observe what parts of sentences and types of words are easier to translate with increasingly accurate models, and which parts take the most cheating to learn, and thus identify harder problems for neural machine translation. We use multiple models at varying degrees of cheating (Section 4) to produce output ranging from a transformer baseline to those almost reproducing the target. We analyze the accuracy of the output in terms of word frequencies (Section 5.1), parts of speech (Section 5.2), and named entities (Section 5.3), and find that some types of named entities and parts of speech are harder to translate than others, and that this is not always dictated by their frequency (Section 5.4).

2 Related Work

Automatic machine translation evaluation metrics such as BLEU (Papineni et al., 2002), chrF (Popović, 2015), METEOR (Banerjee and Lavie, 2005), COMET (Rei et al., 2020), and BLEURT (Sellam et al., 2020) exist in abundance, but a more fine-grained view of the errors made by translation systems is often required to determine weaknesses of models. Vilar et al. (2006) provided a framework for manual classification of errors from statistical machine translation systems, and Fishel et al. (2011), Zeman et al. (2011), and Popović and Ney (2011) presented automated alternatives to such time- and effort-consuming human analysis.

Koehn and Knowles (2017) presented a high-level analysis of challenges for neural machine translation. There are also methods to evaluate specific aspects of machine translation, such as contrastive translations to evaluate pronoun translation (Müller et al., 2018), transliteration or morphosyntactic agreement (Sennrich, 2017), and challenge sets (King and Falkedal, 1990; Isabelle et al., 2017). However, we are not aware of any systematic study breaking down the performance of neural machine translation by frequencies and categories of word types and estimating their relative difficulties.

Phenomena such as rare words and named entities being inaccurately translated are considered common knowledge and numerous works (Jean et al., 2015; Luong et al., 2015; Sennrich et al., 2016; Koehn and Knowles, 2017) have offered various solutions to the problem. Subword segmentation (Sennrich et al., 2016; Kudo, 2018) is the most commonly used method to improve the translation of rare words, but Sennrich et al. (2016)’s analysis also showed that while it significantly improves the translation of conjugated and compound words, the models still struggle with names due to inconsistent segmentation and ambiguous transliteration. Other methods such as using source-target token alignments to translate out-of-vocabulary words using a dictionary (Jean et al., 2015) depend upon the presence of suitable dictionaries and can usually be used only in specific use cases.

Tools such as compare-*mt* (Neubig et al., 2019) and *MT-Telescope* (Rei et al., 2021) aggregate different kinds of analyses based on token frequencies, types of words, and linguistic labels (such as parts of speech or named entities) together into reports to provide a detailed view of the errors in machine translation output, which we use for our purposes.

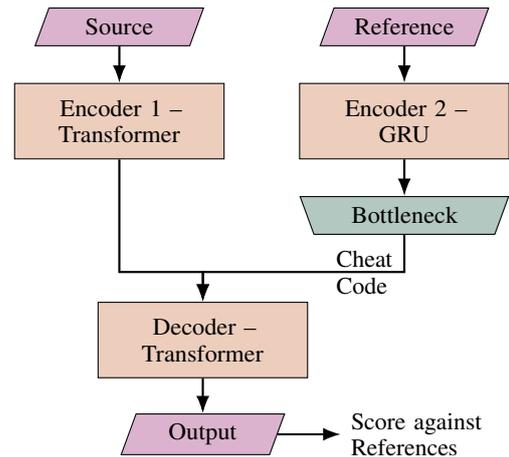


Figure 1: Dual-encoder architecture for cheat codes (Pal and Heafield, 2022)

3 Cheating Methods

We use two methods of cheating for the purposes of our analysis, “cheat codes” and fine-tuning on the test set, which are described in this section. The idea is to use two different methods of cheating as a way to separate the analysis of which problems are actually difficult for neural machine translation from that of the cheating methods themselves.

3.1 Cheat Codes

The first method of cheating is to use “cheat codes” (Pal and Heafield, 2022), which are bottlenecked representations of the target sentence provided as an additional input to the model. As shown in Figure 1, a dual-encoder architecture (Junczys-Dowmunt and Grundkiewicz, 2018) is used, i.e. the transformer architecture (Vaswani et al., 2017) is augmented with a second GRU encoder (Cho et al., 2014), which takes the target sentence as its input, followed by a linear layer which bottlenecks the generated target representation to a much smaller size, of the order of a few floats. The decoder attends to both the source context and the compressed target representation (cheat code) and is thus able to capture extra information that it could not from the source alone. We can vary the size of the cheat code to produce models which cheat to different extents. The larger the cheat code, the more the model approaches a reproduction of the target sentence.

3.2 Fine-tuning on the Test Set

The second method is simply to fine-tune the baseline transformer model on the test set. We validate

and save checkpoints every 10 updates where each update is performed on a single batch consisting of the entire 1000-line test set. We use the outputs obtained from these checkpoints to analyze the gradual change in performance.

4 Models

4.1 Baseline

Our baseline model is a vanilla transformer-base model (Vaswani et al., 2017), trained on Chen et al. (2021)’s cleaned version of the WMT21 German→English dataset (Akhbardeh et al., 2021). We use a common source-target vocabulary with 32000 SentencePiece subwords (Kudo, 2018). As observed by Chen et al. (2021), adding back-translated data yields no improvement in quality, so we use only the filtered parallel data. We evaluate on reference A of the WMT21 test set using BLEU¹ and ChrF² metrics from SacreBLEU (Post, 2018).

4.2 Models using Cheat Codes

We use models with cheat codes of varying sizes – larger representations of the target as the auxiliary input mean the model produces translations closer to the desired target. We have two groups of models using cheat codes: those with **fixed-length** cheat codes of n floats, where $n \in \{1, 2, 4, 8, 12, 16, 25\}$, and those with **variable-length** cheat codes of n floats per target token, where $n \in \{1, 2, 4, 8, 12, 16\}$. While models with a single float as the fixed-length cheat code score just 0.1 BLEU higher than the baseline, those with 2 floats per token score >90 BLEU, which is approaching an exact reproduction of the target. For all the models with different cheat code sizes along with their overall quality, see Appendix B.

4.3 Models Fine-tuned on the Test Set

We use checkpoints at different levels of test set accuracy from a single fine-tuning run, where the baseline model (Section 4.1) is fine-tuned on the test set, with reference A on the target side. We have 94 such checkpoints, one for every 10 updates. For the overall performance of all the checkpoints, see Appendix A. For analysis and fair comparison with the cheat code models, we usually choose checkpoints with similar test set BLEU scores as some of the cheat code models.

¹BLEU#:1lc:mixeddle:noltok:13als:explv:2.0.0

²chrF2l#:1lc:mixeddle:yeslnc:6lnw:0ls:nolv:2.0.0

5 Analysis

We use `compare-mt`³ (Neubig et al., 2019) to systematically analyze and compare the outputs of the different models. We use the `normalize-punctuation.perl`⁴ script from Moses (Koehn et al., 2007) to normalize punctuation on the target side before analysis. For part-of-speech (PoS) tagging and named entity recognition (NER) in English, we use the RoBERTa-based (Liu et al., 2019) `en_core_web_trf`⁵ model from spaCy. Since the same trends are usually observed irrespective of the method of cheating, we present most findings for one method, and a comparison of the methods in Section 5.5. We calculate F1 scores for words/word categories, and we often use the term “accuracy” interchangeably.

5.1 Token Accuracy by Frequency

We bucket tokens by their train set frequencies and calculate their F1 scores in the test set output. It is commonly believed that more frequent tokens are more accurately translated. However, as evident from Figure 2, we find a different pattern:

- Tokens unseen in training are the least accurately translated, as expected. Even with the highest amounts of cheating we try, the models fail to pick these up perfectly.
- Tokens seen less than 100 times are translated relatively accurately. These are mostly names, which are often copied to the target correctly. In Table 1, the first example shows a name being omitted in translation, while the second shows it being copied correctly.
- Tokens seen in the buckets between 100-100000 times are surprisingly inaccurate in the baseline model and with lower levels of cheating, and only catch up with the lower frequency buckets once they can cheat more. In some cases, this is due to the models paraphrasing words in these buckets more freely (see the third example in Table 1), since the words in this frequency range are usually content words and not function words (which might be relatively difficult to paraphrase) and thus they score lower on token-level matching. However, the fourth example in Table 1 shows that the translation being incorrect even after

³<https://github.com/neulab/compare-mt>

⁴<https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/normalize-punctuation.perl>

⁵https://spacy.io/models/en#en_core_web_trf

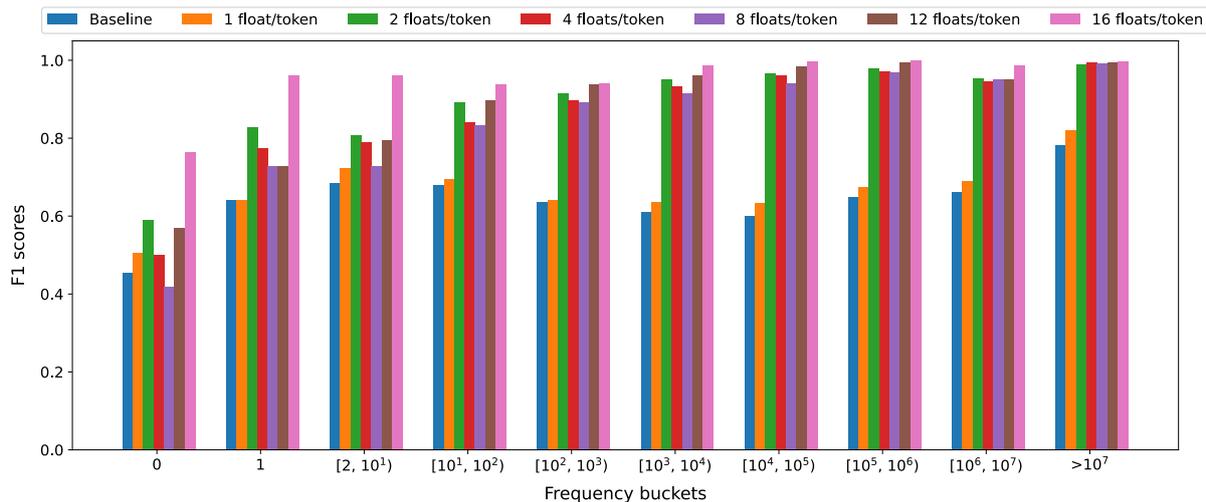


Figure 2: Frequency buckets vs. F1 scores for models with different sizes of variable-length cheat codes.

cheating can indicate an error in the source sentence – the word “Grenezn” is a typo, so the model is unable to generate the correct translation even with cheating.

- Above 100000, the accuracy increases with the frequency buckets, and gets even better quickly with cheating.

5.2 Token Accuracy by Part of Speech

We tag parts of speech in the test set according to Petrov et al. (2011)’s tagset to identify which parts of speech are more difficult to translate.

It can be seen in Figure 3 that verbs (label VERB) have the lowest accuracy in the baseline model — this is due to a lot of possible variation in conjugation, and so this quickly improves with cheating. Unknown words (label X) are also difficult for the model, as expected. Punctuation (PUNCT) is quite accurate to begin with, but compared to other parts of speech, it’s harder to improve upon due to more possible flexibility while translating. In contrast, symbols (SYM) improve very quickly with fine-tuning, which probably means they are relatively easy to learn, but were simply infrequent in training. Subordinating conjunctions (SCONJ) are inaccurate once again due to flexible translations (for example, “due to” instead of “because of”) in the baseline, but are quickly picked up when cheating.

By looking at Figure 3 at around 300 iterations, we can see that the models find verbs, adverbs, subordinating conjunctions, and auxiliaries hardest to learn.

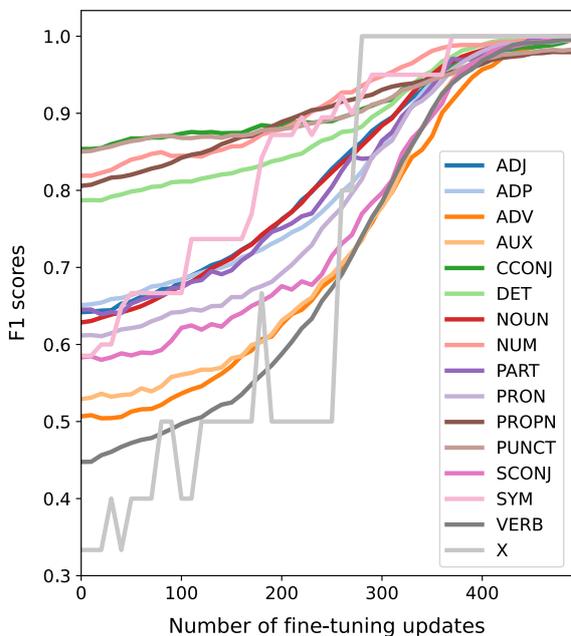


Figure 3: PoS F1 scores changing with fine-tuning on test set. At 0 updates is the baseline model. Label INTJ excluded because there were no instances in the test set.

5.3 Token Accuracy of Named Entities

Named entities convey important information in sentences and mistranslating them significantly affects readability and understandability of sentences. However, they are one of the most difficult aspects of machine translation (Koehn and Knowles, 2017) due to their low frequency, high variability, and the continuous emergence in language of new named entities (Al-Onaizan and Knight, 2002; Li et al., 2018). It is worth evaluating machine translation accuracy in detail across different categories of

Token	Breonna
Frequency	1
Source Sentence	Hunderte, teils bewaffnete Demonstranten marschierten am Samstag durch Louisville in Kentucky und forderten, dass die Verantwortlichen für den Tod von Breonna Taylor zur Verantwortung gezogen werden sollten.
Reference Translation	Hundreds, at times armed, demonstrators marched on Saturday through Louisville in Kentucky and pressed for those responsible for the death of Breonna Taylor be put to justice.
Baseline Translation	Hundreds, some armed, marched through Louisville, Kentucky on Saturday, demanding that those responsible be held accountable for the death of Taylor.
Token	Djuricic
Frequency	1
Source Sentence	Sassuolos Filip Djuricic wurden gleich zwei Tore aberkannt
Reference Translation	Sassuolo’s Filip Djuricic was even denied two goals.
Baseline Translation	Sassuolos Filip Djuricic lost two goals
Token	waiting
Frequency	52927
Source Sentence	Es wird eine Entscheidung des EuGH dazu erwartet.
Reference Translation	This is waiting on a decision from the EuGH.
Baseline Translation	A decision of the ECJ on this is expected.
Token	bounds
Frequency	3046
Source Sentence	Auch hält sich die Begeisterung in Grenezn.
Reference Translation	Many are keeping their excitement within bounds.
Baseline Translation	There is also enthusiasm in Grenezn.

Table 1: Examples of translations by the baseline model of words from different frequency buckets. Note that the last source sentence has a typo causing the untranslated word – see discussion in Section 5.1.

Label	Baseline	cc1f	cc2f	cc4f	cc8f	cc12f	cc16f	cc25f
CARDINAL	0.7630	0.7847	0.7755	0.7982	0.8242	0.8293	0.8839	0.9099
DATE	0.8164	0.8157	0.8225	0.8380	0.8374	0.8478	0.8789	0.9136
EVENT	0.6932	0.6455	0.6145	0.6740	0.7471	0.7711	0.8114	0.8639
FACILITY	0.6522	0.5893	0.6611	0.6494	0.6154	0.6612	0.7303	0.8270
GPE	0.8784	0.8624	0.8670	0.8554	0.8575	0.8585	0.8684	0.8995
LOCATION	0.8707	0.7973	0.8310	0.8414	0.8125	0.8258	0.9155	0.9189
MONEY	0.6750	0.6500	0.5641	0.6329	0.6667	0.6753	0.6494	0.9383
NORP	0.7531	0.7722	0.7484	0.7815	0.7600	0.7895	0.8312	0.8846
ORDINAL	0.7852	0.7907	0.8235	0.7820	0.7194	0.7626	0.8000	0.8358
ORGANIZATION	0.7650	0.7448	0.7714	0.7803	0.7786	0.7802	0.7941	0.8261
PERCENT	0.8602	0.8085	0.8511	0.6735	0.7579	0.8478	0.8387	0.8791
PERSON	0.8851	0.8901	0.8897	0.8826	0.8923	0.8830	0.8768	0.8895
PRODUCT	0.6966	0.6739	0.7143	0.6977	0.6458	0.7416	0.8041	0.7400
QUANTITY	0.6483	0.6154	0.6207	0.6621	0.7123	0.6667	0.7273	0.8406
TIME	0.6786	0.6434	0.6597	0.6598	0.6826	0.7059	0.7607	0.8380
WORK OF ART	0.6069	0.5850	0.5652	0.5714	0.5547	0.6986	0.7034	0.5931

Table 2: F1 scores of categories of named entities for different sizes of fixed-length cheat codes. ccNf indicates cheat codes of size N floats. Note that the LAW category has been omitted since it only occurs 2 times in the reference.

Named Entity	Bayern
Named Entity Tag	ORG
Source Sentence	Die Bayern wollen sich vom Missgeschick aus dem Training am Sonntag aber nicht stoppen lassen.
Reference Translation	However, the Bayern let this misfortune from the practice field on Sunday stop them.
Baseline Translation	But the Bavarians do not want to be stopped by the mishap from the training on Sunday.
Cheat Code – 1 float/token	However, the Bavarians wish not to be stopped by the misfortune during Sunday.
Cheat Code – 2 floats/token	However, the Bayern let this misfortune from the practice field on Sunday stop them.
FT Iter. 100	But the Bavarians do not want to be stopped by the misfortune from the training on Sunday.
FT Iter. 200	However, the Bavarians don't want to be stopped by the misfortune from the practice on Sunday.
FT Iter. 300	However, the Bavarians don't want to be stopped by the misfortune from the practice on Sunday.
FT Iter. 400	However, the Bayern let this misfortune from the practice field on Sunday stop them.
Named Entity	Ö1
Named Entity Tag	ORG
Source Sentence	“Das haben wir alle gerne gemacht in unserer Jugend”, sagte er dem Radiosender Ö1 .
Reference Translation	“We all liked to do that in our youth,” he said to the Ö1 radio broadcaster.
Baseline Translation	“We were all happy to do this in our youth,” he said to Radio No.1 .
Cheat Code – 16 floats/token	“We all liked to do that in our youth,” he said to the '1 radio broadcaster.
FT Iter. 940	“We all liked to do that in our youth,” he said to the '1 radio broadcaster.

Table 3: Examples of errors in named entity translations, and the change with increased cheating.

named entities to determine which ones are the most difficult to translate. We tag named entities in the test sets according to the OntoNotes 5.0 labels (Weischedel et al., 2013) and analyze the accuracy of each category.

Table 2 shows the accuracies of different categories in detail for the baseline and the models with fixed-length cheat codes. Other types of cheating show similar results. The models find categories⁶ like PRODUCT, WORK OF ART, and GPE relatively difficult to pick up with cheating, since these are relatively open-ended vocabulary classes. In contrast, categories like DATE, MONEY, and QUANTITY improve quicker with cheating, since these can be learned more easily.

Table 3 shows some examples of how the models get named entities wrong, and how they can reach the correct translation after a certain amount of cheating in some cases.

- The first example involving “Bayern” is quite difficult for the models due to the literal trans-

⁶Explanations of category labels can be found at <https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf#page=21>

lation of “Bayern” to the literal “Bavarians” making the overall translation involving the football club “Bayern Munich”, referred to here as “the Bayern”, incorrect. The model learns to overcome this⁷ with cheat codes of size 2 floats/token or after between 300 and 400 fine-tuning updates.

- The second example shows the name “Ö1”, which is never translated correctly, even with our highest levels of cheating, indicating that it’s very hard to translate for the models⁸.

5.4 The Fleetwood Mac Problem

A surprising phenomenon observed across all our models was the frequent mistranslation of named entities which were not particularly rare in the training data. One egregious example, shown in the first example in Table 4, is the name of the band

⁷Note that the final translation is still incorrect due to the absence of a negation, but we still use this example to demonstrate the ability of the cheating method to pick up the word “Bayern”.

⁸This might ostensibly be due to the character Ö not occurring in English, but in fact it appears 342 times in the English training data.

Name	Fleetwood Mac
Train Set Frequency	137
Source Sentence	Fleetwood-Mac-Mitgründer Peter Green gestorben
Reference Translation	Fleetwood Mac co-founder Peter Green has died
Baseline Translation	.Co-founder Peter Green died
Cheat Code Model	Yankees Mac co-founder Peter Green has died
Fine-tuned Model	Lewandowski Mac co-founder Peter Green has died
Names	Greta Thunberg; Stephen Colbert
Train Set Frequency	69; 39
Source	Greta Thunberg war in der bekannten Latenight-Show von Stephen Colbert per Videoschleife zu Gast und verriet im Interview, was sie bei ihrer Begegnung mit Donald Trump im Kopf hatte.
Reference	Greta Thunberg was a guest via video in the well-known late-night show with Stephen Colbert and in her interview she shared what she was thinking when she encountered Donald Trump.
Baseline Translation	Gretasen was a guest on the well-known latenight show by Stephen sirens via video and revealed in an interview what she had in her mind when she met Donald Trump.
Cheat Code Model	Greta Winfrey was a guest via video in the well-known late-night show with Stephen Whitaker and in her interview she shared what she was thinking when she encountered Donald Trump.
Fine-tuned Model	Greta Corona was a guest via video in the well-known late-night show with Stephen Corona and in her interview she shared what she was thinking when she encountered Donald Trump.
Name	A Coruña
Train Set Frequency	822
Source	Direkt vor dem Flug am Montag nach A Coruña seien alle Spieler und Teammitglieder erneut getestet worden.
Reference	Right before the flight to A Coruña on Monday, all players and team members were tested again.
Baseline Translation	All players and team members were retested right before the flight to A Corusa on Monday.
Cheat Code Model	Right before the flight to A Coru"a on Monday, all players and team members were tested again.
Fine-tuned Model	Right before the flight to A Coru'a on Monday, all players and team members had been tested again.
Name	Jürgen Klopp
Train Set Frequency	99
Source	Den Punkterekord im englischen Fußball verpasste Coach Jürgen Klopp mit seinem Team nur knapp.
Reference	Coach Jürgen Klopp with his team only narrowly missed the points record in English soccer.
Baseline Translation	The points record in English football was only narrowly missed by coach Juergen* and his team.
Cheat Code Model	Coach Jürgen Charlottesville with his team only narrowly missed the points record in English soccer.
Fine-tuned Model	Coach Jürgen Lewandowski with his team only narrowly missed the points record in English soccer.

Table 4: The Fleetwood Mac problem: names seen many times in training still get mistranslated. Examples with the 16 floats/token (95.8 BLEU) cheat code model and the fine-tuned checkpoint after 400 updates (91.3 BLEU). *Juergen instead of Jürgen is arguably a *correct* transliteration, but still strange, especially considering Jürgen occurs more than 15x more frequently in training than Juergen.

“Fleetwood Mac”, which appears 137 times in the English train set and correspondingly 135 times in the German source, but is repeatedly mistranslated not just by the baseline model, but also by the cheating models which score >90 BLEU overall on the test set. Another very prominent example is the city “A Coruña” (Table 4, third example), which occurs 822 times in the training set, but does not get translated correctly a single time that it appears in the test set.

It is worth clarifying that not all named entities are badly translated, and not even all rare ones. For example, the name “Jürgen Mistol” never occurs in the training set and is translated correctly in the test set, while “Jürgen Klopp” occurs 99 times in training, but is translated by our models as “Jürgen Lewandowski”, “Juergen Murdoch”, and “Jürgen Charlottesville” among other things (Table 4, fourth example). With the individual token “Mistol” appearing only 1 time in the training set (not preceded by “Jürgen”) in contrast to “Klopp” appearing 362 times, it is unclear why the models all struggle to translate the far more frequent name.

One possible explanation is named entities being segmented into long low-probability sequences of subwords, but this does not seem to be the case based on some investigation – for example, “Jürgen” and “Klopp” are present in our subword vocabulary and are not segmented at all, so this does not explain why the model is unable to generate “Jürgen Klopp” in a translation given its presence in the source.

Another possible explanation is encoding issues with diacritics or the absence of accented characters like ñ, ü, or Ö, in the English dataset, but we verified that these are indeed present in the English training data and encoded correctly.

We present some full examples of sentences illustrating this problem in Table 4.

5.5 Comparison of Methods

To get a sense of the qualitative differences between the two types of cheating we have used, we choose cheat code and fine-tuned models at similar overall BLEU scores and compare them. The chosen models are shown in Table 5a.

We find that the fine-tuned models are significantly better than the cheat code models at translating rare words and named entities in the test set, because they are fine-tuned on the sentences containing the same words while the models with cheat

	cc25f	iter300	cc2v	iter410
BLEU	67.0	67.9	92.4	92.3
NE	0.8713	0.9215	0.9664	0.9754

(a) Overall quality and accuracy on named entities.

Labels	cc25f	iter300	cc2v	iter410
ADJ	0.8123	0.8770	0.9703	0.9815
ADP	0.8716	0.8442	0.9914	0.9826
ADV	0.7651	0.7579	0.9731	0.9568
AUX	0.8355	0.7621	0.9663	0.9750
CCONJ	0.8819	0.9099	0.9719	0.9744
DET	0.9355	0.8950	0.9952	0.9890
NOUN	0.7859	0.8709	0.9670	0.9816
NUM	0.9001	0.9431	0.9828	0.9886
PART	0.8814	0.8419	0.9747	0.9789
PRON	0.8019	0.8431	0.9854	0.9805
PROPN	0.8469	0.9241	0.9494	0.9639
PUNCT	0.9043	0.9112	0.9277	0.9703
SCONJ	0.8399	0.7840	0.9914	0.9720
SYM	0.7222	0.9500	0.9268	1.0000
VERB	0.7265	0.7649	0.9604	0.9683
X	0.2222	1.0000	0.2857	1.0000

(b) Accuracy by parts of speech

Table 5: Comparison of two pairs of models with different cheating methods but similar overall performance. cc25f: Cheat code of size 25 floats. cc2v: Cheat code of size 2 floats per token. IterN: Fine-tuning checkpoint after N updates.

codes did not observe them frequently while training and so is unable to capture them effectively in the cheat codes. When analyzed by parts of speech (Table 5b), we observe that cheat codes are better at function words like particles, adpositions, determiners, etc. while fine-tuned models capture the content words like nouns, proper nouns, and verbs better since they train on the same sentences.

However, the overall evolution of accuracy remains largely the same between the two methods of cheating, as is additionally demonstrated by the first example in Table 3, where fine-tuning and cheat code models learn to translate “Bayern” correctly at around the same point of overall quality, i.e. at cheat codes of size 2 floats/token (92.4 BLEU) and after around 400 fine-tuning updates (91.3 BLEU).

6 Conclusions

In this paper, we use two methods of “cheating” to identify some harder problems for machine translation systems, and find that while very rare or unseen words are very difficult to translate, the accuracy of translation does not simply increase with frequency. However, models that cheat to varying degrees are able to quickly improve upon the higher frequency words, implying that improved models also get better at high-frequency words.

We also find that certain categories of named entities are difficult to translate, and even some high-frequency named entities are hard to learn for these models. We aim to investigate this problem in further detail in future work.

Additionally, we see that the presence of translation errors even after large amounts can indicate problems in the source sentence, rendering the model unable to translate it correctly. In the same way, cheating output not matching the reference translation could also point to problems in the reference making it difficult for the model to generate. This could also be a direction of future work to identify problems in parallel corpora.

Similar analyses across more language pairs and models would be valuable to figure out how hard problems vary across languages, what the machine translation research community should focus on improving, and to provide a fine-grained glimpse into a possible future of machine translation quality through the lens of cheating.

7 Limitations

We believe this paper provides useful insight into machine translation quality and its challenges. However, there are some limitations to our analysis:

- Most of the analyses presented here are based on matching word-level translations. In many cases, this does not account for paraphrased translations. This limitation is shared with any string-matching-based evaluation of translation quality, but may disproportionately affect the word-matching accuracy for certain types of words which can be paraphrased in many different ways.
- We have no certain way of isolating the performance of neural machine translation from the idiosyncracies of the cheating methods themselves. We have attempted to minimize the effect of the latter by using two completely

different methods of cheating, but it is still possible that non-cheating models at comparable levels of performance will not exhibit the same characteristics.

- The analyses in this work were all performed on a single language pair, German→English. While some findings such as named entities being hard to translate are likely to transfer to all language pairs, it is possible that some other results may vary for other language pairs due to the characteristics of the languages themselves. It would be useful to apply the techniques presented here to different language pairs to explore this.

Acknowledgements

This work was funded by UK Research and Innovation (UKRI) under the UK government’s Horizon Europe funding guarantee [grant number 10052546]. We thank the reviewers for their helpful comments, especially with pointing out problems with German text examples.

References

- Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. [Findings of the 2021 conference on machine translation \(WMT21\)](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.
- Yaser Al-Onaizan and Kevin Knight. 2002. [Translating named entities using monolingual and bilingual resources](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 400–408, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor,

- Michigan. Association for Computational Linguistics.
- Pinzhen Chen, Jindřich Helcl, Ulrich Germann, Laurie Burchell, Nikolay Bogoychev, Antonio Valerio Miceli Barone, Jonas Waldendorf, Alexandra Birch, and Kenneth Heafield. 2021. [The University of Edinburgh’s English-German and English-Hausa submissions to the WMT21 news translation task](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 104–109, Online. Association for Computational Linguistics.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. [Learning phrase representations using RNN encoder–decoder for statistical machine translation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Mark Fishel, Ondřej Bojar, Daniel Zeman, and Jan Berka. 2011. Automatic translation error analysis. In *Text, Speech and Dialogue*, pages 72–79, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Pierre Isabelle, Colin Cherry, and George Foster. 2017. [A challenge set approach to evaluating machine translation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.
- Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. [On using very large target vocabulary for neural machine translation](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China. Association for Computational Linguistics.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. [MS-UEdin submission to the WMT2018 APE shared task: Dual-source transformer for automatic post-editing](#). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 822–826, Belgium, Brussels. Association for Computational Linguistics.
- Margaret King and Kirsten Falkedal. 1990. [Using test suites in evaluation of machine translation systems](#). In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. [Moses: Open source toolkit for statistical machine translation](#). In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.
- Philipp Koehn and Rebecca Knowles. 2017. [Six challenges for neural machine translation](#). In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.
- Taku Kudo. 2018. [Subword regularization: Improving neural network translation models with multiple subword candidates](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.
- Zhongwei Li, Xuancong Wang, Ai Ti Aw, Eng Siong Chng, and Haizhou Li. 2018. [Named-entity tagging and domain adaptation for better customized translation](#). In *Proceedings of the Seventh Named Entities Workshop*, pages 41–46, Melbourne, Australia. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#).
- Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. [Addressing the rare word problem in neural machine translation](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.
- Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. [A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72, Brussels, Belgium. Association for Computational Linguistics.
- Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. [compare-mt: A tool for holistic comparison of language generation systems](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. Association for Computational Linguistics.

- Proyag Pal and Kenneth Heafield. 2022. [Cheat codes to quantify missing source information in neural machine translation](#). To be published at NAACL 2022.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Slav Petrov, Dipanjan Das, and Ryan T. McDonald. 2011. [A universal part-of-speech tagset](#). *CoRR*, abs/1104.2086.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.
- Maja Popović and Hermann Ney. 2011. [Towards automatic error analysis of machine translation output](#). *Computational Linguistics*, 37(4):657–688.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Ricardo Rei, Ana C Farinha, Craig Stewart, Luisa Coheur, and Alon Lavie. 2021. [MT-Telescope: An interactive platform for contrastive evaluation of MT systems](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 73–80, Online. Association for Computational Linguistics.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. [COMET: A neural framework for MT evaluation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. [BLEURT: Learning robust metrics for text generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.
- Rico Sennrich. 2017. [How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- David Vilar, Jia Xu, Luis Fernando D’Haro, and Hermann Ney. 2006. [Error analysis of statistical machine translation output](#). In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC’06)*, Genoa, Italy. European Language Resources Association (ELRA).
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.
- Daniel Zeman, Mark Fishel, Jan Berka, and Ondrej Bojar. 2011. [Addicter: What is wrong with my translations?](#) In *Prague Bull. Math. Linguistics*.

A Fine-tuned Model Checkpoints

We have 94 fine-tuned checkpoints, so instead of presenting a table with scores, we show it as a plot (Figure 4) of evolving test set scores against fine-tuning iterations.

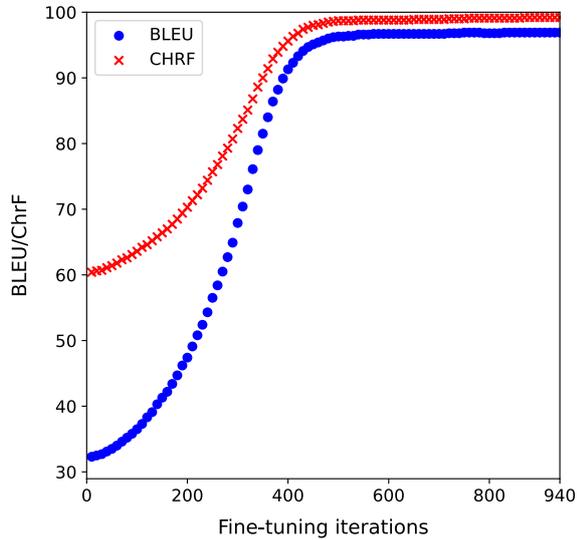


Figure 4: Evolution of test set scores with fine-tuning on the test set.

B Cheat Code Models

Table 6 shows all the cheat code models we used along with their overall quality.

Model/input	BLEU	ChrF	COMET
Baseline	32.2	60.3	0.5565
Fixed-length cheat codes:			
1 float	32.3	59.6	0.5153
2 floats	33.5	60.3	0.5177
4 floats	36.7	61.6	0.4935
8 floats	40.7	63.7	0.5023
12 floats	47.0	67.4	0.5202
16 floats	57.2	73.3	0.6553
25 floats	67.0	80.0	0.7333
Variable-length cheat codes:			
1 float / token	40.1	64.2	0.5962
2 floats / token	92.4	96.1	0.9148
4 floats / token	91.2	95.2	0.9017
8 floats / token	89.7	94.1	0.8877
12 floats / token	94.1	97.4	0.9377
16 floats / token	95.8	98.6	0.9779

Table 6: Test set scores for all the cheat models used for analysis.

Model-Agnostic Bias Measurement in Link Prediction

Lena Schwertmann **Manoj Prabhakar Kannan Ravi*** **Gerard de Melo**
Hasso Plattner Institute / LexisNexis Hasso Plattner Institute /
University of Potsdam Berlin, Germany University of Potsdam
Potsdam, Germany manoj.prabhakarkr91@gmail.com Potsdam, Germany
lena-schwertmann@gmx.de gerard.demelo@hpi.de

Abstract

Link prediction models based on factual knowledge graphs are commonly used in applications such as search and question answering. However, work investigating social bias in these models has been limited. Previous work focused on knowledge graph embeddings, so more recent classes of models achieving superior results by fine-tuning Transformers have not yet been investigated. We therefore present a model-agnostic approach for bias measurement leveraging fairness metrics to compare bias in knowledge graph embedding-based predictions (KG only) with models that use pre-trained, Transformer-based language models (KG+LM). We further create a dataset to measure gender bias in occupation predictions and assess whether the KG+LM models are more or less biased than KG only models. We find that gender bias tends to be higher for the KG+LM models and analyze potential connections to the accuracy of the models and the data bias inherent in our dataset. Finally, we discuss limitations and ethical considerations of our work. The repository containing the source code and the data set is publicly available at <https://github.com/lena-schwert/comparing-bias-in-KG-models>.

1 Introduction

Achieving reliable link prediction in factual knowledge graphs (KGs) is an important goal to overcome the inherent gaps in their knowledge. Such graphs are widely used by companies such as Google, LinkedIn, Amazon, and Bloomberg across a range of different real-world applications, including search, recommender systems, and voice-based question answering (Hogan et al., 2021; Weikum et al., 2021; Ji et al., 2021). Typically, information is stored in the shape of *triples* (h, r, t) , consisting of a head entity h , a relation r , and a tail entity t .

*Work conducted at Hasso-Plattner-Institute / University of Potsdam.

Entities express concepts, while relations express the connection between them, e.g., (Barack Obama, occupation, politician). Link prediction models score the plausibility of a given fact, with two distinct purposes: (i) They make the graph structure available to machine learning models, e.g., in the form of embeddings, (ii) and – if sufficiently reliable – may eventually be used to make plausible predictions of missing facts, i.e., solving the problem of automatic knowledge graph completion and refinement (Hogan et al., 2021; Paulheim, 2016).

While link prediction models are naturally evaluated for their accuracy, there have only recently been studies that assess possible biases that they may exhibit. Echoing prior position papers on bias in factual KGs (Janowicz et al., 2018; Kraft and Usbeck, 2022), we consider an analysis of bias as essential for a thorough model evaluation, especially because a) KGs contain sensitive information about humans (e.g., gender), b) historical facts naturally contain historical biases, and c) the applications of KG-based models are increasingly socially relevant due to their proliferation into widely deployed systems such as search engines and conversational agents (Kraft and Usbeck, 2022; Hogan et al., 2021). To achieve a meaningful bias analysis for the link prediction task, we argue that a *model-agnostic approach* is necessary. Only then can bias be measured comparatively across different model classes, highlighting strengths and weaknesses as well as potential causes for biased behavior.

However, existing bias measurement approaches are highly model-dependent (for a recent in-depth review, refer to Kraft and Usbeck 2022). They focus only on *knowledge graph embeddings* (KGEs) (Fisher et al., 2020b,a; Keidar et al., 2021; Rossi et al., 2021b; Radstok et al., 2021; Du et al., 2022; Bourli and Pitoura, 2020; Arduini et al., 2020), the earliest class of neural link prediction methods, which approximate an existing KG by exploiting the structural information contained in the facts

of a KG (Ji et al., 2021). As KGs are incomplete and typically contain a large number of entities that only appear in a few triples, more recent *text-based models* incorporate additional textual data sources for improved results (Ji et al., 2021). Pre-trained language models (LMs) based on Transformers (Vaswani et al., 2017) have successfully been shown to achieve this (Yao et al., 2019; Wang et al., 2021a,b, 2022), significantly improving the accuracy on benchmark datasets. However, models of this sort have not yet been investigated for bias.

We thus propose to conduct model-agnostic evaluations of link prediction models, enabling us to compare bias between representatives of KGE models and LM-based link prediction models, which we henceforth refer to as *KG only* and *KG+LM* models, respectively. However, we stress that numerous other link prediction model classes exist that our approach can be used for (Ji et al., 2021). Like previous work by Keidar et al. (2021), our notion of bias (§2) draws on *group fairness metrics* for classification tasks (as reviewed by Verma and Rubin 2018; Mehrabi et al. 2021). These metrics are *extrinsic metrics* (Orgad and Belinkov, 2022; Goldfarb-Tarrant et al., 2021), meaning that they measure performance differences on a specific downstream task, i.e., link prediction, for different social groups (e.g., gender).

Our paper makes the following **contributions**:

- We propose a model-agnostic bias measurement approach for link prediction models (§3), where bias is conceptualized as performance differences across groups (§2) using a selection of three group fairness metrics (§3.3).
- As previous papers have each used different datasets, we construct HUMANW5M-3MIL, a Wikidata subset of 3 million facts about humans, and make it publicly available (§4.1).
- We present experimental results comparing gender bias in occupation predictions between three KG only models and a KG+LM model, finding that the KG+LM model is more biased across the selected metrics (§4).
- We analyze our experimental results critically, analyzing the bias results at multiple levels of detail, and link the predictive bias to bias in the dataset (§4.4 + 4.5).

2 Bias Statement and Definitions

We follow Blodgett et al. (2020) who stress the importance of making the authors’ understanding of *bias* explicit whenever it is investigated, following the taxonomy of harms (Barocas et al., 2017). Our understanding of *bias in link prediction* is based on the idea of *representational harm*, more specifically, “differences in system performance for different social groups” (Blodgett et al., 2020, p. 5456). For example, in the case of gender bias in occupation predictions, a link prediction model that predicts the occupations of women less accurately than those of men would be deemed as behaving harmfully. We consider this behavior as harmful, because deploying such a biased model in a downstream application can make it less useful for women than men. The extent of harm that is caused depends on the societal relevance of the application, e.g., it might be used for job applications or credit approval.

Beyond measuring bias in link prediction model predictions, we also investigate *data bias* in the knowledge graph (KG) datasets that we use. We consider the data to be biased if it is highly imbalanced with regard to some ideal distribution across social groups. An example of such an imbalance would be a dataset that contained significantly more facts about men than women and no facts about individuals with other gender identities.

The examples above also show that we measure bias in a specific context, defined by a *sensitive attribute* and a *target property*. A sensitive attribute is an inherent characteristic of an entity worthy of (legal or other) protection. Typical examples are gender, race, ethnicity, religion or worldview, disability, age, and sexual identity. Each sensitive attribute usually defines multiple *groups*, categorical options that a person can belong to, e.g., female gender. We view a target property as some notable property or achievement of an entity. Typical examples are their occupation, awards received, degrees, or where a person was educated. Both the target property and the sensitive attribute need to be expressed by one specific relation in the dataset. This means that the dataset-specific meaning of the respective relation matters: Here, we want to measure the influence of gender on the accuracy of occupation predictions. Following Keyes et al. (2021), we define *gender* as a multiplicitous concept expressing, e.g., identity and behaviors, going beyond bodily attributes that determine the biological *sex*

of a person. While we only discuss gender in the following, we note that for the gender identities “female” and “male” that we analyze, the distinction between gender and sex is not explicit in Wikidata, because the same entity is used to express both concepts.¹ In addition, due to data scarcity, we cannot analyze gender bias for other gender identities, such as “non-binary”.² Due to a lack of reliable information, we do not distinguish between cisgender and transgender individuals when performing our analysis of gender bias for women and men. We refer to the HCI Gender Guidelines for further information about the terminology and concepts discussed above.³

3 A Model-Agnostic Approach for Measuring Bias in Link Prediction

3.1 Our Main Idea

Prior work on bias in link predictions has studied bias at the level of embeddings (Kraft and Usbeck, 2022). We instead propose to assess such bias in a model-agnostic manner by measuring bias directly on the test set predictions that each link prediction model produces. This allows us to measure bias on link prediction model classes that have not yet been evaluated with respect to bias. We focus on tail entity predictions for a given target relation, for example (Barack Obama, occupation, ?). The predictions serve as the input for group fairness metrics that measure extrinsic performance, meaning that we do not access any internal representations of the model. As group fairness metrics are usually defined for classification tasks (Verma and Rubin, 2018), we need to reframe the link prediction task accordingly (§3.2). Using these metrics allows us to investigate bias as a notion of metric-specific performance differences (§3.3). The choice of the sensitive attribute and target relation is largely subject to their prevalence in the dataset (§3.5). Due to limitations in the data (explained in Appendix B + C), our experiments focus on gender as a sensitive attribute using two gender identities and occupation as the target relation. Without loss of generality, we also use these as examples while explaining our method, but stress that our approach can be used for attributes and targets with more than two groups (§3.3).

¹e.g. <https://www.wikidata.org/wiki/Q6581072>

²The reasons for this are further discussed in §3.5, §4.1, §6, §7, and Appendix B.

³<https://www.morgan-klaus.com/gender-guidelines.html> (Version 1.1)

3.2 Recasting Link Prediction as a Multi-Class Classification Task

Link prediction is typically defined as a ranking task, where each model produces continuous plausibility score values. For tail entity predictions – which we use for bias measurement – the model scores each possible entity in the dataset when given a combination of a head entity and a relation $(h, r, ?)$. A model that has learned the task well should thus emit high scores for the entities that are most plausible and the highest one for the entity that is the true tail entity. This enables us to reframe link prediction as a *multi-class classification* task, by defining each tail entity as a separate *class*. For example, when predicting occupation relationships, the set of all occupations in the dataset is the set of candidate tail entities, which can be viewed as class labels. A model is expected to predict true occupations, i.e., the *true label*, of a given person provided as the head entity. We define the tail entity that receives the highest plausibility score, i.e., rank 1, as the *predicted label*.

We note that this framing of the link prediction task best applies to one-to-one and many-to-one relations. For one-to-many and many-to-many relations, link prediction is technically a multi-label multi-class classification task. For a given human as head entity, there can be multiple true labels.⁴ However, we cannot account for this in our approach as we want to avoid data leakage between the training/validation and test set: Our bias measurement is solely based on the test set, so the evaluation of our model predictions on the test set should be independent of the training set. We therefore only consider the occupations in the test set as true labels. We discuss a related aspect in regard to extracting information about our sensitive attribute in §3.5. This limitation applies to all sensitive attribute groups, so we assume that this does not influence the relative pattern in the bias scores between the groups, only their absolute values.

3.3 Selection of Fairness Metrics

Fairness metrics measure the performance for a given classifier, i.e., a link prediction model, using the following definitions: The predicted tail entity is denoted by \hat{y} , while the true one is indicated by y . The set of classes \mathcal{Y} consists of candidate tail

⁴For instance, our target relation occupation is a many-to-many relationship, as a person can have multiple occupations and multiple persons can have the same occupation.

entities $t \in \{t_1, \dots, t_{T-1}\}$ as well as the OTHER class t_T , where T is the total number of classes. We focus on sensitive attributes s with two values, $s \in \{0, 1\}$.⁵ In the following, we introduce the three fairness metrics that we selected. The equations show how the *performance gap* G (Orgad and Belinkov, 2022) is measured for each target property class t , e.g., the absolute difference between the metric calculated for men versus women for a given occupation. We establish this notion of bias in our bias statement (§2).

Demographic Parity (DP) measures the *selection rate* (SelR), i.e., the probability of a given class to be predicted. For example, it answers the question “Which percentage of the persons predicted to be lawyers are men versus women?” It does not use information about the true class of the respective predictions (Verma and Rubin, 2018).

$$\begin{aligned} \text{DPG}(s, t) &= |P(\hat{y} = t | s = 1) - P(\hat{y} = t | s = 0)| \\ &= |\text{SelR}(s = 1, t) - \text{SelR}(s = 0, t)| \end{aligned} \quad (1)$$

This metric can show a potential general imbalance in the predictions. Such prediction imbalances may also reflect data bias, allowing us to analyze this connection. For using this metric in the context of binary classification and debiasing, we refer to work discussing the strengths and weaknesses of this metric (Dwork et al., 2012; Hardt et al., 2016).

Predictive Parity (PP) measures the positive predictive value (PPV), also known as *precision* (Prec), for a given class (Verma and Rubin, 2018; Chouldechova, 2017). Precision is a well-known evaluation metric that accounts for the percentage of correct predictions (true positives, TP) out of all persons predicted as belonging to that class, i.e., out of all true positives and false positives (FP). Achieving high precision for a class therefore means that if a classifier predicts a class, it is very likely that this prediction truly belongs to this class (Sokolova and Lapalme, 2009). The gap is:

$$\begin{aligned} \text{PPG}(s, t) &= |P(y = t | \hat{y} = t, s = 1) - \\ &P(y = t | \hat{y} = t, s = 0)| \quad (2) \\ &= |\text{Prec}(s = 1, t) - \text{Prec}(s = 0, t)| \end{aligned}$$

We choose this and the following metric because they are well-established in the algorithmic fairness

⁵In theory, the approach can be extended to the non-binary case. This is shown by Keidar et al. (2021), however they do not discuss how the choice of their averaging strategy influences the interpretation of the bias score.

community (Verma and Rubin, 2018; Hutchinson and Mitchell, 2019; Barocas et al., 2019), and each focuses on different capabilities of a classifier. For instance, the *precision–recall trade-off* implies a trade-off between Predictive Parity and Equality of Opportunity (Buckland and Gey, 1994). Also, an impossibility theorem from the algorithmic fairness community (Chouldechova, 2017) proves that these two notions of fairness cannot be achieved simultaneously in non-trivial scenarios.

Equality of Opportunity (EO) measures the true positive rate (TPR), also known as *recall* (Rec) (Hardt et al., 2016). For a given class, it measures the percentage of correct predictions (true positives) out of all persons that actually belong to that class. Achieving high recall for a class therefore means that the classifier identified most of the persons that truly belong to this class (Sokolova and Lapalme, 2009). The corresponding gap is:

$$\begin{aligned} \text{EOG}(s, t) &= |P(\hat{y} = t | y = t, s = 1) - \\ &P(\hat{y} = t | y = t, s = 0)| \quad (3) \\ &= |\text{Rec}(s = 1, t) - \text{Rec}(s = 0, t)| \end{aligned}$$

3.4 Analyzing Bias at Three Levels of Detail

In order to conduct a comprehensive and critical analysis, we calculate the above metrics at three levels of granularity. Each highlights a different aspect of model behavior: (i) the broadest one provides **one score per model**, (ii) a more detailed view yields **one score for each sensitive attribute group**, e.g., men vs. women, and (iii) the most detailed one provides **one score for each target property class and sensitive attribute group**, e.g., female lawyers. For (iii), we calculate the metric (selection rate, precision, or recall) for each individual target property class without averaging, e.g., $\text{Rec}(s = 1, t = 0)$. For (ii), we calculate the arithmetic mean for a specific sensitive attribute group, e.g., $s = 1$, across all T target property classes:

$$\text{EOG}(s = 1) = \frac{1}{T} \sum_{i=1}^T \text{Rec}(s = 1, t_i) \quad (4)$$

Calculating the average in this way means that we use *macro-averaging* assigning all classes equal importance (Sokolova and Lapalme, 2009). For (i), we invoke Equations 1–3 and average the results, again using macro-averaging:

$$\text{EOG}(s) = \frac{1}{T} \sum_{i=1}^T \text{EOG}(s, t_i) \quad (5)$$

Table 1: Data for measuring link prediction bias on both datasets using **occupation** as the target property and **gender** as the sensitive attribute. Based on the prevalence in the test set of HUMANW5M-3MIL, we use a **minimum count threshold** of 100. This means that we consider all occupations with more than 100 occurrences as separate classes, aggregating the remaining facts in the class OTHER.

Occupation	in Test Set	with Gender	Thereof Men		Thereof Women	
other	1,534	399	336	84%	63	16%
politician	1,070	308	274	89%	34	11%
writer	262	69	58	84%	11	16%
lawyer	253	78	72	92%	6	8%
actor	158	47	34	72%	13	28%
association football player	142	32	31	97%	1	3%
poet	129	28	21	75%	7	25%
novelist	109	33	19	58%	14	42%
screenwriter	106	26	26	100%	0	0%
sum over all occupations	3,763	1,020	871	85%	149	15%

3.5 Data-Driven Choice of Target Property Classes and Sensitive Attribute

For all link prediction models, the long tail distribution typical for knowledge graph (KG) datasets (Zhang et al., 2020) presents a challenge: A small set of entities appears often, while most entities appear only a handful of times, even in large datasets. We account for this by choosing the target property classes, the sensitive attribute, and its groups based on their prevalence in the dataset. This means that each class needs to be properly represented for each sensitive attribute group, as it is also discussed in similar work in other domains (Seyyed-Kalantari et al., 2020; De-Arteaga et al., 2019). To achieve this, we reduce the number of classes significantly by **aggregating occupations below a minimum count threshold in the class OTHER**, similar to Keidar et al. (2021). The count threshold is **based on the test set of the dataset** since only this part is used for bias measurement. Using only the test set is necessary to avoid data leakage, as we directly use a model’s predictions of the target property facts as input for our measurement:

Given a trained link prediction model, we extract only the facts concerned with our selected target property from the test set tail entity predictions, i.e., the (personXY, occupation, ?) facts. For each person – corresponding to the head entity – we then search the entire dataset for their sensitive attribute information, e.g., a fact stating their gender. We argue that retrieving the sensitive attribute information from the entire dataset is reasonable and does not constitute data leakage, since we only extract ground truth facts from the dataset. To be clear, **we never predict the sensitive attribute** of a person, only their target property. This means that the *data*

basis for the bias measurement consists of persons with a **target property fact in the test set** and a **sensitive attribute fact somewhere in the dataset**.

While ensuring a sufficient data basis is necessary for a valid bias measurement, using a **minimum count threshold is also connected to the issues of data scarcity and data bias** (§2): (i) Facts about members of minority groups will naturally be less frequent than for those of the majority group. In addition, (ii) groups might be underrepresented due to biased selection processes in society that contributed to the creation of the data. In our case, the threshold leads to us only considering female and male as identities, while having to disregard other gender identities due to data scarcity and likely representation bias, as well. We argue that a bias analysis can still be performed under these circumstances, but that the **data basis and limitations should be clearly acknowledged**.

4 Experiments

4.1 Creating the HUMANW5M-3MIL Dataset

We created HUMANW5M-3MIL, a modified subset of Wikidata5M (Wang et al., 2021b) based on Wikidata (Vrandečić and Krötzsch, 2014), consisting of 3 million facts about humans, meaning that the head entity of each triple is always a human entity. For each entity in the dataset, a textual *description* consisting of the first section of the corresponding Wikipedia article in English is available as well as a short English *label* for each entity and relation (Wang et al., 2021b). We argue that a smaller dataset only consisting of human facts is useful to reduce the noise in the dataset and the time required to train and evaluate the models. This approach follows previous work (Bourli

Table 2: Prediction quality of all trained link prediction models on the test set measured using typical **accuracy metrics**. We report the metrics averaged over head and tail entity predictions and separately only for tail entity predictions. The best scores are highlighted in bold. The arrows express whether a high or a low value of the metric corresponds to high accuracy.

	Model	Prediction Type	MR ↓	MRR ↑	Hits@1 ↑	Hits@3 ↑	Hits@10 ↑
KG only	TransE	averaged	188,784	19.21	16.02	20.74	24.47
		tail	11,620	38.29	32.04	41.29	48.63
	DistMult	averaged	176,300	15.62	11.14	18.42	22.44
		tail	8,405	30.76	22.00	36.34	44.09
	RotatE	averaged	221,341	14.80	11.44	17.08	19.50
		tail	19,552	29.56	22.86	34.12	38.92
KG + LM	SimKGC _{IB}	averaged	91,588	32.96	30.19	34.08	38.02
		tail	255	64.79	60.06	67.04	73.60
	SimKGC _{IB+SN+PB}	averaged	91,737	32.91	30.31	33.93	37.60
		tail	276	64.75	60.14	66.89	73.24

and Pitoura, 2020; Keidar et al., 2021), however the respective datasets are not publicly available. We also created this dataset due to issues we find in Wikidata5M: (i) The relation P21⁶, which expresses human *sex or gender*, is not contained in the dataset, despite gender being the most frequently investigated sensitive attribute (Costa-jussà, 2019). (ii) An exploratory analysis revealed data quality issues in the entity labels such as typos or labels not matching the current English Wikidata labels. To address these issues, we merge the human facts of Wikidata5M with gender facts and English labels taken from a current Wikidata version (the truthy triples file from January 2, 2022). We ensure that each entity has a label and a description, meaning that we exclude entities that only have one or the other. For all remaining human entities, we extract the gender facts, if they exist. We limit our analysis to male and female gender, as data on non-binary gender identities and intersex people is very scarce in Wikidata (Klein et al., 2016; Zhang and Terveen, 2021)). In our case, other gender identities and intersex people are only represented by fewer than 500 occurrences combined. As the entities expressing human gender are not part of Wikidata5M and therefore lack a description, we use the first section of the Wikipedia articles for masculinity⁷ and femininity⁸. The resulting dataset contains ca. 11 million triples, which we randomly sample down to 3,101,160 triples, to reduce the dataset size. The resulting dataset, HUMANW5M-3MIL, contains 1,396,220 unique entities – 1,269,907 thereof human – and 225 relations (Table 7). Table 8 shows that HUMANW5M-3MIL is representative of the larger raw dataset,

when considering the manually selected candidate relations that express sensitive attributes or target properties. For instance, the *sex or gender* relation comprises ca. 13.5% of each dataset. We use comparably large evaluation sets, as our bias score calculation is only based on the test set, specifically a [0.9, 0.05, 0.05] train/validation/test random split (compared to [99.9995, 0.00025, 0.00025] for Wikidata5M), as the evaluation split size of ca. 155,000 triples is still manageable for all models we train on our dataset. Further details about the creation process of the dataset are given in Appendix B. We make the code for creating the dataset along with the data files available.⁹

4.2 Models and Training Details

We demonstrate our model-agnostic approach by comparing two model classes: knowledge graph embeddings (KGEs) that learn only from the structure contained in the knowledge graph dataset (*KG only*) and language model (LM)-based models that further also have access to the entity descriptions and relation labels (*KG+LM*).

KG only models: TransE, DistMult and RotatE. KGEs learn a dense embedding for each entity and relation in the dataset, capturing relationships between entities in a latent space (Nguyen, 2021). We choose TransE (Bordes et al., 2013) and DistMult (Yang et al., 2015) because they are common baseline models from different model families (Rossi et al., 2021a). RotatE (Sun et al., 2019) is an expressive state-of-the-art model from the same model class as TransE. We use the self-adversarial negative sampling loss (Sun et al., 2019) for all models. After hyperparameter tuning (Appendix A), we train all models for 400 epochs, using a

⁶<https://www.wikidata.org/wiki/Property:P21>

⁷<https://en.wikipedia.org/wiki/Masculinity>

⁸<https://en.wikipedia.org/wiki/Femininity>

⁹<https://github.com/lena-schwert/comparing-bias-in-KG-models>

Table 3: Bias in occupation predictions **averaged across all occupation classes**. The bias score correspond to performance gaps between predictions for men and women. The highest bias scores per fairness metric are highlighted in bold. DPG: Demographic Parity Gap, PPG: Predictive Parity Gap, EOG: Equality of Opportunity Gap. *: 3,763 occupation facts were available in total for HUMANW5M-3MIL.

Model Class	Model	DPG (Selection Rate)	PPG (Precision)	EOG (Recall)	# of Facts Used*
KG only	TransE	0.51	0.001	0.03	3,735
	DistMult	0.47	0.004	0.001	3,758
	RotatE	0.32	0.003	0.04	3,709
KG + LM	SimKGC _{IB}	0.57	0.04	0.08	3,726
	SimKGC _{IB+SN+PB}	0.54	0.02	0.12	3,721

Table 4: Link prediction bias results **separated for men and women** showing the absolute fairness metric scores. In some cases the absolute difference of the male and female score does not exactly match the *gap* scores in Table 3, because all results were rounded to two or three decimals. We highlight the entries with the highest difference in bold, i.e., the same entries as in Table 3. DP: Demographic Parity, PP: Predictive Parity, EO: Equality of Opportunity

Model Class	Model	DP (Selection Rate)		PP (Precision)		EO (Recall)	
		Male	Female	Male	Female	Male	Female
KG only	TransE	0.76	0.24	0.043	0.042	0.09	0.05
	DistMult	0.74	0.26	0.043	0.047	0.109	0.108
	RotatE	0.66	0.34	0.041	0.044	0.08	0.05
KG+LM	SimKGC _{IB}	0.79	0.21	0.49	0.45	0.32	0.41
	SimKGC _{IB+SN+PB}	0.77	0.23	0.51	0.49	0.31	0.43

Table 5: **Deviation of predicted occupations for women from the data distribution** using the KG+LM model SimKGC_{IB+SN+PB}. For each of the nine occupation classes, we calculate the difference between the selection rate and the distribution of the occupations in the test set of HUMANW5M-3MIL.

	Selection Rate	Data Distribution	Difference
averaged	0.23	0.15	+ 0.08
other	0.12	0.16	- 0.04
politician	0.15	0.11	+ 0.04
writer	0.25	0.16	+ 0.09
lawyer	0.12	0.08	+ 0.04
actor	0.33	0.28	+ 0.05
assoc. football player	0.08	0.03	+ 0.05
poet	0.40	0.25	+ 0.15
novelist	0.42	0.42	± 0.00
screenwriter	0.20	0.00	+ 0.20

batch size of 1,024, an embedding dimensionality of 512, and 32 negative samples per training triple. For TransE and DistMult a learning rate of 0.001 and for RotatE a learning rate of 0.01 is used.

KG+LM model: SimKGC LM-based models utilize pre-trained Transformers (Vaswani et al., 2017) that are fine-tuned on a knowledge graph dataset. To that end, an input sequence is created out of the entity descriptions instead of using the entity and relation IDs. We choose SimKGC (Wang et al., 2022), as it significantly outperforms earlier models with respect to accuracy and computational efficiency. It has a bi-encoder architecture using the pre-trained BERT-base (Devlin et al., 2019). One encoder learns relation-aware

head entity embeddings and the other one tail entity embeddings. The plausibility scoring of triples is then simply achieved using cosine similarity. We train the SimKGC_{IB} and the SimKGC_{IB+SN+PB} model variants to investigate whether they exhibit different bias behavior. We do not conduct hyperparameter tuning, as the parameters for Wikidata5M used in the original paper (Wang et al., 2022) deliver strong results on our validation set. SimKGC uses the InfoNCE loss with an additive margin (Le-Khac et al., 2020). We train for 1 epoch using a batch size of 1,024, a learning rate of 3×10^{-5} and a weight decay of 0.0001. We provide further details for reproducing the experiments in Appendix A.

4.3 Evaluation Protocol

We evaluate our link prediction models for accuracy using mean rank (MR), mean reciprocal rank (MRR) as well as Hits@1, Hits@3, and Hits@10 (Rossi et al., 2021a). We calculate the ranks using the *filtered setting* (Bordes et al., 2013). Since we only use tail entity predictions for measuring bias, we compute the metrics (i) averaged across head and tail entity predictions and separately (ii) only for tail entity predictions. Following §3.5, we choose gender as a sensitive attribute and occupation as the target property for measuring bias in the trained models. We describe in Appendix C how other combinations of sensitive attributes and target property are not analyzed due to data scarcity.

4.4 Model Accuracy and Data Bias Results

Referring to Table 2, we note that the Hits@1 accuracy for tail entity predictions is the most relevant metric for bias measurement, since the tail entities with rank = 1 are used as the predicted class labels, i.e., the predicted occupation. The performance on tail entity predictions is clearly higher than the one averaged across head and tail entity predictions since there are fewer unique tail than head entities, making this prediction easier. Performance on tail entity predictions varies between 11.14 (DistMult) and 60.14 (SimKGC_{IB+SN+PB}). When comparing the two model classes, the KG+LM models clearly outperform the KG only models. Among the KG only models, TransE obtains the best Hits@1 result (16.02), thus outperforming the two other more recent and complex models.

Table 1 shows the absolute counts and the relative distributions of the occupation classes over the two considered gender identities (male, female). It also shows that we choose a *minimum count threshold* of 100 facts per occupation, resulting in eight distinct occupations, aggregating the remaining ones in the class OTHER. When comparing the relative distribution of facts per gender, it is evident that the data is biased: Out of the 1,020 facts that we use for bias measurement, 85% are about men and only 15% about women, while a 50%–50% distribution would be unbiased when considering these two gender identities. The occupation with the largest gender bias in the data is *screenwriter* (100% men) and the one with the smallest bias is *novelist* (58% men, 42% women). In addition, we note again that gender identities beyond women and men are severely underrepresented in the data,

constituting only 0.005% of the gender facts, which is significantly lower than the 0.1–2% estimated by Goodman et al. (2019).

4.5 Results on Gender Bias in Occupation Predictions

For describing and analyzing the gender bias that our models exhibit in its occupation predictions, we consider the *three levels of detail* as introduced in §3.4. Tables 3, 4, and 5 refer to levels of detail (i), (ii), and (iii), respectively. These allow us to answer three different research questions.

Q1: Are KG+LM models more biased than KG only models?

As Table 3 shows, the bias scores are generally higher for the KG+LM models than for KG only models. Comparing the difference between the most biased models for each class shows that it is most pronounced for the demographic parity gap (DPG): $0.57 - 0.47 = 0.1$, followed by the equality of opportunity gap (EOG) $0.12 - 0.04 = 0.08$. These results suggest that the additional textual data the KG+LM models have access to leads to biased occupation predictions and that this has the most pronounced effect on DPG and EOG. The KG only models, in contrast, here manage to obtain fairly unbiased results, with scores close to zero. We note that the column “# of facts used” shows how many facts contributed to the score, since a fact can only be considered when the predicted tail entity is an occupation and not another type of entity.

Q2: Does the bias originate in higher-quality predictions for men or women?

To answer this question, we refer to Table 4, which shows the previously described results separately for men and women. For DP, we observe that the selection rate for predictions for men is generally higher. We connect this to the data distribution in Q3. For PP and EO we make two observations: First, most KG only models – which obtain essentially unbiased results – obtain dismal precision and recall scores (they only appear strong enough when evaluated using ranking metrics). Second, for the KG+LM models we observe opposing trends: With regard to precision, the prediction quality for men is slightly higher, while for recall, the prediction quality for women is noticeably higher. Especially the latter trend is surprising since the data for women is more limited. These observations show why this level of detail is important for a comprehensive

bias analysis: While KG only models exhibit far less bias, they predict occupations inaccurately despite an acceptable overall accuracy (Table 2). In addition, we conclude that predictions for men are not necessarily more accurate than those for the women, despite the significantly larger amount of data for men (85% of all occupation facts.).

Q3: Are there occupation classes that are predicted more often than expected based on their distribution in the data?

We may consider the demographic parity results for SimKGC_{IB+SN+PB}, our most accurate model, as an example. As explained earlier (§3.3), DP measures selection rate imbalances that we expect to mirror the data bias. Table 5 shows the per-class differences between the selection rate (predicted occupation) and the respective distribution in the data (actual occupation) when predicting the occupation of women. Whenever the difference is positive, the model predicts the given occupation for more women than expected (and vice versa). On average, the probability of women having a given occupation is overestimated by 0.08, with the occupations “poet” and “screenwriter” contributing the most to this score. Despite this, the model does predict this occupation for some women, as female screenwriters do exist in the training dataset. This might be due to the entity descriptions that this KG+LM model has access to, potentially because the person’s occupation might be similar to a screenwriter or mention related words.

5 Conclusion

We present a model-agnostic approach for measuring bias in link prediction along with the first experimental study that measures bias in language model (LM)-based link prediction models (KG+LM), comparing it with bias in knowledge graph embedding (KGE) models (KG only). Using a selection of fairness metrics and analyzing our results at three levels of detail, we find that the KG+LM models are more biased. We discuss the relationship between bias, link prediction accuracy metrics and data bias. For our experiments, we create HUMANW5M-3MIL, a subset of 3 million facts about humans contained in Wikidata (Vrandečić and Krötzsch, 2014). We have made our code and the dataset available to the public to encourage further research on these topics.

6 Limitations

In the following we discuss the limitations of our work and how they might be addressed.

Our study considers a single sensitive attribute, gender, limited to two gender identities, female and male. We also note that the approach can be extended to sensitive attributes with more than two groups, requiring additional decisions on how to average the bias scores across the sensitive attribute groups in an interpretable way. This limitation is caused by data scarcity, as we describe in §3.5, §4.1 and Appendix B + C.

Using only the test set of a dataset for bias measurement has a few methodological implications: First, the bias in the test set might not be representative of the bias contained in the other splits of the data set. In our approach we used simple random splitting, where all facts are randomly distributed over the three splits, meaning that the distribution of the relations might not be the same in all splits. This approach is called the *transductive setting*, which is currently the most prevalent method of splitting knowledge graph datasets (Wang et al., 2021b). To rule out differences between the splits to some degree, a potential solution is a stratified split, conditioned on the relations in the dataset. This would enforce, for instance, that each split has the same relative amount of gender and occupation facts. This solution is however only applicable when the researcher has control over the dataset split creation process. Second, in order to have a sufficient data basis for each target property class across sensitive attribute groups (§3.5), the dataset or the test split size needs to be quite large. However, training models on large datasets requires the availability of adequate computational resources that many researchers do not have access to.

Using fairness metrics for bias measurement means that the notion of bias is closely connected to what is considered as a “misclassification”. We note that we do not take into account the severity of misclassifications, e.g., that predicting a novelist to be a writer is less wrong than predicting them to be a diplomat. This would require a semantic analysis of the labels of both the true and the predicted tail entities. This might also be addressed by clustering entities with similar meanings together, e.g., predicting groups of occupations instead of single occupations.

7 Ethical Considerations

For our analysis of gender bias, we rely on factual statements contained in Wikidata¹⁰, a crowd-sourced, public knowledge graph. This means that we utilize gender information that was added to the platform by largely anonymous editors. These statements – and other statements describing demographics – might therefore not correspond to the self-identification of the respective persons or they might be incorrect, especially if human or automated data quality control mechanisms fail (Heindorf et al., 2019).

In addition, we acknowledge that knowledge graphs reflect a limited world view, because their creation process is subject to various biases, such as representation bias, popularity bias, and sampling bias (following the definitions by Mehrabi et al. 2021). In the field of knowledge graphs, these problems were first described by Janowicz et al. (2018) and recently reviewed by Kraft and Usbeck (2022). For example, facts about the non-Western world are underrepresented and persons with occupations in arts, sports, and science and technology are overrepresented (Radstok et al., 2021; Beytía et al., 2022).

One consequence of the biases mentioned above is our decision to only consider male and female gender in our analysis, as all other gender identities combined, such as non-binary, amount to fewer than 500 facts in the entire dataset. To analyze bias for these gender identities, a larger dataset or a different approach than ours would be necessary. We discuss these limitations and our understanding of gender in §2.

As described in our bias statement (§2), our notion of bias focuses on performance differences for different social groups. We note that this is a very specific, limited conceptualization of bias that could be extended by considering real-world distributions or normative connotations such as stereotypes. However, we believe that the contribution of our work is still useful for analyzing whether link prediction models work as intended, especially because it allows for comparing different model classes.

To conclude, we stress that the intended use of our approach is to identify concerning model behavior in a specific context defined by a sensitive attribute and a target property. We emphasize that the selected fairness metrics should not, e.g., be

used as constraints during model training without a deeper analysis of what notions of fairness are suitable in the context of how the model will be used.

Acknowledgements

We thank Lisa Gotzian and Steffen Berhorst for discussions about the bias measurement approach and for feedback on the manuscript. We further thank the three anonymous reviewers for their thoughtful comments.

References

- Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. 2021. *PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings*. *Journal of Machine Learning Research*, 22(82):1–6.
- Mario Arduini, Lorenzo Noci, Federico Pirovano, Ce Zhang, Yash Raj Shrestha, and Bibek Paudel. 2020. Adversarial learning for debiasing knowledge graph embeddings. *MLG 2020: 16th International Workshop on Mining and Learning with Graphs - A Workshop at the KDD Conference*.
- Solon Barocas, Kate Crawford, Aaron Shapiro, and Hanna Wallach. 2017. The problem with bias: Allocative versus representational harms in machine learning. *SIGCIS Conference*.
- Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. *Fairness and Machine Learning*. fairmlbook.org. <http://www.fairmlbook.org>.
- Pablo Beytía, Pushkal Agarwal, Miriam Redi, and Vivek K Singh. 2022. *Visual gender biases in Wikipedia: A systematic evaluation across the ten most spoken languages*. *Proceedings of the International AAAI Conference on Web and Social Media*, 16:43–54.
- Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. 2020. *Fairlearn: A toolkit for assessing and improving fairness in ai*. Technical Report MSR-TR-2020-32, Microsoft.
- Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of “bias” in nlp. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5454–5476.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems*.

¹⁰<https://www.wikidata.org>

- Styliani Bourli and Evaggelia Pitoura. 2020. [Bias in knowledge graph embeddings](#). *Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 6–10.
- Michael Buckland and Fredric Gey. 1994. The relationship between recall and precision. *Journal of the American Society for Information Science*, 45:12–19.
- Alexandra Chouldechova. 2017. [Fair prediction with disparate impact: A study of bias in recidivism prediction instruments](#). *Big Data*, 5:153–163.
- Marta R. Costa-jussà. 2019. [An analysis of gender bias studies in natural language processing](#). *Nature Machine Intelligence*, 1:495–496.
- Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019. [Bias in bios: A case study of semantic representation bias in a high-stakes setting](#). *Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, pages 120–128.
- Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1:4171–4186.
- Yupe Du, Qi Zheng, Yuanbin Wu, Man Lan, Yan Yang, and Meirong Ma. 2022. Understanding gender bias in knowledge base embeddings. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1:1381–1395.
- Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. [Fairness through awareness](#). *ITCS 2012 - Innovations in Theoretical Computer Science Conference*, pages 214–226.
- Joseph Fisher, Arpit Mittal, Dave Palfrey, and Christos Christodoulopoulos. 2020a. [Debiasing knowledge graph embeddings](#). *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7332–7345.
- Joseph Fisher, Dave Palfrey, Christos Christodoulopoulos, and Arpit Mittal. 2020b. Measuring social bias in knowledge graph embeddings. *Proceedings of the Knowledge-Graph Bias Workshop*, page 7332–734.
- Seraphina Goldfarb-Tarrant, Rebecca Marchant, Ricardo Muñoz Sánchez, Mugdha Pandya, and Adam Lopez. 2021. [Intrinsic bias metrics do not correlate with application bias](#). *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 1926–1940.
- Michael Goodman, Noah Adams, Trevor Corneil, Baudewijntje Kreukels, Joz Motmans, and Eli Coleman. 2019. [Size and distribution of transgender and gender nonconforming populations: A narrative review](#). *Endocrinology and Metabolism Clinics of North America*, 48(2):303–321. Transgender Medicine.
- Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems (NIPS)*, pages 1–22.
- Stefan Heindorf, Yan Scholten, Gregor Engels, and Martin Potthast. 2019. [Debiasing vandalism detection models at Wikidata](#). *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, 2:670–680.
- Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D’Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. [Knowledge graphs](#). *Synthesis Lectures on Data, Semantics, and Knowledge, No. 22*.
- Ben Hutchinson and Margaret Mitchell. 2019. [50 years of test \(un\)fairness: Lessons for machine learning](#). *Proceedings of FAT* 2019: Conference on Fairness, Accountability and Transparency*, pages 49–58.
- Krzysztof Janowicz, Bo Yan, Blake Regalia, Rui Zhu, and Gengchen Mai. 2018. Debiasing knowledge graphs: Why female presidents are not like female popes. *CEUR Workshop Proceedings*, 2180:1–5.
- Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.
- Daphna Keidar, Mian Zhong, Ce Zhang, Yash Raj Shrestha, and Bibek Paudel. 2021. [Towards automatic bias detection in knowledge graphs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3804–3811, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Os Keyes, Chandler May, and Annabelle Carrell. 2021. [You keep using that word: Ways of thinking about gender in computing research](#). *Proc. ACM Hum.-Comput. Interact.*, 5(CSCW1).
- Maximilian Klein, Harsh Gupta, Vivek Rai, Piotr Konieczny, and Haiyi Zhu. 2016. [Monitoring the gender gap with Wikidata human gender indicators](#). In *Proceedings of the 12th International Symposium on Open Collaboration, OpenSym ’16*, New York, NY, USA. Association for Computing Machinery.

- Angelie Kraft and Ricardo Usbeck. 2022. [The Lifecycle of "Facts": A Survey of Social Bias in Knowledge Graphs](#). *arXiv*.
- Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. 2020. [Contrastive representation learning: A framework and review](#). *IEEE Access*, 8:193907–193934.
- Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. [A survey on bias and fairness in machine learning](#). *ACM Computing Surveys*, 54.
- Dat Quoc Nguyen. 2021. [A survey of embedding models of entities and relationships for knowledge graph completion](#). *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 1–14.
- Hadas Orgad and Yonatan Belinkov. 2022. [Choose your lenses: Flaws in gender bias evaluation](#). In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 151–167, Seattle, Washington. Association for Computational Linguistics.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [PyTorch: An imperative style, high-performance deep learning library](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- Heiko Paulheim. 2016. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8:489–508.
- Wessel Radstok, Melisachew Wudage Chekol, and Mirko Tobias Schafer. 2021. Are knowledge graph embedding models biased, or is it the data that they are trained on? *CEUR Workshop Proceedings*, 2982.
- Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. 2021a. [Knowledge graph embedding for link prediction: A comparative analysis](#). *ACM Transactions on Knowledge Discovery from Data*, 15:1–49.
- Andrea Rossi, Donatella Firmani, and Paolo Merialdo. 2021b. [Knowledge graph embeddings or bias graph embeddings? A study of bias in link prediction models](#). *DLAKG 2021: Workshop on Deep Learning for Knowledge Graphs, held as part of ISWC 2021: The 20th International Semantic Web Conference*.
- Laleh Seyyed-Kalantari, Guanxiong Liu, Matthew McDermott, Irene Y Chen, and Marzyeh Ghassemi. 2020. [CheXclusion: Fairness gaps in deep chest X-ray classifiers](#). *Pacific Symposium On Biocomputing*, 26:232–242.
- Marina Sokolova and Guy Lapalme. 2009. [A systematic analysis of performance measures for classification tasks](#). *Information Processing and Management*, 45:427–437.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. [RotatE: Knowledge graph embedding by relational rotation in complex space](#). *Proceedings of the International Conference on Learning Representations*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, pages 6000–6010.
- Sahil Verma and Julia Rubin. 2018. [Fairness definitions explained](#). *ACM/IEEE International Workshop on Software Fairness Fairness (FairWare)*, pages 1–7.
- Denny Vrandečić and Markus Krötzsch. 2014. [Wiki-data: A free collaborative knowledgebase](#). *Commun. ACM*, 57(10):78–85.
- Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021a. [Structure-augmented text representation learning for efficient knowledge graph completion](#). *Proceedings of the Web Conference 2021 (WWW '21)*, pages 1737–1748.
- Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. [SimKGC: Simple contrastive knowledge graph completion with pre-trained language models](#). *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294.
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. [KEPLER: A unified model for knowledge embedding and pre-trained language representation](#). *Transactions of the Association for Computational Linguistics*, 9:176–194.
- Gerhard Weikum, Luna Xin Dong, Simon Razniewski, and Fabian Suchanek. 2021. [Machine knowledge: Creation and curation of comprehensive knowledge bases](#). *arXiv preprint*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

- Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. [Embedding entities and relations for learning and inference in knowledge bases](#). *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. [KG-BERT: BERT for knowledge graph completion](#). *arXiv preprint*.
- Charles Chuankai Zhang and Loren Terveen. 2021. [Quantifying the gap: A case study of Wikidata gender disparities](#). In *Proceedings of the 17th International Symposium on Open Collaboration, OpenSym '21*, New York, NY, USA. Association for Computing Machinery.
- Chuxu Zhang, Huaxiu Yao, Chao Huang, Meng Jiang, Zhenhui Li, and Nitesh V. Chawla. 2020. [Few-shot knowledge graph completion](#). *Proceedings of the AAAI Conference on Artificial Intelligence - AAAI Technical Track: Knowledge Representation and Reasoning*, pages 3041–3048.

A Additional Details for Reproducing the Experiments

We list the training and evaluation time as well as the hardware used for all our models in Table 6.

KG only: TransE, DistMult, RotatE. For training the KG only models on HUMANW5M-3MIL, we largely use the parameters contained in the Graphvite configuration files from the official Wikidata5M benchmark¹¹ published by Wang et al. (2021b). To adapt to our dataset, we conduct minor additional hyperparameter tuning. We explore a small grid testing 6 hyperparameter combinations for each model: batch size $\in \{512, 1024\}$, learning rate $\in \{0.1, 0.01, 0.001\}$. We train TransE and DistMult for 50 epochs (training time: ca. 3.5-7 h, evaluation time: ca. 35 min) and RotatE for 20 epochs (training time: ca. 2.5-5 h, evaluation time: ca. 65 min). We choose our final parameter configuration based on the mean reciprocal rank (MRR) on the validation set, as it has been observed to be the most stable metric among the link prediction metrics (Rossi et al., 2021a).

For all models, we use a batch size of 1,024, an embedding dimensionality of 512, and 32 negative samples per training triple. For TransE and DistMult a learning rate of 0.001, for RotatE a learning rate of 0.01 is used. We train the models for 500 epochs, evaluating after each 100 epochs. Finally, the models trained for 400 epochs are used, since the MRR performance drops slightly afterwards. We use the self-adversarial negative sampling loss (Sun et al., 2019) for all models. For TransE, we use margin $\gamma = 12$ and adversarial temperature = 0.5. For DistMult, we use margin $\gamma = 0$ and adversarial temperature = 2. Again following the Graphvite configuration files, we also apply L3 regularization with a weight of 0.002. For RotatE, we use margin $\gamma = 6$ and adversarial temperature = 0.2.

KG + LM: SimKGC. We use the pre-trained BERT-base in its “uncased” variant (Devlin et al., 2019). Since the authors trained their model on Wikidata5M, a superset of our dataset, we try using the exact same parameters as the original paper (Wang et al., 2022). We use two of their model variants to investigate whether using the self-negative (SN) and pre-batch (PB) sample types lead to different bias behavior compared to the “basic” in-batch (IB) model variant. We therefore train the SimKGC_{IB} and the SimKGC_{IB+SN+PB} model

¹¹<https://graphvite.io/docs/latest/benchmark.html>

variants, using 2 pre-batch negatives for the latter. We train for 1 epoch using a batch size of 1,024, a learning rate of 3×10^{-5} and a weight decay of 0.0001. The remaining parameters are: 400 warmup steps for the linear learning rate scheduler, gradient clipping of 10.0, dropout 0.1, temperature τ is initialized with 0.05, additive margin γ for the InfoNCE loss is 0.02, $\alpha = 0.05$ for graph-based re-ranking is used, 2-hop neighbors are considered, and a maximal token length of 50 for the entity descriptions is used. As these parameters deliver good results on the validation set, we do not conduct hyperparameter tuning.

Implementation details. All implementations are done in Python. The code and data including files to re-create the conda environment are contained in the accompanying GitHub repository¹². All models are based on the deep-learning framework PyTorch (Paszke et al., 2019).

- **KG only: Knowledge Graph Embeddings:** For training models on HUMANW5M-3MIL we use the model implementations and the training pipeline of the v1.8.1 PyKEEN library (Ali et al., 2021). This framework enables single-GPU training and the calculation of evaluation metrics.
- **KG + LM: SimKGC:** We use the implementation that was published alongside the paper of Wang et al. (2022). Their code¹³ includes the calculation of evaluation metrics. The implementations use the Huggingface Transformers library v4.15 (Wolf et al., 2020).
- **Data Bias:** We use our own Python implementation.
- **Link Prediction Bias:** For calculating the predictive parity and equality of opportunity, we use Microsoft’s fairlearn library (Bird et al., 2020), that wraps around scikit-learn’s evaluation metrics. For calculating demographic parity, we modify code from the repository published by Keidar et al. (2021).¹⁴

B Additional Details About the Creation of the Dataset

This section describes how we create the raw version of HUMANW5M-3MIL. This raw version

¹²<https://github.com/lena-schwert/comparing-bias-in-KG-models>

¹³<https://github.com/intfloat/SimKGC/>

¹⁴<https://github.com/mianzng/kgbiasdetec>

Table 6: Training runtime, evaluation runtime and hardware used for training all of our models. *: NVIDIA A100-SXM-80GB, †: AMD EPYC 7502 32-Core CPU.

Model Class	Model	Train. Time	Eval. Time	GPU(s) Used	Other Hardware
KG only	TransE	27 h	35 min	1x NVIDIA A100*	10 GB RAM, 32 CPUs [†]
	DistMult	30 h	35 min	1x NVIDIA A100*	10 GB RAM, 32 CPUs [†]
	RotatE	55 h	66 min	1x NVIDIA A100*	10 GB RAM, 32 CPUs [†]
KG+LM	SimKGC _{IB}	45 min	180 min	4x NVIDIA A100*	15 GB RAM, 50 CPUs [†]
	SimKGC _{IB+SN+PB}	45 min	180 min	4x NVIDIA A100*	15 GB RAM, 50 CPUs [†]

Table 7: Dataset statistics for the raw version of our dataset and the subsampled dataset HUMANW5M-3MIL that we use in our experiments.

	11mil. Raw Dataset	HUMANW5M-3MIL
# of entities	1,732,021	1,396,220
# of human entities	1,503,491	1,269,907
# of relations	292	225
# of train triples	-	2,791,044
# of validation triples	-	155,058
# of test triples	-	155,058
total # of triples	11,114,797	3,101,160

Table 8: Manually selected Wikidata relations of general interest for a bias analysis. This large selection can be considered as candidate relations, since they must exist in sufficient quantity to enable a robust bias analysis. We show the triple counts for each relation and the proportion of this count of the total size of each dataset. The raw dataset version contains 11,114,797 triples. Our final dataset, HUMANW5M-3MIL contains 3,101,160 triples. We ultimately only use the relations "gender" and "occupation" for our bias analysis.

Wikidata Label	Wikidata ID	Relation Expresses...	11mil. Raw Dataset		HUMANW5M-3MIL	
sex or gender	P21	gender	1,501,938	(13.51 %)	418,622	(13.50 %)
country of citizenship	P27	nationality	1,143,007	(10.28 %)	319,123	(10.29 %)
place of birth	P19	nationality	854,080	(7.68 %)	238,162	(7.68 %)
religion	P140	religion	27,805	(0.25 %)	7,827	(0.25 %)
ethnic group	P172	ethnicity	27,235	(0.25 %)	7,751	(0.25 %)
native language	P103	nationality	19,771	(0.18 %)	5,495	(0.18 %)
medical condition	P1050	disability	3,824	(0.03 %)	1,049	(0.03 %)
sexual orientation	P91	sexual orientation	484	(0.004 %)	123	(0.004 %)
occupation	P106	target property	1,095,357	(9.85 %)	305,806	(9.86 %)
educated at	P69	target property	438,207	(3.94 %)	122,195	(3.94 %)
award received	P166	target property	169,758	(1.53 %)	47,661	(1.54 %)
member of political party	P102	target property	126,285	(1.14 %)	35,233	(1.14 %)
employer	P108	target property	79,781	(0.72 %)	22,103	(0.71 %)
position held	P39	target property	75,909	(0.68 %)	21,069	(0.68 %)
field of work	P101	target property	17,757	(0.16 %)	4,956	(0.16 %)
military rank	P410	target property	16,330	(0.15 %)	4,510	(0.15 %)
nominated for	P1411	target property	12,854	(0.12 %)	3,627	(0.12 %)
academic degree	P512	target property	5,315	(0.05 %)	1,558	(0.05 %)
doctoral student	P185	target property	1,415	(0.01 %)	424	(0.01 %)

contains 11,114,797 triples, 1,732,021 entities – thereof 1,503,491 human entities – and 292 relations. To reduce dataset size, we sample it down to 3,101,160 triples, creating HUMANW5M-3MIL.

Details on extracting the labels. As an alternative to using the textual descriptions for entities, i.e., the first section of the corresponding Wikipedia article, we propose using the shorter Wikidata labels. As these contain less information for the KG+LM model to process, using labels instead of descrip-

tions reduces the model runtime. We considered using the alias files provided with Wikidata5M¹⁵, but found that the entity aliases have quality issues such as typos (e.g., for the ‘human’ entity Q5, the first alias is ‘Huamn’) or aliases that do not have the same meaning as the current label (e.g., for the ‘universe’ entity Q1, the first alias is ‘Earth’s universe’). After correspondence with the first author of the

¹⁵<https://deepgraphlearning.github.io/project/wikidata5m>

paper that introduced Wikidata5M (Wang et al., 2021b), we learned that they created the alias files using the “pageterms” property of the MediaWiki API¹⁶. The faulty aliases are thus likely a result of the use of that data source and do not represent genuine entity labels. We therefore extract entity and relation labels from the January 2, 2022 truthy triples Wikidata dump¹⁷. The Wikidata dump files are updated every week and contain the most recent state of Wikidata in different data formats¹⁸. We use the truthy triples file specifically, because it only contains non-deprecated triples, which reduces the amount of metadata contained and therefore the file size.

Details on creating the subset of human facts.

We extract all facts that have a human head entity from the raw triples file provided with the original Wikidata5M files. A human head entity is identified by its “instance of human” (QXX P31 Q5) statement. This initially leads to a subset of 9,804,421 facts about humans. In order to be able to compare KG+LM models using either the shorter labels or the longer descriptions as text input, we only keep entities and relations that have both labels and descriptions. This leads to a removal of 17,528 entities and 2 relations from the dataset, due to deletions and additions that happened between the creation of Wikidata5M (based on the July 2019 Wikidata dump, Wang et al. 2021b) and the extraction of the labels (January 2022 Wikidata truthy triples dump). Removing these entities and relations means that we remove all facts that contain them, leading to 191,178 facts that are excluded in total.

Details on adding gender facts from current Wikidata. In an exploratory analysis of Wikidata5M before creating our dataset, we counted the occurrences of facts that we considered to be of general interest for a bias analysis. With respect to the gender relation (PID: P21) we found that its count is unexpectedly low (about 4,000) compared to the number of human entities in the dataset (1.5 million). Furthermore, we found that these facts express animal sex and not human gender, because the head and tail entities are non-human (tail enti-

ties: Q44148, Q43445). When filtering for gender facts in the human facts subset, we only found 384 facts overall. Through correspondence, the first author of the paper that introduced Wikidata5M (Wang et al., 2021b) informed us that they used Wikidata’s “wbgetentities” API¹⁹ to align Wikidata and Wikipedia entries. Since the Wikidata entities for male²⁰ and female gender²¹ are linked to the same Wikipedia page describing gender²², the API might have therefore omitted facts containing these entities. We therefore use the January 2022 truthy triples dump to extract the gender facts as well. We extract 1,243,734 facts with gender *male* and 258,204 facts with gender *female*. Persons with other gender identities, such as *non-binary*, or intersex people have fewer than 500 occurrences in the entire dataset.

We therefore consider only two gender identities within the context of this study, as the data scarcity would not allow our models to properly represent the other gender identities contained in the dataset.

Adding the gender facts for women and men entails adding two new (tail) entities to the dataset (Q6581072, Q6581097). As these entities do not have descriptions in the original dataset, we use the first section of the Wikipedia articles for masculinity²³ and femininity²⁴.

C Considering Additional Sensitive Attributes and Target Properties

Beyond measuring gender bias in occupation prediction, we did consider using other target properties and sensitive attributes for the analysis of the HUMANW5M-3MIL subset. However – in contrast to using “gender” and “occupation” – we found the respective data bases to be lacking.

The relation “educated at” is the target property with the second-highest counts in HumanWikidata5M. In total, the 438,207 facts have 9,330 different tail entities, i.e., educational institutions such as universities. In the test set, the 8,684 “educated at” facts still have 1,919 different tail entities, only 3 of those with more than 100 occurrences. If the minimum count threshold were set at 100, this would result in an “other” class with 8,439 facts, leading to a very imbalanced class distribution. In

¹⁶<https://www.mediawiki.org/w/api.php?action=help&modules=query%2Bpageterms>

¹⁷<https://dumps.wikimedia.org/wikidatawiki/entities/>

¹⁸https://www.wikidata.org/wiki/Wikidata:Database_download

¹⁹<https://www.mediawiki.org/wiki/Wikibase/API>

²⁰<https://www.wikidata.org/wiki/Q6581097>

²¹<https://www.wikidata.org/wiki/Q6581072>

²²<https://en.wikipedia.org/wiki/Gender>

²³<https://en.wikipedia.org/wiki/Masculinity>

²⁴<https://en.wikipedia.org/wiki/Femininity>

Table 9: Data basis for measuring link prediction bias using **occupation** as the target property and **country of citizenship** as the sensitive attribute. This shows the insufficient data basis for a bias analysis: Even the three best represented countries of citizenship (sum over all occupations ≥ 50) are not sufficiently represented across the individual occupations in the test set of HUMANW5M-3MIL.

Occupation	in Test Set	with Citizenship	USA	France	UK	Other
other	1,534	437	146	33	41	217
politician	1,070	278	104	11	9	154
writer	262	68	9	13	4	42
lawyer	253	76	50	1	1	24
actor	158	44	18	5	2	19
association football player	142	34	2	1	10	21
poet	129	38	2	8	2	26
novelist	109	29	17	3	3	6
screenwriter	106	38	9	5	2	22
Sum over all occupations	3,763	1,042	357	80	74	531

addition, the three most frequent tail entities are “Harvard University” (270 facts), “Yale University” (121 facts), and “University of Michigan” (104 facts), which represent a very limited selection of all educational institutions contained in the dataset. We therefore disregard “educated at” as a target property.

Moving on to additional potential sensitive attributes, the relation “country of citizenship” is the most promising candidate with 1,143,007 facts in HumanWikidata5M. However, when creating an overview of counts per occupation class similar to Table 1, it becomes evident that the data for each sensitive attribute group, i.e., country, is very limited (Table 9). While there are in total 18,396 country of citizenship facts in the test set, this information is only available for 1,020 of the 3,763 occupation facts. The three countries with the highest counts are all Western countries, namely USA (357 facts), France (80 facts), and the UK (74 facts). Even for these countries, the majority of the occupation classes are only represented by 0 to 5 facts. The 110 other countries represented in the test set are all aggregated in the “other” class, which is again the largest class with 531 facts. This means that the sensitive attribute groups are already quite homogeneous, while the “other” group contains the majority of diverse information about “citizenship”. Compared to using two groups for “gender” as the sensitive attribute, choosing the sensitive attribute groups as above would thus result in an unrealistic and uninformative comparison. We hence decided against including “country of citizenship” as a sensitive attribute.

Similar considerations apply to the other relations of interest listed in Table 8, since these also have too few facts per target property class or a very

broad distribution over sensitive attribute groups.

Divergence-Based Domain Transferability for Zero-Shot Classification

Alexander Pugantsov and Richard McCreadie
School of Computing Science, University of Glasgow, UK

Abstract

Transferring learned patterns from pretrained neural language models has been shown to significantly improve effectiveness across a variety of language-based tasks, meanwhile further tuning on intermediate tasks has been demonstrated to provide additional performance benefits, provided the intermediate task is sufficiently related to the target task. However, how to identify related tasks is an open problem, and brute-force searching effective task combinations is prohibitively expensive. Hence, the question arises, *are we able to improve the effectiveness and efficiency of tasks with no training examples through selective fine-tuning?* In this paper, we explore statistical measures that approximate the divergence between domain representations as a means to estimate whether tuning using one task pair will exhibit performance benefits over tuning another. This estimation can then be used to reduce the number of task pairs that need to be tested by eliminating pairs that are unlikely to provide benefits. Through experimentation over 58 tasks and over 6,600 task pair combinations, we demonstrate that statistical measures can distinguish effective task pairs, and the resulting estimates can reduce end-to-end runtime by up to 40%.

1 Introduction

As the accuracy of neural models continues to increase, so does the computational cost of training and storing them. One approach of mitigating such cost is through using pretrained models to enhance performance on a downstream task, a paradigm commonly referred to as *transfer learning*. However, when and why transfer learning works is not concretely understood. Traditionally, selecting the best settings, i.e. tasks and hyperparameters, for transfer often involves an extensive trial-and-error process over many combinations and can quickly make the prospect of applying transfer learning undesirable. As such, it would be valuable to estimate whether a task pair combination

will be effective pre-training, i.e. estimate the *transferability* of a source task to a target task.

The most optimal transferability metric would be resource-efficient, such that it is capable of accurately predicting the final performance of the model whilst minimising the amount of processing required to compute it. To this end, several works (Van Asch and Daelemans, 2010; Ruder and Plank, 2017; Ramesh Kashyap et al., 2021) have focused on estimating transferability prior to fine-tuning, using statistical measures of divergence between the underlying feature spaces of model pairs. Domain divergence measures are used to produce a notion of distance between pairs of domains by comparing their representations and have seen significant usage in works which investigate the correlation between their estimations and performance change (Van Asch and Daelemans, 2010; Ramesh Kashyap et al., 2021).

Subsequent transfer learning works have also demonstrated that competitive model performance can be achieved on some target tasks even if no training samples for that task are available, an approach known as *zero-data/shot learning* (Larochelle et al., 2008). In this work, we investigate the effectiveness of domain divergence measures in estimating the performance of zero-shot classification models, wherein models further tuned on one source task are used to directly predict on the test set of a target task without any target training samples. Specifically, we leverage the information captured by these measures as features to an auxiliary learner, whose outputs are used to rank the most effective source model for transfer to a given target task. Through the analysis of 58 sentiment classification domains, we: (1) perform a correlation analysis between each independent measure and each source-target, macro-averaged F_1 -score performance output; (2) and, for each target task, we train a series of auxiliary regression models to predict their projected performance;

(3) we then convert these into rankings of source–target pairs and evaluate the capability of our learners to find the best source model for each given target domain.

2 Experiment Setup

Measures: Ramesh Kashyap et al. (2021) provide categories of divergence measures; two of which we use in our work: *Geometric* measures which calculate distances between continuous representations such as word embeddings and *Information-theoretic* measures which capture the distance between representations such as frequency-based distributions over co-occurring n-grams. We do not report higher-order measures as in the aforementioned work, but instead report *moments*-based features, which better describe the characteristics of our individual term distributions—namely the mean, variance, skewness, and kurtosis of our distributions—as features to our learner. Following prior work (Tsvetkov et al., 2016; Ruder and Plank, 2017), we further complement the above measures by making use of several metrics that capture diversity and prototypicality such as entropy-based features; in our work, these measures are used with probability distributions, and are, as such, categorised here as information-theoretic. Specifically, we use the following metrics:

- **Geometric:** Cosine distance, l_1 - (or Manhattan dist.) and l_2 -norm (or Euclidean dist.).
- **Information-theoretic:** Rényi and Jensen-Shannon divergences (Wong and You, 1985; Rényi et al., 1961), Bhattacharyya Coeff. (Bhattacharyya, 1943), Wasserstein distance (Kantorovich, 1960), Entropy and Rényi Entropy (Shannon, 1948; Rényi et al., 1961), Simpson’s Index (Simpson, 1949).
- **Moments-based:** Mean, variance, skewness, and kurtosis (σ^n where $n \in [1..4]$).

Representations: To compute the above metrics, we use two different representations from prior work by Ruder and Plank (2017), specifically 1) discrete probabilities of the most common terms across domains, using a fixed-size vocabulary V , where $|V| = 10,000$; and 2) a summation over probability-weighted term embeddings in each document, averaged to produce a single vector:

- (1) **Term Distributions (TD)** (Plank and van Noord, 2011): $t \in \mathbb{R}^{|V|}$ where t_i is the probability of the i -th word in the vocabulary V .

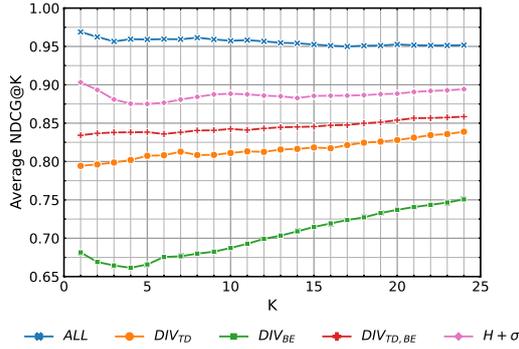
- (2) **BERT Embeddings (BE)** (Devlin et al., 2018): $\frac{1}{n} \sum_i v_{w_i} \sqrt{\frac{a}{p(w_i)}}$ where n is the number of words with embeddings in the document, v_{w_i} is the pretrained embedding of the i -th term, $p(w_i)$ its probability, and a is a smoothing factor used to discount frequent probabilities. Following guidelines by Ruder and Plank (2017), we use this representation with geometric-based measures only, as embedding vectors can be negative.

Generally, since we are using these representations in a zero-shot setting, we compute divergences between the source-task training set (D_S) and the target-task test set (D_T). Entropy and moments-based measures are not used to estimate divergence between domains but used only to compute within-domain characteristics, i.e. on individual term distributions.

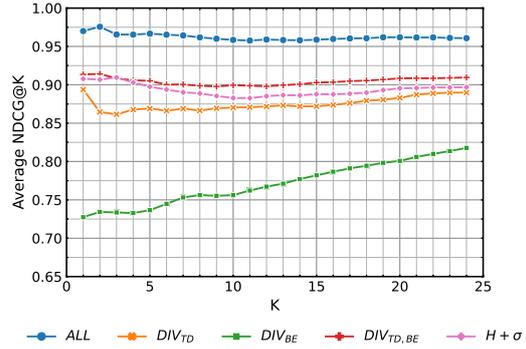
Datasets and Domains: We make use of two ratings prediction datasets with classes in the range 1-5 and, similarly to Zhang et al. (2015), reformulate the task as a binary sentiment classification task by merging the provided labels; 1-2: negative and 3-4: positive. We focus on similar, within-task (i.e. sentiment classification) datasets to (1) remove task variation as a variable, (2) and to highlight the effectiveness of using statistical measures to compute divergence between similar domains which may have very minute differences in semantics and other linguistic phenomena. The first is the *Amazon Product Reviews* dataset, using the review title and review content fields as features and divide the dataset by the product category labels. As a supplementary contribution to our work, we create the *Multi-Domain Yelp Business Reviews* dataset by extending the original *Reviews* and *Business* datasets provided by the *Yelp Dataset Challenge*, mapping top-level¹ categories of businesses to their respective reviews. After filtering out low-sample ($\leq 30,000$) domains, we have 42 and 16 domains for the Amazon and Yelp datasets, respectively.

Implementation Details: We use BERT_{base} (Devlin et al., 2018) as our base model in the experiments. With both runtime- and storage-efficiency in mind, we make use of adapter modules (Pfeiffer et al., 2020) and train each of the domains as a source task adapter, leaving the rest of BERT’s

¹We determined which categories were top-level based on an article written by Yelp



(a) Average NDCG@K for $N_S = 1000$.



(b) Average NDCG@K for $N_S = 25000$.

Figure 1: F1@K averaged across tasks vs. Total Runtime@K of source-task adapters. Higher is better. Runtime is reported in hours.

parameters frozen. More implementation and hyperparameter details can be found in Appendix A.

We divide our experiments into two separate settings by source-task sample size, $N_S \in [1000, 25000]$. We train 116 source-task adapters ($58 D_S \times 2 N_S$ settings), and evaluate a total of 6,612 source-target combinations for analysis. For our auxiliary learner, we use an XGBoost (Chen and Guestrin, 2016) regression model. We split our training and test sets by the target task and train 2,900 regression models (for each of the 58 target domains, 2 sample sizes settings, 5 feature sets, and over 5 random seeds).

3 Experiments and Results

Category	Measure	Term Distributions		BERT Embeddings	
		1K	25K	1K	25K
Geometric	Cosine Dist.	-0.3683*	-0.4801*	-0.3078*	-0.5792*
	L_1 Dist.	-0.3699*	-0.6243*	-0.0792*	-0.4045*
	L_2 Dist.	-0.3345*	-0.3551*	-0.0923*	-0.4228*
Info. Theoretic	Rényi Div.	-0.4766*	-0.4273*		
	Jensen-Shannon Div.	-0.3726*	-0.5914*		
	Wasserstein Dist.	-0.2225*	-0.3266*		
	Bhattacharyya Coeff.	0.3700*	0.5743*		
	Entropy (D_S)	0.1838*	0.2275*		
	Entropy (D_T)	-0.1603*	0.0486*		
	Rényi Entropy (D_S)	-0.1836*	-0.2284*		
	Rényi Entropy (D_T)	0.1618*	-0.0503*		
	Simpson's Index (D_S)	0.0842*	0.1359*		
	Simpson's Index (D_T)	-0.3127*	-0.1442*		
Moments Based	$\sigma^1(D_S)$	-0.1321*	-0.1792*		
	$\sigma^1(D_T)$	-0.1245*	-0.2227*		
	$\sigma^2(D_S)$	-0.1289*	-0.1523*		
	$\sigma^2(D_T)$	-0.1749*	-0.2549*		
	$\sigma^3(D_S)$	0.0106	0.0287		
	$\sigma^3(D_T)$	-0.3823*	-0.2643*		
	$\sigma^4(D_S)$	0.0006	0.0234		
	$\sigma^4(D_T)$	-0.3491*	-0.2473*		

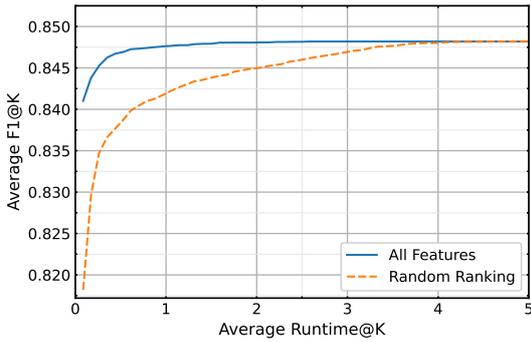
Table 1: Spearman's ρ correlations between each measure and source-target macro-averaged F_1 -score performance. Asterisk denotes measure was statistically significant ($P \leq 0.05$).

To evaluate whether the aforementioned statistical measures are predictive of task pair transferability, we perform a correlation analysis between the

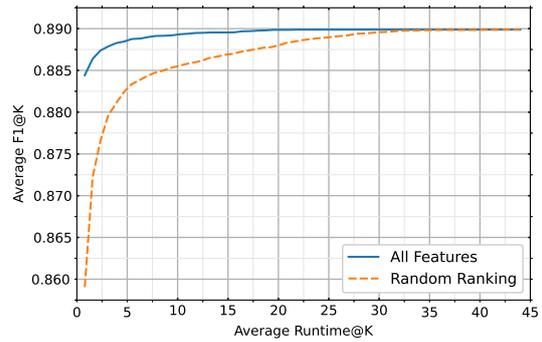
source-target pairs within each domain, where we contrast the statistical measure (which provides information about D_S , D_T , or the differences between them) and the resultant performance (measured using macro-averaged F_1) when using D_S to tune a model for application on task T . Table 1 reports Spearman's Rho (ρ) across all sample size settings for each statistical measure. Higher correlations (distance from 0) indicate increasing predictiveness of the statistical measure of transferability.

Using the interpretation of Spearman's Rho (ρ) correlation coefficients by Dancy and Reidy (2007), we make the following observations: (1) Geometric measures exhibited a moderate-to-strong correlation for Term Distributions across both sample size settings, and strong correlations at $N_S = 25000$ for BERT Embeddings; (2) Between-domain Information-theoretic measures also showed moderate-to-strong performance correlations; (3) All entropy-based measures (aside from Simpson's Index for D_T) had a weak or negligible correlation with performance; (4) Out of all of the higher-order moments of Term Distributions, only the skewness and kurtosis of D_T (σ^3 and σ^4) seemed to have a moderate relationship at $N_S = 1000$, and, generally, the moments of D_T seemed to be more correlated than that of D_S .

Overall, divergence measures with both representations seemed to be more predictive of source-target performances than with entropy or moments-based metrics. However, since it is unlikely that each measure was independently capable of predicting performance, we trained a series of regression models for each target task, combining these measures. Specifically, we train an XGBoost (Chen and Guestrin, 2016) regression model (XGBRegressor) with each of the feature sets as our inputs, over five



(a) Average F1@K vs. Runtime@K for $N_S = 1000$.



(b) Average F1@K vs. Runtime@K for $N_S = 25000$.

Figure 2: F1@K averaged across tasks vs. Total Runtime@K of source-task adapters. Higher is better. Runtime is reported in hours.

random seeds, for each of the 58 target domains and 2 sample size settings, producing 2,900 models for evaluation.

Figure 1 shows the *Average NDCG@K* values for each of these feature sets. We average the NDCG@K values across each of the 58 domains, and again over each of the 5 seeds. For both models, we achieve the best quality ranking using all of the features (*ALL*). Moreover, using divergence measures with both sets of representations (*DIV_{TD, BE}*) achieved a better ranking than using them in isolation (*DIV_{TD}* or *DIV_{BE}*) for both settings. It is also interesting to note that the feature set containing only the entropy and moments-based ($H + \sigma$) values achieve better performance than that of those estimated via divergence measures when the source sample size is significantly limited, coinciding with patterns found in our correlation analysis (Table 1); it may be the case that these features are more discriminative in cases where divergence measures are not as expressive.

Finally, we evaluate the practical, downstream application of our regression models by considering how they may be used to reduce the search time in finding appropriate source models for transfer. For this experiment, we assume the user has a particular training budget K to train task pairs for transfer. The more task combinations that are tried, the more likely the user is to find a better-performing model for a particular task. We use our regression models to determine the order of task pairs to be tried, using the best feature set from our prior experiments (See Fig. 1). We compare with a random ordering of source-task models, which we average over five random seeds to reduce variance. Figure 2 shows the results of our experiments. For $N_S = 1000$, the best macro-averaged F_1

performance score over all tasks is 0.8482 which, with a grid search over all task combinations, would require 4.7 hours of training. With our approach, we can achieve a 44% reduction in training time from 4.7 to 2.6 hours to achieve the same performance. For $N_S = 25000$, we can achieve the maximum score of 0.8899 through a grid search of all source-target combinations at a cost of 42.4 hours of training time. With our approach, we can achieve the same score with only 24.9 hours of training or a 41% reduction in training time.

In determining the overall runtime of our approach, we factor in the computational cost associated with generating the features required to train our regression models. Our feature generation process consists of three stages: (1) the generation of term distributions and embedding representations, (2) the computation of statistical measures in Table 1, (3) and the execution of regression experiments using the *ALL* feature set. A total of 232 term distributions and an equivalent number of embedding representations (58 target domains each with separate training and test sets, in two different sample size settings) were generated. The generation of both sets of representations takes 5.7 minutes at $N_S = 1000$ and 45.9 minutes at $N_S = 25000$. The time taken to compute all statistical measures across both representations is 3 minutes at $N_S = 1000$ and 6.6 minutes at $N_S = 25000$. Finally, the time taken to run the regression experiments was 5.4 minutes in total. Despite the added computational cost, our approach has resulted in a substantial reduction in end-to-end runtime, boasting a 40% reduction at $N_S = 1000$ and a 39% reduction at $N_S = 25000$, demonstrating the efficiency of our approach and the value-add of predicting which task pairs are transferable beforehand.

4 Conclusions and Future Work

In this paper, we have shown that domain divergence measures and other statistical quantities are predictive of zero-shot transferability between tasks, and that this can be used to markedly reduce time when developing effective zero-shot models. Indeed, by predicting which source-target task pairs were likely transferable pre-tuning, we were able to reduce the end-to-end time taken to find the best source-target task pairs (trained on 1,000 source-task samples) by 40%. On the other hand, while we have demonstrated the value of using these metrics in performance estimation, there are a number of further directions worth investigating, namely: (1) examine the transferability across a wider range of domain and task types; (2) investigate more complex, higher-order measures such as those outlined by Ramesh Kashyap et al. (2021); (3) and to experiment with few-shot and other limited data settings.

Limitations

The most pronounced limitation in our work is the small variance in performance scores. As can be seen in Figure 2, the difference between the lower and maximum performances is small. The difference between the minimum and maximum average performance is 0.0305 and 0.0320 for $N_S = 1000$ and $N_S = 25000$, respectively. Even at the individual, source-target model level, the standard deviation of performance scores at each source-task sample size setting is 0.0363 and 0.0311. As such, the benefits of zero-shot transfer are not as apparent between these domains as they would be where the domains are more textually distinct. Nevertheless, we believe it is notable that statistical measures of domain divergence and the other metrics were sufficiently capable of discerning between more effective source-task pairs, even when the domains were similar, illustrating the promise of this approach.

References

- Anil Bhattacharyya. 1943. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109.
- Tianqi Chen and Carlos Guestrin. 2016. [Xgboost: A scalable tree boosting system](#). *CoRR*, abs/1603.02754.
- Christine P Dancey and John Reidy. 2007. *Statistics without maths for psychology*. Pearson education.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. [On the effectiveness of adapter-based tuning for pretrained language model adaptation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online. Association for Computational Linguistics.

Leonid V Kantorovich. 1960. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422.

Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *AAAI*, 2, page 3.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. [AdapterFusion: Non-destructive task composition for transfer learning](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. [Adapterhub: A framework for adapting transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Barbara Plank and Gertjan van Noord. 2011. [Effective measures of domain similarity for parsing](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1566–1576, Portland, Oregon, USA. Association for Computational Linguistics.

Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2021. [Domain divergences: A survey and empirical analysis](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1830–1849, Online. Association for Computational Linguistics.

Alfréd Rényi et al. 1961. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, 547-561. Berkeley, California, USA.

- Sebastian Ruder and Barbara Plank. 2017. [Learning to select data for transfer learning with Bayesian optimization](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.
- Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.
- Edward H Simpson. 1949. Measurement of diversity. *nature*, 163(4148):688–688.
- Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. [Learning the curriculum with Bayesian optimization for task-specific word representation learning](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 130–139, Berlin, Germany. Association for Computational Linguistics.
- Vincent Van Asch and Walter Daelemans. 2010. [Using domain similarity for performance estimation](#). In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 31–36, Uppsala, Sweden. Association for Computational Linguistics.
- Andrew K. C. Wong and Manlai You. 1985. [Entropy and distance of random graphs with application to structural pattern recognition](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(5):599–609.
- Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. [Character-level convolutional networks for text classification](#). *CoRR*, abs/1509.01626.

A Implementation Details

Data Preparation. For each task, we sample a holdout validation set for early stopping and a test set, both of size 2,500 (10% of the maximum sample size), which remains fixed across both sets of experiments. After filtering by sample size, i.e. dropping domains with less than 30,000 samples, we had a total of 58 domains for comparison. Prior to sampling, the total number of domains are 43 and 22 for Amazon and Yelp datasets, respectively. We use the same five seeds for both data and model training.

Hyperparameters. We largely follow the recommended learning rate setting of $1e-4$ (Pfeiffer et al., 2020; He et al., 2021) for adapter training. We set the max number of epochs to 50 and an early stopping patience of 5 non-decreasing epochs. We set the maximum input length to 256 and use a batch size of 32. For the adapter configuration, we make use of the PfeifferConfig (Pfeiffer et al., 2021) with default settings.

Computing Infrastructure. We run our experiments on $3 \times$ NVIDIA® TITAN™ RTX GPUs, with 130 Tensor TFLOPs of performance, 576 tensor cores, and 24 GB of GDDR6 memory. For the CPU, We use 4.5 cores of Intel® Xeon® Gold 5222 Processor (16.5MB Cache, 3.80 GHz) and 96GB of RAM, split across three docker containers.

EDU-level Extractive Summarization with Varying Summary Lengths

Yuping Wu, Ching-Hsun Tseng, Jiayu Shang, Shengzhong Mao,
Goran Nenadic, Xiao-Jun Zeng*

Department of Computer Science, University of Manchester
{yuping.wu-2, ching-hsun.tseng, jiayu.shang,
shengzhong.mao}@postgrad.manchester.ac.uk
gnenadic, x.zeng@manchester.ac.uk

Abstract

Extractive models usually formulate text summarization as extracting fixed top- k salient sentences from the document as a summary. Few works exploited extracting finer-grained Elementary Discourse Unit (EDU) with little analysis and justification for the extractive unit selection. Further, the selection strategy of the fixed top- k salient sentences fits the summarization need poorly, as the number of salient sentences in different documents varies and therefore a common or best k does not exist in reality. To fill these gaps, this paper first conducts the comparison analysis of oracle summaries based on EDUs and sentences, which provides evidence from both theoretical and experimental perspectives to justify and quantify that EDUs make summaries with higher automatic evaluation scores than sentences. Then, considering this merit of EDUs, this paper further proposes an EDU-level extractive model with Varying summary Lengths (EDU-VL¹) and develops the corresponding learning algorithm. EDU-VL learns to encode and predict probabilities of EDUs in the document, generate multiple candidate summaries with varying lengths based on various k values, and encode and score candidate summaries, in an end-to-end training manner. Finally, EDU-VL is experimented on single and multi-document benchmark datasets and shows improved performances on ROUGE scores in comparison with state-of-the-art extractive models, and further human evaluation suggests that EDU-constituent summaries maintain good grammaticality and readability.

1 Introduction

Automatic text summarization aims at aggregating information in long document(s) into a shorter piece of text while keeping important information. Extractive summarization and abstractive summarization are two categories of it. This paper focuses

Document: (...) [The second audio,] [taken from dash cam video from inside a patrol car,] [captures a phone call between Slager and someone] [CNN believes] [is his wife.] (...)

Reference Summary: The second audio captures a phone call between Slager and someone CNN believes is his wife.

Table 1: Example to demonstrate redundant information in sentence. Content within [] indicates an EDU.

only on the extractive task which formulates summarization as identifying salient textual segments in document (Lunh, 1958). Under the supervised learning framework, this task is further formulated as a label classification task, i.e., encoding textual segments and predicting labels on the encoded vectors. Recent state-of-the-art models (Liu and Lapata, 2019; Zhong et al., 2020; Liu et al., 2021; Ruan et al., 2022) on this task tend to be Transformer-based since BERT (Devlin et al., 2019) shows significantly better performance than RNN on most natural language understanding tasks.

Most existing works extract sentences from the document and some works further (Xu and Durrett, 2019) propose post-processing steps to prune the generated summary. The only exception is the few works (Liu and Chen, 2019; Huang and Kurohashi, 2021), which extract finer-grained textual segments, i.e., discourse-level text or EDU, with little justification. The intuition is that a sentence consisting of multiple clauses is inevitable to contain less important information. As demonstrated in Table 1, partially removing a clause in the sentence is conducive to generating a summary. Certainly, such an intuitive explanation does not provide enough evidence and support to justify the use of finer-grained textual segments such as EDU to substitute sentences. Considering such a gap in existing research, the first main motivation of this paper is to propose and conduct the comparison analysis be-

*Corresponding author.

¹<https://github.com/yuping-wu/EDU-VL>

tween sentences and EDUs to disclose and justify whether using EDU is a theoretically advanced and application-advantaged extractive unit.

When selecting textual segments, the top- k strategy with k fixed for all documents is dominant in deciding the length of the generated summary. Some works (Zhong et al., 2020; Chen et al., 2021) manage to output summaries with different lengths, i.e., various numbers of extracted segments, via formulating the problem as deriving a subset of sentences from the combination of top- k sentences. Due to the foreseeing explosion of the combination of sentences to form subsets, these approaches are limited to generating summaries with relatively small values of k . To overcome such a weakness, the second main motivation of this paper is to propose and develop an approach allowing varying lengths for extractive summarization without explicit limitation on the maximum value of k , i.e., the maximum length.

Following the above motivations, the comparison analysis between EDUs and sentences ascertains that EDU is a better text unit for the extractive task because EDU-level summaries achieve higher automatic evaluation scores than sentence-level summaries. This conclusion is justified from two perspectives. Theoretically, a formal theorem about this conclusion could be derived from the property that EDU is essentially part of a sentence. Experimentally, results of comprehensive analysis about oracle summaries of five datasets further quantify this conclusion, i.e., how much the ROUGE scores of EDU-level oracle summary are higher than sentence-level oracle summary.

Based on the aforementioned conclusion and foundation, this paper further proposes and develops an EDU-level extractive model and algorithm, which generates summaries with varying lengths, i.e., EDU-VL. We extend Transformer-based pre-trained language model with an extra classification layer to encode EDUs in a document and predict the corresponding probabilities. Multiple k values are provided to the model to generate a set of candidate summaries under the flexible top- k strategy for the document. Multiple Transformer encoder layers encode the full document and candidate summaries individually. Finally, a similarity score with the encoded document is calculated for each candidate summary and the one with the highest score is the final output of EDU-VL.

Experiments are conducted on five benchmark

datasets from different domains and with various writing styles. The experimental results suggest that EDU-VL achieves better performance than all state-of-the-art extractive baselines on single-document summarization datasets CNN/DailyMail, XSum, Reddit, and WikiHow, in terms of three ROUGE metrics. With direct comparison to the multi-document model, EDU-VL still achieves comparable performance on the multi-document summarization dataset Multi-News. Human evaluation is further carried for the summaries generated by EDU-VL to assess the syntax structure of EDU-constituent summaries. The results provide evidence for the good grammaticality and readability of EDU-constituent summaries and therefore justify the applicability.

The contributions of this paper are threefold:

- 1) We justify and quantify that EDU-level achieves higher automatic evaluation scores than sentence-level oracle summary from both theoretical and experimental perspectives, indicating that setting EDU as the extractive text unit is exploitable and superior in applications.
- 2) We propose a varying summary lengths-enabled extractive model with EDU-level text unit. Such a model and its learning algorithm encodes EDUs in a document and outputs a summary with varying length by making k in the top- k extraction strategy varying.
- 3) Our proposed model achieves superior performance on four single-document summarization datasets on three ROUGE metrics. Human evaluations show that the generated EDU-constituent summaries maintain good grammaticality and readability.

2 Related Work

2.1 Neural Extractive Summarization

The extractive text summarization task aims at extracting salient textual segments from the original document(s) as a summary. A tendency observed among extractive neural models is that the architecture changes from RNN (Nallapati et al., 2017; Xu and Durrett, 2019) to Transformer-based models, e.g., BERT (Zhang et al., 2019; Liu and Lapata, 2019) and Longformer (Liu et al., 2021; Ruan et al., 2022). GNN also gained extensive attention in recent years and is usually stacked after

an RNN (Wang et al., 2020; Jing et al., 2021) or Transformer-based encoder (Cui et al., 2020; Kwon et al., 2021) to supplement graph-based features. Some research works integrated neural networks with reinforcement learning (Dong et al., 2018; Gu et al., 2022) or unsupervised learning frameworks (Liang et al., 2021). In general, it can be said that taking a pre-trained Transformer-based language model as the starting point to encode textual segments in a document is currently the state-of-the-art approach among neural extractive models. Therefore, the Transformer-based models, i.e., RoBERTa (Liu et al., 2019) and BART (Lewis et al., 2020), are used as the basic building blocks in this paper.

2.2 Sub-sentential Extractive Summarization

Most previous works about the extractive task focused on generating sentence-level summaries, though some of them (Xiao et al., 2020; Cho et al., 2020; Ernst et al., 2022) utilized sub-sentential features. Early works by Marcu (1999); Alonso i Alemany and Fuentes Fort (2003); Yoshida et al. (2014); Li et al. (2016) exploited extracting discourse-level textual segments as the summary but those approaches were tested on small datasets. More recent works by Liu and Chen (2019); Xu et al. (2020); Huang and Kurohashi (2021) were evaluated on relatively larger datasets. However, whether the discourse-level textual segments are a better alternative than sentences as the extractive text unit was not justified in those works. To fill this gap, we provide justification for this research question from both theoretical and experimental perspectives in this paper.

2.3 Flexible Extractive Summarization

Extractive summarization task is usually formulated as extracting the top- k number of salient textual segments from a document. The fixed k value for all documents results in the lack of variety in the length of the generated summary. Few works (Jia et al., 2020; Zhong et al., 2020; Chen et al., 2021) managed to output summaries with varying lengths. However, either it requires extra effort for hyper-parameter searching on validation dataset to find a valid threshold, or formulating the problem as selecting a subset of top- k sentences makes the variety of lengths limited to small lengths due to the explosive nature of combination. In this paper, we propose a model with varying k values but without explicit limitation on the length or the need to do hyper-parameter searching.

3 Oracle Analysis of EDUs and Sentences

Oracle analysis refers to the analysis of oracle summary whose definition is stated in Section 3.1. We conducted oracle analysis from both theoretical and experimental perspectives to justify and quantify that discourse-level summary achieves higher scores on automatic evaluation metrics than sentence-level summary.

3.1 Theoretical Formulation

Elementary Discourse Unit (EDU), the discourse-level textual segment in this paper, refers to the terminal node in the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) tree which describes the discourse structure of a piece of text. EDUs are non-overlapping and adjacent text spans in the piece of text and a single EDU is essentially a segment of a complete sentence, i.e., the sentence itself or a clause in the sentence (Zeldes et al., 2019). Namely, a sentence can always be expressed with multiple EDUs, i.e., for the s -th sentence in a document, there is $sent_s = [edu_{s_1}, \dots, edu_{s_m}]$. Consequently, a one-way property from sentence to EDU regarding expressiveness is derived.

Expressiveness Property For any given subset of sentences in a document, i.e., $[sent_i, \dots, sent_j, \dots, sent_k]$, there is always a subset of EDUs in the document, i.e., $[edu_{i_1}, \dots, edu_{i_m}, \dots, edu_{j_1}, \dots, edu_{j_m}, \dots, edu_{k_1}, \dots, edu_{k_m}]$, having identical content.

Oracle Summary The set of salient textual segments that have greedily the highest ROUGE score(s) with the reference summary is the oracle summary for a document. It signifies the upper bound of performance that an extractive summarization model could achieve on ROUGE metrics.

Denote the sentence-level oracle summary as \mathcal{OS}_{sent} and the EDU-level oracle summary as \mathcal{OS}_{edu} . Based on the aforementioned property and definition, Theorem 1 can be derived and its detailed proof is provided below.

Theorem 1. *Given a document \mathcal{D} and its reference summary \mathcal{R} , for any derived \mathcal{OS}_{sent} , there is always an \mathcal{OS}_{edu} having $ROUGE_{F_1}(\mathcal{R}, \mathcal{OS}_{edu}) \geq ROUGE_{F_1}(\mathcal{R}, \mathcal{OS}_{sent})$.*

Proof. For ROUGE-N, let f_n be a function that generates the set of n-grams for the string s and g be a function that calculates the number of overlapping elements between two sets x and y ,

i.e.,

$$\begin{aligned} f_n(s) &= n\text{-gram}(s), \\ g(x, y) &= \text{match}(x, y). \end{aligned}$$

The recall and precision formulas of the ROUGE-N metric between the reference summary \mathcal{R} and sentence-level oracle summary \mathcal{OS}_{sent} are

$$\begin{aligned} \text{R-N}_{\text{recall}, \mathcal{OS}_{sent}} &= \frac{g(f_n(\mathcal{R}), f_n(\mathcal{OS}_{sent}))}{|f_n(\mathcal{R})|}, \\ \text{R-N}_{\text{precision}, \mathcal{OS}_{sent}} &= \frac{g(f_n(\mathcal{R}), f_n(\mathcal{OS}_{sent}))}{|f_n(\mathcal{OS}_{sent})|}. \end{aligned}$$

There is always an EDU-level summary \mathcal{S}_{edu} having $\mathcal{S}_{edu} = \mathcal{OS}_{sent}$. Let \mathcal{S}_{edu}^{sub} be the subset of EDUs in \mathcal{S}_{edu} having equivalent number of overlapping n-grams as \mathcal{S}_{edu} , i.e.,

$$\mathcal{S}_{edu}^{sub} \subseteq \mathcal{S}_{edu} = \mathcal{OS}_{sent}$$

and

$$g(f_n(\mathcal{R}), f_n(\mathcal{S}_{edu}^{sub})) = g(f_n(\mathcal{R}), f_n(\mathcal{OS}_{sent})).$$

The number of words in \mathcal{S}_{edu}^{sub} is smaller than or equal to the number of words in \mathcal{OS}_{sent} , i.e.,

$$|\mathcal{S}_{edu}^{sub}| \leq |\mathcal{OS}_{sent}|,$$

and consequently, the number of n-grams is correspondingly smaller or equal, i.e.,

$$|f_n(\mathcal{S}_{edu}^{sub})| \leq |f_n(\mathcal{OS}_{sent})|.$$

Therefore, the precision score for \mathcal{S}_{edu}^{sub} is larger than or equal to \mathcal{OS}_{sent} and their recall scores are the same, i.e.,

$$\begin{aligned} \text{R-N}_{\text{precision}, \mathcal{S}_{edu}^{sub}} &= \frac{g(f_n(\mathcal{R}), f_n(\mathcal{S}_{edu}^{sub}))}{|f_n(\mathcal{S}_{edu}^{sub})|} \\ &\geq \frac{g(f_n(\mathcal{R}), f_n(\mathcal{OS}_{sent}))}{|f_n(\mathcal{OS}_{sent})|} \\ \text{R-N}_{\text{precision}, \mathcal{OS}_{sent}} &= \frac{g(f_n(\mathcal{R}), f_n(\mathcal{OS}_{sent}))}{|f_n(\mathcal{OS}_{sent})|} \end{aligned}$$

and

$$\text{R-N}_{\text{recall}, \mathcal{S}_{edu}^{sub}} = \text{R-N}_{\text{recall}, \mathcal{OS}_{sent}}$$

Therefore, the EDU-level subset of \mathcal{OS}_{sent} , i.e., \mathcal{S}_{edu}^{sub} , is found to have higher or equal F1-scores on ROUGE-N metrics than \mathcal{OS}_{sent} , i.e.,

$$\text{R-N}_{\text{F}_1, \mathcal{S}_{edu}^{sub}} \geq \text{R-N}_{\text{F}_1, \mathcal{OS}_{sent}}$$

That is to say, it is guaranteed to have an EDU-level summary having higher or equal R-N scores than \mathcal{OS}_{sent} . By taking this \mathcal{S}_{edu}^{sub} as \mathcal{OS}_{edu} , we have $\text{R-N}_{\text{F}_1, \mathcal{OS}_{edu}} \geq \text{R-N}_{\text{F}_1, \mathcal{OS}_{sent}}$.

A similar proof process can be conducted

Text Unit	R-1	R-2	R-L
CNN/DailyMail			
Sentence	53.33	31.09	49.67
EDU	61.02	37.16	58.63
XSum			
Sentence	29.13	8.70	22.32
EDU	36.07	11.74	30.95
WikiHow			
Sentence	37.98	13.76	35.18
EDU	44.28	17.94	42.56
Reddit			
Sentence	30.58	10.95	24.57
EDU	40.62	16.01	35.95
Multi-News			
Sentence	49.65	22.20	44.99
EDU	51.35	23.99	48.70

Table 2: ROUGE F1-scores of sentence-level and EDU-level oracle summaries on training datasets.

on ROUGE-L. Therefore, for any \mathcal{OS}_{sent} , there is always an \mathcal{OS}_{edu} having $\text{ROUGE}_{\text{F}_1}(\mathcal{R}, \mathcal{OS}_{edu}) \geq \text{ROUGE}_{\text{F}_1}(\mathcal{R}, \mathcal{OS}_{sent})$. \square

3.2 Empirical Justification

Five datasets from different domains were analyzed from the experimental perspective and experimental settings are listed in Appendix A. Table 2 presents the ROUGE scores of \mathcal{OS}_{sent} and \mathcal{OS}_{edu} on training datasets. \mathcal{OS}_{edu} gains significantly higher ROUGE scores on all datasets. Larger improvements are observed on ROUGE-1 (6.3-10.04) and ROUGE-L (7.38-11.38) on the majority of datasets, and improvement on ROUGE-2 is smaller but there is still an increase.

Figure 1 shows the comparison of breakdown ROUGE scores between two text units on the CNN/DailyMail training dataset and details about other datasets could be found in Appendix B. Recall scores on all three metrics are approximately equal between the two text units, suggesting that the amount of salient information in both is equal. However, precision scores are observed with a significantly higher value on \mathcal{OS}_{edu} , suggesting the length of \mathcal{OS}_{edu} is smaller.

The experimental results quantify the potential gains that EDU-level oracle summary could achieve on five datasets and the breakdown scores indicate that EDU-level oracle summary is less redundant than sentence-level oracle summary.

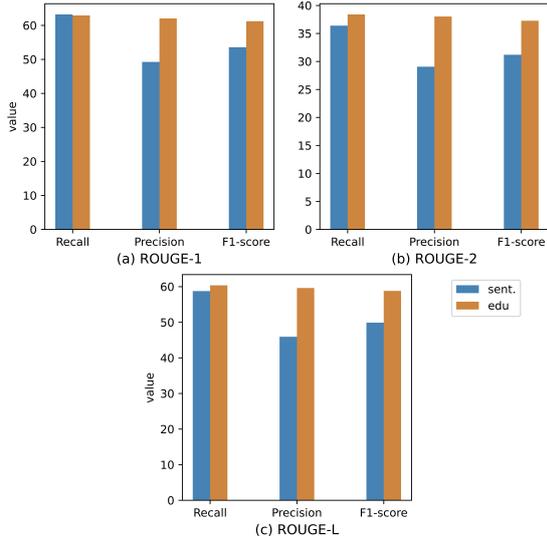


Figure 1: Breakdown ROUGE scores of sentence/EDU-level oracle summaries on CNN/DM training dataset.

4 EDU-level Extractive Model with Varying Summary Lengths

4.1 Problem Formulation

Suppose a document \mathcal{D} consists of m EDUs, i.e., $\mathcal{D} = [edu_1, \dots, edu_m]$, the i -th EDU consists of n_i words, i.e., $edu_i = [w_{i1}, \dots, w_{in_i}]$, and the reference summary wrote by human is denoted as \mathcal{R} . The set of ground truth labels for each EDU could be derived from \mathcal{R} , i.e., $L = [l_1, \dots, l_m]$, via a greedy algorithm as previous works did. Our proposed model aims to generate a summary via selecting one summary from the set of candidate summaries \mathcal{C} where $\mathcal{C} = [cand_1, \dots, cand_c]$ and $cand_j$ consists of EDUs with top- k_j probabilities that are also predicted by the proposed model.

4.2 Model

Figure 2 illustrates the architecture of our proposed model. From bottom to top, firstly, the EDU-level block generates a representation vector and probability for each EDU in a document. Secondly, the candidate summary generator aggregates EDU representation vectors to generate several candidate summaries with varying lengths by specifying different k values. Different from the previous top- k strategy where k is a fixed value, multiple k values are provided to the proposed model, allowing different numbers of EDUs being extracted to form different candidate summaries with varying lengths for the same document. Lastly, the document-level block encodes each candidate summary and selects one of the candidate summaries as the final model

output. In this way, the proposed model decides the most suitable summary length, i.e., k , for each document.

EDU-level Block Given input document $\mathcal{D} = [w_{11}, \dots, w_{mn_m}]$ where w_{ij} denotes j -th word in i -th EDU, [CLS] and [SEP] tokens are inserted into \mathcal{D} at the start and end of each EDU. We adapt the pre-trained Transformer-based language model (PLM) as the EDU encoder, e.g., RoBERTa. The hidden states of [CLS] tokens derived from the PLM are taken as EDU representations, i.e., edu^E in Equation (1). A classification layer is further applied on EDU representations to predict probabilities, i.e., \mathbf{P} in Equation (2).

$$[edu_1^E, \dots, edu_m^E] = \text{PLM}_\theta(\mathcal{D}) \quad (1)$$

$$P_i(y_i = 1) = \sigma(\mathbf{W}^c edu_i^E + \mathbf{b}^c), \quad (2)$$

where θ is the set of all trainable parameters in PLM; \mathbf{W}^c and \mathbf{b}^c are trainable parameters in classification layer, and $\sigma(\cdot)$ denotes sigmoid function.

Candidate Summary Generator Given a pre-defined extraction lengths set $\mathcal{K} = [k_1, \dots, k_c]$, the s -th candidate summary, $cand_s$, consists of EDUs whose probabilities are in top- $k_s(\mathbf{P})$, i.e., $[edu_{i_1}, \dots, edu_{i_j}, \dots, edu_{i_k}]$ where $i_j \leq m$ and $P_{i_j} \in \text{top-}k_s(\mathbf{P}), j = 1, 2, \dots, k_s$. The initial representation vector, $cand_s^C$, for $cand_s$ is the concatenation of representation vectors of EDUs in it. The initial document representation vector, \mathcal{D}^C , is aggregated from the representation vectors of all EDUs.

Document-level Block Multiple Transformer encoder layers (MTL) are stacked to encode document-level information for document \mathcal{D}^C , and all candidate summaries, e.g., $cand_s^C$, separately, and generate \mathcal{D}^D and $cand_s^D$ in Equation (3). Then cosine similarity, i.e., sim_s in Equation (4), is computed between the encoded document representation and the encoded s -th candidate summary representation. The candidate summary with the highest similarity with the document is taken as the final model-generated summary.

$$[\mathcal{D}^D, cand_s^D] = [\text{MTL}_\eta(\mathcal{D}^C), \text{MTL}_\eta(cand_s^C)] \quad (3)$$

$$sim_s = \text{cosine}(\mathcal{D}^D, cand_s^D), \quad (4)$$

where η is the set of trainable parameters in MTL.

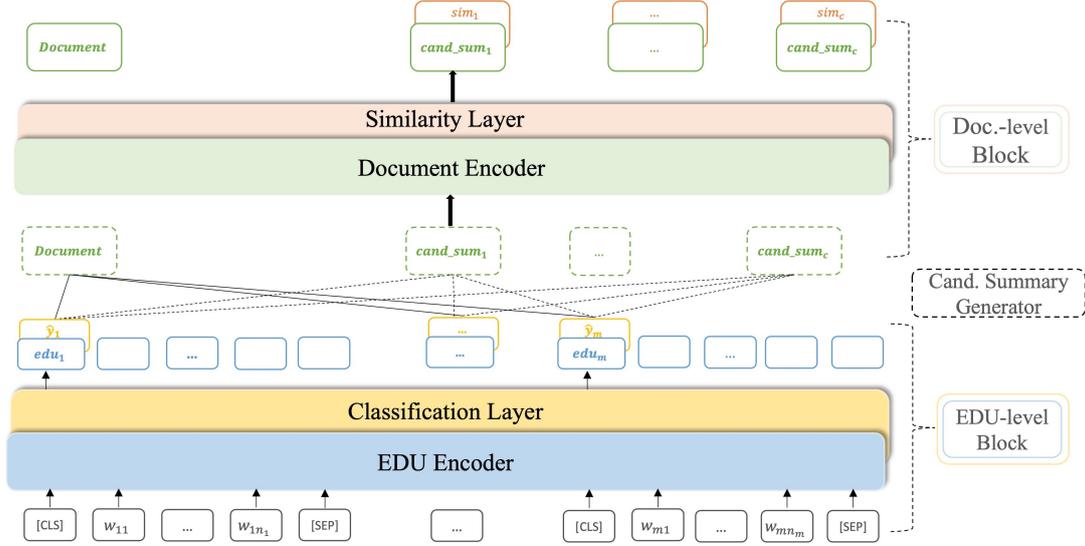


Figure 2: Model architecture. The EDU-level block encodes and predicts a probability value for each EDU in the input document; The candidate summary generator generates a set of candidate summaries based on the predicted probability values; The document-level block encodes the whole document and candidate summaries and generates similarity values between them. The final output is the candidate summary with the highest similarity score.

Training Algorithm 1 summarizes the model learning procedure. The model encodes EDUs in the document and predicts the probability for each EDU (lines 1-2), generates indices of EDUs for candidate summaries with different lengths which are derived from different k values (lines 3-4), encodes the whole document and candidate summaries and calculates similarity scores (lines 7-10), and selects the best candidate summary (line 16) in an end-to-end manner. Inspired by Zhong et al. (2020) that the candidate summary having a higher ROUGE score with the reference summary is expected to have a higher similarity score with the whole document, during training, ROUGE scores for each $(\mathcal{R}, cand_s)$ pair are calculated and used to sort the set \mathcal{C} in descending order (lines 5-6) to align with the loss function in Equation (7). Besides, to better emphasize those important EDUs, the EDU-level oracle summary, denoted as $cand_{gt}$ here, is introduced to the training process and assumed to have the highest ROUGE score (lines 12-13).

4.3 Objective Function

Binary cross entropy is calculated on the outputs of the classification layer in the EDU-level block, as in Equation (6). Contrastive learning loss is calculated on the outputs of the similarity layer in the document-level block, as in Equations (7-9). The final training loss \mathcal{L} in Equation (5) is

Algorithm 1 Model Learning Algorithm

Input: $\mathcal{D}_1^m, \mathcal{K}_1^c, L_1^m$

Output: candSumIdx

- 1: $eduRep_1^m \leftarrow PLM_\theta(\mathcal{D})$
 - 2: $P_1^m \leftarrow classification_{w,b}(eduRep_1^m)$
 - 3: **for** $i \leftarrow 1$ to c **do**
 - 4: $selIdx_i \leftarrow$ indices of top- $\mathcal{K}_i(P_1^m)$
 - 5: **if training then**
 - 6: $selIdx_1^c \leftarrow$ sort based on ROUGE scores
 - 7: $docRep \leftarrow MTL_\eta(eduRep_1^m)$
 - 8: **for** $j \leftarrow 1$ to c **do**
 - 9: $candRep_j \leftarrow MTL_\eta(eduRep_{selIdx_j})$
 - 10: $sim_j \leftarrow cosine(docRep, candRep_j)$
 - 11: **if training then**
 - 12: $gtIdx \leftarrow$ indices of 1 in L_1^m
 - 13: $sim_{gt} \leftarrow$ repeat 9-10
 - 14: $\mathcal{L} \leftarrow$ loss from $P_1^m, L_1^m, sim_1^c, sim_{gt}$
 - 15: $\theta, w, b, \eta \leftarrow$ parameters updated by \mathcal{L}
 - 16: $candSumIdx \leftarrow selIdx_{index_max(sim_1^c)}$
 - 17: **return** candSumIdx
-

calculated as a weighted summation between them.

$$\mathcal{L} = \mathcal{L}_{bce} + \rho * \mathcal{L}_{con}, \quad (5)$$

where

$$\mathcal{L}_{bce} = -\sum_{i=1}^m (l_i \log(P_i) + (1 - l_i) \log(1 - P_i)) \quad (6)$$

$$\mathcal{L}_{con} = \mathcal{L}_1 + \mathcal{L}_2, \quad (7)$$

where

$$\mathcal{L}_1 = \sum_{s=1}^c \max(0, \text{sim}_s - \text{sim}_{gt} + \gamma_1) \quad (8)$$

$$\mathcal{L}_2 = \sum_{i < j}^c \max(0, \text{sim}_j - \text{sim}_i + (j - i) * \gamma_2) \quad (9)$$

5 Experiments

5.1 Datasets

CNN/DailyMail (Hermann et al., 2015) is the most commonly used news dataset for the extractive task with human-written highlights as reference summary. The non-anonymized version was used in our experiments. **XSum** (Narayan et al., 2018) is another news dataset with the first introductory sentence in the article as the reference summary. **Reddit** (Kim et al., 2019) is a dataset crawled from the social media forum with the content in the section TL;DR as the reference summary. Experiments were conducted on the TIFU-long version. **WikiHow** (Koupaee and Wang, 2018) is a dataset crawled from the question-answering website with the first sentence in each paragraph as the reference summary. **Multi-News** (Fabbri et al., 2019) is a multi-document dataset with one summary for a cluster of documents. We follow Zhong et al.’s (2020) setting to split Reddit and Multi-News datasets and concatenate multiple documents into one single document. The detailed statistics of the five datasets in our experiments can be found in Appendix C.

5.2 Baselines

Various extractive models are selected as baselines. **HETFORMER** (Liu et al., 2021) modifies Longformer with longer input lengths to implement multi-granularity attention and selects sentences. Among models generating summaries with varying lengths, **MATCHSUM** (Zhong et al., 2020) selects among a set of candidate summaries derived from a trained sentence-level extractive model; **HAHSUM** (Jia et al., 2020) transforms a document into a heterogeneous hierarchical graph and flexibly selects sentences based on a threshold. Among models with sub-sentential segments as input, the **Proposed** model by Huang and Kurohashi (2021) is another Longformer-based model but extracts EDUs based on the constructed heterogeneous graph; **DISCOBERT** (Xu et al., 2020) and **D-SUM** (Liu and Chen, 2019) are models extracting discourse-level textual segments but they differ in whether integrating GNN into the model.

SGSUM (Chen et al., 2021) is a multi-document model by encoding all documents within one cluster individually and selecting the best sub-graph. **FAR** (Liang et al., 2021) is an unsupervised ranking model considering facet-specific information.

5.3 Experimental Setting

EDU segmentation of sentences in the document is conducted by NeuralEDUSegmentation² (Wang et al., 2018). To facilitate the training process, the calculation of ROUGE scores is avoided by pre-selecting the set of candidate summaries based on the predicted probabilities by the fine-tuned RoBERTa on the extractive task for each dataset. The pre-trained “roberta-base” or “bart-base” is adapted as the EDU encoder and enlarged to handle the first 768 BPEs of each document. The number of Transformer encoder layers is 4 by default. Following Liu and Lapata (2019), a similar greedy algorithm is applied to generate ground truth labels for EDUs (also for oracle summaries in Section 3.2) and the pseudo-code is in Appendix D. The trigram strategy is applied when forming the final EDU-constituent summary during validating and testing.

We follow Zhong et al.’s (2020) setting to set up $\gamma_1 = 0$ and $\gamma_2 = 0.01$. ρ is set as 100 based on our observation during training. Adam optimizer is used. The batch size is 5 to fit the GPU memory limit during training and 60 during validating or testing. Every 6k steps are defined as one epoch; the training process could take up to 100 epochs and early stopping is activated with patience as 10 epochs and R-2 as the metric. Experiments are conducted on a single Nvidia-v100-16GB GPU. The F1-scores of ROUGE-1/2/L³ (Lin, 2004) are taken as the automatic evaluation metrics. More details are provided in Appendix E.

5.4 Experimental Results

CNN/DailyMail Table 3 shows the results. The top section includes F1-scores of oracle summaries and the Lead-3 method. The second section presents the F1-scores reported in the original papers of all baselines. The last section lists the F1-scores of our proposed model.

Our proposed model outperforms the unsupervised baseline, FAR, by a large margin, aligning with the observation from other supervised

²<https://github.com/PKU-TANGENT/NeuralEDUSeg>

³<https://github.com/bheinzerling/pyrouge>

Model	R-1	R-2	R-L
ORACLE (EDU)	62.50	38.67	60.16
ORACLE (sentence)	55.31	32.73	51.63
LEAD-3 (sentence)	39.96	17.39	36.27
D-SUM (Liu and Chen, 2019)	42.78	20.23	-
DISCOBERT (Xu et al., 2020)	43.77	20.85	40.67
Proposed (Huang and Kurohashi, 2021)	43.61	20.81	41.12
HAHSUM (Jia et al., 2020)	44.68	21.30	40.75
MATCHSUM (Zhong et al., 2020)	44.41	20.86	40.55
HETFORMER (Liu et al., 2021)	44.55	20.82	40.37
FAR (Liang et al., 2021)	40.83	17.85	36.91
EDU-VL _{ROBERTA}	44.80	21.66	42.56
EUD-VL _{BART}	44.70	21.63	42.46

Table 3: F1-scores on CNN/DailyMail test dataset.

Model	R-1	R-2	R-L
XSum			
ORACLE (EDU)	36.16	11.74	31.02
ORACLE (sentence)	29.11	8.66	22.29
LEAD-3 (sentence)	19.41	2.65	15.05
MATCHSUM (Zhong et al., 2020)	24.86	4.66	18.41
EDU-VL _{ROBERTA}	26.48	5.74	22.33
EDU-VL _{BART}	26.43	5.78	22.35
Reddit			
ORACLE (EDU)	44.49	18.53	38.87
ORACLE (sentence)	34.36	12.97	26.98
LEAD-3 (sentence)	18.39	3.01	14.12
MATCHSUM (Zhong et al., 2020)	25.09	6.17	20.13
EUD-VL _{ROBERTA}	27.04	6.87	22.64
EDU-VL _{BART}	27.01	7.06	22.70

Table 4: F1-score results on test dataset of XSum and Reddit. The number of Transformer encoder layers in BART version of XSum is 6 and 2 for both versions of Reddit.

baselines. Compared with discourse-level baselines, i.e., D-SUM, DISCOBERT and Proposed, our proposed model achieves an improvement of at least 1.03/0.81/1.44 on R-1/2/L. When compared against other two varying lengths-enabled models, i.e., HAHSUM and MATCHSUM, our proposed model achieves better R-1 result on a small scale (0.12) and R-2/L on a larger scale (0.8/1.81). Our proposed model also beats HETFORMER which allows longer input length by a similar scale pattern. It is observed that the RoBERTa version of our proposed model performs slightly better than the BART version. The experimental results suggest that our proposed model achieves better performance than all baselines on the R-1/2/L.

XSum and Reddit The results in Table 4 show that our proposed model outperforms the baseline model, MATCHSUM, by a large margin on all three metrics (1.57/1.12/3.94 and 1.92/0.89/2.57 on R-1/2/L for XSum and Reddit, respectively). The RoBERTa version of our model only achieves

Model	R-1	R-2	R-L
WikiHow			
ORACLE (EDU)	44.13	17.90	42.38
ORACLE (sentence)	37.89	13.80	35.13
LEAD-3 (sentence)	23.97	5.37	22.22
FAR (Liang et al., 2021)	27.54	6.17	25.46
MATCHSUM (Zhong et al., 2020)	31.85	8.98	29.58
EDU-VL _{ROBERTA}	33.94	10.31	32.55
EDU-VL _{BART}	34.01	10.45	32.66
Multi-News			
ORACLE (EDU)	51.60	24.24	48.92
ORACLE (sentence)	49.87	22.43	45.18
LEAD-3 (sentence)	28.40	8.63	24.93
HETFORMER (Liu et al., 2021)	46.21	17.49	42.43
SGSUM (Chen et al., 2021)	47.53	18.75	43.31
FAR (Liang et al., 2021)	43.48	16.87	44.00
MATCHSUM (Zhong et al., 2020)	46.20	16.51	41.89
EDU-VL _{ROBERTA}	46.82	17.05	44.36
EDU-VL _{BART}	47.56	17.64	45.05

Table 5: F1-score results on test dataset of WikiHow and Multi-News.

Model	R-1	R-2	R-L
EDU-VL _{ROBERTA}	44.80	21.66	42.56
w/o EDU	43.89	20.79	40.18
w/o VL	44.32	21.38	42.12

Table 6: Ablation analysis on test dataset of CNN/DM.

slightly better result on R-1 than the BART version.

WikiHow and Multi-News As shown in Table 5, our proposed model achieves significantly better performance on WikiHow dataset, beating both MATCHSUM and FAR by at least 2.16/1.47/3.08 on R-1/2/L. For the Multi-News dataset, our proposed model outperforms HETFORMER, MATCHSUM and FAR. It is noteworthy that SGSUM is initially designed to incorporate multiple documents, meaning that its input document is more complete than ours. Though our proposed model underperforms SGSUM on R-2, our proposed model achieves comparable result on R-1 and better result on R-L. The BART version of our proposed model outperforms the RoBERTa version on all three metrics on both datasets. To sum up, our proposed model performs better on WikiHow dataset and comparably on Multi-News dataset when compared against the corresponding state-of-the-art baselines.

5.5 Analysis

Ablation Analysis We further conduct ablation analysis by removing specific characteristics in our model and the result is presented in Table 6. Both letting the model extract sentences under the same

architecture and removing the document-block to disable the varying lengths characteristic reduce model performance on all three metrics. A larger decrease is observed in the sentence-level model.

Human Evaluation We randomly sample 50 summaries generated by our model from the CNN/DailyMail test dataset and conduct detailed qualitative analysis. For each summary, we combine EDUs from the same sentence together as one textual segment. Then referring to the dependency tree of the corresponding sentence, we evaluate the syntactical completeness of the extracted textual segment. Out of 221 extracted textual segments in all 50 summaries, 68% are syntactically complete and 32% are not. It is noteworthy that about half of those incomplete ones are subordinate clauses, whose syntax structure is close to being complete. Out of these complete ones, 44.7% are the whole sentence itself because all EDUs in that sentence are extracted; 55.3% maintain complete syntax structure after dropping some EDU(s) in that sentence (as the example shown in Table 7). Therefore, it is safe to believe that even sentences split into multiple EDUs, the model is capable to maintain the syntax structure by choosing multiple EDUs in a sentence and in some cases, filtering out some redundant information without breaking the completeness of the syntax.

Generated Summary Examples Table 7 provides an example of a summary generated by our proposed model, which illustrates that the model manages to selectively drop redundant information in sentences by operating on the EDU-level while maintaining an informative and readable summary.

6 Conclusion

In this paper, we verify and quantify the argument that the EDU-level summary achieves higher automatic evaluation scores than sentence-level summary from both theoretical and experimental perspectives. We further propose an EDU-level extractive summarization model and develop its learning algorithm, which generates summaries with different lengths for different documents. The experimental results demonstrate that our model achieves superior performance on four single-document summarization datasets and comparable performance for multi-document summarization with direct comparison with the multi-document model. In the future, we will explore integrating the EDU-

Document: (...) [*Arnold Breitenbach of St. George wanted to get ‘CIB-69’ put on a license plate,*]₂₁ [the Spectrum newspaper of St. George reported.]₂₂ [*That would have commemorated both Breitenbach getting the Purple Heart in 1969 and his Combat Infantryman’s Badge,*]₃₁ [according to the newspaper.]₃₂ (...) [*The Utah DMV denied his request,*]₅₁ [*citing state regulations*]₅₂ [*prohibiting the use of the number 69*]₅₃ [*because of its sexual connotations*]₅₄ (...)

Reference Summary: Arnold Breitenbach of St. George, Utah, wanted to get ‘CIB-69’ put on a license plate. That would have commemorated both Breitenbach getting the Purple Heart in 1969 and his Combat Infantryman’s Badge. The Utah DMV denied his request, citing state regulations prohibiting the use of the number 69 because of its sexual connotations.

Table 7: Example from model-generated summary. Content within [] represents an EDU and subscript number ij indicates it is the j -th EDU in the i -th sentence in the document. Each color represents information in a sentence in reference summary. Italic denotes content selected by *our proposed model*.

level summary generated by our model into the abstractive summarization model.

Limitations

Though EDU is defined as a clause in a sentence, current EDU segmenters are still underdeveloped due to the limited training dataset and usually split a sentence into consecutive EDUs, which breaks the syntactic structure. Occasionally some extracted EDUs from a sentence fail to recover a complete syntactic structure. Therefore, a more sophisticated segmenter could further improve the segmentation, or some post-processing treatments could be developed to address such a potential issue specifically.

Acknowledgements

We would like to acknowledge the assistance given by Research IT and the use of the Computational Shared Facility at The University of Manchester. We thank the anonymous reviewers for their helpful comments.

References

- Laura Alonso i Alemany and Maria Fuentes Fort. 2003. [Integrating cohesion and coherence for automatic summarization](#). In *Proceedings of EAACL2003*, page 1.
- Moye Chen, Wei Li, Jiachen Liu, Xinyan Xiao, Hua Wu, and Haifeng Wang. 2021. [SgSum:transforming multi-document summarization into sub-graph selection](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4063–4074, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Sangwoo Cho, Kaiqiang Song, Chen Li, Dong Yu, Hassan Foroosh, and Fei Liu. 2020. [Better highlighting: Creating sub-sentence summary highlights](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6282–6300, Online. Association for Computational Linguistics.
- Peng Cui, Le Hu, and Yuanchao Liu. 2020. [Enhancing extractive text summarization with topic-aware graph neural networks](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5360–5371, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. [Bandit-Sum: Extractive summarization as a contextual bandit](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748, Brussels, Belgium. Association for Computational Linguistics.
- Ori Ernst, Avi Caciularu, Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Jacob Goldberger, and Ido Dagan. 2022. [Proposition-Level Clustering for Multi-Document Summarization](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1765–1779, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. [Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.
- Nianlong Gu, Elliott Ash, and Richard Hahnloser. 2022. [MemSum: Extractive summarization of long documents using multi-step episodic Markov decision processes](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6507–6522, Dublin, Ireland. Association for Computational Linguistics.
- Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems*, 2015-Janua:1693–1701.
- Yin Jou Huang and Sadao Kurohashi. 2021. [Extractive summarization considering discourse and coreference relations based on heterogeneous graph](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3046–3052, Online. Association for Computational Linguistics.
- Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. [Neural extractive summarization with hierarchical attentive heterogeneous graph network](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631, Online. Association for Computational Linguistics.
- Baoyu Jing, Zeyu You, Tao Yang, Wei Fan, and Hanghang Tong. 2021. [Multiplex graph neural network for extractive text summarization](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 133–139, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 2519–2531.
- Mahnaz Koupaee and William Yang Wang. 2018. [WikiHow: A Large Scale Text Summarization Dataset](#). In *arXiv preprint arXiv:1810.09305*, pages 1–5.
- Jingun Kwon, Naoki Kobayashi, Hidetaka Kamigaito, and Manabu Okumura. 2021. [Considering nested tree structure in sentence extractive summarization with pre-trained transformer](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4039–4044, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*,

- pages 7871–7880, Online. Association for Computational Linguistics.
- Junyi Jessy Li, Kapil Thadani, and Amanda Stent. 2016. [The role of discourse units in near-extractive summarization](#). In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 137–147, Los Angeles. Association for Computational Linguistics.
- Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. [Improving unsupervised extractive summarization with facet-aware modeling](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1685–1697, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yang Liu and Mirella Lapata. 2019. [Text summarization with pretrained encoders](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.
- Ye Liu, Jianguo Zhang, Yao Wan, Congying Xia, Lifang He, and Philip Yu. 2021. [HETFORMER: Heterogeneous transformer with sparse attention for long-text extractive summarization](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 146–154, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A Robustly Optimized BERT Pretraining Approach](#). *arXiv preprint arXiv:1907.11692*.
- Zhengyuan Liu and Nancy Chen. 2019. [Exploiting discourse-level segmentation for extractive summarization](#). In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 116–121, Hong Kong, China. Association for Computational Linguistics.
- H. P. Luhn. 1958. [The Automatic Creation of Literature Abstracts](#). *IBM Journal of Research Development*, 2(2):159–165.
- William C. Mann and Sandra A. Thompson. 1988. [Rhetorical Structure Theory: Toward a functional theory of text organization](#). *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- Daniel Marcu. 1999. Discourse trees are good indicators of importance in text. In *Advances in automatic text summarization*, pages 123–136.
- Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. [SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents](#). In *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pages 3075–3081.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.
- Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. [HiStruct+: Improving extractive text summarization with hierarchical structure information](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1292–1308, Dublin, Ireland. Association for Computational Linguistics.
- Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. [Heterogeneous graph neural networks for extractive document summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.
- Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. [Toward fast and accurate neural discourse segmentation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 962–967, Brussels, Belgium. Association for Computational Linguistics.
- Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2020. [Do we really need that many parameters in transformer for extractive summarization? discourse can help !](#) In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 124–134, Online. Association for Computational Linguistics.
- Jiacheng Xu and Greg Durrett. 2019. [Neural extractive text summarization with syntactic compression](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3292–3303, Hong Kong, China. Association for Computational Linguistics.
- Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. [Discourse-aware neural extractive text summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.
- Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. 2014. [Dependency-based discourse parser for single-document summarization](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1834–1839, Doha, Qatar. Association for Computational Linguistics.

Amir Zeldes, Debopam Das, Erick Galani Maziero, Juliano Antonio, and Mikel Iruskieta. 2019. [The DIS-RPT 2019 shared task on elementary discourse unit segmentation and connective detection](#). In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 97–104, Minneapolis, MN. Association for Computational Linguistics.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. [HiBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. [Extractive summarization as text matching](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

A Parameters for Oracle Summaries

Table 8 presents parameters for oracle summaries.

Dataset	# Sentences	# EDUs
CNN/DM	5	8
XSum	5	8
Reddit	5	8
WikiHow	5	8
Multi-News	15	30

Table 8: Maximum number of textual segments allowed to be extracted in oracle summaries.

B Breakdown Comparison on ROUGE scores

Table 9 presents the breakdown ROUGE scores of other four datasets.

C Statistics of Datasets

Table 10 presents the statistics of the five datasets.

D Greedy Selection Algorithm

Algorithm 2 presents the pseudo-code of the algorithm of selecting salient textual segments, which is used to generate oracle summary and ground truth labels.

E Supplementary Experimental Settings and Results

Table 11 and Table 12 present detailed experimental settings and results, respectively.

Metric	Sentence		EDU	
	recall	precision	recall	precision
XSum				
R-1	40.18	25.77	40.16	36.54
R-2	11.70	7.95	12.86	12.26
R-L	30.68	19.79	34.31	31.44
WikiHow				
R-1	45.28	36.90	44.41	49.25
R-2	16.45	13.44	18.01	19.99
R-L	41.96	34.17	42.71	47.29
Reddit				
R-1	44.70	26.71	45.40	40.39
R-2	15.63	10.02	17.62	16.48
R-L	35.86	21.56	40.19	35.75
Multi-News				
R-1	45.09	58.87	42.45	68.35
R-2	19.96	26.72	19.86	31.79
R-L	40.77	53.44	40.24	64.86

Table 9: Breakdown ROUGE scores of sentence/EDU-level oracle summaries on XSum, WikiHow, Reddit, and Multi-News training datasets.

Dataset	# word	# EDU	# sent.	# EDU/sent.
CNN/DM	733.98	94.25	36.23	2.67
XSum	431.12	52.02	19.76	2.63
Reddit	443.46	65.28	23.44	3.01
WikiHow	581.15	75.72	29.42	2.58
Multi-News	503.33	58.33	18.13	3.35

Table 10: Statistics of datasets. #word, #EDU and #sent. refer to the average number of words, EDUs and sentences, respectively, of documents in the dataset. #EDU/sent. refers to the average number of EDUs per sentence.

Model Statistics		
model	#params	runtime per epoch
EDU-VL _{ROBERTA}	147M	1h 20min
EDU-VL _{BART}	161M	1h 30min
Pre-processing Setting		
dataset	#min	#max
CNN/DM	6	10
XSum	3	7
Reddit	4	8
WikiHow	6	10
Multi-News	27	31

Table 11: Supplementary information of experimental settings. #params refers to the total number of trainable parameters in the model (here both versions are calculated with 4 MTLs). #min and #max refer to the range of lengths (k values in the top- k strategy) of candidate summaries generated by the model, respectively.

Algorithm 2 Greedy Selection Algorithm

Input: Doc, Ref, k $\triangleright k$: # of selections
Output: sel_idx \triangleright selected indices

```
1:  $sel\_idx \leftarrow []$   $\triangleright$  empty list
2:  $C \leftarrow []$   $\triangleright$  candidate: empty list
3: while  $k \geq 0$  do
4:   end  $\leftarrow$  TRUE
5:   for  $i \leftarrow 0$  to  $len(Doc)$  do
6:      $tmp\_C \leftarrow C + [Doc_i]$ 
7:      $score \leftarrow ROUGE(tmp\_C, Ref)$ 
8:     if  $score$  increases then
9:        $sel\_idx \leftarrow sel\_idx + [i]$ 
10:       $C \leftarrow tmp\_C$ 
11:       $k \leftarrow k - 1$ 
12:     end  $\leftarrow$  FALSE
13:   break
14:   if end then
15:     break
16: return  $sel\_idx$ 
```

Model	R-1	R-2	R-L
CNN/DM			
EDU-VL _{ROBERTA}	45.45	22.10	43.23
EDU-VL _{BART}	45.29	22.08	41.11
XSum			
EDU-VL _{ROBERTA}	26.58	5.83	22.34
EDU-VL _{BART}	26.66	5.97	22.51
Reddit			
EDU-VL _{ROBERTA}	28.20	7.84	23.58
EDU-VL _{BART}	28.40	7.81	23.89
WikiHow			
EDU-VL _{ROBERTA}	33.90	10.19	32.53
EDU-VL _{BART}	33.95	10.31	32.59
Multi-News			
EDU-VL _{ROBERTA}	46.58	17.00	44.14
EDU-VL _{BART}	47.29	17.49	44.82

Table 12: Experimental results of ROUGE F1-scores on the corresponding validation datasets.

Chère maison or maison chère? Transformer-based Prediction of Adjective Placement in French

Eleni Metheniti ^{① ②}

Tim Van de Cruys ^③

Wissam Kerkri ^④

^① CLLE-CNRS

^④ Université Toulouse -

Jean Jaurès (UT2J)

firstname.lastname@{univ-tlse2.fr, irit.fr, kuleuven.be}

Juliette Thuilier ^①

^② IRIT

Université Toulouse -

Paul Sabatier (UT3)

Nabil Hathout ^①

^③ UK Leuven

Department of Linguistics

Leuven.AI institute

Abstract

In French, the placement of the adjective within a noun phrase is subject to variation: it can appear either before or after the noun. We conduct experiments to assess whether transformer-based language models are able to learn the adjective position in noun phrases in French—a position which depends on several linguistic factors. Prior findings have shown that transformer models are insensitive to permuted word order, but in this work, we show that finetuned models are successful at learning and selecting the correct position of the adjective. However, this success can be attributed to the process of finetuning rather than the linguistic knowledge acquired during pretraining, as evidenced by the low accuracy of experiments of classification that make use of pretrained embeddings. Comparing the finetuned models to the choices of native speakers (with a questionnaire), we notice that the models favor context and global syntactic roles, and are weaker with complex structures and fixed expressions.

1 Introduction

In French, the placement of the adjective is subject to a considerable amount of variation—a phenomenon that has been under close scrutiny among linguists. Generally speaking, adjective placement in anteposition or postposition is attributed to many intertwining, linguistic processes, rather than a few rigid grammatical rules. However, the order of the adjective can be crucial to the meaning of the noun phrase; in the titular example, *chère maison* means “dear house” but *maison chère* means “expensive house”.

Meanwhile, natural language processing researchers investigate whether language models built by transformer architectures are able to capture some of the inner workings of human language during their learning process. So far, research has shown that the high performance of such models does not imply the understanding of basic concepts

such as grammatical order because the transformer architecture is non-sequential by design.

We are exploring whether transformer-based language models are capable of perceiving the adjective’s position in a sentence with regard to its head noun, with a variety of experiments. Our goal is not to set a new state-of-the-art, but to explore if and how this information on word order is learned and used in tandem with the contextual word embedding information. While previous work has shown that transformer models are insensitive to word order (Pham et al., 2021; Gupta et al., 2021), finetuned models were successful in classifying adjective position (Sinha et al., 2021b). We also tested variations of finetuning training sizes and the use of attention masks to hide either the context of the sentence or the head noun and adjective.

For most adjectives, classifying their position is a relatively easy decision based on frequency; to observe the models’ underlying competencies in more complex cases, we carried out an error analysis and additional experiments and visualizations on the pretrained versions of the models. We also had the opportunity to conduct an experiment with native French speakers, to compare their choices in challenging cases of adjective placement to the models’ predictions.

Our findings show that finetuned models are capable of learning word order and efficiently classifying it; this knowledge is fainter in pretrained embeddings, but some layers demonstrate some specialization. Finetuning a model helps to learn these variations in adjective position and very successfully select the correct one. CamemBERT models were more successful than FlauBERT models over all experiments and captured more positional information in the finetuned adjective embedding. However, all transformers models show weaknesses (to different degrees) in complex cases of adjective/noun dependent phrases and fixed expressions.

2 Position of adjective in French noun phrases

While traditional grammar proposes that adjectives in French follow the noun, in a noun phrase, linguistic analysis supports that adjectives are mobile, i.e. can be placed in anteposition or postposition relative to the noun (Abeillé and Godard, 1999). However, most adjectives tend to appear in specific positions; adjectives that accept only anteposition, only postposition, and those that alternate position (Benzitoun, 2013). For example, ordinal adjectives in *-ième* (e.g. *troisième* ‘third’), are almost always anteposed to the noun, the adjectives *exotique* ‘exotic’, *idéal* ‘ideal’, *populaire* ‘popular’, *moderne* ‘modern’, *géant* ‘giant’, *naturel* ‘natural’ are always postposed, and the adjectives *énorme* ‘huge’, *immense* ‘immense’, *superbe* ‘superb’ alternate between the two possible position (Larsson, 1994; Benzitoun, 2014).

The preferred position of an adjective depends on its features and frequency; for example, Benzitoun (2014) claims that the adjective *prochain* ‘next’ in plural form does not occur in postposition (based on corpora statistics), but the singular does. Wilmet (1980, 1981) calculated that the most frequent adjectives in a corpus of literary works tend to precede the noun. However, chromatic adjectives (e.g. *rouge* ‘red’) which are of high frequency are always postposed to nouns when not a part of a multi-word expression. Adjectives derived from nouns and adjectives have a very strong tendency to be postposed (Forsgren, 2016; Goes, 1999). Wilmet (1981) and Forsgren (1978) support that the length of the adjective affects its position; short adjectives (e.g. *bon* ‘good’, *beau* ‘pretty’) tend to antepose, while longer adjectives and derivatives can only be postposed.

Semantic factors may also affect the position of an adjective with respect to its head word. For example, adjectives with multiple meanings may have different meanings in different positions; e.g. *propre* when anteposed refers to possession ‘own’, but when postposed means ‘clean’ (Thuilier, 2013). Benzitoun (2014) also presents the concept of adjective-noun pairs, where the meaning of the noun influences the position of the adjective. These pairs differ from fixed expressions because it is possible to create a pair with a different order (and different meaning), while fixed expressions are lexicalized and do not allow the existence of a variation with a different meaning. For example, the

lexicalized phrase *arts premiers* (where *premier* is postposed) has a very specific meaning (‘arts of the non-Western world’) compared to *premiers arts* ‘first arts’ where it used in its literal sense and is not a lexicalized phrase.

The presence of more dependents in the noun phrase also affects the position of the adjective. The presence of an adverbial modifier to the adjective may force the adjective phrase to postposition or increase the occurrence of the adjective in postposition, or at least allow more flexible positioning of the adjective phrase relative to the noun (Forsgren, 1978; Thuilier, 2013). A definitive case of postposition happens when an adjective has a multi-word modifier, e.g. a prepositional phrase (Thuilier, 2013). Postposition is also favored when there are multiple adjectives defining the noun. Thuilier (2013) also suggests that elements in the syntactic phrase are ordered by increasing length (known as *increasing* or *relative mass*). However, it may not apply to high-frequency adjectives such as *magnifique* ‘magnificent’ (Larsson, 1994).

3 Word order and transformer models

There has been extensive work on analyzing the syntactic and semantic capabilities of transformer models and their pretrained word embeddings, with positive and negative findings on the abilities of these models to capture linguistically salient word relations. In this review, we focus on word position and word order findings.

The addition of absolute word order (i.e. the sequential order of words) to the training process of contextual word embedding models has proven quite beneficial. Transformer models with bidirectional training, which captures adjacent word order, showed improvement compared to the original self-attention neural networks (Yang et al., 2019). Transformer models trained with masked language modeling, such as BERT and RoBERTa, are able to learn absolute word positions, but they also learn structural word positions (i.e. phrase position in hierarchical tree structures) and make use of them (Wang et al., 2019; Wang and Chen, 2020). Multiple experiments combine absolute and structural word positions to create better-informed and better-performing word embeddings (Wang et al., 2020; He et al., 2021; Chang et al., 2021; Wang et al., 2021).

However, experiments on already pretrained language models and shuffled word order tell a dif-

ferent story. [Pham et al. \(2021\)](#) conducted experiments on BERT-based models (BERT, RoBERTa, ALBERT) with GLUE classification tasks, and showed that tasks such as sentiment analysis were not affected by shuffled word order, except for the grammatical correctness task. [O’Connor and Andreas \(2021\)](#) conducted experiments on the effect that context variation has on transformer models’ usable information, and discovered that word shuffling has a negative effect, whether the shuffling was implemented on short or long distances among words. [Gupta et al. \(2021\)](#) conducted similar experiments with GLUE tasks and observed that model performance was lower on shuffled word orders (in methods that render a sequence ungrammatical and incomprehensible to humans) but close enough to support that models rely more on embedding information rather than sequential context. [Sinha et al. \(2021b\)](#) confirm that pretrained language models are insensitive to word order in tasks of Natural Language Inference and show that, on some occasions, classification is successful only with certain (random) word order variations of an input sequence. They also conducted experiments on finetuned models and noted finetuning’s positive influence on learning word order. Finetuning improved performance on tasks of inference and grammaticality as well (even with models pretrained with scrambled word order) ([Sinha et al., 2021a](#)). For French, [Li et al. \(2021\)](#) conducted experiments, on the transformer models’ capacity to capture long-range object-verb agreement and word order (in one of their experiments). They observed that models performed worse with scrambled inputs, and increasingly worse, for increasingly complex relations.

4 Experiment 1: Finetuning and classification of adjective position

4.1 Methodology

Given the findings from previous work, highlighting the syntactic and semantic capacities of transformer models as well as also their weakness in learning word order, we want to test whether transformer models are able to classify the position of the adjective in a sentence.

In order to provide the two possible positions that the adjective may have in a noun phrase, we provide a pair of sentences as input: the first sentence of the input has the adjective always anteposed to the noun, and the second sentence al-

ways postposed. We label the two-sentence sequences with ‘0’ if the first sentence is correct (i.e. the correct order is anteposition) and ‘1’ if the second sentence is correct (i.e. postposition)—see example in [Table 1](#). The sentences are separated by the specific end-of-sequence token of each model. With this task, we aim to observe if word order is insignificant to the models or if they are able to capture the preferred word order between two sentences with identical tokens and different word order. We finetuned the transformer models for 4 epochs based on the guidelines by [Devlin et al. \(2019\)](#) and [McCormick and Ryan \(2019\)](#) (see [Section 4.2](#) for datasets and details).

We also run the same experiment with a one-sentence input, with the original sentence without any permutations. The models were finetuned for 4 epochs as well, with the original sentence and its label of ante-/postposition. This method is less informative, as the model is not aware of the different possible positions of the adjective, and will only predict correctness.

In order to further study the contribution of different tokens in the input sequence, we also finetuned the models with blocked attention to certain tokens; we used the attention mask, which is an array that instructs the model’s self-attention mechanism to attend to specific tokens of the input sequence, by assigning 1s to the “visible” tokens and 0s to the “invisible” ones. In addition to the *default* setting of attending to all tokens, we tested a *pair* setting, in which all tokens are masked except for the adjective and its head noun, and a *context* setting, in which the adjective and noun are masked and all the other tokens are visible. Our goal is to observe whether the adjective-noun pair is significant enough to encapsulate their preferred positions or not, and whether the context contains (enough) information on preferred adjective-noun positions even without explicit information on the pair. We present a visualization of what an input sentence looks like in these settings in [Table 2](#).

4.2 Datasets

We extracted sentences with correct adjective-noun pairs from two parsed corpora: the frWaC corpus ([Baroni et al., 2009](#)) and the French corpora of Universal Dependencies 2.9 (UD; [Zeman et al., 2021](#)), in different combinations¹.

¹The list of corpora can be found at <https://universaldependencies.org/fr/>

On construit les éléments de plus haut niveau .	
↓	
SENTENCE	LABEL
On construit les éléments de plus haut niveau.	0
</s>On construit les éléments de plus niveau haut.	

Table 1: An example input of two sentences for the original sentence *On construit les éléments de plus haut niveau* ‘We build the higher level elements’. We only shift the position of the adjective-noun pair in the noun phrase, without affecting any other elements of the phrase (e.g. the dependent adjective *plus*).

MASK	TOKENS
Default	on construit les éléments de plus haut niveau
Pair	on construit les éléments de plus haut niveau
Context	on construit les éléments de plus

Table 2: Use of attention masks for the sentence: *On construit les éléments de plus haut niveau*. In this sentence, the adjective-noun pair is *haut niveau* (the adjective is before the noun). The label for all three inputs is [0]. For the double-sentence input, the same process will be followed for the second sentence of the input *On construit les éléments de plus niveau haut*.

We used all UD sentences and selected 120K relevant sentences from frWaC, with a 2/3 ratio of anteposition/postposition, which is roughly the ratio documented in the literature and the one that occurs in our corpora² –this ratio is also beneficial since anteposed adjectives are fewer but more frequent. However, we excluded the adjectives and words which were incorrectly parsed as adjectives, such as numerals, pronouns such as *autre*, *certain*, *chacun*, *quelque* which may have other linguistic functions than an adjective. In addition, we also excluded the adjectives and the nouns which were tokenized into subwords by the transformer model tokenizers, in order to create the attention mask described in Section 4.1.

The sentences of the two datasets were combined and used in various ways. In one setting, we trained the model only with frWaC, and used the UD sentences as an additional test set. In another one, we added a subset of the UD sentences to the train set and tested on the rest of UD; we also finetuned the model just with the (significantly) smaller UD dataset. When applicable, we tested both with frWaC and UD sentences. The size of the datasets is presented in Table 3.

²Measured on 1M frWaC sentences and the entire UD corpora.

Dataset	Train	Val.	frWaC test set	UD test set (entire)	UD test set (part)
frWaC	76,164	7,672	7,740	19,437	5,151
frWaC + UD	91,615	7,672	7,740	-	5,151
UD	13,905	1,546	7,740	-	5,151

Table 3: Dataset sizes for the finetuned models.

4.3 Transformer models

We used two monolingual French transformer-based models, available from the HuggingFace Python library (Wolf et al., 2020), CamemBERT (Martin et al., 2020) and FlauBERT (Le et al., 2020). CamemBERT is the pioneering monolingual French model and is built based on the RoBERTa architecture and trained on monolingual data. Experiments showed its advantage on traditional NLP tasks over multilingual transformer models. The authors also highlight the base version’s high performance with a fraction of the size of the large version. FlauBERT is a monolingual French BERT-based model trained with multiple, heterogeneous corpora and a more extensive tokenization procedure. It has been shown to (slightly) outperform CamemBERT on French benchmark tasks.

4.4 Baselines

The simplest baseline we can establish is based on frequency in our corpus: we assign each adjective a label of ante-/postposition based on its most frequent position in the training set. We also performed classification with more classical NLP methods, namely a logistic regression model on bag-of-words, implemented with `scikit-learn` (Pedregosa et al., 2011), and a CNN-based classifier, more sensitive to word order, implemented with PyTorch (Paszke et al., 2019).

4.5 Results

The results for the two-sentence input experiment can be found in Table 4 (and for the one-input in Table 7 in Appendix A). We can already observe that frequency yields a quite high accuracy, the bigger the training set is and the smaller the test set is. The CNN classifier is very successful when the training set is large enough. Therefore, it comes as no surprise that the finetuned transformer models made very few mistakes, with the overall accuracy being close to 100%. The results were consistently high, even when testing with a different dataset (frWaC and UD). However, with a much

Model	frWaC train			frWaC+UD train		UD train	
	frWaC	UD-full	UD-test	frWaC	UD-test	frWaC	UD-test
camembert-base	0.99	0.93	0.93	0.99	0.99	0.93	0.95
camembert-large	0.99	0.91	0.93	0.99	0.99	0.98	<i>0.66</i>
flaubert-small-cased	0.99	0.90	0.90	0.99	0.99	0.62	<i>0.66</i>
flaubert-base-cased	0.99	0.90	0.87	0.99	0.97	0.96	0.96
flaubert-base-uncased	0.99	0.90	0.91	0.99	0.99	0.95	0.95
flaubert-large-cased	0.99	0.93	0.88	0.99	0.99	0.91	0.87
Position frequency	0.91	0.77	0.93	0.91	0.94	0.45	0.62
Logistic Regression	<i>0.45</i>	<i>0.68</i>	<i>0.66</i>	<i>0.45</i>	0.65	0.82	0.87
CNN	0.94	0.48	0.94	0.96	0.95	0.55	0.72

Table 4: Classification results for the finetuned models and baselines, with the different training and test sets. Values in *italics* indicate that the model completely failed to classify.

Model	Attention mask: hidden context						Attention mask: hidden adj + noun											
	frWaC train			frWaC+UD train			UD train			frWaC train			frWaC+UD train			UD train		
	frWaC	UD-full	UD-test	frWaC	UD-test	frWaC	UD-test	frWaC	UD-test	frWaC	UD-full	UD-test	frWaC	UD-test	frWaC	UD-test	frWaC	UD-test
camembert-base	0.99	0.80	0.83	0.99	0.99	0.78	0.83	0.99	0.45	0.57	0.99	0.98	0.45	0.66	0.45	0.66	0.45	0.66
camembert-large	0.98	0.76	0.76	0.98	0.99	0.87	0.91	<i>0.45</i>	<i>0.66</i>	<i>0.66</i>	<i>0.45</i>	<i>0.66</i>	0.45	0.63	0.45	0.63	0.45	0.63
flaubert-small-cased	<i>0.45</i>	<i>0.68</i>	<i>0.68</i>	<i>0.45</i>	<i>0.66</i>	<i>0.45</i>	<i>0.66</i>	0.99	0.52	0.52	0.99	0.98	0.47	0.64	0.47	0.64	0.47	0.64
flaubert-base-cased	<i>0.45</i>	<i>0.68</i>	<i>0.68</i>	<i>0.45</i>	<i>0.66</i>	<i>0.45</i>	<i>0.66</i>	0.99	0.47	0.47	0.99	0.99	0.58	0.68	0.47	0.62	0.47	0.62
flaubert-base-uncased	<i>0.45</i>	<i>0.68</i>	<i>0.68</i>	<i>0.45</i>	<i>0.66</i>	<i>0.45</i>	<i>0.66</i>	0.99	0.61	0.61	0.99	0.99	0.47	0.62	0.47	0.62	0.47	0.62
flaubert-large-cased	<i>0.45</i>	<i>0.68</i>	<i>0.68</i>	<i>0.45</i>	<i>0.66</i>	<i>0.45</i>	<i>0.66</i>	0.99	0.54	0.54	0.99	0.99	0.50	0.64	0.50	0.64	0.50	0.64

Table 5: Classification results of the finetuned models with attention masks. Values in *italics* indicate that the model completely failed to classify.

smaller training set, results were slightly lower (as expected; finetuning guidelines recommend a training set of at least 100K inputs). In comparison, the accuracy of the one-sentence finetuning experiment is 11-12% lower, which is even lower than the frequency-based baseline and the CNN classifier.

The results of the experiments with attention masks are presented in Table 5 (and Table 8 in Appendix A for the one-input finetuning). In these experiments, the models’ attention mechanism had access to only certain tokens. When attention was only allowed to the adjective and noun pair, the Flaubert models were unable to classify, while the Camembert models showed equally outstanding performance with the frWaC sentences (but lower performance with the UD test sets). Meanwhile, masking the adjective and noun pair and only allowing attention to the rest of the sequence was surprisingly successful for the finetuned models with the larger training sets (except for camembert-large), reaching similar accuracies to those of the no-mask finetuned models. For the one-input finetuning experiment, we notice that, for the masked context scenario, performance rose drastically only for CamemBERT models and only in the frWaC domain, while the Flaubert models were again unsuccessful. For the masked adjective-pair scenario and for the UD domain, the performance is significantly lower.

4.6 Qualitative analysis

In most cases, the models make very few mistakes, which are not consistent among models. Moreover, the models are very confident in their choices, assigning high probabilities to all predictions (see Figure 1 in Appendix).

Focusing on the frWaC training set with the UD dataset as the test set, we notice that most of the sentences that were mislabeled are ones where the adjective could possibly be in a different position, with a different meaning than the original one (i.e. the utterance remains grammatical when the adjective-noun order is reversed). For example, the sentence *Une école a ouvert dans une ancienne église en 1950* ‘A school opened in a former church’ remains correct with *ancienne* postposed to the noun, but the meaning of the adjective becomes ‘old’. The context provided by the sentence is not sufficient to decipher the actual meaning, and native French speakers agree that both sentences are grammatical. On the other hand, mistakes in the classification of sentences such as *Les créations sensuelles, modernes et orientales se font remarquer* ‘The sensual, modern and oriental creations stand out’ uncover the models’ shallow perception of syntactic relations –these mistakes were, however, very rare. Finally, we notice a few badly-parsed and badly-formed sentences in the dataset, which were not enough to warrant a redesign, but were confusing to the models.

5 Experiment 2: Pre-existing knowledge in pretrained embeddings

The previous experiment shows that the finetuned transformer models are quite successful in classifying the adjective’s position when asked to distinguish between two possible positions. The following part of this research aims to observe whether this capability is given by the finetuning, or whether the pretrained models had already learned enough information on the adjective’s preferred position, with regard to its context.

5.1 Classification with adjective embeddings

The layers of a transformer model specialize in creating different dynamic word embeddings, which capture and interact with a word’s context in a different way than the previous layer. Therefore, the adjective embedding might contain the syntactic, contextual, and semantic information that determine its position with regard to the noun. We extracted the word embeddings for the adjective of each sentence, per layer, and we trained a simple logistic regression model –built in the same way as in Section 4.4. We used the frWaC training set and tested on the frWaC test set and on the entire UD dataset.

The results of the classification for the two test sets can be seen in Figure 2 in Appendix C. The classification results for the frWaC test set are quite low –close to being non-classifiable– except for the `flaubert_base_uncased` model, which unexpectedly reached 97% accuracy on the last layer. Results for the UD test set were more unpredictable, with a few layers of *camembert-base* reaching a very high accuracy, but the final layer having the lowest accuracy. On the other hand, the `flaubert` models had a progressively better performance, but they are not as good as their finetuned counterparts nor as the baselines.

5.2 Adjective [MASK] probabilities with Masked Language Models

Pretrained models can predict the tokens that can fill a masked position in a sequence. We use this method to retrieve the probability that the models have assigned to the adjective in the sentence, specifically in the position it was found in. We make use of the sentences of the frWaC test set. (These probabilities are presented in Figure 3 in Appendix D.) We observe that overall the models assigned higher probabilities to anteposed adjectives

being in anteposition, than to postposed adjectives in postposition; apart from the stricter linguistic constraints for anteposed adjectives, this could also be due to the fact that transformer models favor token frequency, and most of the most frequent adjectives in French are anteposed, while postposition harbors far more adjectives. Additionally, we observe that CamemBERT models give higher probabilities in the predictions of both anteposed and postposed adjectives.

When we shifted the [MASK] position from its original place to the opposite one, and asked the models to assign the adjective’s probability in the “wrong” position, the probability of the adjectives was close to zero for at least 85% of the cases, even for anteposed adjectives which are more versatile.

6 Experiment 3: Human vs Transformers judgments of adjective order

6.1 Methodology and Dataset

We had the opportunity to carry out an additional experiment on adjective word order, in which we studied how native French speakers and the finetuned transformer models dealt with challenging cases of adjective position, caused by structural or semantic idiosyncrasies. As observed in the previous experiments, the models demonstrated weaknesses in cases of adjacent adjectives that did not belong to the noun phrase, and their choices did not always align with the original sentence in cases of semantic ambiguity relating to the adjective position. These cases cannot always be coined as errors, since native speakers may also make similar choices whether intentionally (e.g. different comprehension of context) or unintentionally (e.g. haste, lack of attention).

The structure of the experiment is the same as in Experiment 1, where speakers and the finetuned models were presented with a sentence containing a noun-adjective pairing, and its variation having the target adjective in the opposite position. Regardless of the original order, each sentence pair of the two positions was presented in the order of anteposition-postposition. We created 89 prompt sentences, written by a native French speaker or extracted and modified from frWaC, and evaluated by French speakers. The full dataset can be found in Appendix F. The sentences are split into four categories based on the type of relations that the adjective has with the noun, or the context of the sentence:

1. *Presence of adjective/noun dependent*: The only categorical constraint that governs the position of the adjective in French is the presence of a dependent to the adjective, which forces the position of the adjective to postposition. However, if the dependent is to the noun, the position of the adjective is not restricted. We included sentences with the same adjectives and dependents either to the adjective or the noun.

2. *Fixed expressions*: Adjectives in fixed expressions will always have a fixed position in this specific context and meaning. Apart from sentences with fixed expressions we selected, we added sentences with the adjectives found in those expressions, but not in restrictive structures.

3. *Structural persistence*: Speakers are sensitive and tend to reuse repeating syntactic constructions (*syntactic priming*, (Branigan et al., 1995)). The presence of a noun phrase with an adjective in a certain position may influence the processing of the next noun phrase, especially if it contains the same adjective. We want to test the extent of this effect on native speakers and our models.

4. *Blocked and mobile adjectives*: In this category, we are including adjectives which are (almost) always found in postposition, and adjectives with free position depending on the meaning (*propre, ancien*). This category serves both as a control group, but could also provide unexpected results.

6.2 Questionnaire diffusion

While the finetuned models received all sentences as a test set, we divided the prompt sentences in 3 questionnaires, ensuring that there is equal distribution of the four categories in each. The participants were asked to select the sentence that sounded “most natural” to them, out of the two position variations. In order to eliminate outliers or non-native speakers of French, at the start of each questionnaire we asked for input of first language, and to confirm that they were native speakers of French (and also to acclimatize the participants with the experiment) there was a mini-tutorial with two sentence pairs which could not be mistaken by French speakers. The questionnaire was built with LimeSurvey³ and distributed to French university students and French locals. Out of the 71 participants who completed the questionnaire and were

³<https://www.limesurvey.org/>

Model	Micro avg.	Macro avg.
camembert-base	0.3326	0.1629
camembert-large	0.5801	0.4673
flaubert_small_cased	0.6014	0.3711
flaubert_base_cased	0.433	0.3446
flaubert_base_uncased	0.5192	0.3298
flaubert_large_cased	0.3688	0.3554

Table 6: Correlation between the average choice of the speakers and each model’s output. Micro-averaged is aggregating all sentences regardless of category while macro-average is category-sensitive.

not considered outliers, each version of the questionnaire had 22-25 participants, i.e. each sentence pair was evaluated by at least 22 speakers.

6.3 Quantitative and Qualitative Results

We calculated the average selection over all speakers and used this as the baseline to make judgments for our models. In Table 6 we are presenting the Pearson correlation between the speakers’ and the models’ choices, in order to see which of the models was closer to the behavior of the speakers. The model that achieved the highest micro- and macro-averaged correlation was `camembert-large`, although `flaubert_small_cased` model was slightly better at micro-averaged correlation – an interesting finding, since this model is created for debugging purposes and its results are unreliable. The `camembert-base` and `flaubert_large_cased` models showed the lowest correlations, and all models except for `camembert-large` did not show a strong positive correlation (>0.4) in the macro-averaged correlation.

We also examined the speakers’ decisions and the models’ predictions per category and performed error analysis. For the *presence of adjective/noun dependent* category, the speakers preferred longer adjectives in postposition, even when the dependent phrase was attached to the noun: for example, the speakers unanimously chose the postposed variation of the sentence *Ils vivent une différente relation sans amour*. “They lived a different relationship without love.” and so did most of the models. However, for shorter adjectives, the speakers chose anteposition when there was a noun dependent and postposition when there was an adjective dependent. The models however did not present a uniform behavior, with some models mostly preferring postposition (`camembert-large`) or anteposition (`flaubert-large-cased`), while the more successful ones made mistakes on the shorter adjectives.

In the *fixed expressions* category, the speakers naturally did not make any mistakes on the fixed expressions, and were able to differentiate between the fixed and the free position of the same adjective in different contexts. However, the models made several mistakes on very common fixed expressions, e.g. *la grasse matinée* “the morning of sleeping in”, but were not mistaken on expressions with a short adjective, e.g. *bénéfice net* “net benefit” (i.e. the short adjective was not anteposed, while its variations in non-fixed phrases are commonly anteposed). In the category of *structural persistence*, the speakers were able to make their choices for the adjective position despite being primed by a previous noun phrase with the opposite adjective position, e.g. they preferred the variation *Il lui a offert des volumineuses plantes à fleurs volumineuses*. “He offered them voluminous plants with voluminous flowers.” for the noun phrase *fleurs volumineuses*. However, all the models predicted anteposition, and this could have been affected by the adjectives being in the same wordform. Finally, in the *blocked/mobile* adjectives category, the speakers did not make any inexplicable choices, and always preferred postposition for the postposed adjectives (e.g. *chromatic*) and both positions for the mobile adjectives (despite the length). The only model which made mistakes on the postposed adjectives was `flaubert-large-cased`, while the other models made very few mistakes on mobile adjectives—decisions which are to some extent acceptable, since the meaning may be different but still grammatical.

7 Discussion

Previous work on exploring transformer models has supported that their success in NLP tasks is heavily based on their vast training data and efficient learning of frequencies. Our experiments, compared to a frequency-based uninformed baseline, show that there are more complex operations in play. Transformers were more efficient than sequential-order-learning neural networks, and were in fact able to differentiate between two sentences with identical tokens and slightly different word order. Finetuning is more efficient with a larger training corpus and different domains, but can still be successful with a smaller dataset if necessary.

When the models’ attention mechanism only has access to the context, and not to the adjective-noun pair itself, they were still quite capable of classi-

fying adjective position even without attending to it. This observation is consistent with the linguistic description that supports that adjective position is also determined by context and not solely by the noun phrase. However, the fact that CamemBERT models were extremely successful in identifying position without the use of context, while Flaubert models failed completely, is caused by the models’ different architectures and choices in the way the tokens are handled. In our more detailed experiments, we saw that CamemBERT models assign an overall higher probability to adjectives, regardless of their position, and that, at least for the UD dataset, the adjective embeddings were, in some layers, very informed on the preferred word position. This knowledge is correlated to the learned contextual word embeddings, rather than the word itself, as we observed a lack of semantic similarity in the visualization.

Regarding the models’ mistakes in the testing phase, they were either caused by low-frequency adjectives, bad parsing, or ambiguous meaning which may be grammatical and acceptable in both adjective positions. However, comparing the models to human performance showed their true strengths and weaknesses; when they are successful, the models tend to follow a more rigid syntactic structure and favor postposition, as it is the most frequent adjective position over all adjectives. They showed severe problems in recognizing some fixed expressions, and were more easily swayed than humans by being primed with the same adjective. In cases where both positions were possible, they usually preferred the more “traditional” postposition. These findings may demonstrate that the models base their decisions on adjectives more on frequency rather than the syntactic and semantic information of a particular adjective, and are impervious to factors that affect speakers’ decisions such as length, difficulty of processing with regard to cognitive load, and substantial or subtle semantic differences.

8 Conclusion

In this work, we aimed to study the capabilities of transformer-based language models in understanding word order, specifically the order of adjectives in a noun phrase in French. Our findings, for pretrained models, confirmed previous ones which claimed that these models are agnostic to word position. However, the process of finetun-

ing and classification with two variations of the sentence (one correct and one with permuted adjective order) was very successful, which proves that the models are capable of learning and becoming sensitive to word order. Concerning the use of attention masks, the CamemBERT models were very capable of classifying word order by only attending to the adjective and noun, while for the Flaubert models it was impossible. The differences between the two architectures were also reflected in our study of the pretrained word embeddings and the adjective probabilities, where we noticed that CamemBERT's adjective embeddings were better informed. The adjective embeddings themselves, for all models, seem to contain more contextual than word-specific information, which makes different iterations of an adjective differ from each other. In our experiment comparing native speakers to the models' preferences, we observed that the models showed weakness in structures with dependents, fixed expressions, and priming, and reverted to the grammatically-established postposition more than humans. Therefore, the models' understanding of the position relies both on context and on shallow syntactic roles, but is lacking semantic nuances. We also observed that the information on position is specialized in some layers –and easily learned via finetuning.

Limitations

This work has been conducted in the French language, due to the available language resources and transformer models in this high-resource, in addition to the authors' adept knowledge of the language and its linguistic properties. We decided to focus on the specific phenomenon of adjective placement because it offers the possibility to study the models' sensitivity to word order on pairs with one grammatical and one ungrammatical sentence, but also with pairs where both sentences were grammatical. The finetuning of the transformers models, especially of the large versions, was made possible with the use of a server with GPU clusters, provided by our institution.

References

Anne Abeillé and Danielle Godard. 1999. *La position de l'adjectif épithète en français: le poids des mots*. *Recherches linguistiques de Vincennes*, (28):9–32.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. *The WaCky wide web: a*

collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Christophe Benzitoun. 2013. *Adjectifs épithètes alternants en français parlé: premiers résultats*. *TIPA. Travaux interdisciplinaires sur la parole et le langage*, (29).

Christophe Benzitoun. 2014. *La place de l'adjectif épithète en français: ce que nous apprennent les corpus oraux*. In *SHS Web of Conferences*, volume 8, pages 2333–2348. EDP Sciences.

Holly P Branigan, Martin J Pickering, Simon P Liv-ersedge, Andrew J Stewart, and Thomas P Urbach. 1995. *Syntactic priming: Investigating the mental representation of language*. *Journal of Psycholinguistic Research*, 24(6):489–506.

Tyler Chang, Yifan Xu, Weijian Xu, and Zhuowen Tu. 2021. *Convolutions and self-attention: Re-interpreting relative positions in pre-trained language models*. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4322–4333, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mats Forsgren. 1978. *La place de l'adjectif épithète en français contemporain: étude quantitative et sémantique*. Ph.D. thesis, Acta Universitatis Upsaliensis.

Mats Forsgren. 2016. *La place de l'adjectif épithète*. *Encyclopédie grammaticale du français (online)*. Accessed on Jan 04, 2022.

Jan Goes. 1999. *L'adjectif: entre nom et verbe*, volume 777. De Boeck Supérieur.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. *Bert & family eat word salad: Experiments with text understanding*. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12946–12954.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. *Deberta: Decoding-enhanced bert with disentangled attention*. In *International Conference on Learning Representations*.

Björn Larsson. 1994. *La place et le sens des adjectifs épithètes de valorisation positive: Une étude de 113 adjectifs d'emploi fréquent dans la langue du tourisme et dans d'autres types de prose non-littéraire*. Lund University Press.

- Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Al-lauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. **FlauBERT: Unsupervised language model pre-training for French**. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.
- Bingzhi Li, Guillaume Wisniewski, and Benoit Crabbé. 2021. **Are Transformers a modern version of ELIZA? Observations on French object verb agreement**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4599–4610, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamel Seddah, and Benoît Sagot. 2020. **CamemBERT: a tasty French language model**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.
- Chris McCormick and Nick Ryan. 2019. **BERT Fine-Tuning Tutorial with PyTorch**. <https://mccormickml.com/2019/07/22/BERT-fine-tuning/>. Retrieved January 24, 2021.
- Joe O’Connor and Jacob Andreas. 2021. **What context features can transformer language models use?** In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 851–864, Online. Association for Computational Linguistics.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. **Scikit-learn: Machine Learning in Python**. *Journal of Machine Learning Research*, 12:2825–2830.
- Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. **Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks?** In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1145–1160.
- Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021a. **Masked language modeling and the distributional hypothesis: Order word matters pre-training for little**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021b. **UnNatural Language Inference**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online. Association for Computational Linguistics.
- Juliette Thuilier. 2013. **Syntaxe du français parlé vs. écrit: le cas de la position de l’adjectif épithète par rapport au nom**. *TIPA. Travaux interdisciplinaires sur la parole et le langage*, (29).
- Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. 2021. **On position embeddings in {bert}**. In *International Conference on Learning Representations*.
- Benyou Wang, Donghao Zhao, Christina Lioma, Qiuchi Li, Peng Zhang, and Jakob Grue Simonsen. 2020. **Encoding word order in complex embeddings**. In *International Conference on Learning Representations*.
- Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019. **Self-attention with structural position representations**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1403–1409, Hong Kong, China. Association for Computational Linguistics.
- Yu-An Wang and Yun-Nung Chen. 2020. **What Do Position Embeddings Learn? An Empirical Study of Pre-Trained Language Model Positional Encoding**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6840–6849, Online. Association for Computational Linguistics.
- Marc Wilmet. 1980. **Antéposition et postposition de l’épithète qualificative en français contemporain**. *Travaux de linguistique*, 7:179–201.
- Marc Wilmet. 1981. **La place de l’épithète qualificative en français contemporain. étude grammaticale et stylistique**. *Revue de Linguistique Romane Lyon*, (177-178):17–73.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz,

Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. *Transformers: State-of-the-art natural language processing*, pages 38–45.

Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. 2019. *Assessing the ability of self-attention networks to learn word order*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3635–3644, Florence, Italy. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielë Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, Hórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaz Erjavec, Aline Etienne, Wograinne Evelyn, Sidney Facundes, Richárd Farkas, Jannatul Ferdousi, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guino-

vart, Berta González Saavedra, Bernadeta Griciūtė, Matias Gironi, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yulistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Olájídé Ishola, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Oğuzhan Kuyrukçü, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, Lorena Martín-Rodríguez, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, Amiraeid Mojiri Foroushani, Judit Molnár, Amiraeid Moolodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhle, Juan Ignacio Navarro Horriacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayo Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Per-

rier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalinina, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahman, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksí Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Shafi Sourov, Carolyn Spadine, Rachele Sprugnoli, Steinhóór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021. [Universal dependencies 2.9](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

A Results for finetuning with one sentence input

Model	frWaC train			frWaC+UD train		UD train	
	frWaC	UD-full	UD-test	frWaC	UD-test	frWaC	UD-test
camembert-base	0.89	0.8	0.8	0.89	0.87	0.84	0.87
camembert-large	0.89	0.8	0.8	0.89	0.87	0.84	0.87
flaubert-small-cased	0.88	0.81	0.81	0.88	0.87	0.84	0.85
flaubert-base-cased	0.89	0.81	0.81	0.89	0.87	0.82	0.87
flaubert-base-uncased	0.89	0.82	0.82	0.88	0.87	0.82	0.87
flaubert-large-cased	0.89	0.81	0.81	0.89	0.87	0.83	0.87
Logistic Regression	<i>0.45</i>	<i>0.68</i>	<i>0.66</i>	<i>0.45</i>	<i>0.65</i>	<i>0.45</i>	<i>0.65</i>
CNN				0.8	0.84	0.68	0.79

Table 7: Classification results for finetuning models and baselines, with only one sentence as input, with our different training and test sets. Values in *italics* indicate that the model failed completely to classify

Model	Attention mask: hidden context							Attention mask: hidden adj + noun						
	frWaC train			frWaC+UD train		UD train		frWaC train			frWaC+UD train		UD train	
	frWaC	UD-full	UD-test	frWaC	UD-test	frWaC	UD-test	frWaC	UD-full	UD-test	frWaC	UD-test	frWaC	UD-test
camembert-base	0.99	0.99	0.99	0.8	0.8			0.79	0.77	0.77	0.79	0.89	0.67	0.82
camembert-large	0.97	0.98	0.98	0.97	0.98			<i>0.45</i>	<i>0.66</i>	<i>0.66</i>			<i>0.45</i>	<i>0.66</i>
flaubert-small-cased	<i>0.45</i>	<i>0.68</i>	<i>0.68</i>	<i>0.76</i>	<i>0.79</i>	<i>0.45</i>	<i>0.66</i>	0.76	0.75	0.75	0.76	0.82	0.59	0.74
flaubert-base-cased	<i>0.45</i>	<i>0.68</i>	<i>0.68</i>	0.8	0.8	<i>0.45</i>	<i>0.66</i>	0.8	0.69	0.69			0.7	0.86
flaubert-base-uncased	<i>0.45</i>	<i>0.68</i>	<i>0.68</i>	0.8	0.8	<i>0.45</i>	<i>0.66</i>	0.81	0.76	0.76			0.7	0.86
flaubert-large-cased	<i>0.45</i>	<i>0.68</i>	<i>0.68</i>	<i>0.45</i>	<i>0.66</i>	<i>0.45</i>	<i>0.66</i>	0.82	0.79	0.79			0.69	0.83

Table 8: Classification results of finetuning models with only one sentence as input and with attention masks. Values in *italics* indicate that the model failed completely to classify.

B Probabilities of predicted labels during classification

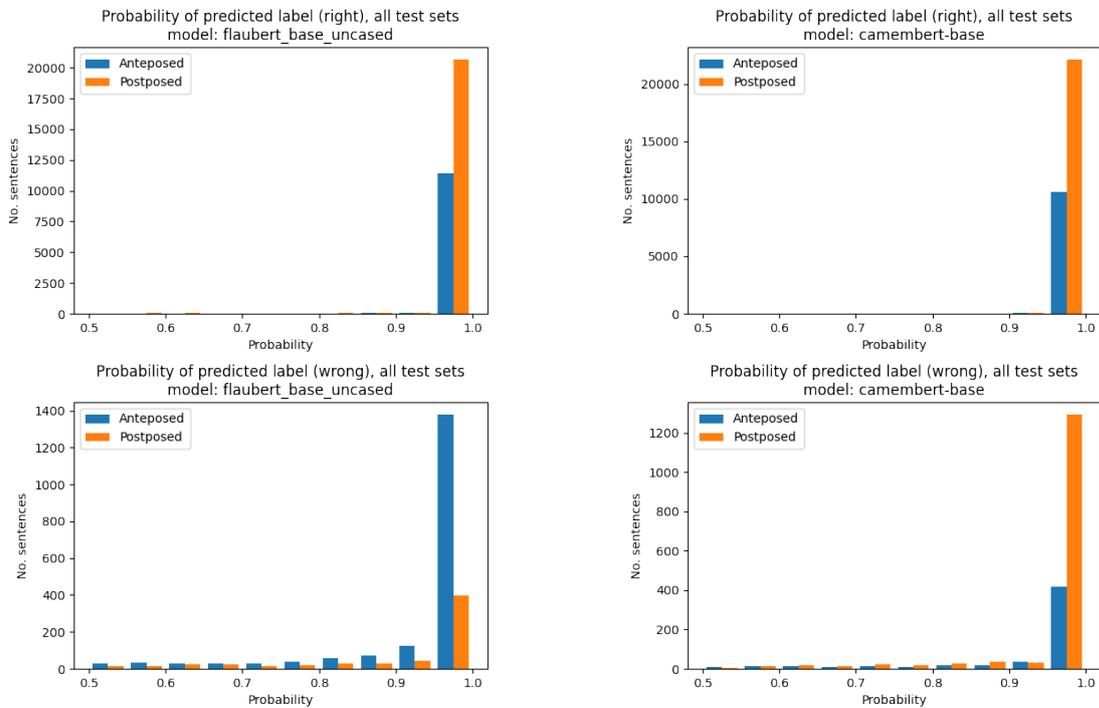


Figure 1: The probability of predicted labels, for wrong and correct predictions, from the frWaC train set and both test sets.

C Classification based on the adjective’s pretrained embedding, with logistic regression

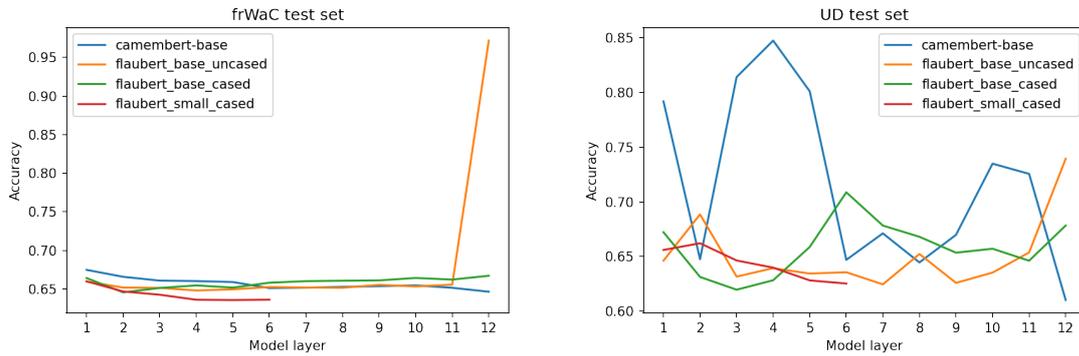


Figure 2: Logistic regression accuracy trained with layer-specific adjective embeddings, with our two large test sets.

D Adjective [MASK] probabilities with Masked Language Models

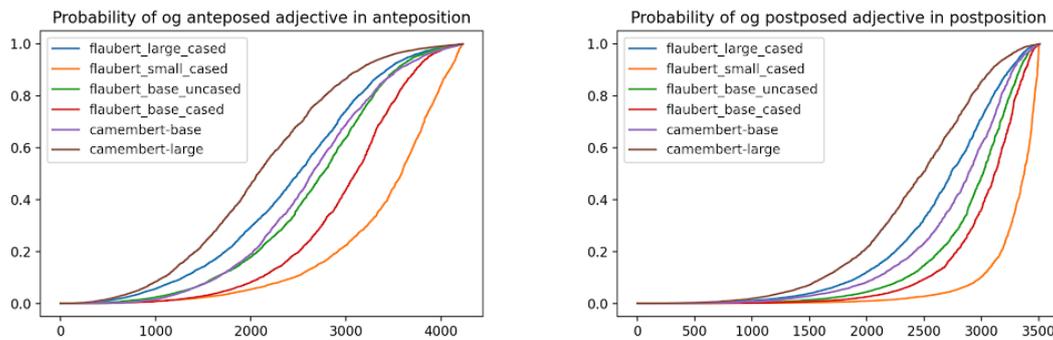


Figure 3: The assigned probability of each adjective instance, when placed in its original position which has been masked (anteponition/postponition), for each model.

E Visualizing adjective pretrained embeddings per layer

We extract layer-specific embeddings of some transformer models, and use them to visualize static embeddings by reducing their dimensions and plotting them on a 2-dimensional space, in order to observe their closest neighbors and possible clusters or patterns emerge. We selected a few frequent adjectives from the literature, either with a preferred position or without: *grand*, *petit* for always-anteponed, *naturel* for always-postposed, *ancien* for ambivalent. All the embeddings of each adjective (from the different sentences it appeared in) were used and plotted per layer.

We reduced the embeddings’ dimensional with t-distributed Stochastic Neighbor Embedding (t-SNE) from `scikit-learn` and plotted with `matplotlib`. Some of the plots are presented in Figure 4. Our intuition was that the anteponed and postposed adjectives would form clusters. However, we could not observe discernible clusters in any of the data –the closest being for some early layers, for some adjectives, and for complex word forms rather than the base ones.

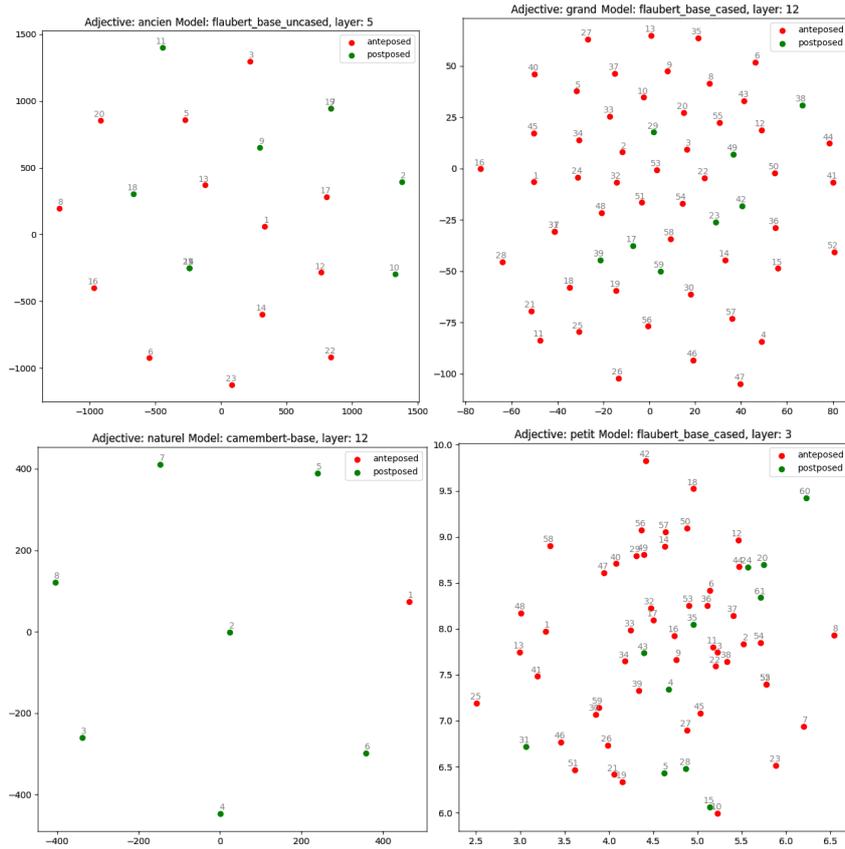


Figure 4: Embedding projections for base-form adjectives *ancien* ‘old’, *grand* ‘large’, *naturel* ‘natural’, *petit* ‘small’ –from various layers and models. The numbers correspond to the sentence id.

F Dataset for questionnaire (with English translations)

We have annotated in italics the sentences for which the French speakers preferred anteposition.

Anteposition	Postposition	Translation
<i>Ces fiers époux attendent avec impatience le jour J.</i>	<i>Ces époux fiers attendent avec impatience le jour J.</i>	<i>These proud spouses are eagerly awaiting the go time.</i>
<i>Cette fière équipe de travail se hâte de présenter son projet.</i>	<i>Cette équipe fière de travail se hâte de présenter son projet.</i>	<i>This proud work team is eager to present its project.</i>
<i>Cette longue saison de football a été intense.</i>	<i>Cette saison longue de football a été intense.</i>	<i>This long football season has been intense.</i>
<i>Elle connaît ce fier artiste depuis des années.</i>	<i>Elle connaît cet artiste fier depuis des années.</i>	<i>She has known this proud artist for years.</i>
<i>Il a écrit un long article de linguistique.</i>	<i>Il a écrit un article long de linguistique.</i>	<i>He wrote a long article on linguistics.</i>
<i>Ils ont emprunté un long chemin sans visibilité.</i>	<i>Ils ont emprunté un chemin long sans visibilité.</i>	<i>They took a long path without visibility.</i>
<i>J’ai lu un long roman comme je les aime.</i>	<i>J’ai lu un roman long comme je les aime.</i>	<i>I read a long novel as I like them.</i>
<i>Les fiers ouvriers déjeunent actuellement.</i>	<i>Les ouvriers fiers déjeunent actuellement.</i>	<i>The proud workers are currently lunching.</i>
<i>Ma tante est une fière cuisinière de renom.</i>	<i>Ma tante est une cuisinière fière de renom.</i>	<i>My aunt is a proud cook of renown.</i>
<i>Elle a participé à un long séminaire de quelques jours.</i>	<i>Elle a participé à un séminaire long de quelques jours.</i>	<i>She participated in a seminar lasting a few days.</i>
<i>Il a écrit un long article de 50 pages.</i>	<i>Il a écrit un article long de 50 pages.</i>	<i>He wrote a 50 page long article.</i>
<i>Ils ont emprunté un long chemin de plusieurs kilomètres.</i>	<i>Ils ont emprunté un chemin long de plusieurs kilomètres.</i>	<i>They took a path several kilometers long.</i>
<i>J’ai lu un long roman de plusieurs tomes.</i>	<i>J’ai lu un roman long de plusieurs tomes.</i>	<i>I read a novel several volumes long.</i>
<i>Elle annote un différent segment de 32 caractères.</i>	<i>Elle annote un segment différent de 32 caractères.</i>	<i>She annotates a different segment of 32 characters.</i>
<i>Ils vivent une différente relation sans amour.</i>	<i>Ils vivent une relation différente sans amour.</i>	<i>They live a different relationship without love.</i>
<i>L’architecte a construit une différente maison dans le sud.</i>	<i>L’architecte a construit une maison différente dans le sud.</i>	<i>The architect built a different house in the south.</i>
<i>Tu as acheté un différent cahier pour dessiner.</i>	<i>Tu as acheté un cahier différent pour dessiner.</i>	<i>You bought a different notebook to draw.</i>
<i>Vous avez couru un différent marathon toujours populaire.</i>	<i>Vous avez couru un marathon différent toujours populaire.</i>	<i>You ran a different, ever-popular marathon.</i>
<i>Ces fiers époux de leurs préparatifs attendent avec impatience.</i>	<i>Ces époux fiers de leurs préparatifs attendent avec impatience.</i>	<i>These spouses proud of their preparations are waiting impatiently.</i>
<i>Cette fière équipe de son projet se hâte de le présenter.</i>	<i>Cette équipe fière de son projet se hâte de le présenter.</i>	<i>This team, proud of its project, is eager to present it.</i>
<i>Cette longue saison de 4 mois a été intense.</i>	<i>Cette saison longue de 4 mois a été intense.</i>	<i>This 4 month long season has been intense.</i>
<i>Elle annote un différent segment du précédent.</i>	<i>Elle annote un segment différent du précédent.</i>	<i>It annotates a different segment from the previous one.</i>
<i>Elle connaît ce fier artiste de sa création.</i>	<i>Elle connaît cet artiste fier de sa création.</i>	<i>She knows this artist who is proud of his creation.</i>
<i>Ils vivent une différente relation de la suivante.</i>	<i>Ils vivent une relation différente de la suivante.</i>	<i>They live a different relationship than the following one.</i>
<i>L’architecte a construit une différente maison de celle prévue.</i>	<i>L’architecte a construit une maison différente de celle prévue.</i>	<i>The architect built a different house than planned.</i>
<i>Les fiers ouvriers de leur avancement s’accordent une pause.</i>	<i>Les ouvriers fiers de leur avancement s’accordent une pause.</i>	<i>The workers, proud of their advancement, take a break.</i>
<i>Ma tante est une fière cuisinière de ses talents.</i>	<i>Ma tante est une cuisinière fière de ses talents.</i>	<i>My aunt is a cook proud of her talent.</i>
<i>Tu as acheté un différent cahier du sien.</i>	<i>Tu as acheté un cahier différent du sien.</i>	<i>You bought a notebook different from his.</i>
<i>Vous avez couru un différent marathon de celui de Toulouse.</i>	<i>Vous avez couru un marathon différent de celui de Toulouse.</i>	<i>You ran a different marathon than that of Toulouse.</i>

Table 9: Sentences in the *Presence of adjective/noun dependent category*.

Anteposition	Postposition	Translation
<i>Dimanche, ils ont pu faire la grasse matinée.</i>	<i>Dimanche, ils ont pu faire la matinée grasse.</i>	<i>On Sunday, they were able to sleep in.</i>
<i>Elle a écrit un vibrant hommage pour sa mère décédée.</i>	<i>Elle a écrit un hommage vibrant pour sa mère décédée.</i>	<i>She wrote a moving tribute for her late mother.</i>
<i>Elle aime la grasse matinée du lundi.</i>	<i>Elle aime la matinée grasse du lundi.</i>	<i>She loves sleeping in on Mondays.</i>
<i>Il a passé une dure semaine.</i>	<i>Il a passé une semaine dure.</i>	<i>He's had a tough week.</i>
<i>Il admet son net avantage sur les autres.</i>	<i>Il admet son avantage net sur les autres.</i>	<i>He admits his clear advantage over others.</i>
<i>Il ne retient pas ses diverses leçons.</i>	<i>Il ne retient pas ses leçons diverses.</i>	<i>He does not retain his various lessons.</i>
<i>Ils ont rendu un vibrant hommage à ce digne soldat.</i>	<i>Ils ont rendu un hommage vibrant à ce digne soldat.</i>	<i>They paid a vibrant tribute to this worthy soldier.</i>
<i>J'avais des doubles objectifs précis.</i>	<i>J'avais des objectifs doubles précis.</i>	<i>I had specific dual objectives.</i>
<i>Nous effectuons diverses expériences.</i>	<i>Nous effectuons des expériences diverses.</i>	<i>We perform various experiments.</i>
<i>Elle a fait un net bénéfice ce mois-ci.</i>	<i>Elle a fait un bénéfice net ce mois-ci.</i>	<i>She made a net profit this month.</i>
<i>Depuis la mort de son hamster, il a le dur cœur.</i>	<i>Depuis la mort de son hamster, il a le cœur dur.</i>	<i>Since the death of his hamster, he has had a hard heart.</i>
<i>Depuis la mort de son hamster, il a une dure vie.</i>	<i>Depuis la mort de son hamster, il a une vie dure.</i>	<i>Since the death of his hamster, he has had a hard life.</i>
<i>Dimanche, ils ont mangé des gras plats.</i>	<i>Dimanche, ils ont mangé des plats gras.</i>	<i>On Sunday, they ate fatty dishes.</i>
<i>Elle essaiera par elle-même pour en avoir le net cœur.</i>	<i>Elle essaiera par elle-même pour en avoir le cœur net.</i>	<i>She will try on her own to find out for sure.</i>
<i>Elle n'aime pas laver la grasse boîte.</i>	<i>Elle n'aime pas laver la boîte grasse.</i>	<i>She doesn't like to wash the greasy box.</i>
<i>Il est adepte de divers faits.</i>	<i>Il est adepte de faits divers.</i>	<i>He is adept at various facts.</i>
<i>Il n'a pas accepté sa défaite, il a le dur cœur.</i>	<i>Il n'a pas accepté sa défaite, il a le cœur dur.</i>	<i>He did not accept his defeat, he has a hard heart.</i>
<i>Ils ont acheté un vibrant fauteuil pour leur salon.</i>	<i>Ils ont acheté un fauteuil vibrant pour leur salon.</i>	<i>They bought a vibrating armchair for their living room.</i>
<i>J'ai mis les doubles bouchées pour arriver à temps.</i>	<i>J'ai mis les bouchées doubles pour arriver à temps.</i>	<i>I worked hard to get there on time.</i>
<i>Nous suivons les divers faits à la télévision.</i>	<i>Nous suivons les faits divers à la télévision.</i>	<i>We follow the news on television.</i>
<i>Vous avez mis les doubles bouchées pour terminer.</i>	<i>Vous avez mis les bouchées doubles pour terminer.</i>	<i>You worked hard to finish.</i>

Table 10: Sentences in the *Fixed expressions* category.

Anteposition	Postposition	Translation
<i>A nouvelle année, nouveaux dynamismes pour cette entreprise.</i>	<i>A nouvelle année, dynamismes nouveaux pour cette entreprise.</i>	<i>A new year, new dynamics for this company.</i>
<i>Fabuleux amis, fabuleux camarades : l'ennemi n'est pas à l'intérieur !</i>	<i>Fabuleux amis, camarades fabuleux : l'ennemi n'est pas à l'intérieur !</i>	<i>Fabulous friends, fabulous comrades: the enemy is not within!</i>
<i>J'ai aimé le concept : bonne ambiance, bonne musique, les gens sont contents.</i>	<i>J'ai aimé le concept : bonne ambiance, musique bonne, les gens sont contents.</i>	<i>I liked the concept: good atmosphere, good music, people are happy.</i>
<i>Ce document vise à expliquer le déficit véritable, la véritable dette dans son ensemble.</i>	<i>Ce document vise à expliquer le déficit véritable, la dette véritable dans son ensemble.</i>	<i>This document aims to explain the real deficit, the real debt as a whole.</i>
<i>Nous avons adopté pour des stratégies communes, actions communes et positions communes.</i>	<i>Nous avons adopté pour des stratégies communes, actions communes et communes positions.</i>	<i>We have adopted for common strategies, common actions and common positions.</i>
<i>Avec la merveilleuse sélection et de merveilleux essais, ils ont trouvé les résultats qu'ils cherchaient.</i>	<i>Avec la merveilleuse sélection et des essais merveilleux, ils ont trouvé les résultats qu'ils cherchaient.</i>	<i>With the wonderful selection and wonderful testing, they found the results they were looking for.</i>
<i>Il lui a offert des volumineuses plantes à volumineuses fleurs.</i>	<i>Il lui a offert des volumineuses plantes à fleurs volumineuses.</i>	<i>He gave her bulky plants with bulky flowers.</i>
<i>Je suis d'accord avec eux : à événement exceptionnel, exceptionnel dispositif.</i>	<i>Je suis d'accord avec eux : à événement exceptionnel, dispositif exceptionnel.</i>	<i>I agree with them: for an exceptional event, an exceptional device.</i>
<i>Cette année, ils préparent un diplôme professionnel en professionnel lycée.</i>	<i>Cette année, ils préparent un diplôme professionnel en lycée professionnel.</i>	<i>This year, they are preparing a professional diploma in vocational high school.</i>
<i>Concernant la protection des données personnelles, aucune personnelle information n'est collectée.</i>	<i>Concernant la protection des données personnelles, aucune information personnelle n'est collectée.</i>	<i>Regarding the protection of personal data, no personal information is collected.</i>
<i>Elle a procédé à l'étude de quelques instruments pitoyables et pitoyables illusions.</i>	<i>Elle a procédé à l'étude de quelques instruments pitoyables et illusions pitoyables.</i>	<i>She proceeded to study some pitiful instruments and pitiful illusions.</i>
<i>Ce bâtiment n'a pas changé depuis sa construction : lumineuses couleurs, lumineux lampadaires.</i>	<i>Ce bâtiment n'a pas changé depuis sa construction : lumineuses couleurs, lampadaires lumineux.</i>	<i>This building has not changed since its construction: bright colors, bright streetlights.</i>

Table 11: Sentences in the *Structural persistence* category.

Anteposition	Postposition	Translation
<i>Elle préfère son propre pantalon à celui de sa sœur.</i>	<i>Elle préfère son pantalon propre à celui de sa sœur.</i>	<i>She prefers her own pants to her sister's.</i>
<i>Nous nous sommes rejoins autour d'un chaleureux repas.</i>	<i>Nous nous sommes rejoins autour d'un repas chaleureux.</i>	<i>We came together for a hearty meal.</i>
<i>Tu m'as fait part de ta fabuleuse idée.</i>	<i>Tu m'as fait part de ton idée fabuleuse.</i>	<i>You told me about your fabulous idea.</i>
<i>Cet ancien fer n'est plus utilisé.</i>	<i>Ce fer ancien n'est plus utilisé.</i>	<i>This old iron is no longer used.</i>
<i>C'était un fabuleux voyage que nous avons organisé.</i>	<i>C'était un voyage fabuleux que nous avons organisé.</i>	<i>It was a fabulous trip that we organized.</i>
<i>Ce chaleureux accueil m'a fait chaud au cœur.</i>	<i>Cet accueil chaleureux m'a fait chaud au cœur.</i>	<i>This warm welcome warmed my heart.</i>
<i>Ce légendaire récit me tourmente chaque jour.</i>	<i>Ce récit légendaire me tourmente chaque jour.</i>	<i>This legendary tale torments me every day.</i>
<i>Ce puéril discours lui a porté préjudice.</i>	<i>Ce discours puéril lui a porté préjudice.</i>	<i>This childish speech harmed him.</i>
<i>Cette fermière entreprise n'est plus aussi familiale que dans le temps.</i>	<i>Cette entreprise fermière n'est plus aussi familiale que dans le temps.</i>	<i>This farm business is no longer as family-run as it used to be.</i>
<i>Cette jaune chaise est très tendance.</i>	<i>Cette chaise jaune est très tendance.</i>	<i>This yellow chair is very trendy.</i>
<i>Cette puérile plaisanterie ne l'a pas fait rire.</i>	<i>Cette plaisanterie puérile ne l'a pas fait rire.</i>	<i>This childish joke did not make him laugh.</i>
<i>Elle m'a fourni la volumineuse archive.</i>	<i>Elle m'a fourni l'archive volumineuse.</i>	<i>She provided me with the voluminous archive.</i>
<i>Il m'a apporté une bleue gourde.</i>	<i>Il m'a apporté une gourde bleue.</i>	<i>He brought me a blue water bottle.</i>
<i>Il mange des roses bonbons.</i>	<i>Il mange des bonbons roses.</i>	<i>He eats pink candies.</i>
<i>Ils n'ont pas pu télécharger le volumineux fichier.</i>	<i>Ils n'ont pas pu télécharger le fichier volumineux.</i>	<i>They were unable to download the large file.</i>
<i>J'ai écrit sur une bleue feuille.</i>	<i>J'ai écrit sur une feuille bleue.</i>	<i>I wrote on a blue sheet.</i>
<i>La jaune trousse contient ses feutres.</i>	<i>La trousse jaune contient ses feutres.</i>	<i>The yellow pencil case contains her markers.</i>
<i>La pétrolière industrie ne m'attire pas du tout.</i>	<i>L'industrie pétrolière ne m'attire pas du tout.</i>	<i>The oil industry does not appeal to me at all.</i>
<i>Le ferroviaire transport est voué à s'étendre.</i>	<i>Le transport ferroviaire est voué à s'étendre.</i>	<i>Rail transport is destined to expand.</i>
<i>Le ministériel arrêté a confirmé les mesures prises.</i>	<i>L'arrêté ministériel a confirmé les mesures prises.</i>	<i>The ministerial decree confirmed the measures taken.</i>
<i>Les filles ont opté pour une mauve couverture.</i>	<i>Les filles ont opté pour une couverture mauve.</i>	<i>The girls opted for a purple blanket.</i>
<i>Leur financière situation s'aggrave de jour en jour.</i>	<i>Leur situation financière s'aggrave de jour en jour.</i>	<i>Their financial situation is getting worse day by day.</i>
<i>Ma sœur porte des mauve lunettes.</i>	<i>Ma sœur porte des lunettes mauve.</i>	<i>My sister wears purple glasses.</i>
<i>Mon bureau est décoré d'un vert panier.</i>	<i>Mon bureau est décoré d'un panier vert.</i>	<i>My office is decorated with a green basket.</i>
<i>Sa rose poubelle lui plaît énormément.</i>	<i>Sa poubelle rose lui plaît énormément.</i>	<i>His pink trash can pleases him enormously.</i>
<i>Son doudou est une verte peluche.</i>	<i>Son doudou est une peluche verte.</i>	<i>His cuddly toy is a green plush.</i>
<i>Elle a acheté un vibrant jouet pour son fils.</i>	<i>Elle a acheté un jouet vibrant pour son fils.</i>	<i>She bought a vibrating toy for her son.</i>

Table 12: Sentences in the *Blocked and mobile adjectives* category.

On the Role of Reviewer Expertise in Temporal Review Helpfulness Prediction

Mir Tafseer Nayeem
University of Alberta
mnayeem@ualberta.ca

Davood Rafiei
University of Alberta
drafie@ualberta.ca

Abstract

Helpful reviews have been essential for the success of e-commerce services, as they help customers make quick purchase decisions and benefit the merchants in their sales. While many reviews are informative, others provide little value and may contain spam, excessive appraisal, or unexpected biases. With the large volume of reviews and their uneven quality, the problem of detecting helpful reviews has drawn much attention lately. Existing methods for identifying helpful reviews primarily focus on review text and ignore the two key factors of (1) **who** post the reviews and (2) **when** the reviews are posted. Moreover, the helpfulness votes suffer from scarcity for less popular products and recently submitted (a.k.a., cold-start) reviews. To address these challenges, we introduce a dataset and develop a model that integrates the reviewer's expertise, derived from the past review history of the reviewers, and the temporal dynamics of the reviews to automatically assess review helpfulness. We conduct experiments on our dataset to demonstrate the effectiveness of incorporating these factors and report improved results compared to several well-established baselines.

1 Introduction

Many customers rely on online reviews from non-professionals, on daily basis, to decide what products to buy (e.g., *Amazon*), what hotels to stay at (e.g., *TripAdvisor*), what restaurants to eat (e.g., *Yelp*) and even what books to read (e.g., *Goodreads*). A recent survey of Bizrate Insights reward members found that approximately 98% of online shoppers research a vendor via online reviews before making a purchase decision (Kats, 2018). Since the reviews are expected to describe the actual experiences and opinions of users, they can provide a reliable source of reference, improving other customers' confidence, comfort, and the overall shopping experience (Foo et al., 2017; Gamzu

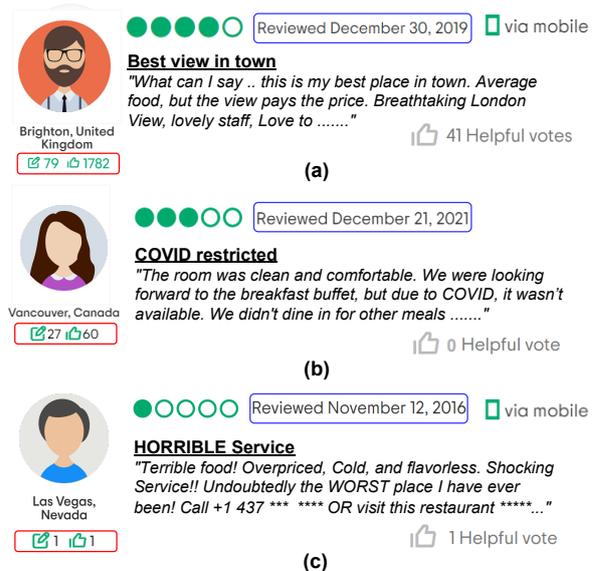


Figure 1: A snapshot of three reviews with the reviewers' history information: Review *a* has accumulated more helpful votes but is posted almost two years before Review *b*; on the other hand, Review *b* (a.k.a., cold-start review) contains time-sensitive information, describing the current conditions and Review *c* is likely a spam review. Photos of the reviewers are replaced with avatars for privacy reasons.

et al., 2021). However, despite their tremendous benefits, online reviews are often of mixed qualities. While many reviews are informative, others provide little value and may contain excessive appraisal or spam (see Figure 1-c). There are multiple factors that affect the quality of a review, including the reviewers' life experience, educational background, and the motive for writing the review (Du et al., 2021), and these factors are not usually explicit in the review text. All these pose challenges for customers who are less experienced in a subject area and need the reviews the most, simply because there is less incentive for more experienced users to use the reviews. Moreover, customers usually have limited patience for reading reviews – most customers read less than 10 reviews before mak-

ing a purchase decision about an item (Murphy, 2016). The large volume of reviews and their unpredictable quality and the limited customer patience demand better review utilization strategies to manage the information overload.

One standard method to identify more informative reviews is to ask for feedback from customers or site visitors who read them. By asking, “*Was this review helpful to you?*,” or “*Did you find this review helpful?*” at the end of each review, online platforms can crowdsource helpfulness votes from other customers. As a result, user reviews that gain the most helpful votes are shown first to the potential buyers to make the decision easier. However, the voting data suffers from scarcity (Siersdorfer et al., 2010) since only a tiny proportion of customers are willing to cast helpfulness votes. The scarcity is even more severe in reviews of less popular products and more recently submitted reviews (a.k.a., cold-start reviews) (Liu et al., 2008), despite the fact that more recent reviews may in fact contain more relevant and time-sensitive information (e.g., “*New COVID Restrictions*” or “*Dirty Pool Area*”) as shown in Figure 1-b but no helpfulness vote.

In this paper, we study the confluence of the reviewing history of reviewers and the review text for helpfulness identification. First, we observe that people who post more reviews and earn more helpful votes are more likely to be better reviewers. Second, trustworthy reviewers (e.g., Figure 1-a) are less likely to be posting fake or biased reviews, and their reviews are more likely to earn more helpful votes; otherwise, they will be ruining their reputation. Third, those who have been to more hotels or restaurants across different cities have a better basis for comparison and writing critical reviews. To the best of our knowledge, existing works only focus on review content and neglect the reviewers and their reviewing history. Integrating the review text with the reviewing history of the reviewers is the problem studied in this paper.

Our main contributions are summarized as follows:

- We introduce a new dataset with both review text and reviewer’s history, to highlight the importance of integrating the two sources for review helpfulness.
- We propose a model incorporating the reviewer’s expertise and temporal information of reviews in helpfulness prediction.

- We present a detailed case-study to interpret the model behavior and highlight potential directions to be addressed in the future.

2 Related work

More traditional approaches on review helpfulness prediction focus solely on the text of reviews, and some consider both text and images to guide the prediction. In general, the task can be addressed using a predictive model based on hand-crafted features such as structural (Susan and David, 2010; Xiong and Litman, 2014), lexical (Kim et al., 2006; Xiong and Litman, 2011), syntactic (Kim et al., 2006), emotional (Martin and Pu, 2014), semantic (Yang et al., 2015), and arguments (Liu et al., 2017) from the review text. These features may be fed into a conventional classifier such as SVM, Random Forest, or gradient boosting to identify helpful reviews. These methods heavily rely on manual feature engineering, which is labor-intensive and time-consuming.

Inspired by the remarkable progress of deep neural networks, more recent studies make use of deep neural models, which can learn both intrinsic and extrinsic features given labeled data. Chen et al. (2018) uses a text-based CNN model to automatically capture the character-level, word-level, and topic-level features for helpfulness prediction. Fan et al. (2018) uses an end-to-end multi-task neural architecture with the help of an auxiliary task, such as rating regression, to boost the performance of the review helpfulness identification. Liu et al. (2021) and Han et al. (2022) use both text and images to guide the review helpfulness prediction. Since the image field is usually optional in reviews, a large volume of reviews contain only text, for which these multimodal models would produce inconsistent results.

3 Review Helpfulness Prediction

3.1 Dataset

To the best of our knowledge, there is no human-annotated dataset that is publicly available for the task of review helpfulness prediction with the reviewers’ attributes and review date. Therefore, we build our dataset by scraping reviews from TripAdvisor¹. Out of 225,664 reviews retrieved, close to one third have no helpful votes. We filter such reviews, and this reduces the number of reviews to 161,541. Table 1 presents the summary of

¹<https://www.tripadvisor.com>

	Train	Valid	Test
Total #Samples	145,381	8,080	8,080
Avg. #Sentences	7.82	7.80	7.81
Avg. #Words	152.37	152.25	148.90

Table 1: Our dataset statistics.

our dataset with train, validation, and test splits². Following (Liu et al., 2021), we leverage a logarithmic scale ($\lfloor \log_2 n_{\text{votes}} \rfloor$) to categorize the reviews based on the number of votes received. Specifically, we map the number of votes into five intervals (i.e., $[1, 2)$, $[2, 4)$, $[4, 8)$, $[8, 16)$, $[16, \infty)$), each corresponding to a helpfulness score $Y \in \{1, 2, 3, 4, 5\}$, where the higher the score, the more helpful the review.

3.2 Proposed Model

Review Helpfulness Prediction (RHP) can be modeled as a supervised machine learning task where the input contains information about the reviews (\mathcal{R}) and the reviewers (\mathcal{U}). Let $\mathcal{R}_i = ([s_1, \dots, s_N], t_i)$ denote a review posted at time t_i with sentences s_1, \dots, s_N , and $\mathcal{U}_i = (n_i, m_i)$ denote a reviewer who posts n_i reviews and earns a total of m_i helpful votes. We formulate the review helpfulness prediction as a multi-class classification where we seek to find a model f that minimizes the loss function \mathcal{L} , i.e.

$$\min_{\theta} \mathcal{L}(f(\theta, \mathcal{R}, \mathcal{U}), Y), \quad (1)$$

where Y is the ground-truth, θ is the model parameter and the output of the model is a helpfulness class $\hat{Y} \in \{1, 2, 3, 4, 5\}$. The learning task is to find the best parameter that minimizes the above equation.

We encode the review sentences using BERT (Devlin et al., 2019; Xu et al., 2019). We concatenate the review sentences together while inserting a [CLS] token at the start and a [SEP] token at the end. If $\mathbf{h}^{[\text{CLS}]}$ denotes the embedding vector of the special [CLS] token and $\mathbf{h}^{(i)}$ denotes the embedding vector of the i -th token, we extract the last hidden state of $\mathbf{h}_i^{[\text{CLS}]}$ to represent the review sentences and apply a linear transformation to get a final contextualized representation $x_h \in \mathbb{R}^K$, where Θ is a non-linear activation function.

$$[\mathbf{h}^{[\text{CLS}]}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \dots] = \mathbf{BERT}([\text{CLS}] s_1, \dots, s_N [\text{SEP}]), \quad (2)$$

²We present our dataset construction details in Section A of the Appendix.

$$x_h = \Theta(\mathbf{MLP}(\mathbf{h}_i^{[\text{CLS}]])). \quad (3)$$

Generally, users who post more reviews and earn more helpful votes are likely to be better reviewers. Such users may have been to more hotels and restaurants across the globe and have a better basis for comparison. We define the term *reviewer expertise* as the mean number of helpful votes received per review, written as $e_s = m/n$ for a reviewer who posts m reviews and earns n overall helpfulness votes. We use a linear layer to get a weighted representation of the expertise score (h_s).

$$h_s = \mathbf{MLP}(e_s) \quad (4)$$

Previous approaches for this task fail to consider the temporal nature of the reviews. Older reviews are more likely to accumulate more helpfulness votes than newer reviews but are not necessarily the most relevant describing the current conditions (e.g., *new COVID restrictions*). One-time problems such as broken bathrooms and dirty pool area are likely to be addressed and to be less relevant. Let t_d be the relative age of a review in days, for example, as of the day the reviews are scraped. We use a linear layer to get a weighted representation of the relative review age.

$$h_t = \mathbf{MLP}(t_d). \quad (5)$$

It should be noted that both the review age and the reviewer expertise are normalized to a fixed range $[a, b]$ before being used in the linear layers in Equations 4 and 5. If \mathcal{X} denotes a set of scores (e.g., reviewers expertise score), a score $x_i \in \mathcal{X}$ is normalized into z_i as follows:

$$z_i = (b - a) \frac{x_i - \min(\mathcal{X})}{\max(\mathcal{X}) - \min(\mathcal{X})} + a \quad (6)$$

In our case, both review age and reviewer expertise are scaled into the interval $[0, 1]$.

We concatenate the textual representation (x_h), expertise representation (h_s), and temporal representation (h_t) to get a final embedding

$$o_{\text{final}} = h_s \oplus x_h \oplus h_t, \quad (7)$$

where \oplus is a concatenation operator. The final helpfulness prediction layer feeds o_{final} into a linear layer and use softmax activation to get the final predicted helpfulness class \hat{Y} .

$$\hat{Y} = \mathbf{softmax}(W_r \cdot o_{\text{final}} + b_r), \quad (8)$$

Baseline Models	Acc. (↑)	MAE (↓)	MSE (↓)
ARH	58.73	0.476	0.619
UGR + BGR	62.76	0.464	0.674
TextCNN	62.82	0.444	0.608
MTNL	62.77	0.458	0.653
BERTHelp	63.03	0.432	0.591
Our Ablations	Acc. (↑)	MAE (↓)	MSE (↓)
RHP (ours)	65.18[†]	0.393[†]	0.491[†]
- <i>w/o Expertise</i>	63.87	0.421 [†]	0.550 [†]
- <i>w/o Temporal</i>	63.40	0.437 [†]	0.592
- <i>w/o Expertise + Temporal</i>	62.92	0.446	0.617

Table 2: Performance compared to our baseline models and the result of our ablation study (↑ indicates higher values for a better performance and ↓ indicates lower values for a better performance). † reported results are statistically significant in paired t-test by taking BERTHelp (Xu et al., 2020) as a reference with the confidence of 95% (p -value < 0.05).

$$\mathcal{L} = \mathcal{L}_{CE}(\hat{Y}, Y) \quad (9)$$

where $W_r \in \mathbb{R}^{K \times K}$ and $b_r \in \mathbb{R}^K$ denote the projection parameter and a bias term respectively. We use the cross-entropy loss function \mathcal{L}_{CE} with respect to the ground truths helpfulness class (Y).

3.3 Experiments

We evaluated the performance of the proposed model³ compared to well-established baselines. We compare our system with ARH (Kim et al., 2006), UGR + BGR (Xiong and Litman, 2011), TextCNN (Chen et al., 2018), MTNL (Fan et al., 2018), and BERTHelp (Xu et al., 2020). We didn’t perform any explicit preprocessing of the review text. We discuss the baseline systems, preprocessing, and hyperparameters used for our experiments in Appendix (Section B & Section C).

3.3.1 Results

As part of a detailed evaluation of our algorithm, we report our model’s performance compared with the baselines in terms of Accuracy (**Acc.**), Mean Average Error (**MAE**), and Mean Squared Error (**MSE**). As shown in Table 2, our final model outperforms the baselines in terms of all the metrics. Our ground-truth values consist of 5 classes which correspond to five helpfulness scores $\{1, 2, 3, 4, 5\}$, where the higher the score, the more helpful the review. To gain more insights into the performance of our prediction model, we also evaluate our algorithm in terms of **MAE** and **MSE**, which assess the fine-grained differences between the ground-truth

and the predicted helpfulness scores. Our **RHP** model consistently outperforms the baselines with a good margin, which means when misclassified, our model predictions are very close to the actual helpfulness scores. We conduct detailed ablation studies to demonstrate the effects of different components of our **RHP** model by removing expertise (denoted as *w/o Expertise*) and removing temporal information (denoted as *w/o Temporal*). The ablation test results on our dataset are summarized in Table 2. We can observe that the temporal feature has the largest impact on the performance of our model, and the impact of expertise is also significant. This suggests that the reviewer’s expertise and temporal information of the reviews play a key role in review helpfulness prediction. Therefore, it is no surprise that combining all components achieves the best performance on our proposed dataset.

3.3.2 Analysis

We also present a detailed analysis to provide more supportive evidence of our arguments. To this end, we randomly selected m examples for each class of reviews considering helpfulness votes. Then, we extract Top K (where $K = 5$) n -grams from each class of reviews to identify the most relevant keywords or topics in reviews to assess what aspects are most talked about the items (e.g., hotels or restaurants).

Preprocessing Our preprocessing step includes tokenization, lemmatization, removal of stopwords, Part-Of-Speech (POS) tagging, and filtering punctuation marks. We use the NLTK⁴ to preprocess each sentence and obtain a more accurate representation of the information. Moreover, we also add ‘hotel’ and ‘restaurant’ in the stopwords list as they frequently occur in every review and are not meaningful in our context.

Extracting Candidate n -grams We remove the sentiment words and emojis using VADER⁵ (Hutto and Gilbert, 2014), a “gold-standard” sentiment lexicon especially attuned to microblog-like contexts. As the sentiment expressed in reviews are highly subjective, we are interested in extracting only the aspects or topics (e.g., *room*, *location*, *customer service* etc.) for which the opinions are

³Code, dataset, and model checkpoints: <https://github.com/tafseer-nayeem/RHP>

⁴<https://www.nltk.org/>

⁵<https://github.com/cjhutto/vaderSentiment>

Helpfulness Class	Unigram	Bigram
Class #1 Helpful Votes [1, 2)	'room'	'front desk'
	'staff'	'coffee maker'
	'location'	'breakfast buffet'
	'time'	'sofa bed'
	'service'	'swim pool'
Class #2 Helpful Votes [2, 4)	'room'	'front desk'
	'staff'	'shampoo conditioner'
	'service'	'customer service'
	'location'	'resort fee'
	'time'	'pool area'
Class #3 Helpful Votes [4, 8)	'room'	'front desk'
	'staff'	'resort fee'
	'time'	'customer service'
	'service'	'coffee maker'
	'view'	'city view'
Class #4 Helpful Votes [8, 16)	'room'	'front desk'
	'staff'	'resort fee'
	'service'	'customer service'
	'time'	'minute walk'
	'pool'	'life jacket'
Class #5 Helpful Votes [16, ∞)	'room'	'front desk'
	'time'	'resort fee'
	'service'	'bed bug'
	'staff'	'beach chair'
	'pool'	'cable car'

Table 3: Top 5 unigrams and bigrams extracted from five different classes of reviews divided according to helpfulness votes. For each column, green color indicates the overlap with all 5 classes, whereas blue for 4, orange for 3, and red for 2 overlaps.

expressed. Therefore, we keep only the nouns⁶ (with POS tags 'NN' and 'NNS') for extracting the aspects or topics.

Ranking Candidate n -grams We extract the unigrams and bigram collocations for each of the review classes. Then, we rank the unigrams by counting the frequency of occurrences and bigrams using likelihood ratios (Manning and Schütze, 1999) to obtain Top K . We present the Top 5 unigrams and bigrams in Table 3 grouped according to helpfulness classes and ordered by descending ranking scores.

Table 3 shows a high overlap of n -grams among different classes of reviews, which further strengthens our argument that helpfulness does not entirely depend on the review text but rather the confluence of the review text, reviewing history of reviewers (*who post the reviews*), review age (*when the reviews are posted*). Generally, older reviews (i.e., review age) were present longer than the newer reviews in the platform and had more time to accumulate helpful votes.

⁶As adjectives and adverbs may contain sentiment towards aspects.

[Free WiFi, Free parking, Location, Room, Staffs, Front Desk, Food, swimming pools, foods, Bar, Air conditioning, Non-smoking rooms, Fitness center, ATM on site, Shuttle service, Room service, Spa,]

 Aspects / Facilities

[CLS] We could not have been happier with our choice for our family's 3 night stay in Las Vegas recently. The location was perfect. We stayed in a 2 bedroom villa, which was so spacious and had a great view of the Vegas lights and airportThe bathroom to the main bedroom had a fabulous big bath. The beds very comfortable. Dinner in the restaurant in the lobby one night, the food and service were both great. We particularly liked the restaurant and bar next to the pool on level 5, very relaxing for lunch [SEP]

 Review Text

Figure 2: Top 10 ranked tokens of the RHP model shown in green colors with the color intensity indicating the importance of the tokens in the overall prediction.

3.4 Case Study

To gain more insights into the review helpfulness prediction task, we present a detailed case-study to interpret the model behavior and highlight the most important features of this task. Models are interpretable when humans can readily comprehend the reasoning behind model predictions and decisions made (Kim et al., 2016). To this end, we randomly selected a sample with Helpfulness Class = 3 from our test set and used Captum⁷ to interpret the words/tokens that contributed the most to the prediction. As can be seen in Figure 2, the top-ranked words are highly representative of the aspects/facilities listed on the restaurant page. We can conclude from this observation that users tend to look for specific aspects in reviews to find them helpful. We also notice that the use of personal pronouns (e.g., I, we, they, etc.), describing personal experiences, contributes to the helpfulness prediction. People often find reviews useful if it comes from others' experiences and personal pronouns are a good indicator of it.

4 Conclusion and Future Work

In this paper, we develop a model incorporating the reviewer's expertise and temporal information in reviews to predict the helpfulness, especially for unreliable and cold-start reviews. Furthermore, we present a detailed analysis to interpret the model behavior and provide reasoning behind model predictions. For future work, we will look into the problem of personalized review helpfulness prediction to model the demographics and cultural differences of the reviewers.

⁷Captum (<https://captum.ai/>) is an open-source, extensible library for model interpretability that uses the integrated gradients method (Sundararajan et al., 2017).

Limitations

Despite the effectiveness of incorporating the reviewer's history and temporal information of the reviews in helpfulness prediction, our current studies still have several limitations, which can pave the path for future research.

For simplicity, like existing works, we assume that all the users rate reviews unanimously. However, the diversity of demographics, age, and cultural background also affect how users give, receive, and understand the sentiments expressed in reviews. Users may focus on different review aspects based on their preferences (i.e., "5 stars, party every night" vs "5 stars, always quiet and peaceful"). It would be interesting to see how to incorporate personal preferences for the helpfulness prediction task.

Another limitation of our work is that we only worked with reviews written in English. As a result, we filter out the reviews written in other languages and notice code-switched reviews when the reviewers alternate between two or more languages in a single review. We aim to extend this work to support more languages.

Ethics Statement

In our data scraping process, we took into account ethical considerations. We obtained data at an appropriate pace, avoiding any potential DDoS attacks. Additionally, we eliminated any Personal Identifying Information, such as names, telephone numbers, and email addresses, from the data set.

Acknowledgements

We thank all the anonymous reviewers for their valuable feedback and constructive suggestions for improving this work. This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by a grant from Huawei. Mir Tafseer Nayeem is also supported by a Huawei Doctoral Scholarship.

References

Cen Chen, Yinfei Yang, Jun Zhou, Xiaolong Li, and Forrest Sheng Bao. 2018. [Cross-domain review helpfulness prediction based on convolutional neural networks with auxiliary domain discriminators](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 602–607, New Orleans,

Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiahua Du, Jia Rong, Hua Wang, and Yanchun Zhang. 2021. [Neighbor-aware review helpfulness prediction](#). *Decision Support Systems*, 148.

Miao Fan, Yue Feng, Mingming Sun, Ping Li, Haifeng Wang, and Jianmin Wang. 2018. [Multi-task neural learning architecture for end-to-end identification of helpful reviews](#). In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '18*, page 343–350. IEEE Press.

Sheng Khoo Foo, Lee Teh Phoey, and Boon Ooi Pei. 2017. [Consistency of online consumers' perceptions of posted comments: An analysis of tripadvisor reviews](#). *Journal of information and Communication Technology*, 16(2):374–393.

Iftah Gamzu, Hila Gonen, Gilad Kutiel, Ran Levy, and Eugene Agichtein. 2021. [Identifying helpful sentences in product reviews](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 678–691, Online. Association for Computational Linguistics.

Wei Han, Hui Chen, Zhen Hai, Soujanya Poria, and Lidong Bing. 2022. [SANCL: Multimodal review helpfulness prediction with selective attention and natural contrastive learning](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5666–5677, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ruining He and Julian McAuley. 2016. [Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering](#). In *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, page 507–517, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

C. Hutto and Eric Gilbert. 2014. [Vader: A parsimonious rule-based model for sentiment analysis of social media text](#). *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Rimma Kats. 2018. [Surprise! most consumers look at reviews before a purchase](#). Accessed: May 10, 2022.

- Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. 2016. [Examples are not enough, learn to criticize! criticism for interpretability.](#) In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, page 2288–2296, Red Hook, NY, USA. Curran Associates Inc.
- Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. 2006. [Automatically assessing review helpfulness.](#) In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06*, page 423–430, USA. Association for Computational Linguistics.
- Yoon Kim. 2014. [Convolutional neural networks for sentence classification.](#) In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization.](#) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*
- Haijing Liu, Yang Gao, Pin Lv, Mengxue Li, Shiqiang Geng, Minglan Li, and Hao Wang. 2017. [Using argument-based features to predict and analyse review helpfulness.](#) In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1363, Copenhagen, Denmark. Association for Computational Linguistics.
- Junhao Liu, Zhen Hai, Min Yang, and Lidong Bing. 2021. [Multi-perspective coherent reasoning for helpfulness prediction of multimodal reviews.](#) In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5927–5936, Online. Association for Computational Linguistics.
- Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. 2008. [Modeling and predicting the helpfulness of online reviews.](#) In *2008 Eighth IEEE International Conference on Data Mining*, pages 443–452.
- Christopher Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing.* MIT press.
- Lionel Martin and Pearl Pu. 2014. [Prediction of helpful reviews using emotions extraction.](#) In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.
- Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. [Image-based recommendations on styles and substitutes.](#) In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15*, page 43–52, New York, NY, USA. Association for Computing Machinery.
- Rosie Murphy. 2016. [Local consumer review survey 2016.](#) Accessed: May 10, 2022.
- Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. 2010. [How useful are your comments? analyzing and predicting youtube comments and comment ratings.](#) In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, page 891–900, New York, NY, USA. Association for Computing Machinery.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. [Axiomatic attribution for deep networks.](#) In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.
- M Mudambi Susan and Schoff David. 2010. [What makes a helpful online review? a study of customer reviews on amazon.com.](#) *MIS Quarterly*, 34(1):185–200.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing.](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. [Google's neural machine translation system: Bridging the gap between human and machine translation.](#) *arXiv preprint arXiv:1609.08144.*
- Wenting Xiong and Diane Litman. 2011. [Automatically predicting peer-review helpfulness.](#) In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 502–507, Portland, Oregon, USA. Association for Computational Linguistics.
- Wenting Xiong and Diane Litman. 2014. [Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews.](#) In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1985–1995, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. [BERT post-training for review reading comprehension and aspect-based sentiment analysis.](#) In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Shuzhe Xu, Salvador E. Barbosa, and Don Hong. 2020. Bert feature based model for predicting the helpfulness scores of online customers reviews. In *Advances in Information and Communication*, pages 270–281, Cham. Springer International Publishing.

Yinfei Yang, Yaowei Yan, Minghui Qiu, and Forrest Bao. 2015. Semantic analysis and helpfulness prediction of text for online product reviews. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 38–44.

A Dataset Construction

Publicly available datasets which are mostly used for this task are Amazon⁸ (He and McAuley, 2016; McAuley et al., 2015) and Yelp⁹. In Yelp dataset, the user votes are distributed among three categories such as “Useful”, “Funny” or “Cool”, where “Useful” voting feature was introduced much later than the other two categories. Therefore, many good reviews already in the dataset may not have been marked useful. On the other hand, the Amazon dataset does not contain the reviewers’ reviewing history and helpfulness votes to evaluate our hypothesis studied in this paper. Moreover, for Amazon, the samples come from various categories such as Books, Electronics, Clothing, Beauty, Shoes and Jewelry, Grocery, Pet Supplies, etc – the total helpfulness votes for the reviewers are coming from different categories and it’s not explicit in the fields from Amazon website. Therefore, it’s hard to devise expertise because of domain diversity.

We build our dataset by scraping reviews from TripAdvisor¹⁰, a travel site that offers online hotel and restaurant reservations and a platform for sharing the travel experiences of users. We take reviews from January 1st, 2015 until January 1st, 2020, and extract only those written in English. For each review, we extract the review text, the total helpfulness votes and the posting time, and for each reviewer, we extract the number of reviews contributed and the cumulative helpfulness votes. The attributes we extracted are summarized as follows:

⁸http://jmcauley.ucsd.edu/data/amazon/index_2014.html

⁹<https://www.yelp.com/dataset>

¹⁰<https://www.tripadvisor.com>

• Reviews

- Review Text
- Total Review Helpful Votes
- Review Posting Time

• Reviewers

- Total Number of Reviews Contributed
- Cumulative Helpful Votes

B Baseline Systems

We compare our system performance with the following baselines.

- **ARH** (Kim et al., 2006) & **UGR + BGR** (Xiong and Litman, 2011) use machine learning-based methods with hand-crafted features such as *structural*, *lexical*, *syntactic*, *emotional*, *semantic*, and *meta-data* from the review text to address this task. These features are fed into conventional classifiers such as SVM, Random Forest, and gradient boosting to identify helpful reviews.
- **TextCNN** (Chen et al., 2018) employs a text-based CNN model (Kim, 2014) to automatically capture the character-level, word-level, and topic-level features for helpfulness prediction.
- **MTNL** (Fan et al., 2018) utilizes end-to-end multi-task neural learning (MTNL) architecture for classifying helpful reviews. They take the help of an auxiliary task, such as rating regression, to boost the performance of the original task, which is review helpfulness identification.
- **BERTHelp** (Xu et al., 2020) develop their helpfulness prediction model using pre-trained BERT (Devlin et al., 2019). They design a regression model using BERT-based features extracted from review texts, star rating, and product type information from Amazon product review dataset (He and McAuley, 2016).

C Preprocessing & Hyperparameters

Preprocessing We didn’t perform any explicit preprocessing of the review text. Instead, we use BertTokenizer to avoid the out-of-vocabulary (OOV) problem, which uses WordPiece (Wu et al.,

2016) for tokenizing the sentences into words or subwords. In addition, we add special tokens to the start (e.g., [CLS]) and end of each review text (e.g., [SEP]) and truncate all sentences to a single constant length (e.g., 512).

Hyperparameters We use Adam optimizer (Kingma and Ba, 2015) with a learning rate of $3 \times e^{-5}$ and a batch size of 32. We use BERT_{BASE} (Wolf et al., 2020) pre-trained model with a fixed vocabulary. We run the training for 5 epochs and check the improvement of validation (*dev set*) loss to save the latest best model during training.

Towards a Unified Model for Generating Answers and Explanations in Visual Question Answering

Chenxi Whitehouse, Tillman Weyde, Pranava Madhyastha

City, University of London

{chenxi.whitehouse, t.e.veyde, pranava.madhyastha}@city.ac.uk

Abstract

The field of visual question answering (VQA) has recently seen a surge in research focused on providing explanations for predicted answers. However, current systems mostly rely on separate models to predict answers and generate explanations, leading to less grounded and frequently inconsistent results. To address this, we propose a multitask learning approach towards a **Unified Model for Answer and Explanation generation (UMAE)**. Our approach involves the addition of artificial prompt tokens to training data and fine-tuning a multimodal encoder-decoder model on a variety of VQA-related tasks. In our experiments, UMAE models surpass the prior state-of-the-art answer accuracy on A-OKVQA by 10~15%, show competitive results on OK-VQA, achieve new state-of-the-art explanation scores on A-OKVQA and VCR, and demonstrate promising out-of-domain performance on VQA-X.¹

1 Introduction

Contemporary models for visual question answering (VQA) and commonsense reasoning are typically trained discriminatively to select the best answers from Multiple-Choice questions or to classify single-word answers to a predetermined vocabulary (e.g. Anderson et al., 2018). Such settings often lead to limitations such as encouraging models to find superficial correlations (Ye and Kovashka, 2021) or penalising model performance even when the answers are plausible (e.g. synonyms and multi-word expressions, and morphological variations are not considered correct). Most current explanation generation models are trained independently of the QA model and the explanations are usually generated after the QA model has provided an answer. As a result, these explanation models lack access to the process that generated the answer and thus

the grounding of the explanation is limited to the answer text.

We posit that a unified model that simultaneously performs answer prediction and explanation generation is a more effective and consistent approach for VQA. Generative models, such as GPT-3 (Brown et al., 2020), T5 (Raffel et al., 2020), or OFA (Wang et al., 2022a), have been shown to be successful at rapidly adapting to downstream tasks and generating high-quality open-ended text, and hence are suitable candidates for this unified approach.

We propose a multitask learning approach for multimodal transformer-based encoder-decoder models, towards a **United Model for Answer and Explanation generation (UMAE)**. In addition to the current trend of separate answer prediction and explanation generation based on the answers, our approach adds the capability of jointly generating answers and explanations together. Inspired by the success of artificial prompt tokens in Neural Machine Translation (NMT) (Johnson et al., 2017), we extend and demonstrate the efficacy of the artificial prompt-based method for VQA in a multitask setup. We augment training instances with artificial prompt tokens, enabling the model to distinguish different tasks while learning shared semantic features. Experiments on a combination of three knowledge-intensive VQA datasets, OK-VQA (Marino et al., 2019), A-OKVQA (Schwenk et al., 2022), and VCR (Zellers et al., 2019), show that the UMAE models achieve a new state-of-the-art (SOTA) answer accuracy on A-OKVQA, new SOTA explanation score on VCR, and competitive out-of-domain performance on VQA-X (Park et al., 2018). UMAE supports the generation of the answer to a question, the explanation for a given question and answer, and both together jointly, making the model efficient and flexible. An illustration of the training setup is shown in Figure 1.

In summary, our main contributions are as follows: (1) the UMAE framework where answers and

¹Code is available at: <https://github.com/chenxwh/UMAE>.

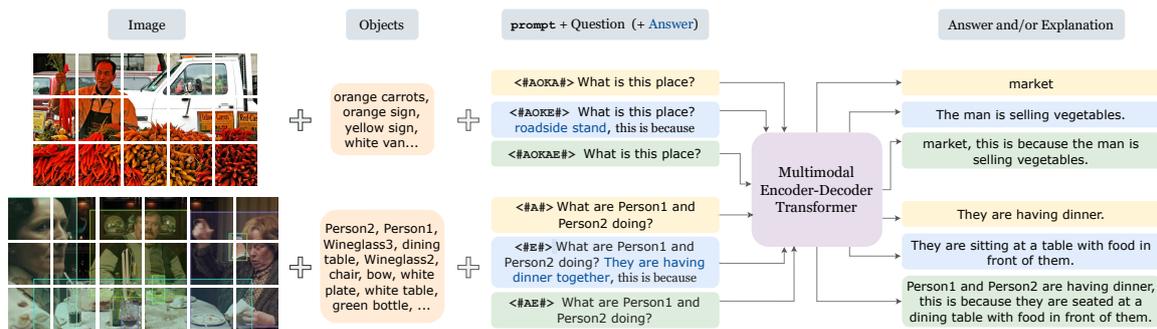


Figure 1: Illustration of UMAE: we train a multimodal encoder-decoder model on the mix of VQA tasks for jointly optimising answer and explanation, where we distinguish the training instances and target output with artificial prompt tokens (e.g. `<#AOKA#>`). The top and bottom examples are from A-OKVQA and VCR, respectively.

explanations can be generated by a single unified model (§3.1); (2) a simple and efficient training approach that uses multitask learning with artificial prompts and demonstrates its ability to generalise across domains (§4); (3) a method to map generated answers to Multiple-Choice options via evaluating the perplexity of the generation (§3.2); (4) new SOTA results by UMAE, particularly for explanation generation and promising out-of-domain performance (§5).

2 Related Work

Multimodal Transformer-based Models achieve SOTA performance on various vision-language tasks (Chen et al., 2020; Li et al., 2020; Cho et al., 2021; Wang et al., 2022c; Zhang et al., 2021). They showcase the possibility of capturing richer multimodal semantic coherence than discriminatively trained models and are further capable of generating self-explanations. Pretrained on multitask settings with natural language instructions, e.g. “*what does the region describe?*”, models like OFA (Wang et al., 2022a) are claimed to have the capability to transfer to unseen tasks and domains via similar instructions. However, contrary to these claims, we observe that pretrained OFA is incapable of generating valid explanations through simple natural language instructions (§5).

Artificial Prompt Tokens have previously been explored for NMT by Johnson et al. (2017); Mitzalis et al. (2021). They propose a single model with the traditional NMT model architecture (usually for one language pair) and jointly train on different language pairs with added artificial prompts, e.g. 2_{es} to distinguish the target language. This approach has been found to foster implicit cross-lingual bridging and exhibit zero-shot translation

capability. In this paper, we exploit a similar approach with artificial prompts for answer and explanation generation in VQA with a united model. This enables the model to learn shared features among tasks and datasets in various domains.

Explanation Generation for VQA has gained growing interest in research. However, most recent approaches use separate models to predict answers and generate explanations (Dua et al., 2021). Wu and Mooney (2019) generate explanations with an object detector and a GRU unit for text embedding, then train on a subset of VQA-X in which the explanations contain the objects most attended to by the model. Kayser et al. (2021) develop an e-UG model combining UNITER (Chen et al., 2020) for processing multimodal input and GPT-2 (Radford et al., 2019) for generation. In contrast, in this paper, we propose using a single united model for more grounded answer and explanation generation.

3 Methodology

3.1 Multitask Learning with Artificial Prompt

We formulate three generation settings: $Q \rightarrow A$: answer prediction; $QA \rightarrow E$: explanation generation conditioned on the answer; and $Q \rightarrow AE$: *joint* answer and explanation generation for a given question. We hypothesise that by training the model to generate both the answer and its explanation *simultaneously*, the result answer and explanation will be more grounded and consistent.

We use a pretrained multimodal encoder-decoder transformer as our base model (here we build on the openly released version of OFA as a strong baseline), and finetune the model on a mix of VQA datasets from different domains.

Different from OFA, for each image in the VQA datasets, we first extract objects and attributes us-

MODEL	OK-VQA		A-OKVQA				VCR	
	<i>direct answer</i>	<i>multiple choice</i>			<i>direct answer</i>	<i>multiple choice</i>	BERTSCORE	
	TEST	VAL (<i>ppl</i>)	VAL (<i>GloVe</i>)	TEST	VAL	TEST	VAL (<i>ppl</i>)	VAL
OFA*	40.40	24.54	56.19	47.40	48.09	39.77	33.55	64.55
OFA _{Q->A}	49.93	74.32	65.30	61.71	63.00	53.91	54.89	83.85
UMAE _{ALL}	51.77	74.59	65.67	63.26	63.29	56.14	56.66	85.97
PRIOR-BEST	54.41	–	60.30	53.70	48.60	40.70	(77.10) [†]	–

Table 1: Performance of models for answer generation. Better results are in bold. OFA* refers to the pretrained OFA. Prior-best results for the three datasets are from Gui et al. (2022), Schwenk et al. (2022), Wang et al. (2022b), respectively. † is from a discriminative model and thus not comparable (see Ye and Kovashka, 2021).

DATASET	MODEL	e-ViL SCORES			N-GRAM SCORES					LEARNT SCORE
		S _O	S _T	S _E	BLEU4	ROUGE-L	METEOR	CIDEr	SPICE	BERTSCORE
A-OKVQA	OFA*	4.44	56.19	7.90	0.30	4.45	3.26	4.82	4.62	68.64
	OFA _{Q->A} +OFA _{QA->E}	35.82	74.32	48.29	22.18	48.51	23.56	86.76	22.46	85.96
	UMAE _{A-OKVQA}	37.10	73.97	50.15	27.61	52.23	24.06	104.39	22.88	87.86
	UMAE _{ALL}	37.91	74.59	50.82	27.35	52.56	24.83	101.09	23.33	88.21
VCR	e-UG	19.30	69.80	27.60	4.30	22.50	11.80	32.70	12.60	79.00
	UMAE _{VCR}	22.57	56.68	39.82	12.25	28.87	16.67	48.14	27.36	81.77
	UMAE _{ALL}	22.82	56.66	40.27	13.44	29.53	17.54	47.33	26.45	81.91
VQA-X	e-UG	36.50	80.50	45.40	23.20	45.70	22.10	74.10	20.10	87.00
	UMAE _{ALL}	31.58	77.65	40.67	14.63	35.12	20.29	50.35	19.13	85.40

Table 2: Explanation Scores. OFA* is the pretrained OFA, showing the transferability of OFA for generating explanations with natural language instructions. Results with e-UG are from Kayser et al. (2021). We show the best results of A-OKVQA and VCR in bold. The last row in blue shade shows *out-of-domain* performance.

ing a bottom-up top-down attention-based model, which is crucial for open-domain VQA tasks (Anderson et al., 2018). We then add artificial prompt tokens at the beginning of the textual input to signal the generation task (answer, explanation, or both) and the dataset². For $Q \rightarrow \text{AE}$, we concatenate answers and explanations with a separator in between. Finally, we mix all training instances, each consisting of an image (processed in patches), objects and attributes, and textual input with artificial prompts.

3.2 Perplexity as Multiple Choice Metric

To map the generated output to Multiple-Choice options, in previous work the predictions are loosely matched with options or gold answers using embedding-based methods, such as GloVe embedding similarity (Schwenk et al., 2022). In contrast to these approaches, we propose to evaluate each option as a *text generation* task, by feeding the model the information that was used to generate the answer as a prompt, and calculating the likelihood of each option being generated. Formally, given an option $Y = (y_1, y_2, \dots, y_t)$ with t tokens,

²Artificial prompt tokens are added as special tokens to the tokenizer to avoid bias in the pretrained embeddings. However, we note that these tokens may be biased w.r.t their association with specific tasks after training, which is an intended effect.

we calculate the probability of each token y_i being generated by feeding the image, objects, and question, as well as the first $i - 1$ tokens from Y to the model p_θ . The perplexity is then calculated with: $\text{PPL}(Y) = \exp \left\{ -\frac{1}{t} \sum_i^t \log p_\theta(y_i | y_{<i}) \right\}$, which reflects the probability of option Y being generated by the model. Finally, the option with the lowest perplexity is chosen as the answer.

We also compare the performance of our approach, using perplexity as the metric, with GloVe embedding similarity for A-OKVQA (see Table 1).

4 Experimental Setup

We primarily evaluated our proposed UMAE approach using pretrained OFA³ as the base model on three knowledge-intensive VQA datasets: OK-VQA, A-OKVQA and VCR⁴. We split the original train set into train and validation set (95%-5%) for all three datasets. Since the test set is not publicly available for A-OKVQA and VCR, we use the original validation set for experimental analyses. We prepare training instances⁵ as introduced

³<https://github.com/OFA-Sys/OFA>

⁴See Appendix A for datasets details.

⁵Specifically, we add <#OKA#> for OK-VQA (only answers are available), <#A#>, <#E#>, <#AE#> for VCR, and <#AOKA#>, <#AOKE#>, <#AOKAE#> for A-OKVQA.

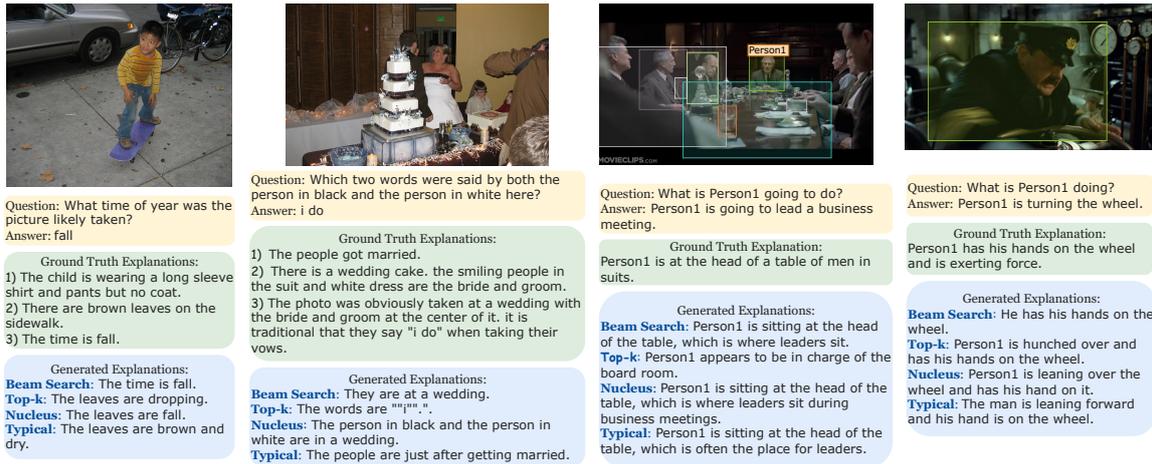


Figure 2: Examples of generated explanations from $UMA_{E_{ALL}}$ model with different decoding strategies. The two examples on the left are from A-OKVQA and the two on the right are from VCR.

in §3.1. Additionally, for VCR, we draw coloured highlights around the referenced entity on the images, following Zellers et al. (2021) (Appendix A). To account for the imbalance in size among the datasets, we up-sample instances in OK-VQA and A-OKVQA, and shuffle all instances to train the $UMA_{E_{ALL}}$ model.

For ablation studies, we finetune OFA for separate answer prediction ($OFA_{Q \rightarrow A}$) and explanation generation conditioned on answers ($OFA_{QA \rightarrow E}$). To better understand the impact of mixing datasets from different domains, we also train $UMA_{E_{A-OKVQA}}$ and $UMA_{E_{VCR}}$, focusing on all three answer and explanation generation tasks but only using data from a single dataset: either with A-OKVQA or with VCR. Details of training parameters are included in Appendix B.

We use beam search for generating answers and additionally experiment with different decoding methods including top-k sampling, Nucleus sampling (Holtzman et al., 2020), and Typical sampling (Meister et al., 2022), for generating explanations. We evaluate answer accuracy as well as explanation quality with automatic NLG metrics and e-ViL scores (Kayser et al., 2021). e-ViL scores consist of S_T (task/answer accuracy), S_E (explanation score), and overall S_O (product of S_T and S_E), where S_E is the harmonic mean of NGRAMScore (the harmonic mean of n-gram scores ROUGE-L (Lin and Och, 2004), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016)) and additionally the BERTScore (Zhang et al., 2020), a learned similarity metric over contextual representations of sentences.

5 Results and Discussion

5.1 Answer Accuracy

Table 1 presents our observations for answer accuracy on $Q \rightarrow A$ task over the three datasets. We also evaluate VCR answers using BERTScore as the answers for VCR are usually sentences. We observe that $UMA_{E_{ALL}}$ outperforms $OFA_{Q \rightarrow A}$ on all datasets, improves the prior SOTA on A-OKVQA by 10~15%, and achieves competitive results on OK-VQA. For models that are finetuned on A-OKVQA, we also see a salient improvement (+9%) with the proposed mapping of options by perplexity in Multiple-Choice, instead of GloVe embeddings similarity⁶. We conducted several ablation studies on the dependency of the modality for the answer accuracy in A-OKVQA, where we find the visual encoder is crucial for performance. Details are included in Appendix C.

5.2 Explanation Evaluation

Table 2 shows e-ViL scores (§4) for explanations using automatic NLG metrics⁷. Following the same setup as in Kayser et al. (2021), an explanation is evaluated only if the answer predicted by the system is correct⁸. We observe that pretrained OFA with natural language prompts, e.g. “*what is the explanation for the answer?*” or “*this is*

⁶Preliminary experiments with NLG metrics (BERTScore and BLEU) for selecting the options given generation were sub-optimal.

⁷Nucleus sampling shows best results and is reported. Detailed scores with different decoding methods are shown in Appendix D.

⁸A limitation of evaluating all explanations is that explanations of wrong answers may get high scores with n-gram metrics, even though they are justifying wrong answers and should be penalised.

MODEL	S_E	BLEU4	R-L	MET.	CIDEr	SPICE	BERTSc.
OFA _{Q->A} +OFA _{QA->E}	42.4	20.0	44.2	19.3	66.7	19.1	85.1
UMA _E -OKVQA	45.8	23.6	47.9	21.7	78.0	20.5	86.9
UMA _E _{ALL}	46.8	24.9	49.5	22.3	84.1	20.8	87.3

Table 3: Explanation scores on the same subset of A-OKVQA.

because” performs poorly, as most generated explanations are words (“*yes/no*”) or short-phrases⁹. We compare UMAE models (on all and individual datasets) with prior best results from e-UG (see §2), and standard separated trained baselines (OFA_{Q->A}+OFA_{QA->E}). UMAE_{ALL} achieves better results across all datasets, showing the advantage of mixing tasks and datasets in different domains. For out-of-domain evaluation on VQA-X, UMAE_{ALL} also shows mostly competitive results. Examples of explanation generation are shown in Figure 2 and Appendix E.

Since e-ViL only evaluates an explanation if a model generates the correct answer, the subset of explanations evaluated varies by model. To fairly compare explanations on the same subset, we propose only using the subset of samples where all models provide correct answers for explanation prediction. Table 3 shows the results on A-OKVQA with such a subset of 770 candidates, where UMAE_{ALL} shows an even higher explanation score. This highlights that UMAE_{ALL} generates explanations that overlap significantly better with gold explanations.

In summary, our experiments demonstrate that the UMAE model leads to improved answer and explanation generation and allows for the flexibility to generate different types of outputs, including answers, explanations, or both. We observe that UMAE exhibits promising results in jointly generating both the answer and explanation. We further provide a comparative evaluation in Appendix F as a first step towards comparison as there is currently no standard evaluation setup for the joint answer and explanation evaluation.

5.3 Error Analysis

To better understand the generated answers and errors, we randomly sample 50 errors in OK-VQA and A-OKVQA. Our analysis reveals the following main error types, where the first three are related to

⁹BERTScore is not representative of the validity of outputs from OFA*. We refer the reader to an exposition of the problems associated with NLG metrics in Caglayan et al. (2020).

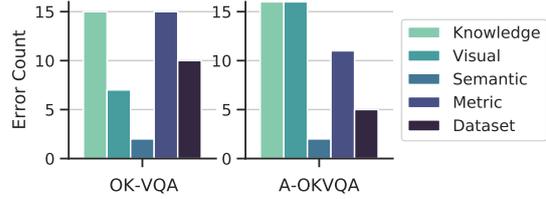


Figure 3: Error type distribution in 100 random samples from A-OKVQA and OK-VQA.

model performance: (1) *Knowledge*: the implicit knowledge learned by the model is insufficient for answering some of the knowledge-intensive questions, such as questions asking *when* a certain sport was invented; (2) *Visual*: the model fails to identify the visual attributes correctly, such as questions about *recognising object shape or material*; (3) *Semantic disassociation*: the model misinterprets questions or fails to match the intended semantic meaning. For example, it may answer what *an object is* instead of a more complex question such as *what is commonly packed in it* (e.g. answering "suitcase" instead of "clothes"); (4) *Metric*: the evaluation metric may penalise some of the plausible answers, especially when searching for exact match answers (mostly due to the difference of singular/plural or phrases with/without space in between); and (5) *Dataset*: errors due to issues in the datasets themselves. We discuss prominent issues in dataset quality briefly in Appendix G and further present the distribution of error types in Figure 3.

6 Conclusions

In this work, we propose UMAE, a unified model that generates answers and explanations in VQA using a multitask learning approach for multimodal encoder-decoder models, where artificial prompt tokens are added to distinguish different tasks while learning shared semantics. Evaluation of our approach on various VQA tasks shows that UMAE outperforms prior best models and separately trained baselines in both answer and explanation scores, where we also demonstrate the benefit of using perplexity as the metric for mapping generated answers to Multiple-Choice options. Additionally, UMAE offers flexibility in output and can generate explanations for datasets without explanations for training, e.g. OK-VQA, while also improving answer quality. Through case studies and error analysis, we identify potential areas for future improvement, including dataset quality.

Limitations

We discuss the limitations of our work in the following two aspects. Firstly, the experiments with our proposed framework and finetuning approach are primarily on the OFA model. We believe our approach applies to any multimodal generative model, however, it would also provide insights to experiment with more models. Secondly, regarding the evaluation of our proposed joint framework, to better evaluate the generated explanation quality, especially to evaluate the difference between explanations generated jointly with answers and generated conditioned on the answers, human judgement would be an important criterion compared to automatic NLG metrics.

Acknowledgements

We acknowledge the support of Apoorv Khandelwal from AI2 for providing us with results for the evaluation of our model predictions over a hidden test set. This was valuable for our earlier draft of the paper. We would like to thank the anonymous reviewers who provided valuable feedback on the previous draft of our paper.

References

- Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic Propositional Image Caption Evaluation. In *European conference on computer vision*, pages 382–398. Springer.
- Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and Top-down Attention for Image Captioning and Causal Question Answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.
- Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual Question Answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.
- Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language Models are Few-shot Learners. *Advances in neural information processing systems*, 33:1877–1901.
- Ozan Caglayan, Pranava Madhyastha, and Lucia Specia. 2020. Curious case of language generation evaluation metrics: A cautionary tale. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2322–2328, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal Image-Text Representation Learning. In *European conference on computer vision*, pages 104–120. Springer.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying Vision-and-Language Tasks via Text Generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR.
- Radhika Dua, Sai Srinivas Kancheti, and Vineeth N Balasubramanian. 2021. Beyond vqa: Generating Multi-Word Answers and Rationales to Visual Questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1623–1632.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.
- Liangke Gui, Borui Wang, Qiuyuan Huang, Alexander Hauptmann, Yonatan Bisk, and Jianfeng Gao. 2022. KAT: A knowledge augmented transformer for vision-and-language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 956–968, Seattle, United States. Association for Computational Linguistics.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Maxime Kayser, Oana-Maria Camburu, Leonard Salewski, Cornelius Emde, Virginie Do, Zeynep Akata, and Thomas Lukasiewicz. 2021. e-ViL: A Dataset and Benchmark for Natural Language Explanations in Vision-Language Tasks. In *Proceedings*

- of the *IEEE/CVF International Conference on Computer Vision*, pages 1244–1254.
- Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.
- Chin-Yew Lin and Franz Josef Och. 2004. [Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics](#). In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft Coco: Common Objects in Context. In *European conference on computer vision*, pages 740–755. Springer.
- Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204.
- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2022. Typical Decoding for Natural Language Generation. *arXiv preprint arXiv:2202.00666*.
- Faidon Mitzalis, Ozan Caglayan, Pranava Madhyastha, and Lucia Specia. 2021. [BERTGen: Multi-task generation through BERT](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6440–6455, Online. Association for Computational Linguistics.
- Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8779–8788.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI Technical Report*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. 2017. Movie Description. *International Journal of Computer Vision*, 123(1):94–120.
- Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-OKVQA: A Benchmark for Visual Question Answering using World Knowledge. *ArXiv*, abs/2206.01718.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based Image Description Evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.
- Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022a. [OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23318–23340. PMLR.
- Yanan Wang, Michihiro Yasunaga, Hongyu Ren, Shinya Wada, and Jure Leskovec. 2022b. VQA-GNN: Reasoning with Multimodal Semantic Graph for Visual Question Answering. *arXiv preprint arXiv:2205.11501*.
- Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2022c. [SimVLM: Simple visual language model pretraining with weak supervision](#). In *International Conference on Learning Representations*.
- Jialin Wu and Raymond Mooney. 2019. [Faithful multimodal explanation for visual question answering](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 103–112, Florence, Italy. Association for Computational Linguistics.
- Keren Ye and Adriana Kovashka. 2021. A Case study of the Shortcut Effects in Visual Commonsense Reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 3181–3189.
- Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From Recognition to Cognition: Visual Commonsense Reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition*.
- Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. 2021. [MERLOT: Multimodal Neural Script Knowledge Models](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 23634–23651. Curran Associates, Inc.
- Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting Visual Representations in Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [BERTScore: Evaluating Text Generation with BERT](#). In *International Conference on Learning Representations*.

A Datasets

The datasets used in the paper are as follows:

OK-VQA (Marino et al., 2019) is a knowledge-based VQA dataset that requires outside knowledge beyond the images to answer the questions. It has train and test splits of size 9,009 and 5,046. Each question is provided answers by five annotators. To use the VQA (Antol et al., 2015) metric, each annotated answer is then repeated twice to form a gold answer set with 10 answers. Since no explanation is provided, we only train $Q \rightarrow A$ task on OK-VQA.

A-OKVQA (Schwenk et al., 2022) is currently the largest knowledge-based VQA dataset split into 17.1K, 1.1K, and 6.7K for train, validation, and test, respectively. The questions cover four knowledge types: visual, commonsense, knowledge bases, and physical. For each question, it provides both multiple-choice answers and 10 free-form answers (annotated by 10 different people), as well as three explanations. Images in both OK-VQA and A-OKVQA are from MSCOCO (Lin et al., 2014), and answers in both datasets are in single words or short phrases.

VCR (Zellers et al., 2019) is a large multiple-choice dataset for Visual Commonsense Reasoning. The train, validation, and test splits have 191.6k, 21.3k, and 26.5k instances, respectively. Each question has four answer options in sentences, and the correct answer is further provided with four explanation options. Images in VCR are from movie clips (Rohrbach et al., 2017). Bounding boxes of entities are provided associated with mentions such as `Person1` in questions, answers and explanations. We follow Zellers et al. (2021) and draw coloured highlights around the referenced entity on the images, where entity names and the coloured highlights are consistent in the entire dataset, expecting the model to learn the association between the coloured bounding box and the entity.

VQA-X (Park et al., 2018) contains a subset from the VQA_{v2} (Goyal et al., 2017) dataset and further provides three explanations for each question. The image-question pairs are split into train, validation, and test with 29.5k, 1.5k, and 2k instances, respec-

QUESTION	OBJECTS	IMAGES	ACCURACY
✓	✓	original	50.39
✓	✗	✗	39.16
✓	✗	random	33.48
✓	✓	✗	33.28

Table 4: Ablation on the modality dependency for answer accuracy of A-OKVQA.

tively. We only use the original test set to evaluate the zero-shot performance of the trained models.

B Hyper-Parameters and Training

We begin with the pretrained weights from the original OFA-large¹⁰, which is trained on vision-only tasks including Image Classification, language-only tasks including Sentence Classification, Text Summarisation, as well as various vision-language tasks including Image Captioning, Visual Question Answering and Visual Entailment. Adam is used as the optimizer and cross-entropy is the loss function. We set the learning rate to 10^{-5} , the warm-up ratio to 0.4, and the patch image size to 480. We shuffle all the training examples and use batch size 16. Due to the large size of VCR, we train for 30 epochs on models involving VCR (OFA_{Q->A} for VCR, UMAE_{VCR} and UMAE_{ALL}), and up to 100 epochs for other models. We report the empirical performance with checkpoints that perform best on the validation set (the 5% split from the original train set). For A-OKVQA, we additionally report the answer accuracy on the original test set.

C Ablations on Modality Dependency

We conduct several ablation studies to investigate the dependency of object features and images on the performance of our model UMAE_{ALL} for answer accuracy of A-OKVQA, where we removed images, replaced them with random images, and removed extracted attributes and features. Results in Table 4 show that the visual encoder is crucial for performance and that visual objects alone are not sufficient for answer prediction. Using a random image would introduce noise and therefore performs worse than not including the image at all. We did not test removing the question because we believe the model needs the questions to be able to provide answers.

¹⁰<https://github.com/OFA-Sys/OFA>

DATASET	DECODING	e-ViL	N-GRAM SCORES								LEARNT SC.
		S_E	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE-L	METEOR	CIDEr	SPICE	BERTSCORE
A-OKVQA	BEAMSEARCH	44.71	52.01	36.69	26.72	19.88	40.39	22.06	68.48	20.94	86.05
	TOP-K ($k = 100$)	44.34	52.56	37.06	27.06	19.72	44.45	21.58	73.44	19.38	86.27
	NUCLEUS ($p = 0.4$)	50.82	58.92	44.66	35.06	27.35	52.56	24.83	101.09	23.33	88.21
	TYPICAL ($p = 0.6$)	47.27	54.18	39.39	29.82	22.18	47.78	22.79	84.43	21.47	86.95
VCR	BEAMSEARCH	40.23	26.41	20.15	15.95	12.47	29.13	16.82	49.72	27.70	81.84
	TOP-K ($k = 50$)	33.19	20.98	14.89	11.18	8.33	23.65	13.72	32.73	21.99	80.31
	NUCLEUS ($p = 0.1$)	40.27	31.42	22.95	17.62	13.44	29.53	17.54	47.33	26.45	81.91
	TYPICAL ($p = 0.4$)	35.12	23.42	16.88	12.83	9.64	25.36	14.70	35.85	23.32	80.70
VQA-X	BEAMSEARCH	35.88	37.84	24.91	16.67	10.97	31.32	17.90	38.23	16.23	84.39
	TOP-K ($k = 50$)	33.28	38.35	23.11	14.21	8.45	29.15	17.05	32.89	15.26	83.41
	NUCLEUS ($p = 0.1$)	40.67	47.56	31.44	21.47	14.63	35.12	20.29	50.35	19.13	85.40
	TYPICAL ($p = 0.5$)	36.31	40.85	25.57	16.82	11.14	31.08	18.15	39.71	16.62	83.93

Table 5: Explanation scores with automatic NLG for generated explanations ($QA \rightarrow E$) from $UMAE_{ALL}$ model with different decoding strategies. The last two rows (with blue shadow) indicate out-of-domain performance.

DATASET	DECODING	e-ViL	N-GRAM SCORES								LEARNT SC.
		S_E	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE-L	METEOR	CIDEr	SPICE	BERTSCORE
A-OKVQA	BEAMSEARCH	47.01	54.75	41.39	32.08	24.25	49.75	22.54	86.28	20.68	87.39
	NUCLEUS ($p = 0.5$)	46.72	55.53	41.63	31.91	23.67	49.16	22.48	82.37	20.67	87.18
VCR	BEAMSEARCH	37.02	25.00	18.90	14.87	11.54	27.07	15.66	38.77	25.03	80.68
	NUCLEUS ($p = 0.1$)	35.10	27.41	19.36	14.50	10.73	26.18	15.21	34.99	21.88	80.52
VQA-X	BEAMSEARCH	38.13	39.91	26.30	17.99	12.46	31.69	19.11	42.10	18.15	84.95
	NUCLEUS ($p = 0.1$)	39.67	44.92	28.88	19.04	12.55	33.08	20.07	44.28	19.19	85.21

Table 6: Explanation scores with automatic NLG for generated explanations from $Q \rightarrow AE$ with $UMAE_{ALL}$ model. The last two rows (with blue shadow) indicate out-of-domain performance.

D More Explanation Scores

For decoding, we evaluate the performance of beam search with the size of 5, top-k sampling with k from $\{50, 100, 200, \dots, 1000\}$, and Nucleus and Typical (Meister et al., 2022) sampling, both with p from $\{0.1, 0.2, \dots, 0.9\}$. We show the details of the NLG scores using different decoding strategies for explanations generated from $QA \rightarrow E$ in Table 5, and $Q \rightarrow AE$ in Table 6.

E Examples of Generated Explanations

Examples of the explanations generated with beam search and Nucleus sampling for A-OKVQA are shown in Figure 4, and VCR in Figure 5.

F Joint Generation Performance

We present the results of the proposed $Q \rightarrow AE$ task where answers and explanations are jointly generated. We parse the generated sequence to the answer and the explanation and use the same sets of metrics as the separate generation for evaluation. Results for answers in Table 7 and explanations in Table 8. For answers, since the perplexity metric does not directly compare the generation,

TASK	A-OKVQA	VCR	VQA-X
	MC (GOLVE)	BERTSCORE	DA
Q->A	65.67	81.91	77.65
Q->AE	65.67	82.30	69.60

Table 7: Evaluation of answers generated given questions ($Q \rightarrow A$) and jointly generated with explanations ($Q \rightarrow AE$). MC stands for Multiple Choice, DA for Direct Answer. The last column with a blue shadow indicates out-of-domain performance.

DATASET	S_E		NGRAMSCORE		BERTSCORE	
	QA->E	Q->AE	QA->E	Q->AE	QA->E	Q->AE
A-OKVQA	50.82	47.01	35.69	32.15	88.21	87.39
VCR	40.27	37.02	26.70	24.02	81.91	80.68
VQA-X	40.67	39.67	26.69	25.85	85.40	85.21

Table 8: Scores of explanations generated given answers ($QA \rightarrow E$) and jointly generated with answers ($Q \rightarrow AE$). The last row with a blue shadow indicates out-of-domain performance.

we show the Multiple-Choice accuracy using the Glove metric for A-OKVQA and BERTScore for VCR answer sentences.

	OK-VQA	A-OKVQA	
	DA	MC (GLOVE)	DA
BEST	80.94	80.74	66.20
AVERAGE	54.98	71.53	57.29
WORST	16.37	59.35	41.46

Table 9: Human performance on OK-VQA and A-OKVQA measured from the ground truth answers.

see that the average performance on both datasets is relatively poor, which indicates the noise in the datasets. The quality of the datasets needs to be more carefully inspected so that the model performance evaluated on these datasets can be more meaningful.

G Datasets Quality and Issues

As mentioned in [subsection 5.3](#), during error analysis we found that many errors are due to the issue in the dataset itself. Concretely, we observe the following issues in the existing datasets: (1) wrong answers (2) subjective or unanswerable questions (3) typos or unclear expressions (4) not requiring images or knowledge to answer the question as designed.

Furthermore, since the answer and explanation for a question in VCR are obtained from the same person who authored the question, this may result in severe subjectivity in the answers or explanations. For example, we find that many questions in VCR require knowledge of the *movie plot* from which the image is extracted, rather than *common-sense reasoning* to answer the questions. While human annotators have an implicit understanding of the movies, the dataset itself does not contain relevant contextual information.

We show some of the issues in the datasets below. [Figure 6](#) shows examples from VCR that require an understanding of the movie plot to generate answers. [Figure 7](#) shows examples from OK-VQA where questions and answers are subjective or ambiguous. [Figure 8](#) shows examples from A-OKVQA and VQA-X that either contain wrong answers, questions that do not need visual input or typos which severely impact the model generation (“house” should be “horse”).

To understand the inter-annotator agreement for the datasets, we further measure the best, average and worst human performance on OK-VQA and A-OKVQA by selecting the most common answer, a random answer, and the least common answer, respectively, from the 10 ground truth answers for each question. We calculate the performance using the VQA metric for direct answers, and the GloVe metric for Multiple Choice for simplicity. Note that we also remove the answer selected from the ground truth answers when measuring human performance. From the results in [Table 9](#) we can



Question: Why is the woman wearing goggles?
 Answer: protection

Ground Truth Explanations:
 0) The woman is wearing goggles for protections.
 1) The snow goggles this woman wears protects her eyes from the sun and other bits of debris she might encounter skiing downhill.
 1) There is a lot of sun glare and snow flying up when skiing.

Explanation Generation:
 Beam: The woman needs protection.
 Nucleus: The woman is skiing and needs protection.

Answer + Explanation Generation:
 Beam: protection, this is because The woman wants to protect her eyes.
 Nucleus: protection, this is because The woman is wearing goggles to protect her eyes from the sun.



Question: What time of day is it likely right now?
 Answer: morning

Ground Truth Explanations:
 0) It is sunny out. there are fruit on the table, so it likely is time for breakfast.
 1) The time is morning.
 2) You can see the light shining in through the window and door.

Explanation Generation:
 Beam: The people are using umbrellas because it's raining.
 Nucleus: The people are using umbrellas to keep from getting wet.

Answer + Explanation Generation:
 Beam: morning, this is because The sun is shining through the window.
 Nucleus: morning, this is because There is light coming through the window.



Question: What has caused the elephants to turn brown?
 Answer: dirt

Ground Truth Explanations:
 0) These elephants have been rolling around in the mud.
 1) The elephants are covering themselves in the naturally occurring brown substance that is near the water.
 2) The elephants are dirty.

Explanation Generation:
 Beam: The elephants are in mud.
 Nucleus: The elephants are standing in dirt.

Answer + Explanation Generation:
 Beam: mud, this is because The elephants are muddy.
 Nucleus: mud, this is because The elephants are in the mud.



Question: What are the umbrellas placed in the sand to block?
 Answer: sun

Ground Truth Explanations:
 0) They are on a beach on a sunny day, and the umbrellas give cool and shade from the sun.
 1) The umbrellas block sun.
 2) Umbrellas provide shade.

Explanation Generation:
 Beam: The umbrellas block the sun.
 Nucleus: The umbrellas are blocking the sun.

Answer + Explanation Generation:
 Beam: sun, this is because The umbrellas block the sun.
 Nucleus: sun, this is because The umbrellas are placed in the sand to block the sun.



Question: Why are the people lining up?
 Answer: boarding

Ground Truth Explanations:
 0) The people are near a plane as if they want to get on.
 1) The people are boarding.
 2) The people are lined up facing the plane. people line up facing transportation vehicles when they are preparing to board.

Explanation Generation:
 Beam: The people are lined up to board a plane.
 Nucleus: The people are lined up to board a plane.

Answer + Explanation Generation:
 Beam: board plane, this is because The people are lined up to board the plane.
 Nucleus: board plane, this is because The people are boarding a plane.



Question: What is the white appliance the people are looking in being used to store?
 Answer: food

Ground Truth Explanations:
 0) The appliance is a fridge.
 1) Perishable food is kept in this little refrigerator, which is too small for many people, but just right for one or two.
 2) There are multiple edible items in the fridge already.

Explanation Generation:
 Beam: The appliance is a refrigerator.
 Nucleus: The appliance is a fridge.

Answer + Explanation Generation:
 Beam: food, this is because The appliance is a fridge.
 Nucleus: food, this is because The appliance is a fridge.



Question: Why are the people using umbrellas?
 Answer: it's raining

Ground Truth Explanations:
 0) There is rain and the people are being sheltered by umbrellas.
 1) The people want rain protection.
 2) They are protecting themselves from getting wet from the rain.

Explanation Generation:
 Beam: The people are using umbrellas because it's raining.
 Nucleus: The people are using umbrellas to keep from getting wet.

Answer + Explanation Generation:
 Beam: rain, this is because The people are using umbrellas because it is raining.
 Nucleus: it's raining, this is because The people are using umbrellas to stay dry.



Question: The layout of buildings most resembles which period?
 Answer: historical european

Ground Truth Explanations:
 0) The cars are on the left side of the road, so it is either the united kingdom or Ireland.
 1) The high pitched roofs and the stone architecture is from then
 2) The layout is from the olden days.

Explanation Generation:
 Beam: The buildings are old.
 Nucleus: The buildings are from the late nineteenth century.

Answer + Explanation Generation:
 Beam: medieval, this is because The buildings look like medieval buildings.
 Nucleus: medieval, is because The buildings look like medieval buildings.

Figure 4: Examples of generated answers and explanations for A-OKVQA.



Question: What event are Person2 and Person8 walking away from?
 Answer: A wedding reception.

Ground Truth Explanation:
 Everyone is really dressed up, and the lights give the area a fairy tale ambiance which is typical of a wedding.

Explanation Generation:
 Beam: Person2 is wearing a pink dress and Person8 is wearing a suit.
 Nucleus: Person2 is wearing a pink dress and Person8 is holding hands with her.

Answer + Explanation Generation:
 Beam: Person2 and Person8 are walking away from a wedding, this is because Person2 and Person8 are holding hands and wearing formal clothing.
 Nucleus: They are walking away from a wedding, this is because There are balloons and strings of lights above them.



Question: Is Dog2 a living creature?
 Answer: No, Dog2 is a stuffed animal.

Ground Truth Explanation:
 Dog2 is immobile, and wouldn't be still with so much prey around. Dog2 is also obviously a toy, based on the coloration.

Explanation Generation:
 Beam: Teddybear1 and Teddybear2, and Teddybear5 are stuffed animals, Dog2 is a stuffed animal.
 Nucleus: Dog2 is a stuffed animal, and stuffed animals are not alive.

Answer + Explanation Generation:
 Beam: No, Dog2 is not a living creature, this is because Dog2 is a stuffed animal.
 Nucleus: No, Dog2 is not a living creature, this is because Dog2 is a stuffed animal, stuffed animals are not alive.

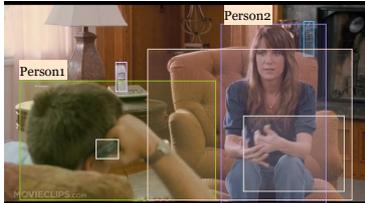


Question: How does Person2 feel about this performance?
 Answer: Person2 is slightly impressed by what she sees.

Ground Truth Explanation:
 Person2 has a slight smirk on her face.

Explanation Generation:
 Beam: Person2 has her arms crossed and a slight frown on her face.
 Nucleus: Person2 has her arms crossed and a slight frown on her face, which suggest she is not impressed.

Answer + Explanation Generation:
 Beam: Person2 does not like what she is hearing, this is because Person2 has her arms crossed and a frown on her face.
 Nucleus: Person2 is not impressed by it, this is because Person2 is looking down and frowning, and her arms are crossed, which is a sign of disapproval.



Question: What is Person2 doing?
 Answer: Telling Person1 something important.

Ground Truth Explanation:
 Person2 is focused on Person1 and waving their hands to emphasize what they're saying.

Explanation Generation:
 Beam: Person2 is sitting across from Person1 who is on the phone.
 Nucleus: Person2 is sitting in a chair and is looking at Person1 who is on the phone.

Answer + Explanation Generation:
 Beam: Person2 is listening to Person1 talk on the phone, this is because Person2 is sitting in a chair facing Person1 who is on the phone.
 Nucleus: Person2 is telling Person1 a story, this is because Person2 is sitting in a chair and Person1 is on the phone.



Question: Why does Person1 have a pointy hood on his cloak?
 Answer: Person1 is a wizard.

Ground Truth Explanation:
 Pointy hoods are usually used to depict a wizard character.

Explanation Generation:
 Beam: Wizards often wear pointy hoods.
 Nucleus: Wizards often wear pointy hoods and robes.

Answer + Explanation Generation:
 Beam: Person1 is trying to hide his identity, this is because Person1 is standing in front of Person2 and Person3, and Person4 who are all looking at him.
 Nucleus: Person1 is a wizard, this is because Wizards often wear pointy hoods and robes.



Question: Does Person1 drink alcohol?
 Answer: Yes Person1 drinks alcohol.

Ground Truth Explanation:
 Person1 has a full beer in front of him.

Explanation Generation:
 Beam: Person1 has a cup of beer in front of him.
 Nucleus: Person1 is drinking from Cup2.

Answer + Explanation Generation:
 Beam: Yes, Person1 drinks alcohol, this is because Person1 has a cup of beer in front of him.
 Nucleus: Yes, he does drink, this is because He has Cup2 in front of him and it is full of beer.

Figure 5: Examples of generated answers and explanations generation for VCR.



Question: Why is Person 3 wearing a life jacket?

Answer Options:
 0) The boat has a leak, and Person3 is scared of drowning.
 1) The boat is sinking and the life jacket will help them float.
 2) Person3 is piloting the ship.
 3) Person9 is wearing a life vest in case the ship sinks.

Generation: Person3 is on a boat.



Question: Why did Person1 drop Person3?

Answer: Person1 dropped Person 3 by accident.

Explanation Options:
 0) Person1 can upon Person3 in the woods, and kissed her; she awoke, and he dropped her off the bier.
 1) Person2 is Person3's mother. Person3 is an infant and can't walk on his own.
 2) Person3 is stuck in the toilet as Person1 is pulling her out.
 3) Person3 is bent over and appears unsteady. Person1 looks concerned for her.

Generation: Person1 is kneeling over the body of Person3.

Figure 6: Questions that require knowledge of the movie plots to generate the answers from VCR.



Question: Is this legal or illegal?

Ground Truth Answers:
 legal (6), illegal (4)

Generation: legal



Question: In which country are the transportation regulations loose enough to allow vehicles like these?

Ground Truth Answers:
 india (8), china (2)

Generation: england



Question: How long does it take to cook?

Ground Truth Answers:
 45 minutes (4), 20 minutes (2), 25 minutes (2), minute (2)

Generation: 1 hour



Question: What nationality is this food?

Ground Truth Answers:
 american (4), mediteranian (2), greek (2), asian (2)

Generation: italian

Figure 7: Examples of subjective questions from OK-VQA.



Question: What country headquarters this plane company?

Answer: usa

Ground Truth Explanations:
 0) The headquarters are the us.
 1) The company name is virgin atlantic that was founded and has headquarters in london england.
 2) The airplane has virgin atlantic livery. this company is based in england.



Question: How long does the average giraffe live?

Answer: 20-30 years

Ground Truth Explanations:
 0) Giraffes can live a long time.
 1) 20-30 years is the lifespan.
 2) I looked up this answer on the internet since there is no way to tell the answer from the picture.



Question: What is the brown house doing?

Answer: walking

Ground Truth Explanations:
 0) it has two legs up and two down and it is moving.
 1) only two feet are touching the ground.
 2) he is moving slowly on a mountain range.

Figure 8: Issues in the datasets that severely impact the model generation: wrong answers (left, from A-OKVQA), questions do not need visual input to answer (middle, from A-OKVQA), and typo (right, from VQA-X).

Machine Translation between Spoken Languages and Signed Languages Represented in SignWriting

Zifan Jiang
University of Zurich
jiang@cl.uzh.ch

Amit Moryossef
Bar-Ilan University
University of Zurich
amitmoryossef@gmail.com

Mathias Müller
University of Zurich
mmueller@cl.uzh.ch

Sarah Ebling
University of Zurich
ebling@cl.uzh.ch

Abstract

This paper presents work on novel machine translation (MT) systems between spoken and signed languages, where signed languages are represented in SignWriting, a sign language writing system. Our work¹ seeks to address the lack of out-of-the-box support for signed languages in current MT systems and is based on the SignBank dataset, which contains pairs of spoken language text and SignWriting content. We introduce novel methods to parse, factorize, decode, and evaluate SignWriting, leveraging ideas from neural factored MT. In a bilingual setup—translating from American Sign Language to (American) English—our method achieves over 30 BLEU, while in two multilingual setups—translating in both directions between spoken languages and signed languages—we achieve over 20 BLEU. We find that common MT techniques used to improve spoken language translation similarly affect the performance of sign language translation. These findings validate our use of an intermediate text representation for signed languages to include them in NLP research.

1 Introduction

Most current machine translation (MT) systems only support spoken language input and output (text or speech), which excludes around 200 different signed languages used by up to 70 million deaf people² worldwide from modern language technology. Since signed languages are also natural languages, Yin et al. (2021) calls for including sign language processing (SLP) in natural language processing (NLP) research.

From a technical point of view, SLP brings novel challenges to NLP due to the visual-gestural modality of sign language and special linguistic features

¹Code and documentation available at <https://github.com/J22Melody/signwriting-translation>

²According to the World Federation of the Deaf: <https://wfdeaf.org/our-work/>

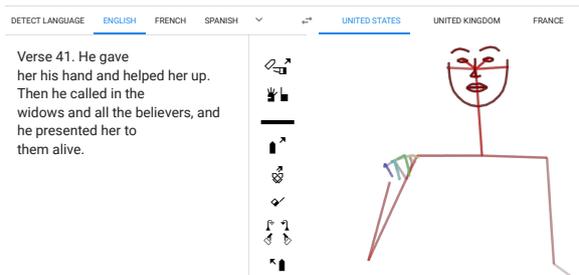


Figure 1: Demo application based on our models, translating from spoken languages to signed languages represented in SignWriting, then to human poses.

(e.g., the use of space, simultaneity, referencing), which requires both computer vision (CV) and NLP technologies. Crucially, the lack of a standardized or widely used written form for signed languages has hindered their inclusion in NLP research.

However, sign language writing systems do exist and are sporadically used (e.g., SignWriting (Sutton, 1990) and HamNoSys (Prillwitz and Zienert, 1990)). Therefore, we adopt the proposal of Yin et al. (2021) to formulate the sign language translation (SLT) task using a sign language writing system as an intermediate step (illustrated by Figure 1): given spoken language text, we propose to translate to sign language in a written form, then transform this intermediate result into a final video or pose output³—and vice versa. According to this multi-step view of SLT, in this work we study translation between signed languages in written form and spoken languages. We use *SignWriting* as the intermediate writing system.

SignWriting has many advantages, like being universal (multilingual), comparatively easy to understand, extensively documented, and computer-supported. In addition, despite looking pictographic, it is a well-defined writing system. Every

³Note that the second step, animation of SignWriting into human poses or video, is not included in this research. In the demo application, spoken language text is translated directly into sign language poses, resulting in low-quality output.

sign can be written as a sequence of symbols (box markers, graphemes, and punctuation marks) and their location on a 2-dimensional plane.

To our knowledge, this work is the first to create automatic SLT systems that use SignWriting. Our main contributions are as follows: (a) we propose methods to parse (§3.3), factorize (§3.4), decode (§4.3), and evaluate (§4.3) SignWriting sequences; (b) we report experiments on multilingual machine translation systems between SignWriting and spoken language text (§4); (c) we demonstrate that common techniques for low-resource MT are beneficial for SignWriting translation systems (§5).

2 Background

2.1 Sign language processing (SLP)

SLP (Bragg et al., 2019; Yin et al., 2021; Moryossef and Goldberg, 2021) is an emerging subfield of both NLP and CV, which focuses on automatic processing and analysis of sign language content. Prominent tasks include pose estimation from sign language videos (Cao et al., 2017, 2021; Güler et al., 2018), gloss transcription (Mesch and Wallin, 2012; Johnston and Beuzeville, 2016; Konrad et al., 2018), sign language detection (Borg and Camilleri, 2019; Moryossef et al., 2020), sign language identification (Gebre et al., 2013; Monteiro et al., 2016), and sign language segmentation (Bull et al., 2020; Farag and Brock, 2019; Santemiz et al., 2009).

Besides, tasks including sign language recognition (Adaloglou et al., 2021), translation, and production involve transforming one sign language representation to another or from/to spoken language text, as shown in Figure 2⁴. We find that existing works cover gloss-to-text (Camgöz et al., 2018; Yin and Read, 2020) (where “text” denotes spoken language text), text-to-gloss (Zhao et al., 2000; Othman and Jemni, 2012), video-to-text (Camgöz et al., 2020b,a), pose-to-text (Ko et al., 2019), and text-to-pose (Saunders et al., 2020a,b,c; Zelinka and Kanis, 2020; Xiao et al., 2020).

2.2 Motivation

Our work is the first to explore translation between spoken language text and sign language content represented in SignWriting⁵. We focus on a sign language writing system for the following reasons:

⁴In the paper, we distinguish between a phonetic “writing system” (e.g., SignWriting) and “glosses” (lexical notation, marking the semantics of each sign with a distinct category).

⁵Related work based on HamNoSys: Morrissey (2011); Sanaullah et al. (2021); Walsh et al. (2022)

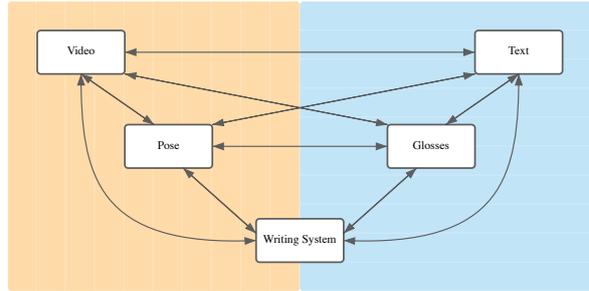


Figure 2: SLP tasks. Every edge on the left side represents a task in CV (language-agnostic). Every edge on the right side represents a task in NLP (language-specific). Every edge crossing both sides represents a task requiring a combination of CV and NLP. Figure taken from Moryossef and Goldberg (2021).

(a) currently an end-to-end (video-to-text/text-to-video) approach is not feasible. State-of-the-art systems either have a BLEU score lower than 1 (Müller et al., 2022a) or work only on a very narrow linguistic domain, e.g., Camgöz et al. (2020b,a) work on the RWTH-PHOENIX-Weather T data set which covers only 1,231 unique signs from weather reports (less than what we use in Table 2); (b) a writing system is lower-dimensional than videos (not all parts of a video are relevant in a linguistic sense), while adequate to encode information of signs; (c) written sign language is a closer fit to current MT pipelines than videos or poses; (d) a phonetic writing system is a more universal solution than glossing since glosses are semantic and therefore language-specific, and are an inadequate representation of meaning (Müller et al., 2022b).

2.3 SignWriting, FSW, and SWU

SignWriting (Sutton, 1990) is a featural and visually iconic sign language writing system (introduced extensively in Appendix A). Previous work explored recognition (Stiehl et al., 2015) and animation (Bouزيد and Jemni, 2013) of SignWriting.

SignWriting has two computerized specifications, Formal SignWriting in ASCII (FSW) and SignWriting in Unicode (SWU). SignWriting is two-dimensional, but FSW and SWU are written linearly, similar to spoken languages. Figure 3 gives an example of the relationship between SignWriting, FSW, and SWU⁶. We use FSW in our research instead of SWU to explore the potential of factorizing SignWriting symbols and utilizing numerical values of their position (§3.3, §3.4).

⁶Online demonstration: <https://slevinski.github.io/SuttonSignWriting/characters/index.html>.

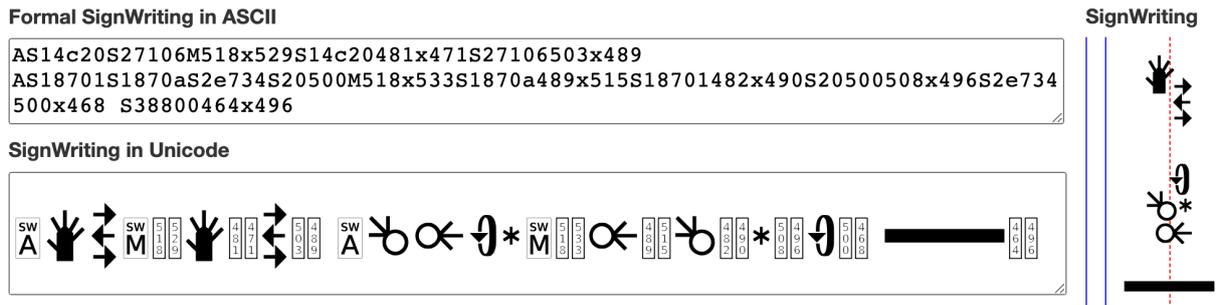


Figure 3: “Hello world.” in FSW, SWU and SignWriting graphics. In FSW/SWU, A/SWA and M/SWM are the box markers (acting as sign boundaries); S14c20 and S27106 (graphemes in SWU) are the symbols; 518 and 529 are the x, y positional numbers on a 2-dimensional plane that denote symbols’ position within a sign box, S38800 (horizontal bold line in SWU) is the punctuation full stop symbol.

3 Data and method

The data source we use for this research is SignBank, the largest repository of *SignPuddles*⁷. A SignPuddle is a community-driven dictionary where users add parallel examples of SignWriting and spoken language text (not necessarily with corresponding videos and glosses). The puddles contain material from various signed languages and linguistic domains (e.g., general literature or Bible) without a strict writing standard. We use the Sign Language Datasets (Moryossef and Müller, 2021) library to load SignBank as a Tensorflow Dataset.

3.1 Data statistics

In SignBank, there are roughly 220k parallel samples from 141 puddles covering 76 language pairs, yet the distribution is unbalanced (full details in Appendix C). Relatively high-resource language pairs (over 10k samples) are listed in Table 1.

Notably, most of the puddles are dictionaries, which we consider less valuable than sentence pairs (instances of continuous signing) for a general MT system. If dictionaries are used as training data, we expect models to memorize word mappings and not learn to generate sentences.

Therefore, we treat the four sentence-pair puddles (Table 2) of the relatively high-resource language pairs as primary data and the other dictionary puddles as auxiliary data. Note that even the language pairs constituting the high-resource pairs of SignBank are low-resource compared to datasets used in mature MT systems for spoken languages, where millions of parallel sentences are commonplace (Akhbardeh et al., 2021).

3.2 Data preprocessing

We first perform general data cleaning to extract the main body of spoken language text and remove irrelevant parts such as HTML tags or samples that are empty or too long (100 words for a dictionary entry). We then learn a byte pair encoding (BPE) segmentation (Sennrich et al., 2016) on the cleaned spoken language text, using a vocabulary size of 2,000.

Multilingual models In our multilingual experiments (§4.2, §4.3), we learn a shared BPE model across all spoken languages.

Following Johnson et al. (2017), we add special tags at the beginning of source sequences to indicate the desired target language and nature of the training data (sentence pair or dictionary). Three types of tags are designed to encode all necessary information: (a) spoken language code; (b) country code⁸; (c) dictionary vs. sentence pair. For example, an English sentence to be translated into American Sign Language is represented as the following:

<2en> <4us> <sent> Hello world.

Data split We shuffle the data and split it into 95%, 3%, and 2% for training, validation, and test sets, respectively.

3.3 FSW parsing

On the sign language side, an appropriate segmentation and tokenization strategy is needed for the FSW data. We parse an original FSW sequence (e.g. Figure 3) into several pieces:

- box markers: A, M, L, R, B;

⁸spoken language code plus country code specifies a one-to-one mapping to a related signed language in our data.

⁷<https://www.signbank.org/signpuddle/>

language pair	#samples	#puddles
en-us (American English & American Sign Language)	43,698	7
pt-br (Brazilian Portuguese & Brazilian Sign Language)	42,454	3
de-de (Standard German & German Sign Language)	24,704	3
fr-ca (Canadian French & Quebec Sign Language)	11,189	3

Table 1: Relatively high-resource language pair statistics.

puddle name	language pair	#samples	#signs	mean sequence len
Literature US	en-us	700	9,922	24
ASL Bible Books NLT	en-us	11,667	51,485	24
ASL Bible Books Shores Deaf Church	en-us	4,321	44,612	31
Literatura Brasil	pt-br	1,884	19,221	13

Table 2: Primary sentence-pair puddles. Mean sequence length is measured by the mean number of words in the spoken language sentences.

• symbols: S1f010, S18720, etc.;	Source	Target
• positional numbers x and y: 515, 483, etc.;	symbol	S1f010
• punctuation marks (special symbols without box markers): S38800, etc.	X	515
	Y	483
	relative X	0 → “Hi”
	relative Y	1
	symbol core	S1f0
	column	1
	row	0

We further factorize each symbol into several parts regarding its orientation (see Figure 7 in Appendix A for an explicit motivation of this step). For example, the symbol S1f010 is split into:

- symbol core: S1f0;
- column number (from 0 to 5): 1;
- row number (from 0 to hex F): 0.

For positional numbers, which have a large range (from 250 to 750) and are encoded discretely, we hypothesise that models might have difficulty understanding their relative order. Therefore, we further calculate two additional factors that denote a symbol’s relative position (based on the absolute numbers) within a sign: relative x and relative y, both ranging from 0 to #symbols - 1.

We provide a full example of the result of FSW parsing in Listing 1 in Appendix C.

3.4 Factored machine translation

We use a factored machine translation system (Koehn and Hoang, 2007; Garcia-Martinez et al., 2016) to encode or decode parsed FSW sequences. We argue that this architecture is suitable because

Figure 4: Representation of translating a FSW symbol together with its factors to English.

concatenating all parsed FSW tokens results in sequences much longer than the maximum length of many Transformer models (e.g., 512).

From another perspective, the essential information units are the symbols. Nevertheless, the positional numbers are necessary to determine how symbols are assembled. The same symbols can be arranged differently in space to convey different meanings.

In our setup, we treat the symbols (including punctuation marks and box markers) as the primary source/target tokens and the rest as source/target factors that are strictly aligned with each source/target token (illustrated by Figure 4).

Depending on the translation direction, factored FSW representations need to be encoded or decoded. For encoding (when FSW is the source), we embed each factor separately and then concatenate

them to the aligned symbol’s embedding. For decoding (when FSW is the target), we use only a subset of factors (absolute x and y) because others are irrelevant for prediction, and additional weighted cross-entropy losses are calculated.

4 Experiments and results

This section introduces three lines of experiments on both bilingual and multilingual SignWriting translation. We use Transformer models (Vaswani et al., 2017) that support source and target factors. See Appendix B for more details on our training configuration.

4.1 Initial exploration with a bilingual model

For a first exploration, we train a bilingual model that translates from American Sign Language (ASL) to English (en-us). The purpose of this experiment is (a) to demonstrate that automatic SignWriting translation is feasible and (b) to explore different strategies for data processing and hyperparameters.

We use roughly 40k parallel training samples comprising roughly 15k sentence pairs and 25k dictionary pairs. The quality of spoken language translation is measured by BLEU (Papineni et al., 2002) and chrF2 (Popović, 2015). Table 3 shows the evaluation results on the test set.

4.2 Multilingual sign-to-spoken translation

Here we extend our initial bilingual model to a multilingual setting, translating from multiple signed languages to multiple spoken languages. We define two data conditions:

- **high-resource:** using roughly 100k parallel training samples (roughly 17k sentence pairs and roughly 83k dictionary pairs covering four language pairs),
- **adding low-resource:** in addition to the high-resource data, use all additional language pairs in SignBank that have at least 1k parallel samples (most of which are dictionaries). The total number of training examples grows to roughly 170k, covering 21 language pairs (Table 7).

The exact factorization strategy and model hyperparameters are informed by our bilingual experiments reported in Table 3.

Evaluating dictionary entries For these multilingual models, many of the training samples are dictionary entries, and so are some test samples. To evaluate the translation quality for dictionary entries, we use top-n accuracy, which tests whether one of the top-n translation candidates from beam search matches the entry from the reference.

Table 4 shows the evaluation results on the test set.

4.3 Multilingual spoken-to-sign translation

Finally, we train multilingual models that translate in the reverse direction, from spoken languages to signed languages. The data and model configuration are the same as for the multilingual sign-to-spoken model under high-resource data condition.

FSW decoding strategies SignWriting utterances are parsed into a factored FSW representation (§3.3, §3.4) and are used for encoding successfully, yet it is not obvious how to best decode to FSW. We try the following strategies: (a) predicting everything (including positional numbers) as target tokens all in one long target sequence, inspired by Chen et al. (2022); (b) predicting symbols only (as a comparative experiment); (c) predicting symbols with positional numbers as target factors.

During decoding in the test phase, we apply beam search only for the main target token prediction. Target factors do not participate in beam search, i.e., each target factor prediction is the argmax of the corresponding output layer distribution. We shift target factors to the right by 1 to condition their prediction on the previously generated target symbol.

Evaluation of FSW output Due to variations of SignWriting symbols based on different orientations that do not change meaning (Figure 7), evaluating FSW output only at the token (symbol) level is not sufficient. Therefore, we evaluate the output symbols (e.g., line 4 in Listing 1) not only with BLEU, but also chrF2++, which captures both word-level and character-level statistics. Additionally, we evaluate the output positional numbers by mean absolute error (MAE) to measure the distance between predicted positional numbers and the ones from the FSW reference (e.g., lines 5 and 6 in Listing 1). Let x be the predicted sequence of positional numbers and y be the gold sequence:

$$MAE(x, y) = \frac{1}{|x|} \sum_{i=1}^{|x|} |x_i - y_i|$$

	model	BLEU	chrF2
E1	baseline (lowercase training and test data)	22.5	-
E2	E1 + dictionary data	25.2	-
E3	E2 + BPE	27.0	46.2
E4	E3 + x,y as factors	27.5	46.5
E5	E4 + symbol core as source + row, col as factors	23.1	41.2
E6	E4 + relative x, y as factors	28.1	47.5
E7	E6 + aggressive dropout + tied softmax	31.4	52.0
E8	E7 + symbol core, row, column as factors	32.0	52.7
E9	E8 + remove lowercasing	30.8	51.2
E10	E9 + smaller BPE vocab 2000 to 1000	29.5	50.8

Table 3: Translation quality of ASL→en-us bilingual models. Note that E1 to E8 are trained and evaluated with all spoken language data lowercased, while from E9 to all later experiments we remove the lowercasing, so we expect a little performance drop for the later experiments. We introduce chrF2 as an evaluation metric starting from E3.

language	metrics	4 language pairs (100k)	21 language pairs (170k)
en-us (40k)	BLEU	29.5	25.0
	chrF2	49.8	47.0
	top-1	0.37	0.33
	top-5	0.52	0.45
en-sg (1k)	top-1	-	0.20
	top-5	-	0.27
pt-br (40k)	BLEU	23.8	6.4
	chrF2	44.3	17.5
	top-1	0.12	0.09
	top-5	0.17	0.15
mt-mt (4k)	BLEU	-	10.1
	chrF2	-	29.8
	top-1	-	0.05
	top-5	-	0.05
de-de (20k)	top-1	0.22	0.15
	top-5	0.31	0.27
de-ch (4k)	top-1	-	0.04
	top-5	-	0.06
fr-ca (10k)	top-1	0.04	0.07
	top-5	0.08	0.10
fr-fr (1k)	top-1	-	0.16
	top-5	-	0.24
fr-ch (8k)	top-1	-	0.07
	top-5	-	0.09

Table 4: Translation quality of multilingual sign-to-spoken models (partial results on the most frequent languages). Languages without sentence pairs are only evaluated by top-n accuracy. Empty cells mean that a language pair is not supported by the model. In the parentheses are the rough numbers of samples.

	model	BLEU	chrF2++	MAE x	MAE y
E1	2symbol+numbers	6.6	23.1	-	-
E2	2symbol	25.6	44.2	-	-
E3	2symbol+factors (w=1)	19.9	39.1	46.5	52.6
E4	2symbol+factors (w=0.5)	21.9	40.8	46.8	52.7
E5	2symbol+factors (w=0.2)	22.9	42.0	47.4	53.0
E6	2symbol+factors (w=0.1)	22.0	41.7	46.4	52.2
E7	2symbol+factors (w=0.01)	21.0	40.9	48.4	58.3

Table 5: Translation quality of multilingual spoken-to-sign models. Evaluated in BLEU (on symbol, higher is better), chrF2++ (on symbol, higher is better), and MAE (on positional numbers, lower is better). w denotes the weight between each factor’s loss and the main target loss.

language	BLEU (on symbols)	chrF2++ (on symbols)
en-us (40k)	35.7	58.4
pt-br (40k)	1.9	14.9
de-de (20k)	17.3	43.2
fr-ca (10k)	5.3	19.1

Table 6: Translation quality of multilingual spoken-to-sign model ($w=0.1$) per language. In the parentheses are the rough numbers of samples per language.

where if the predicted and gold sequences do not have the same length, they are padded with zeros.

Table 5 shows the results of evaluation on the test set. Table 6 shows the results of multilingual evaluation on $E6$ ($w=0.1$) of Table 5.

5 Discussion

5.1 Effect of adding dictionaries, BPE, and low-resource optimizations

As shown in Table 3, enlarging the sentence-level training data (15k sentence pairs) with 25k dictionary pairs improves the translation quality by 2.7 BLEU ($E1$ vs. $E2$). Likewise, applying BPE segmentation to the spoken language side also improves translations by 1.8 BLEU ($E2$ vs. $E3$).

We evaluate several low-resource “tricks” (Senrich and Zhang, 2019) including aggressive dropout and weight tying⁹. These low-resource optimizations borrowed from spoken language MT prove to be effective for sign language translation as well, as they result in an improvement of 3.3 BLEU and 4.5 chrF2 ($E6$ vs. $E7$ in Table 3).

⁹The tying is only between the target embedding and the softmax output matrix since the source and target languages are of a very different nature and therefore cannot be tied.

5.2 Utilizing positional numbers

In earlier sections we introduce novel methods to parse and factorize FSW (§3.3, §3.4). However, from a model training perspective it is unclear how to best utilize additional factors such as positional numbers. In $E4$, $E5$, $E6$, and $E8$ of Table 3, we explore different ways of including factors.

We find that the best strategy is explicitly adding all additional information (x , y , relative x , relative y , symbol core, column number, row number) as source factors while keeping symbols as the primary source tokens. This strategy achieves the state-of-the-art performance of 32.0 BLEU and 52.7 chrF2 in $E8$.

5.3 Generating positional numbers

We explore different ways of generating positional numbers in Table 5. As a first attempt, we treat them as normal target tokens in $E1$, which results in poor performance and overly long target sequences, and long beam search decoding time.

In all subsequent experiments, we treat positional numbers as target factors and generally achieve over 20 BLEU (evaluating on symbols). In $E2$, we translate only the symbols as a baseline. Then we also try translating with target factors and varying the weights between factors and the pri-

mary target, i.e., symbols. Finally, we observe that $E6$ ($w=0.1$) leads to the best trade-off between generating symbols and positional numbers.

5.4 Multilingual performance

We discuss multilingual performance mainly based on Table 4. Generally speaking, the more resources a language has in the multilingual model, the better its performance (Zhou et al., 2021). The two target languages most frequent in the training data—American English (en-us, 40k) and Brazilian Portuguese (pt-br, 40k)—have the highest translation quality.

Multilingual transfer effects We observe examples of both positive and negative multilingual transfer. Evidence shows that a relatively high-resource language can help a related low-resource language. For instance, the performance of Singaporean English (en-sg, 1k) is likely improved by American English (en-us, 40k), which is almost as good as Standard German (de-de, 20k).

The comparable bilingual en-us model ($E9$ in Table 3) outperforms en-us in our multilingual sign-to-spoken model in Table 4 by 1.3 BLEU and 1.4 chrF2. However, when extending the training data from 4 to 21 language pairs, we observe severe degradations: en-us drops by 4.5 BLEU and 2.8 chrF2; Brazilian Portuguese (pt-br) drops by 17.4 BLEU and 26.8 chrF2.

Such findings are in line with previous work on highly multilingual translation systems. For example, Aharoni et al. (2019) finds that average per-language performance drops when the number of languages increases. We conclude that SignWriting translation suffers from a similar *curse of multilinguality*.

5.5 Side-by-side SignWriting example

Finally, to gain intuition for how well the translation model signs, we give a side-by-side example of SignWriting graphics. We compare the reference and model prediction of an ASL utterance corresponding to an American English utterance from the Bible corpus, shown in Figure 5.

We ask an ASL user proficient in SignWriting for a translation of the predicted SignWriting back to English to assess the quality of the prediction.

Similar patterns appear in both: in the beginning, the model signs “Verse 41” in the same way as in the reference; the graphics in the top parts of all the columns are consistent; and we see correct sym-

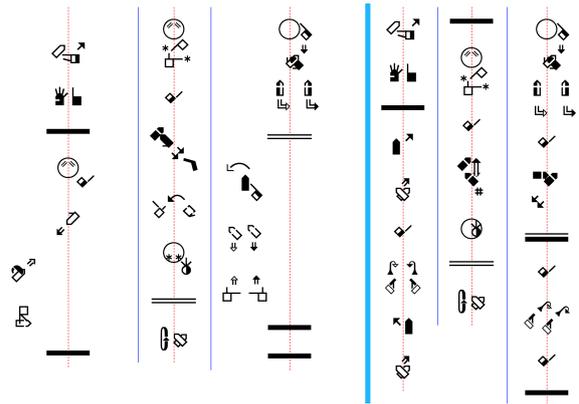


Figure 5: Side-by-side SignWriting example. ASL translation of the English sentence “Verse 41. He gave her his hand and helped her up. Then he called in the widows and all the believers, and he presented her to them alive.” Separated by the vertical bold (light blue) line, the left is the gold sentence, and the predicted sentence is on the right. The predicted sentence translated back to English is “Verse 41. His hand he gives her hand. Then he helped up, all believers he warned: he put there.”

bols sometimes predicted with slightly different positions.

More translation examples can be seen in Appendix D.

6 Conclusion

This work explores building bilingual and multilingual translation systems between spoken and signed languages. Instead of representing sign language as videos (or as continuous features derived from videos) common in previous research, we propose to represent sign language in SignWriting, a sign language writing system. We argue that using a written form is more amenable to well-established NLP techniques.

However, encoding or decoding SignWriting in an MT system requires specialized tools. Therefore, we propose novel methods to parse, factorize, decode, and evaluate SignWriting sequences. Our factorization technique divides SignWriting sequences into meaningful units such as sign symbols and positional numbers. The factors are then encoded or decoded by a factored Transformer model.

As a result, we achieve over 30 BLEU in the bilingual setting and over 20 BLEU for some high-resource language pairs in both directions in the multilingual setting.

Using SignWriting as an intermediate representation enables us to reuse tools (e.g., evaluation met-

rics) from spoken language translation. We also observe striking similarities to spoken language MT in the experiments themselves. For example, low-resource optimizations have a similar impact, and multilingual models exhibit similar transfer effects. These findings validate our use of an intermediate text representation for signed languages to include them in NLP research.

7 Limitations

7.1 A word on top-n accuracy

In the translation of dictionary data, if a dictionary entry has been seen during training, assuming the model has enough capacity, it should memorize and predict it. However, evaluating the translation is tricky, so in §4.2 we resort to using top-n accuracy.

Paradoxically, high top-n accuracy on the test set does not guarantee good generalization and might be associated with overlap between the training and test sets. Conversely, claiming a model is terrible when it performs poorly on the test set is unjustified, as there might be no overlap between the training and test sets. If a model has seen all the words from a language, it should perform well on whatever dictionary test set. However, this is not the case in our low-resource setting.

7.2 Fingerspelling tokenization

Fingerspelling (Battison, 1978; Wilcox, 1992; Brentari and Padden, 2001) is an interesting linguistic phenomenon where a signed language geographically coexists with a spoken language. For words with no associated signs (e.g., names of people, locations, organizations, etc.), sign language users borrow a word of a spoken language by spelling it letter-by-letter with predefined signs for the letters of the alphabet of that language. The fingerspelling (manual) alphabet of a sign language draws on a closed set of hand shapes, which are supported by SignWriting.

As fingerspelling is usually applied on a character level (rarely extending to the level of multiple characters, such as “CH” or “SCH” for the finger alphabet of Swiss German Sign Language), the way BPE segmentation works (on subword level) does not apply perfectly. However, if we could detect fingerspelling during the segmentation/tokenization process, then force fingerspelled words to be split letter-by-letter, our models should be able to learn better the mappings between fingerspelling signs and spoken language letters.

7.3 Towards better multilingual models

As shown by Table 2, the data we use to train our models only contains many sentence pairs for American English and American Sign Language. For other language pairs, we train mainly on dictionary data.

At the time of writing, we find a multilingual parallel corpus created from translations of the Bible¹⁰ (Christodoulopoulos and Steedman, 2014), which, if aligned correctly, can be used to translate the ~15k American Sign Language biblical text to another 100+ spoken languages. We believe we could train better multilingual translation models (at least on the spoken language side) based on them.

7.4 Regression objective for positional numbers

In our experiments, positional numbers are treated as target factors (§5.3), contributing cross-entropy loss to the training process. However, we are aware that the positional numbers are, by nature, numeric values, so a regression objective/loss would possibly work better than the current cross-entropy loss, as it better reflects the numeric relationship between positional numeric values.

As for now, the target factor function we use is only implemented with a classification objective (cross-entropy loss). We envision that custom implementation of the regression objective might improve translation quality in this scenario.

7.5 Possibly flawed positional number evaluation

We note that using MAE for evaluating positional numbers (§4.3) is possibly flawed because the predicted symbol sequences can deviate from the gold symbol sequences. If this is the case, making a token-by-token comparison on the positional numbers is meaningless, as even the sequence length can mismatch.

7.6 Advanced SignWriting evaluation

Finally, we call for advanced and novel methods of SignWriting evaluation, considering its differences from spoken languages.

In our experiments, we separate the evaluation of FSW symbols and positional numbers. For symbols, we borrow BLEU and chrF2/chrF2++ from spoken language evaluation since FSW symbols are the basic graphemes in SignWriting that show

¹⁰<https://github.com/christos-c/bible-corpus>

many similar linguistic features as spoken language words. For positional numbers, MAE is used, and its limitation is discussed in §7.5.

From a broader perspective, FSW is merely a linearized specification of SignWriting, which means we can also evaluate on the original graphical form, as we do manually in §5.5. Moreover, we can exploit CV techniques to do an automatic comparative evaluation between predicted SignWriting graphics and gold SignWriting graphics.

Ideally, a cascading evaluation method is applied to SignWriting: we first evaluate the overall graphics of the signs, then the symbols within the signs, then the position of the symbols, then the factorized representation of the symbols. Finally, a thorough human evaluation is needed to gain better insight.

Note on reproducibility

We will release the source code and documentation to train our models, an API server with the trained models, and a demo Web application. This will allow others to see and consistently reproduce our results with minimal changes. We encourage the community to attempt to reproduce our results and publish the results.

Acknowledgements

This work is funded by the following projects: EASIER (Grant agreement number 101016982) and ICT (Grant agreement number PFFS-21-47). We are grateful for their support. We also thank Rico Sennrich for his suggestions.

References

Nikolaos M. Adaloglou, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas, Georgios Th Papadopoulos, Vassia Zacharopoulou, George Xydopoulos, Klimis Antzakas, Dimitris Papazachariou, and Petros Daras. 2021. [A comprehensive study on deep learning-based methods for sign language recognition](#). *IEEE Transactions on Multimedia*, page 1–1.

Roe Aharoni, Melvin Johnson, and Orhan Firat. 2019. [Massively multilingual neural machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa,

Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khushabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vyrdrin, and Marcos Zampieri. 2021. [Findings of the 2021 conference on machine translation \(WMT21\)](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Robbin Battison. 1978. *Lexical borrowing in American sign language*. ERIC, Linstok Press, Inc., Silver Spring, Maryland 20901.

Mark Borg and Kenneth P Camilleri. 2019. [Sign language detection "in the wild" with recurrent neural networks](#). In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1637–1641. IEEE.

Yosra Bouzid and Mohamed Jemni. 2013. [An avatar based approach for automatically interpreting a sign language notation](#). In *2013 IEEE 13th International Conference on Advanced Learning Technologies*, pages 92–94.

Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, Christian Vogler, and Meredith Ringel Morris. 2019. [Sign language recognition, generation, and translation: An interdisciplinary perspective](#). In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS '19*, page 16–31. Association for Computing Machinery.

Diane Brentari and Carol Padden. 2001. A language with multiple origins: Native and foreign vocabulary in american sign language. *Foreign vocabulary in sign language: A cross-linguistic investigation of word formation*, pages 87–119.

Hannah Bull, Michèle Gouiffès, and Annelies Braffort. 2020. Automatic segmentation of sign language into subtitle-units. In *European Conference on Computer Vision*, pages 186–198. Springer.

Necati Cihan Camgöz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7784–7793.

Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020a. Multi-channel transformers for multi-articulatory sign language translation. In *European Conference on Computer Vision*, pages 301–319.

- Necati Cihan Camgöz, Oscar Koller, Simon Hadfield, and Richard Bowden. 2020b. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10023–10033.
- Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2021. [Openpose: Realtime multi-person 2d pose estimation using part affinity fields](#). *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):172–186.
- Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*.
- Stanley F. Chen and Joshua Goodman. 1996. [An empirical study of smoothing techniques for language modeling](#). In *34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318.
- Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. 2022. [Pix2seq: A language modeling framework for object detection](#). In *International Conference on Learning Representations*.
- Christos Christodoulopoulos and Mark Steedman. 2014. [A massively parallel corpus: the bible in 100 languages](#). *Language Resources and Evaluation*, 49:1–21.
- Iva Farag and Heike Brock. 2019. Learning motion disfluencies for automatic sign language segmentation. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7360–7364. IEEE.
- Mercedes Garcia-Martinez, Loïc Barrault, and Fethi Bougares. 2016. [Factored Neural Machine Translation Architectures](#). In *International Workshop on Spoken Language Translation (IWSLT’16)*.
- Binyam Gebrekidan Gebre, Peter Wittenburg, and Tom Heskes. 2013. Automatic sign language identification. In *2013 IEEE International Conference on Image Processing*, pages 2626–2630. IEEE.
- Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306.
- Felix Hieber, Tobias Domhan, Michael Denkowski, and David Vilar. 2020. [Sockeye 2: A toolkit for neural machine translation](#). In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 457–458. European Association for Machine Translation.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. [Google’s multilingual neural machine translation system: Enabling zero-shot translation](#). *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Trevor Johnston and Louise De Beuzeville. 2016. [Auslan corpus annotation guidelines](#). *Auslan Corpus*.
- Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. 2019. Neural sign language translation based on human keypoint estimation. *Applied Sciences*, 9(13):2683.
- Philipp Koehn and Hieu Hoang. 2007. [Factored translation models](#). In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876. Association for Computational Linguistics.
- Reiner Konrad, Thomas Hanke, Gabriele Langer, Susanne König, Lutz König, Rie Nishio, and Anja Regen. 2018. [Public dgs corpus: Annotation conventions](#). Technical report, Hamburg University.
- Julia Kreutzer, Jasmijn Bastings, and Stefan Riezler. 2019. [Joey NMT: A minimalist NMT toolkit for novices](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 109–114. Association for Computational Linguistics.
- Johanna Mesch and Lars Wallin. 2012. From meaning to signs and back: Lexicography and the swedish sign language corpus. In *Proceedings of the 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon [Language Resources and Evaluation Conference (LREC)]*, pages 123–126.
- Caio DD Monteiro, Christy Maria Mathew, Ricardo Gutierrez-Osuna, and Frank Shipman. 2016. Detecting and identifying sign languages through visual features. In *2016 IEEE International Symposium on Multimedia (ISM)*, pages 287–290. IEEE.
- Sara Morrissey. 2011. [Assessing three representation methods for sign language machine translation and evaluation](#). In *Proceedings of the 15th annual meeting of the European Association for Machine Translation (EAMT 2011), Leuven, Belgium*, pages 137–144.
- Amit Moryossef and Yoav Goldberg. 2021. Sign Language Processing. <https://sign-language-processing.github.io/>.
- Amit Moryossef and Mathias Müller. 2021. Sign language datasets. <https://github.com/sign-language-processing/datasets>.
- Amit Moryossef, Ioannis Tsochantaridis, Roei Yosef Aharoni, Sarah Ebling, and Srini Narayanan. 2020. Real-time sign-language detection using human

- pose estimation. In *SLRTP 2020: The Sign Language Recognition, Translation & Production Workshop*.
- Mathias Müller, Sarah Ebling, Eleftherios Avramidis, Alessia Battisti, Michèle Berger, Richard Bowden, Annelies Braffort, Necati Cihan Camgöz, Cristina España-Bonet, Roman Grundkiewicz, Zifan Jiang, Oscar Koller, Amit Moryossef, Regula Perrollaz, Sabine Reinhard, Annette Rios, Dimitar Shterionov, Sandra Sidler-Miserez, Katja Tissi, and Davy Van Landuyt. 2022a. Findings of the WMT 2022 shared task on sign language translation. In *Proceedings of the Seventh Conference on Machine Translation*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Mathias Müller, Zifan Jiang, Amit Moryossef, Annette Rios, and Sarah Ebling. 2022b. Considerations for meaningful sign language machine translation based on glosses. *arXiv preprint arXiv:2211.15464*.
- Achraf Othman and Mohamed Jemni. 2012. English-asl gloss parallel corpus 2012: Aslg-pc12. In *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon LREC*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. **Bleu: a method for automatic evaluation of machine translation**. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318.
- Maja Popović. 2015. **chrF: character n-gram F-score for automatic MT evaluation**. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395. Association for Computational Linguistics.
- Siegmund Prillwitz and Heiko Zienert. 1990. Hamburg notation system for sign language: Development of a sign writing with computer application. In *Current trends in European Sign Language Research. Proceedings of the 3rd European Congress on Sign Language Research*, pages 355–379.
- Muhammad Sanaullah, Babar Ahmad, Muhammad Kashif, Tauqeer Safdar, Mehdi Hassan, Mohd Hilmi Hasan, and NorShakirah Aziz. 2021. **A real-time automatic translation of text to sign language**. *Computers, Materials and Continua*, 70:2471–2488.
- Pinar Santemiz, Oya Aran, Murat Saraclar, and Lale Akarun. 2009. Automatic sign segmentation from continuous signing via multiple sequence alignment. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2001–2008. IEEE.
- Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020a. Adversarial training for multi-channel sign language production. In *The 31st British Machine Vision Virtual Conference*. British Machine Vision Association.
- Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020b. Everybody sign now: Translating spoken language to photo realistic sign language video. *arXiv preprint arXiv:2011.09846*.
- Ben Saunders, Necati Cihan Camgöz, and Richard Bowden. 2020c. Progressive transformers for end-to-end sign language production. In *European Conference on Computer Vision*, pages 687–705.
- Rico Sennrich. 2012. **Perplexity minimization for translation model domain adaptation in statistical machine translation**. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. **Neural machine translation of rare words with subword units**. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.
- Rico Sennrich and Biao Zhang. 2019. **Revisiting low-resource neural machine translation: A case study**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221.
- Steve Slevinski. 2021. **Formal SignWriting**. Internet-Draft draft-slevinski-formal-signwriting-08, Internet Engineering Task Force.
- D. Stiehl, L. Addams, L. S. Oliveira, C. Guimarães, and A. S. Britto. 2015. **Towards a signwriting recognition system**. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 26–30.
- Valerie Sutton. 1990. *Lessons in sign writing*. SignWriting.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. **Attention is all you need**. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Harry Walsh, Ben Saunders, and Richard Bowden. 2022. Changing the representation: Examining language representation for neural sign language production. In *LREC 2022*.
- Sherman Wilcox. 1992. *The phonetics of fingerspelling*, volume 4. John Benjamins Publishing.
- Qinkun Xiao, Minying Qin, and Yuting Yin. 2020. Skeleton-based chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. *Neural Networks*, 125:41–55.
- Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. **Including signed languages in natural language processing**. In *Proceedings of the 59th Annual Meeting of the*

- Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7347–7360, Online. Association for Computational Linguistics.
- Kayo Yin and Jesse Read. 2020. Better sign language translation with stmc-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989.
- Jan Zelinka and Jakub Kanis. 2020. Neural sign language synthesis: Words are our glosses. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3395–3403.
- Liwei Zhao, Karin Kipper, William Schuler, Christian Vogler, Norman Badler, and Martha Palmer. 2000. A machine translation system from english to american sign language. In *Conference of the Association for Machine Translation in the Americas*, pages 54–67. Springer.
- Chunting Zhou, Daniel Levy, Xian Li, Marjan Ghazvininejad, and Graham Neubig. 2021. [Distributionally robust multilingual machine translation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5664–5674. Association for Computational Linguistics.

A Extended introduction to SignWriting

SignWriting (Sutton, 1990) is a sign language writing system developed by Valerie Sutton¹¹ and currently managed by Steve Slevinski¹². SignWriting is very featural and visually iconic, both in:

- the shapes of the symbols, which are abstract pictures of hand shapes (Figure 6), orientation (Figure 7), body locations, facial expressions, contacts, and movement;
- the symbols’ two-dimensional spatial arrangement in an invisible “sign box” (Figure 8).

Outside each sign, the script is written linearly to reflect the temporal order of signs. Signs are mostly written vertically, arranged from top to bottom within each column, interspersed with special punctuation symbols (horizontal lines), and the columns progress left to right across the page. Within each column, signs may be vertically aligned to the center or shifted left or right to indicate body shifts.

A.1 Formal SignWriting in ASCII (FSW)

In 2012, Formal SignWriting in ASCII (FSW) specification (Slevinski, 2021) was released and documented in an Internet Draft submitted to the IETF.

The design of FSW is computerized so that it can be recognized and processed by programs. While signed languages are natural languages, FSW is a formal language handy in mathematics, computer science, and linguistics.

Although SignWriting is two-dimensional, FSW is written linearly like spoken languages. Each sign is written as first a box marker, then a sequence of symbols, and their relative position, as illustrated by Figure 3.

A.2 SignWriting in Unicode (SWU)

In 2017, SignWriting in Unicode (SWU) specification (Slevinski, 2021) was released, making SignWriting included in the Unicode Standard. The Unicode block for SWU is U+1D800 - U+1DAAF.

As illustrated in Figure 3, SWU is also written linearly. FSW and SWU are isomorphic and interchangeable, and both faithfully encode the complete information of SignWriting.



Figure 6: Hand shapes and their equivalents in SignWriting.

S100	00	10	20	30	40	50
00	□	□	□	□	□	□
01	◇	◇	◇	◇	◇	◇
02	▭	▭	▭	▭	▭	▭
03	◊	◊	◊	◊	◊	◊
04	▮	▮	▮	▮	▮	▮
05	◇	◇	◇	◇	◇	◇
06	▮	▮	▮	▮	▮	▮
07	◇	◇	◇	◇	◇	◇
08	▮	▮	▮	▮	▮	▮
09	◇	◇	◇	◇	◇	◇
0a	▮	▮	▮	▮	▮	▮
0b	◇	◇	◇	◇	◇	◇

Figure 7: Orientation of a symbol in SignWriting in 3D space. Each row applies a rotation of the palm in a 2D space **vertical** to the ground. Each column applies a rotation of the palm in a 2D space **parallel** to the ground. This can be seen as a factorization of the symbol S100xx to its core S100 plus row and column numbers.

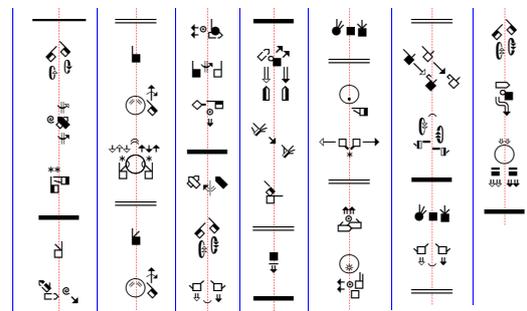


Figure 8: An example of SignWriting written in columns, ASL translation of an introduction to Formal SignWriting in ASCII. The relative positions of the symbols within the box iconically represent the locations of the hands and other parts of the body involved in the sign being represented.

¹¹Valerie Sutton: https://en.wikipedia.org/wiki/Valerie_Sutton

¹²Steve Slevinski: <https://steveslevinski.me/>

B Experimental setup

We performed all our experiments with Python 3.8.11 on an Nvidia Tesla V100 GPU (32GB GPU ram).

B.1 40k sign-to-en-us

Data includes:

- ~15k sentence pairs from 3 en-us puddles: Literature US, ASL Bible Books NLT, ASL Bible Books Shores Deaf Church,
- ~25k dictionary pairs from 3 en-us puddles: Dictionary US, LLCN & SignTyp, ASL Bible Dictionary,

which leads to:

- ~6k source vocabulary size (number of non-factorized symbols),
- ~2k target vocabulary size (determined by BPE).

Final model configuration:

- 6 layers + 8 heads + 512 embedding size (16 for each factor) (0.5 dropout) + 512 hidden size (0.5 dropout) + 2,048 feed forward size (0.5 dropout),
- initial learning rate 0.0001, decrease learning rate by a factor of 0.7 every 5 times validation score (BLEU) not improved,
- batch size 32 sentences, label smoothing 0.2, epochs 300,
- for testing, decoding with a checkpoint with the best validation score, beam size 5, alpha for length penalty 1.

Experiments were conducted with a custom version of Joey NMT (Kreutzer et al., 2019) to support source factors. Each model (~47 million parameters) finished training within 1 day.

B.2 100k sign-to-spoken

Data includes:

- ~17k sentence pairs from 3 en-us puddles (Literature US, ASL Bible Books NLT, ASL Bible Books Shores Deaf Church) and 1 pt-br puddle (Literatura Brasil),

- ~83k dictionary pairs from 3 en-us puddles (Dictionary US, LLCN & SignTyp, ASL Bible Dictionary), 2 pt-br puddles (Dicionário Brasil, Enciclopédia Brasil), 1 de-de puddle (Wörterbuch DE) and 1 fr-ca puddle (Dictionnaire Quebec),

which leads to:

- ~11k source vocabulary size (number of non-factorized symbols),
- ~2k target vocabulary size (determined by BPE).

A little change to the previous configuration to make training more efficient:

- batch size 4,096 tokens.

Experiments were conducted with a custom version of Joey NMT to support source factors. Each model (~50 million parameters) finished training within ~1.5 days and ~3 days, respectively.

B.3 100k spoken-to-sign

Data and model configurations are the same as **100k sign-to-spoken**, except that we use perplexity (Chen and Goodman, 1996; Sennrich, 2012) as validation score instead of BLEU.

Experiments were conducted with Sockeye (Hieber et al., 2020) for the convenience of ready-to-use target factor support. Each model (~60 million parameters) finished training within ~0.5 day.

C Data

Figure 9 visualizes the language pair distribution in SignBank. Table 7 contains an exhaustive list of all 21 language pairs used in this research. Listing 1 shows an example of FSW parsing and factorization.

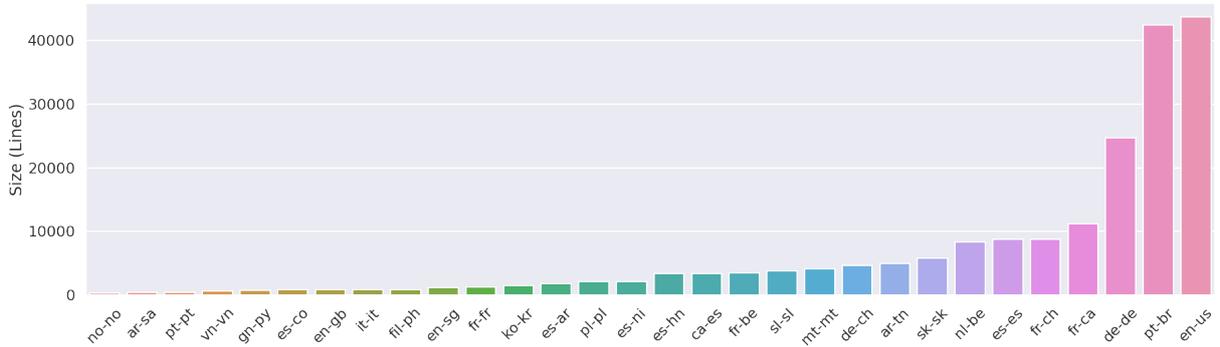


Figure 9: Data distribution (the first 30 language pairs).

language	#samples	#puddles	sentence pairs (>1k)
en-us (American English)	43,698	7	✓
en-sg (Singaporean English)	1,136	2	
pt-br (Brazilian Portuguese)	42,454	3	✓
mt-mt (Maltese Maltese)	4,118	4	✓
de-de (German German)	24,704	3	
de-ch (Swiss German)	4,700	2	
fr-ca (Canadian French)	11,189	3	
fr-ch (Swiss French)	8,806	3	
fr-be (Belgian French)	3,439	1	
fr-fr (French French)	1,299	2	
es-es (Spanish Spanish)	8,806	2	
es-hn (Honduran Spanish)	3,399	1	
es-ni (Nicaraguan Spanish)	2,150	2	
es-ar (Argentinian Spanish)	1,774	2	
ar-tn (Tunisien Arabic)	4,965	2	
ca-es (Spanish Catalan)	3,419	2	
ko-kr (Korean Korean)	1,525	1	
nl-be (Belgian Flemish)	8,301	2	
pl-pl (Polish Polish)	2,130	2	
sk-sk (Czech Czech)	5,780	2	
sl-sl (Slovenian Slovenian)	3,808	2	

Table 7: All 21 language pairs (spoken languages with corresponding signed languages).

```

1 {
2   'fsw': 'M550x535S32a00482x483S15d09455x499S15d01522x497S22114516x484
3     S22114456x484S20f00524x522S20f00451x523 ',
4   'symbol': 'M S32a00 S15d09 S15d01 S22114 S22114 S20f00 S20f00',
5   'feat_x': '550 482 455 522 516 456 524 451',
6   'feat_y': '535 483 499 497 484 484 522 523',
7   'feat_x_rel': '-1 3 1 5 4 2 6 0',
8   'feat_y_rel': '-1 0 4 3 1 2 5 6',
9   'feat_core': 'M S32a S15d S15d S221 S221 S20f S20f',
10  'feat_col': '-1 0 0 0 1 1 0 0',
11  'feat_row': '-1 0 9 1 4 4 0 0',
12 }

```

Listing 1: An example of FSW parsing and factorization.

D More side-by-side SignWriting examples

Separated by the vertical bold (light blue) line, the left is the gold sentence, and the predicted sentence is on the right.

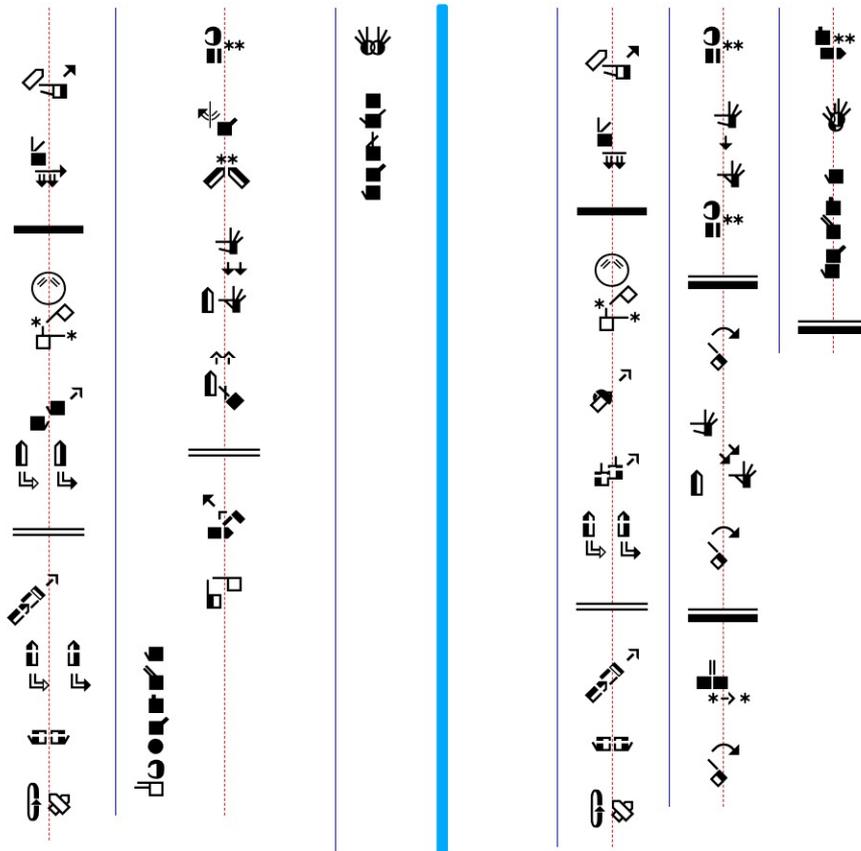


Figure 10: ASL translation of the English sentence “Verse 22. Then the apostles and elders together with the whole church in Jerusalem chose delegates, and they sent them to Antioch of Syria”

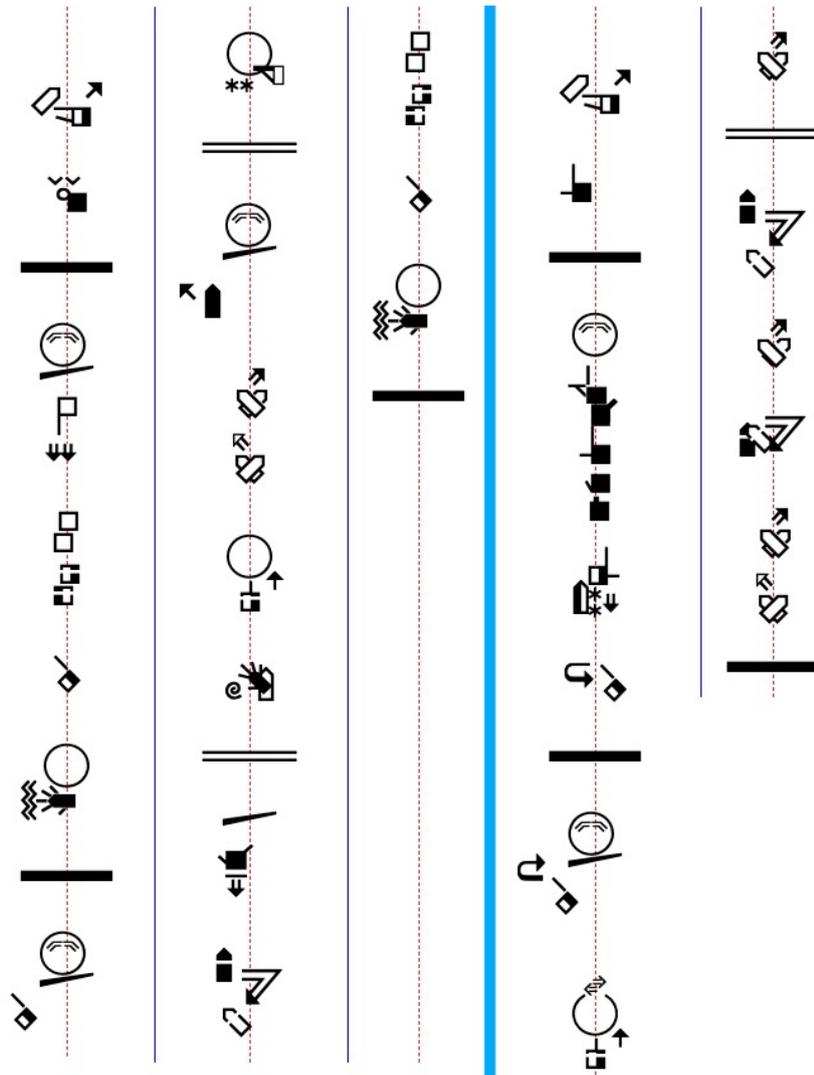


Figure 11: ASL translation of the English sentence “These are what defile you. Eating with unwashed hands will never defile you.”

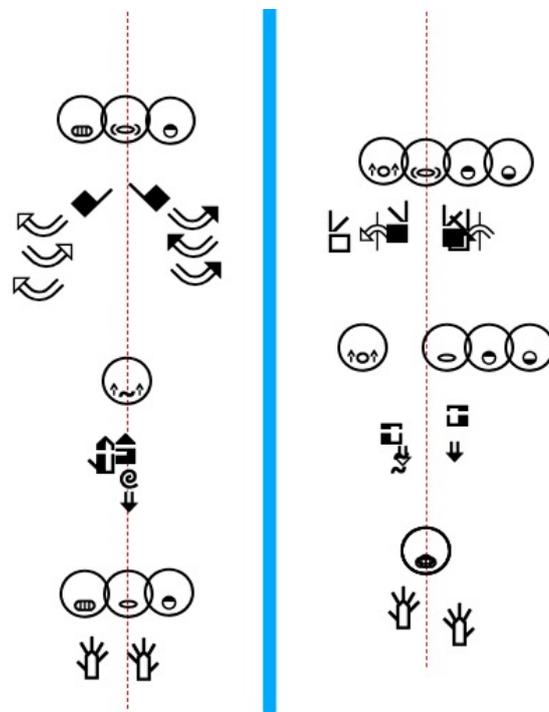


Figure 12: German Sign Language translation of the German words “signen, senken, zehn”

A Multi-dimensional Evaluation of Tokenizer-free Multilingual Pretrained Models

Jimin Sun^{1,2} Patrick Fernandes¹ Xinyi Wang¹ Graham Neubig¹

¹Language Technologies Institute, Carnegie Mellon University ²Kakao Enterprise
{jimins2, pfernand, xinyiw1, gneubig}@cs.cmu.edu

Abstract

Recent works on tokenizer-free multilingual pretrained models show promising results in improving cross-lingual transfer and reducing engineering overhead compared to subword-based alternatives. However, previous work mainly focuses on reporting accuracy on a limited set of tasks and data settings, placing less emphasis on other important factors when tuning and deploying the models in practice, such as memory usage, inference speed, and finetuning data efficiency. We attempt to fill this gap by performing a comprehensive empirical comparison of multilingual tokenizer-free and subword-based models considering the various dimensions. Surprisingly, we find that subword-based models might still be the most practical choice in many settings, achieving better performance for lower inference latency and memory usage. Based on these results, we encourage future work in tokenizer-free methods to consider these factors when designing and evaluating new models.¹

1 Introduction

Several recent results (Clark et al., 2022; Xue et al., 2022) have excited the research community with the possibility of “tokenizer-free” models, character-level and byte-level models, as an alternative to more traditional subword-based models. Tokenizer-free models are especially appealing to practitioners as they can eschew the two-step processing pipeline of subword segmentation and reduce the corresponding difficulties in cross-lingual transfer (Hu et al., 2020; Maronikolakis et al., 2021; Rust et al., 2021; Wang et al., 2021) or domain adaptation (Sato et al., 2020; Liu et al., 2021) due to inconsistent subword units.

However, upon several attempts to apply tokenizer-free methods, our analysis reveals several practical difficulties in applying these methods.

¹We will release code to train and evaluate models upon de-anonymization.

This paper is a chronicle of some of the concerns we uncovered; we highlight some challenges with applying these models and propose best practices for future results reporting in this area.

Specifically, we perform experiments finetuning pretrained multilingual models, evaluating them with respect to finetuning data efficiency, inference time, and memory consumption. Based on these multiple dimensions, we come to the somewhat surprising conclusion that subword-based models, in particular mBERT (Devlin et al., 2019), might still be the most practical choice in most settings, as they perform best while maintaining a relatively low inference cost.

2 Tokenizer-free Multilingual Models

While multilingual pretrained models (Devlin et al., 2019; Lample and Conneau, 2019; Liu et al., 2020; Xue et al., 2021) have led to impressive performance improvements for low-resource languages through cross-lingual transfer, the standard word representation method in these models relies on subword segmentation (Sennrich et al., 2016; Kudo, 2018). In multilingual settings, subword tokenization can be sub-optimal as supporting hundreds of languages with various scripts and vocabularies causes segmentation mismatch between languages and over-segmentation in the lower-resourced languages (Wang et al., 2020).

To alleviate this problem, recent works propose removing the subword segmentation step by using characters or bytes as lexical units (Clark et al., 2022; Xue et al., 2022). In particular, these “tokenizer-free” methods have been applied to both encoder-only and encoder-decoder models. Tab. 1 presents an overview of the different tokenizer-free multilingual models with comparable subword models. Next, we briefly describe the two tokenizer-free models we consider in this work.

CANINE (Clark et al., 2022) is a character-level

Model	Params	Vocab (%)	Non-vocab	Architecture	Enc.	Dec.	Tokenization	↓sample?	Corpus	Langs
mBERT	178M	92M (52%)	86M	Enc-only	12	-	Subword	✗	Wikipedia	104
CANINE	132M	25M (19%)	107M	Enc-only	12	-	Character	✓	Wikipedia	104
mT5 (Small)	300M	256M (85%)	44M	Enc-dec	8	8	Subword	✗	mC4	101
ByT5 (Small)	300M	1.1M (0.3%)	298.5M	Enc-dec	12	4	UTF-8 bytes	✗	mC4	101

Table 1: Configuration of the pretrained models used. From left to right: number of parameters, number and ratio of vocabulary-related parameters, number of non-vocabulary parameters, architecture, encoder / decoder depth, tokenization scheme, whether downsampling was used, pretrained corpus, number of pretrained languages.

encoder suggested as an alternative to mBERT (Devlin et al., 2019). CANINE operates on raw characters and is pretrained using the masked language modeling objective. To compensate for the computational efficiency loss due to increased sequence length, CANINE uses convolutions to downsample the sequence before passing the representations to the transformer layers. The two weight variants of CANINE (CANINE-S, CANINE-C) have the same architecture but slightly different pretraining objectives using either subwords or characters at the last layer. As both variants performed similarly in our experiments and Clark et al. (2022), we only include CANINE-S for the main discussion, leaving CANINE-C results in § B.3.

ByT5 (Xue et al., 2022) is an encoder-decoder transformer model similar to the mT5 (Xue et al., 2021) model. ByT5 operates on the raw UTF-8 bytes of the input without any downsampling, leading to a longer sequence length while having a much smaller vocabulary size than mT5. Both ByT5 and mT5 are pretrained on the mC4 corpus² using the span reconstruction objective proposed by Raffel et al. (2020).

To keep the parameter count fixed between mT5 and ByT5, ByT5 allocates the parameters saved from the embedding layer to additional encoder layers. Although adding more depth to the encoder is a reasonable design choice, our results in § 4 show that ByT5 suffers from a much higher inference cost due to the deeper encoder, especially when input/output sequence lengths are longer.

3 Experimental settings

We conduct a multi-dimensional evaluation focusing on two aspects: finetuning data efficiency (§ 4.1) and inference cost (§ 4.2) to provide a better understanding of the practical applicability of tokenizer-free models. We finetune and evaluate

²<https://www.tensorflow.org/datasets/catalog/c4#c4multilingual>

two subword-based models (mBERT, mT5) and two tokenizer-free models (CANINE, ByT5), as mBERT-CANINE and mT5-ByT5 are directly comparable counterparts in terms of their pretraining corpus as shown in Tab. 1. For the T5 models, we consider only the small models of both mT5 and ByT5 as the focus of our work is in the practical implication of using multilingual pretrained models at relatively resource-constrained settings.

Specifically, we finetune the models on three multilingual natural language understanding tasks adopted from the XTREME benchmark (Hu et al., 2020). The three tasks we choose cover various input, output formats – sequence-level classification (XNLI), token-level classification (NER), and extractive question answering (TyDi QA-GoldP).

3.1 Tasks

XNLI The Cross-lingual Natural Language Inference (Conneau et al., 2018) is a sequence classification task in which the model predicts whether the hypothesis sentence is an entailment, contradiction, or neutral given the premise sentence. The task is provided in 15 languages.

NER Named Entity Recognition (NER) is a structured prediction task, where the model predicts a tag (location, person, organization) in IOB2 format for each token in the input sentence. We use the WikiAnn dataset (Pan et al., 2017) and select 20 out of 282 languages for multilingual training based on linguistic diversity and the language availability in the other two tasks we consider.

TyDi QA-GoldP The Typologically Diverse Question Answering (Clark et al., 2020) dataset is an extractive QA benchmark in 11 languages. While the original dataset includes two “primary” tasks (SelectP, MinSpan), the secondary GoldP task is the most widely adopted as it is compatible with other SQuAD-style QA tasks (Rajpurkar et al., 2016; Artetxe et al., 2020). For this reason, we mainly compare models on TyDi QA-GoldP

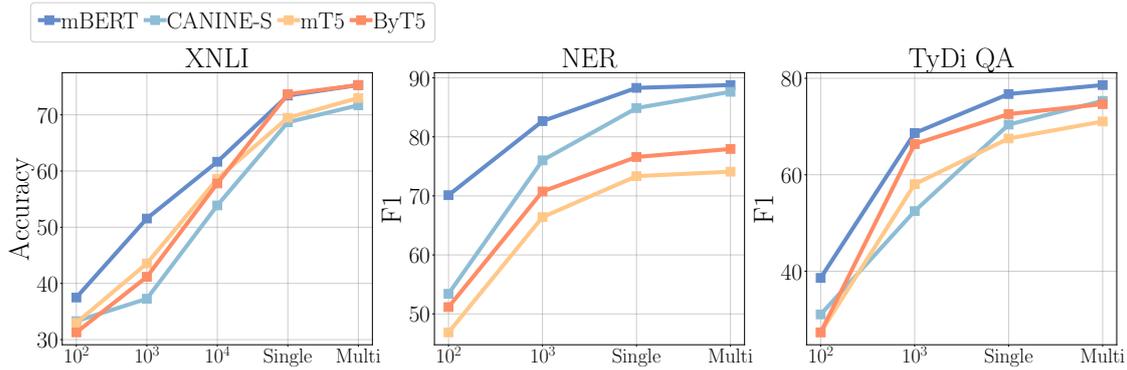


Figure 1: Average XNLI, NER, TyDi performance when each pretrained model is finetuned with varying numbers of in-language finetuning data (10^2 , 10^3 , 10^4), all in-language samples (Single), or the entire multilingual dataset (Multi). The exact numbers can be found in the Appendix (Tab. 2).

and discuss primary task results briefly through our replication experiment of Clark et al. (2022).

3.2 Details of Hardware and Measurements

We use a single Tesla V100 (32GB) GPU for all experiments regarding inference cost measurements. To obtain the peak GPU memory and inference latency, we randomly select 100 samples from the English test set for each task and measure the average cost of predicting one example at a time.

4 A Multi-dimensional Evaluation

4.1 Finetuning data efficiency

Most work presenting multilingual pretrained models evaluates downstream task performance under multilingual finetuning or zero-shot scenarios. In practice, however, downstream task datasets are often available in the language of interest. Thus, in addition to multilingual training, we compare models tuned on different data sizes *within* a single language to evaluate their finetuning data efficiency.

Specifically, we finetune the four pretrained models with varying numbers of task examples – 10^2 , 10^3 , 10^4 (when available), all target language samples (Single), and multilingual training (Multi) to incorporate situations where the task dataset is available in multiple languages. We experiment with four downstream task languages – English, Arabic, Russian, and Swahili – chosen based on both linguistic diversity and various pretraining resource conditions.³ While the controlled experiments are done on a subset of languages, we report the task performance in all languages for zero-shot evaluation, single language training, and multilin-

gual training in § B.3 for comprehensiveness.⁴

In Fig. 1, we report the models’ task performance averaged over languages under different finetuning settings. Notably, we find that mBERT achieves the highest score for most settings. The only exception is on XNLI Single and Multi, where ByT5 slightly outperforms mBERT. As the dataset size decreases, it becomes more evident that mBERT is the most sample efficient, especially in the most data-scarce scenarios where only 100 finetuning examples are available. The fact that mBERT outperforms mT5 and ByT5 on smaller datasets is quite surprising, as one might expect T5 models to generalize better in low-resource settings given their much larger pretraining corpus.

Interestingly, we find that CANINE performs poorly compared to mBERT in all three tasks, and the performance gap increases as fewer finetuning data are available. To explain this phenomenon, we hypothesize that character-level models have the additional burden of learning to compose characters into semantically meaningful units and thus require more data to learn task-specific higher-level semantics. These results align with the NER results on the CoNLL and MasakhaNER dataset in Clark et al. (2022), where mBERT outperformed CANINE in all languages except Amharic, a language not covered by mBERT’s vocabulary.

However, mBERT’s stronger performance in TyDi QA-GoldP was unexpected as CANINE performed better at the TyDi QA primary tasks in Clark et al. (2022). Through replication experiments to reconcile the contradictory findings, we found that mBERT outperforms CANINE also in the primary tasks when finetuned for more epochs with our codebase, suggesting that the previous

³The pretraining corpus sizes are noted in § B.4 (Tab. 8).

⁴Hyperparameters for all experiments are in Appendix A.

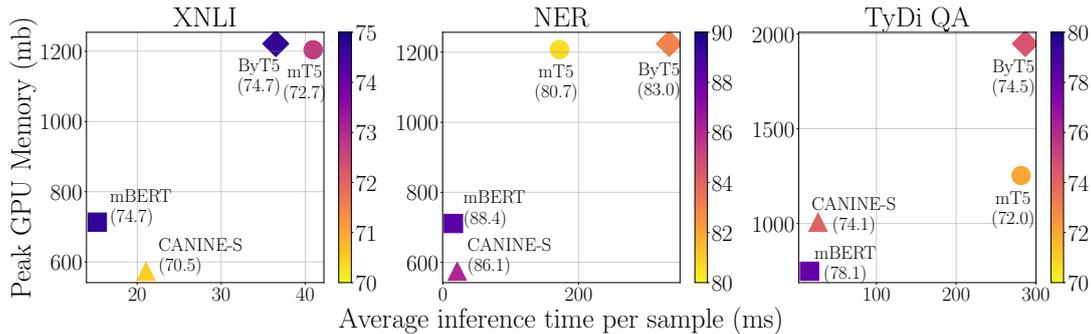


Figure 2: The inference cost of the four models (■: mBERT, ▲: CANINE, ●: mT5, ◆: ByT5) in each task. The x-axis denotes the average inference time while the y-axis shows the peak GPU memory consumption. Thus, models located near the bottom left corner are more cost efficient. The colors represent the model’s best task performance (XNLI: Accuracy, NER: F1, TyDi QA: F1). The numbers used to generate the plot can be found in § B.2 (Tab. 3).

mBERT baseline was potentially undertrained.⁵

For mT5 and ByT5, we find that the two models perform comparably in smaller datasets, while on larger sets, ByT5 consistently outperforms mT5 on all tasks. We note that the mT5-Small model could have been penalized in terms of capacity as 85% of the parameters are allocated to embeddings as shown in Tab. 1, leaving only 44M parameters for the non-vocabulary layers. This is even less than that of mBERT (86M), and drastically smaller compared to ByT5-Small, which assigns 298.5M parameters to the non-vocabulary layers. Also, given that the tasks concerned are not generation-heavy, the extra depth on the encoder (12 for ByT5 vs. 8 for mT5) might have favored ByT5 over mT5.

4.2 Inference cost

Another key concern in utilizing pretrained models for downstream applications is the inference cost, such as memory consumption and latency. In Fig. 2, we plot each model’s inference latency and peak memory consumption, color-coding their task performance to provide a comprehensive view of the trade-offs of deploying each model in practice.

In general, the encoder-only models, mBERT and CANINE, require much less memory and inference latency than mT5 and ByT5. Considering performance alongside inference cost, we find that mBERT is still the most practical choice among the four models, achieving the best performance while maintaining a relatively low inference cost.

While producing longer sequences than mBERT, CANINE does not necessarily incur higher memory or latency costs, as it has fewer parameters than mBERT. This helps CANINE, especially in sentence-level tasks (XNLI, NER) where inputs are

relatively shorter. However, for tasks with much longer inputs (TyDi QA), the computational overhead from the sequence length dominates the parameter reduction, leading to higher memory usage and slower inference for CANINE.

For mT5 and ByT5, inference costs vary according to the task’s input and output length. For tasks with shorter inputs and outputs like XNLI, ByT5 yields better performance than mT5 while retaining similar costs. However, for token-level prediction tasks like NER, ByT5 needs to generate tags autoregressively at the byte level, which drastically slows down the inference time. However, the additional cost is negligible in terms of memory consumption as the inputs are still relatively short. For TyDi QA, we observe an opposite pattern. As the input is a long passage, the extended input sequence significantly increases the memory consumption of ByT5, requiring more effort in tuning the batch size to fit into the GPU memory.

5 Related work

Large-scale NLP models have achieved remarkable performance in various natural language tasks, with the recent ChatGPT demonstrating near human-level language understanding capabilities. While achieving impressive results in standard benchmark settings, the applicability of these models have remained limited mainly due to practical considerations including their high energy consumption and environmental impact (Strubell et al., 2019). Both the NLP and computer vision communities have proposed evaluating models based on practical metrics, such as training/inference efficiency (Canziani et al., 2016; Dehghani et al., 2021; Zhou et al., 2021), energy usage (Henderson et al., 2020), robustness (Ribeiro et al., 2020; Kiela et al.,

⁵We include the finetuning code in our released codebase.

2021; Koh et al., 2021), and expected performance (Dodge et al., 2019). Similarly, a recent study by Liang et al. (2022) suggests a comprehensive evaluation suite for generative NLP models, including measures of robustness, fairness, and efficiency. Our multi-dimensional evaluation is an attempt to expand these evaluation protocols to *multilingual* settings and examine the trade-offs of various tokenization schemes.

6 Conclusion

In this paper, we present a multi-dimensional evaluation of tokenizer-free multilingual models focusing on their efficiency against finetuning dataset size and inference cost. Based on our experiments, we find that mBERT might still be the most cost-effective choice for many tasks, and show that the efficiency trade-offs of model design choices (tokenization, decoder availability) depend heavily on the task’s length statistics. Despite our findings, tokenizer-free models still have a significant advantage in reducing engineering effort and potentially increasing robustness to noisy data. We believe more work should be done in developing *efficient* tokenizer-free models, and encourage the community to consider these criteria of practical applicability when developing and evaluating tokenizer-free pretrained models.

7 Limitations

This paper mainly covers three NLP tasks, focusing on smaller-sized multilingual pretrained models. In future work, it would be interesting to run the multi-dimensional evaluation we suggest on a broader set of tasks and models. Although our results show that subword models are a more practical choice in some tasks, we note that other tasks or datasets may exist where tokenizer-free methods achieve better relative performance. For instance, tokenizer-free models have been reported to excel in word-level tasks, and noisy environments (Xue et al., 2022), and the conclusions we reached may be different in such settings. Moreover, we did not explore more complicated generation tasks like translation or summarization, where the difficulty in decoding and longer decode horizons could paint a different picture in a multi-dimensional evaluation.

Ethics Statement

We hope our results encourage the community to consider the practical concerns of running large lan-

guage models (LLMs) and designing tokenizer-free pretrained models. As the state-of-the-art LLMs are becoming more computationally extensive, it has become increasingly difficult for researchers and practitioners with less resources to utilize these models for downstream applications. We hope our multi-dimensional analysis can help researchers and practitioners with less computational resources decide which model to use in practice.

Acknowledgements

We acknowledge Kakao Enterprise for providing the compute resources for this work. We would like to thank Sanket Vaibhav Mehta, Daniel Fried, Saujas Vaduguru, and the anonymous reviewers for their valuable comments and feedback. Additionally, we would like to thank Jon Clark for answering questions related to the CANINE model. This work was supported in part by grant #2040926 from the National Science Foundation as well as the CMU-Portugal MAIA project.

References

- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *ACL*.
- Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. 2016. An analysis of deep neural network models for practical applications.
- Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an Efficient Tokenization-Free Encoder for Language Representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.
- Jonathan H. Clark, Jennimaria Palomaki, Vitaly Nikolaev, Eunsol Choi, Dan Garrette, Michael Collins, and Tom Kwiatkowski. 2020. Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Trans. Assoc. Comput. Linguistics*, 8:454–470.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2475–2485. Association for Computational Linguistics.
- Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. 2021. The efficiency misnomer. *CoRR*, abs/2110.12894.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North*, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China. Association for Computational Linguistics.
- Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *J. Mach. Learn. Res.*, 21(248):1–43.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.
- Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanus Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR.
- Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.
- Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *NeurIPS*.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D Manning, Christopher Ré, Diana Acosta-Navas, Drew A Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models.
- Xin Liu, Baosong Yang, Dayiheng Liu, Haibo Zhang, Weihua Luo, Min Zhang, Haiying Zhang, and Jin-song Su. 2021. Bridging subword gaps in pretrain-finetune paradigm for natural language generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6001–6011, Online. Association for Computational Linguistics.

- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Antonis Maronikolakis, Philipp Dufter, and Hinrich Schütze. 2021. [Wine is not v i n. on the compatibility of tokenizations across languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2382–2399, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. [Cross-lingual name tagging and linking for 282 languages](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1946–1958. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. [Beyond accuracy: Behavioral testing of NLP models with CheckList](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.
- Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. [How good is your tokenizer? on the monolingual performance of multilingual language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.
- Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. 2020. [Vocabulary adaptation for domain adaptation in neural machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4269–4279, Online. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and policy considerations for deep learning in NLP](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.
- Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021. [Multi-view subword regularization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482, Online. Association for Computational Linguistics.
- Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. [Extending multilingual BERT to low-resource languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.
- Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. [ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models](#). *Transactions of the Association for Computational Linguistics*, 10:291–306.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mt5: A massively multilingual pre-trained text-to-text transformer](#). In *NAACL*.
- Xiyou Zhou, Zhiyu Chen, Xiaoyong Jin, and William Yang Wang. 2021. [HULK: An energy efficiency benchmark platform for responsible natural language processing](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 329–336, Online. Association for Computational Linguistics.

A Tasks

For all tasks and models, we refer to the original papers' codebase for hyperparameters.⁶⁷⁸

XNLI For encoder-only models, the first token ([CLS]) is used to map the sentence representation to the label distribution. For encoder-decoder models, we generate the index of the label (e.g., '0') directly.

NER For encoder-decoder models, we follow the input-output format (e.g., input: 'tag: rick and morty are cool .', output: 'PER: rick \$\$ PER: morty') specified in the mT5 model's original codebase.

B Tables

B.1 Finetuning data efficiency

Tab. 2

B.2 Inference cost

Tab. 3

B.3 Experimental results for all languages (Zero-shot, Single language (full), Multilingual)

XNLI: Tab. 4, NER: Tab. 5, TyDi QA-GoldP: Tab. 6, Tydi QA Primary: Tab. 7

B.4 Pretraining corpus size

Tab. 8

⁶<https://github.com/google-research/language/tree/master/language/canine>

⁷<https://github.com/google-research/multilingual-t5>

⁸<https://github.com/google-research/byt5>

Finetuning setting	XNLI (Accuracy)					NER (F1)				TYDI QA (F1)			
	10 ²	10 ³	10 ⁴	Single	Multi	10 ²	10 ³	Single	Multi	10 ²	10 ³	Single	Multi
Arabic													
mBERT	36.6	51.0	59.5	70.6	73.2	67.3	80.2	89.6	89.6	44.8	70.9	81.0	81.5
CANINE-S	32.8	36.6	53.3	65.8	69.7	46.2	71.2	84.9	88.0	38.4	59.8	79.2	80.5
CANINE-C	34.1	45.3	50.5	66.2	68.5	51.7	71.3	85.1	87.8	34.8	57.8	77.8	80.7
mT5	33.1	44.2	55.0	65.5	70.3	57.3	75.5	86.5	86.8	33.7	62.6	73.1	75.3
ByT5	23.7	42.0	55.2	72.9	73.3	60.6	77.5	85.4	87.7	33.4	67.3	75.8	75.9
English													
mBERT	38.8	58.6	71.2	82.0	83.5	65.1	78.1	84.2	85.4	32.4	67.6	73.6	76.0
CANINE-S	33.6	37.5	59.5	77.7	79.1	49.7	70.3	80.4	84.1	29.2	49.4	64.0	71.6
CANINE-C	34.1	50.7	61.2	77.1	78.0	52.8	70.6	81.1	84.1	27.5	47.8	57.3	71.6
mT5	33.3	50.9	66.4	79.0	79.9	40.1	63.1	71.9	72.5	25.0	52.8	59.4	64.4
ByT5	35.2	39.6	66.2	80.9	81.0	44.1	65.0	73.8	73.5	16.5	63.1	64.6	69.4
Russian													
mBERT	35.7	45.5	52.9	66.3	68.1	81.3	89.9	90.0	90.9	42.9	74.3	79.8	82.4
CANINE-S	33.1	35.9	48.6	61.5	65.0	63.2	86.9	87.7	89.6	29.1	54.0	71.3	77.4
CANINE-C	33.1	42.9	45.4	60.8	64.4	70.0	86.5	86.5	90.0	32.3	58.9	71.4	79.7
mT5	33.0	44.9	58.0	63.2	68.1	54.3	70.6	71.0	72.3	29.0	65.8	71.5	76.6
ByT5	34.3	41.2	56.4	67.5	71.3	68.6	83.5	84.5	84.3	32.5	73.5	78.8	80.3
Swahili													
mBERT	38.9	51.1	63.0	74.8	76.4	66.7	82.4	89.4	89.2	34.3	61.8	72.5	74.4
CANINE-S	33.7	39.2	54.2	69.7	73.0	54.5	75.6	86.5	88.8	27.4	46.6	67.2	71.8
CANINE-C	35.0	46.5	54.1	68.6	71.7	55.4	76.0	87.3	88.9	20.0	46.9	66.5	72.7
mT5	32.6	34.2	55.1	70.3	73.7	35.8	56.4	64.0	64.8	21.6	51.0	66.1	67.9
ByT5	32.0	42.0	53.4	73.4	75.6	31.4	57.0	62.6	66.3	26.9	61.6	71.1	73.0

Table 2: Task performance with varying finetuning data conditions (10², 10³, 10⁴ (for XNLI), full target language dataset, multilingual dataset)

	XNLI			NER			TYDI QA		
	Latency	Memory	Accuracy	Latency	Memory	F1	Latency	Memory	F1
mBERT	15.24	713.33	74.7	15.30	710.97	88.4	16.10	748.34	78.13
CANINE-S	21.04	573.48	70.5	20.96	574.57	86.1	26.89	1006.74	74.13
mT5	40.94	1204.19	72.7	171.99	1207.76	80.7	281.52	1253.13	72.05
ByT5	36.49	1221.54	74.7	333.72	1224.40	83.0	286.76	1948.30	74.48

Table 3: Inference latency (ms), peak GPU memory (mb), best average performance of each model in the three tasks

Model	en	ar	bg	de	el	es	fr	hi	ru	sw	th	tr	ur	vi	zh	avg
Zero-shot (en)																
mBERT	82.0	64.1	67.5	70.4	65.5	73.7	72.8	59.3	67.4	50.2	53.2	60.2	57.5	68.7	68.1	65.4
CANINE-S	77.7	50.1	60.1	62.4	53.7	67.6	66.0	43.7	60.7	40.4	39.6	47.9	41.1	53.1	43.2	53.8
CANINE-C	77.1	53.1	61.4	63.5	58.3	68.5	66.4	47.7	63.3	41.0	39.2	48.8	44.4	53.4	39.1	55.0
mT5-Small	79.0	61.3	66.0	64.4	67.4	65.9	62.4	59.7	66.6	52.2	64.1	57.9	56.4	57.3	63.9	63.0
ByT5-Small	80.9	65.9	70.2	71.2	67.7	76.5	75.0	58.6	67.9	62.4	58.4	63.6	55.6	69.5	64.9	67.2
Single-language																
mBERT	82.0	70.6	76.2	76.6	75.1	77.7	77.4	67.0	74.8	66.3	65.7	72.5	62.9	75.9	76.4	73.1
CANINE-S	77.7	65.8	70.6	72.4	68.6	73.8	73.4	61.2	69.7	61.5	59.9	66.6	58.0	67.4	57.2	66.9
CANINE-C	77.1	66.2	71.1	72.0	69.8	72.8	72.6	62.3	68.6	60.8	57.1	65.7	58.2	67.3	60.0	66.8
mT5-Small	79.0	65.4	69.9	72.0	73.6	73.1	74.8	65.2	70.3	63.2	69.7	67.6	58.9	69.2	71.0	69.5
ByT5-Small	80.9	72.9	75.4	75.8	75.1	77.7	76.4	68.3	73.4	67.5	70.0	72.6	63.0	72.7	72.5	73.0
Multilingual																
mBERT	83.5	73.2	77.7	77.5	75.7	79.8	78.6	70.1	76.4	68.1	67.2	73.8	64.4	76.5	77.9	74.7
CANINE-S	79.1	69.7	75.0	74.9	72.5	76.3	75.3	65.2	73.0	65.0	62.3	68.9	64.1	71.3	65.6	70.5
CANINE-C	78.0	68.5	73.7	74.1	72.9	75.7	74.9	63.8	71.7	64.4	57.7	67.9	62.6	69.7	58.7	69.0
mT5-Small	79.9	70.3	74.7	74.9	74.4	76.5	75.5	67.7	73.7	68.1	71.2	71.9	65.4	72.4	73.2	72.7
ByT5-Small	81.0	73.3	77.8	76.5	76.5	78.5	77.2	70.0	75.6	71.3	71.4	73.6	68.3	75.7	74.1	74.7

Table 4: XNLI Performance (Accuracy)

Model	en	ar	bn	de	el	es	fi	fr	hi	id	ja	ko	ru	sw	ta	te	th	tr	ur	zh	avg
Zero-shot (en)																					
mBERT	84.2	41.7	68.2	78.2	71.4	71.8	77.3	78.0	64.5	51.6	29.2	59.7	65.6	71.4	51.0	50.4	0.4	73.9	33.3	43.1	58.2
CANINE-S	80.8	29.6	49.6	70.7	63.5	66.4	66.7	74.1	41.1	47.3	0.5	29.3	57.7	59.8	28.4	19.7	0.1	55.8	22.0	5.4	43.4
CANINE-C	81.1	38.3	56.9	70.9	66.4	64.8	68.0	73.5	43.4	46.6	1.8	28.7	61.7	58.9	36.9	21.6	0.2	58.9	29.8	8.1	45.8
mT5-Small	71.9	32.9	56.6	67.1	42.3	70.0	65.1	75.3	56.2	45.3	25.5	23.9	36.9	49.0	38.0	35.9	3.6	58.7	58.7	31.3	47.2
ByT5-Small	73.8	45.9	61.5	70.7	67.7	79.4	67.1	77.4	57.1	46.2	31.3	26.2	46.7	60.2	31.9	27.9	9.6	23.3	1.3	32.8	46.9
Single-language																					
mBERT	84.2	89.6	96.1	90.3	91.4	92.5	92.2	91.2	93.6	74.4	88.8	89.4	90.0	86.5	80.4	76.2	93.2	95.7	83.1	88.5	
CANINE-S	80.8	84.9	92.9	88.0	88.6	89.7	89.1	88.9	84.9	90.9	63.3	81.6	86.5	87.7	81.0	49.9	70.5	90.9	91.0	73.2	82.7
CANINE-C	81.1	85.1	93.5	87.5	89.1	89.8	88.4	88.4	84.3	90.6	60.2	79.5	87.3	86.5	79.6	43.0	74.0	90.6	92.4	68.9	82.0
mT5-Small	71.9	86.5	86.6	83.7	83.8	88.0	87.8	86.7	85.5	85.3	65.9	80.2	64.0	71.0	82.6	74.5	64.6	86.3	93.0	75.1	80.1
ByT5-Small	73.8	85.3	88.3	82.4	87.6	86.6	86.4	84.7	83.0	84.5	69.9	83.2	62.6	84.5	80.3	69.1	74.5	83.4	90.5	73.2	80.7
Multilingual																					
mBERT	85.4	89.6	95.9	89.8	91.3	92.9	92.0	91.2	89.3	93.4	74.9	88.1	89.2	90.9	86.0	80.6	76.5	93.1	95.5	82.3	88.4
CANINE-S	84.1	88.0	94.7	89.3	90.7	92.1	91.1	90.9	85.8	92.8	69.3	83.8	88.8	89.6	81.7	71.3	76.2	92.4	94.0	75.7	86.1
CANINE-C	84.1	87.8	95.6	89.2	91.1	92.5	90.7	90.9	88.2	92.6	67.9	81.5	88.9	90.0	81.6	69.5	77.7	92.0	93.7	72.1	85.9
mT5-Small	72.5	86.8	84.5	84.8	83.4	88.7	88.3	87.7	83.6	87.2	70.1	83.1	64.8	72.3	82.3	69.8	67.8	86.9	92.4	76.5	80.7
ByT5-Small	73.5	87.7	88.4	86.1	88.7	90.3	89.9	89.3	84.7	87.3	70.3	83.8	66.3	84.3	81.8	78.0	72.6	88.6	92.6	76.5	83.0

Table 5: NER Performance (F1)

Model	en	ar	bn	fi	id	ko	ru	sw	te	avg
Zero-shot (en)										
mBERT	73.64	60.11	45.1	57.63	63.78	52.16	57.52	56.51	42.15	56.51
CANINE-S	64.78	44.85	20.13	39.73	43.78	13.67	44.49	30.64	31.59	37.07
CANINE-C	63.96	42.19	22.05	43.13	36.87	17.44	42.02	33.3	30.51	36.83
mT5-Small	59.39	43.25	22.51	44.27	48.7	22.05	44.85	33.08	28.77	38.54
ByT5-Small	64.58	56.4	15.86	51.91	55.85	22.21	54.11	35.44	31.43	43.09
Single-language										
mBERT	73.64	79.86	70.78	76.08	79.93	62.76	72.48	79.81	81.21	75.17
CANINE-S	64.78	79.2	55.81	70.13	70.0	49.53	67.15	71.26	81.75	67.73
CANINE-C	63.96	77.79	50.92	67.28	66.26	49.84	66.49	71.39	82.78	66.3
mT5-Small	59.39	73.07	67.92	65.33	73.65	54.93	66.13	71.49	80.93	68.09
ByT5-Small	64.58	75.82	69.91	71.98	80.55	58.65	71.09	78.81	85.39	72.97
Multilingual										
mBERT	76.02	81.49	72.86	80.41	84.87	67.09	74.45	82.42	83.52	78.13
CANINE-S	71.55	80.53	67.24	75.42	78.44	61.25	71.75	77.43	83.53	74.13
CANINE-C	71.56	80.74	62.6	74.21	76.28	65.79	72.66	79.71	84.43	74.22
mT5-Small	64.39	75.34	76.89	70.01	76.73	59.24	67.86	76.62	81.35	72.05
ByT5-Small	69.42	75.86	70.9	74.52	79.78	60.62	73.01	80.32	85.93	74.48

Table 6: TyDi QA-GoldP Performance (F1)

Model	en	ar	bn	fi	id	ja	sw	ko	ru	te	th	avg
MINSPAN												
mBERT	65.1	83.1	66.7	69.0	65.8	53.0	71.7	62.8	66.4	87.1	64.5	69.0
CANINE-S	61.4	83.2	64.7	66.6	63.9	49.5	67.8	56.7	63.0	82.5	61.0	65.9
CANINE-C	58.8	82.6	58.7	64.7	64.3	50.8	65.1	56.2	64.4	83.9	61.5	65.2
SELECTP												
mBERT	51.1	73.6	56.6	59.0	56.8	43.6	64.7	48.2	50.8	83.1	53.4	59.0
CANINE-S	49.2	71.5	56.4	58.3	54.6	41.5	60.1	40.5	49.3	77.2	50.7	56.0
CANINE-C	47.4	71.0	46.5	53.8	54.4	40.2	56.0	34.0	48.8	78.0	49.1	53.2

Table 7: TyDi QA Primary Task Performance (F1)

Language	Wikipedia (Number of docs)	mC4 (Number of examples)
English	2.5M	3B
Russian	319K	756M
Arabic	77K	53M
Swahili	7K	985K

Table 8: Pretraining corpus sizes for languages used in § 4.1 experiments. The number of Wikipedia documents per language can be found here: https://en.wikipedia.org/wiki/Wikipedia:Multilingual_statistics

Neural Ranking with Weak Supervision for Open-Domain Question Answering : A Survey

Xiaoyu Shen^{†1}, Svitlana Vakulenko¹, Marco del Tredici¹, Gianni Barlacchi¹
Bill Byrne^{1,2} and Adrià de Gispert¹

¹Amazon Alexa AI

²Univeristy of Cambridge

[†]gyouu@amazon.com

Abstract

Neural ranking (NR) has become a key component for open-domain question-answering in order to access external knowledge. However, training a good NR model requires substantial amounts of relevance annotations, which is very costly to scale. To address this, a growing body of research works have been proposed to reduce the annotation cost by training the NR model with weak supervision (WS) instead. These works differ in what resources they require and employ a diverse set of WS signals to train the model. Understanding such differences is crucial for choosing the right WS technique. To facilitate this understanding, we provide a structured overview of standard WS signals used for training a NR model. Based on their required resources, we divide them into three main categories: (1) only documents are needed; (2) documents and questions are needed; and (3) documents and question-answer pairs are needed. For every WS signal, we review its general idea and choices. Promising directions are outlined for future research.

1 Introduction

Open-Domain Question Answering (ODQA) aims to provide precise answers in response to the user’s questions by drawing on a large collection of documents (Voorhees et al., 1999). The majority of modern ODQA models follow the retrieve(-rerank)-read architecture: 1) given a question, a set of relevant documents are selected from a large document collection, and 2) the reader model produces an answer given this selected set and the question (Chen et al., 2017; Verga et al., 2021; Lee et al., 2021). Compared with parametric models without access to external knowledge, this architecture can better adapt to updated knowledge, offer easier interpretation and reduce hallucination (Zhu et al., 2021a; Shuster et al., 2021; Guo et al., 2022).

Conventional methods use sparse retrievers (SRs) such as TF-IDF and BM-25 in the first stage

Resource	Weak-Supervision Signal
Documents (§3)	Self Contrastive Learning (§3.1)
	Question Generation (§3.2)
Documents + Questions (§4)	Sparse Retriever (§4)
	Pre-trained Language Model (§4)
Documents + QA Pairs (§5)	Supervised Teacher Model (§4)
	Answer as Document (§5.1)
	Answer-Document Mapping (§5.2)
	Latent-Variable Model (§5.3)

Table 1: Overview of different weak-supervision signals, together with their required resources, that we can apply to train NR models for open-domain question-answering.

to match questions and documents via lexical overlap (Robertson and Walker, 1994), a process that may overlook semantically relevant documents with low lexical overlap with the question. Neural ranking (NR) models resolve this issue by encoding the questions and documents into dense vectors so that synonyms and paraphrases can be mapped to similar vectors through task-specific fine-tuning (Das et al., 2019; Karpukhin et al., 2020). However, training good NR models requires substantial amount of relevance annotations to perform competitively and NR models have been found to generalize poorly across domains (Thakur et al., 2021; Ren et al., 2022). In practice, collecting question-document relevance annotations is time-consuming. For a given question, an extensive annotation effort may be required to find the relevant documents. Repeating this annotation for every language and domain is not feasible (Shen et al., 2022c).

To reduce annotation costs, many techniques have been proposed to train NR models with *weak supervision (WS) signals* instead. This survey aims to provide a clear taxonomy to characterize these WS signals based on their required resources. There are three common resources that can be leveraged: (1) **Document** set, which is a bare minimum for building a ODQA system; (2) **Questions** without ground-truth relevance or answer annotations;

(3) **Question-Answer (QA) Pairs**, which are a set of already answered questions. Different applications have different levels of resource availability. For example, common domains such as e-commerce normally already have large amounts of question-answer pairs from customer services while smaller domains or low-resource languages can only have a document set without any existing questions. For each level of resource availability, we review the applicable WS signals. An overview can be seen in Table 1.

While there have been surveys that describe general neural information retrieval (IR) approaches (Mitra et al., 2018; Guo et al., 2020; Lin et al., 2021a; Zhu et al., 2021a; Guo et al., 2022), we focus specifically on the low-resource scenarios, which makes our contribution unique in this respect. The closest to our work is the BEIR benchmark for zero-shot cross-domain evaluation of IR models (Thakur et al., 2021) and its multiple related studies (Mokrii et al., 2021; Reddy et al., 2021; Wang et al., 2022; Ren et al., 2022). Nonetheless, these studies test specific algorithms but do not provide a holistic overview of how they are related. Our survey can be useful in that: (1) future WS research can use it as a reference book to compare with similar techniques. (2) It can serve as a practical guide for choosing the best WS signals to train NR models given different availability levels of resources. (3) As the retrieve(-rerank)-read paradigm is generic and has been increasingly popular across NLP tasks like machine translation (He et al., 2021) and intent detection (Mehri and Eric, 2021), it can have broader impact in many other applications. Therefore, although this survey illustrates with the use case of ODQA, *the introduced techniques are intended to go beyond specific applications.*

In the following sections we first lay out the necessary background knowledge (§2), then explain popular WS signals when different resources are available in sections 3 to 5. In conclusion, we highlight promising directions for future work (§6).

2 Background

Neural Ranking (NR) for ODQA Let Q , D and A denote the question, document and answer set. Given a question $q \in Q$, the NR model assigns a relevance score $\mathcal{R}(q, d)$ to each $d \in D$ and selects top- k document $D_{topk} \in D$ with the highest relevance scores. Afterwards, a reader will estimate the score $\mathcal{G}(a|q, D_{topk})$ to predict the final

answer $a \in A$ conditioned on both q and D_{topk} . The NR model can be implemented using various architecture with increasing model complexity. For computational efficiency, normally a bi-encoder architecture (Bromley et al., 1993) is first applied to pre-select top candidates from the whole document set, then a more complex cross-interaction model is applied to provide more accurate relevance scores only for the preselected candidates (Lee et al., 2021). The training objective for the NR model \mathcal{R} can be formalized as:

$$\min_{\mathcal{R}} \mathbb{E}_{q, d^+, d_{1 \sim n}^- \in Q \times D} \mathcal{L}(\mathcal{R}, q, d^+, d_{1 \sim n}^-) \quad (1)$$

where $Q \times D$ indicates the full set of question-document pairs, d^+ is a positive (relevant) document for q , $d_{1 \sim n}^-$ is the sampled n negative (irrelevant) documents and \mathcal{L} is the loss function. A common choice for \mathcal{L} is the contrastive loss:

$$\mathcal{L} = -\log \frac{e^{\mathcal{R}(q, d^+)}}{e^{\mathcal{R}(q, d^+)} + \sum_{j=1}^n e^{\mathcal{R}(q, d_j^-)}} \quad (2)$$

Neural Ranking with Weak Supervision In the standard supervised setting we need relevance annotations for $(q, d) \rightarrow \{+, -\}$ to train \mathcal{R} with Eq 1. Obtaining high-quality relevance annotations requires tremendous human labor and is expensive to scale to multiple domains (Del Tredici et al., 2021; Ram et al., 2022). Weak supervision (WS) is a widely-used approach to reduce such cost by leveraging supervision signals from e.g., heuristic rules, knowledge bases or external models (Zhang et al., 2021). WS signals are cheap to obtain but might contain significant noise which will affect the NR performance. Therefore, understanding their working mechanisms and pros and cons are important to obtain a good NR model. We group WS signals into 3 classes by the resources that they need: (1) *Documents*: only document collection D is needed; (2) *Documents + Questions*: document collection D and question set Q are needed; (3) *Documents + QA Pairs*: document collection D and QA pairs (Q, A) are needed. In the next section, we will present the three classes of WS signals and discuss their pros and cons.

3 Resource: Documents

This section discusses two main techniques to produce WS signals requiring only the document set: (1) self-contrastive learning and (2) question generation. This makes the minimum assumption about

Method	Pseudo Question
Perturbation	d with added perturbation
Summary	(Pseudo) summary of d
Proximity	Nearby text of d
Cooccurrence	Text sharing cooccurred spans with d
Hyperlink	Text with a hyperlink to/from d

Table 2: Given a document d , different heuristics to construct pseudo questions. Contrastive samples made from these heuristics serve as WS signals to train the NR model.

resource availability since having the document set is a prerequisite for building an ODQA system.

3.1 Self Contrastive Learning

Self contrastive learning relies on heuristics to construct pseudo question-document pairs $(q', d'^{+/-})$ from D , then uses them to supervise training of a NR model. The objective is:

$$\min_{\mathcal{R}} \mathbb{E}_{q', d'^+, d'_{1 \sim n}^- \in D} \mathcal{L}(\mathcal{R}, q', d'^+, d'_{1 \sim n}^-) \quad (3)$$

where \mathcal{L} is the ranking loss as in Eq 1. Since negative pairs can be easily constructed by random sampling, the main difficulty is to design good heuristics for constructing positive pseudo pairs (q', d'^+) . There are 5 popular heuristics to construct such positive pairs: perturbation-based, summary-based, proximity-based, cooccurrence-based and hyperlink-based. An overview is in Table 2.

Perturbation-based heuristics add perturbations to some text, then treat the perturbed text and the original text as a positive pair. The intuition is that *perturbed text should still be relevant to the original text*. Typical choices of perturbations include word deletion, substitution and permutation (Zhu et al., 2021b; Meng et al., 2021), adding drop out to representation layers (Gao et al., 2021), or passing sentences through different language models (Carlsson et al., 2021), among other.

Summary-based heuristics extract a summary from the document as the pseudo question based on the intuition that *questions should contain representative information about the central topic of the document*. The summary can be the document title (MacAvaney et al., 2017, 2019; Mass and Roitman, 2020), a random sentence from the first section of the document (Chang et al., 2020), randomly sampled ngrams (Gysel et al., 2018) or a set

of keywords generated from a document language model (Ma et al., 2021a).

Proximity-based heuristics utilize the position information in the document to obtain positive pairs based on the intuition that *nearby text should be more relevant to each other*. The most famous one is the inverse-cloze task (Lee et al., 2019), where a sentence from a passage is treated as the question and the original passage, after removing the sentence, is treated as a positive document. They can be combined with typical noise injection methods like adding drop-out masks (Xu et al., 2022), random word chopping or deletion (Izacard et al., 2021) to further improve the model robustness. Other methods include using spans from the same document (Gao and Callan, 2022; Ma et al., 2022), sentences from the same paragraph, paragraphs from the same document as positive samples (Di Liello et al., 2022), etc.

Cooccurrence-based heuristics construct positive samples based on the intuition that *sentences containing cooccurred spans are more likely to be relevant* (Ram et al., 2021). For example, Glass et al. (2020) constructs a pseudo question with a sentence from the corpus. A term from it is treated as the answer and replaced with a special token. Passages retrieved with BM25 which also contains the answer term are treated as pseudo positive documents. Ram et al. (2022) treat a span and its surrounding context as the pseudo question and use another passage that contains the same span as a positive document.

Hyperlink-based heuristics leverage hyperlink information based on the intuition that *hyperlinked text are more likely to be relevant* (Zhang et al., 2020; Ma et al., 2021b). For example, Chang et al. (2020) takes a sentence from the first section of a page p as a pseudo question because it is often the description or summary of the topic. A passage from another page containing hyperlinks to p is treated as a positive document. Yue et al. (2022a) replace an entity word with a question phrase like “what/when” to form a pseudo question. A passage from its hyperlinked document that contains the same entity word is treated as a positive sample. Zhou et al. (2022) build positive samples with two typologies: “dual-link” where two passages have hyperlinks pointed to each other, and “co-mention” where two passages both have a hyperlink to the same third-party document.

Input	Document Document + Answer Document + Answer + Question Type Document + Answer + Question Type + Clue
Question Generator	Rule-based Generator Prompt-based Generator Fine-tuned Generator [◊]
Filter	LM Score Round-trip Consistency Probability from pre-trained QA Influence Function Ensemble Consistency Entailment Score Learning to Reweight [◊] Target-domain Value Estimation [◊]

Table 3: Different choices for a question generation model setup. [◊] means minimal relevance annotations are needed.

3.2 Question Generation

Self contrastive learning relies on sentences already present in D . Question generation leverage a question generator (QG) to generate new questions *not found* in D , which can then be used to provide WS signals for the NR model. It often employs a filter Fil to filter poorly generated questions. The training objective is:

$$\min_{\mathcal{R}} \mathbb{E}_{q=QG(d^+) \& Fil(q, d^+) = 0} \mathcal{L}(\mathcal{R}, q, d^+, d_{1 \sim n}^-)$$

where the expectation is with respect to documents d^+ and $d_{1 \sim n}^-$ drawn from D , the q are generated from d^+ , $Fil(q, d^+) = 0$ requires that these questions not be discarded by Fil , and \mathcal{L} is the standard ranking loss. There are various ways of designing the question generator and filter. We will cover the popular choices in the following section. An overview can be seen in Table 3.

Choices of Input A variety of information can be provided as input for the QG. The most straightforward approach is answer-agnostic which provides only the document (Du and Cardie, 2017; Kumar et al., 2019). In this way, the model can choose to attend to different spans of the document as potential answers and so generate different, corresponding questions. A more common method is answer-aware where an answer span is first extracted from a document, then the QG generates a question based on both the document and answer (Alberti et al., 2019; Shakeri et al., 2020). Finer-grained information can also be provided such as the question type (“what/how/...”) (Cao and Wang, 2021; Gao et al., 2022) as well as additional clues (such as document context to disambiguate the question) (Liu et al., 2020). Adding more information reduces the entropy of the question and

makes it easier for the model to learn, but also increases the possibility of error propagation (Zhang and Bansal, 2019). In practice, well-defined filters should be applied to remove low-quality questions.

Choices of Question Generator There are three popular choices for the question generator. (1) Rule-based methods (Pandey and Rajeswari, 2013; Rakangor and Ghodasara, 2015) rely on hand-crafted templates and features. These are time-consuming to design, domain-specific, and can only cover certain forms of questions. (2) Prompt-based methods relying on pre-trained language models (PLMs). Documents can be presented to a PLM, with an appended prompt such as “Please write a question based on this passage” so that the PLM can continue the generation to produce a question (Bonifacio et al., 2022; Sachan et al., 2022; Dai et al., 2022). (3) Fine-tuned generators that are trained on annotated question-document pairs. When in-domain annotations are not enough, we can leverage out-of-domain (OOD) annotations, if any, to fine-tune the QG. The first two QGs require no training data, but their quality is often inadequate. In practice, we should only consider them when *there is a complete lack of high-quality supervised data for fine-tuning the QG*. When target-domain questions are available, we can also apply semi-supervised techniques such as back-training to adapt the QG to the target domain (Zhao et al., 2019; Kulshreshtha et al., 2021; Shen et al., 2022a).

Choices of Filter Filtering is a crucial part of QG since a significant portion of generated questions could be of low quality and would provide misleading signals when used to train the NR model (Alberti et al., 2019). A typical choice is filtering based on round-trip consistency (Alberti et al., 2019; Dong et al., 2019), where a pre-trained QA system is applied to produce an answer based on the generated question. A question is kept only when the produced answer is consistent with the answer from which the question is generated. We can also relax this strict consistency requirement and manually adjust an acceptance threshold based on the probability from the pre-trained QA system (Zhang and Bansal, 2019; Lewis et al., 2021), LM score from the generator itself (Shakeri et al., 2020; Liang et al., 2020), or an entailment score from a model trained on question-context-answer pairs (Liu et al., 2020). Influence functions (Cook and Weisberg, 1982) can be used to estimate the

effect on the validation loss of including a synthetic example (Yang et al., 2020), but this does not achieve satisfying performances on QA tasks (Bartolo et al., 2021). Bartolo et al. (2021) propose filtering questions based on ensemble consistency, where an ensemble of QA models are trained with different random seeds and only questions agreed by most QA models are selected. When minimal target-domain annotation is available, we can also learn to reweight pseudo samples based on the validation loss (Sun et al., 2021), or use RL to select samples that lead to validation performance gains (value estimation) (Yue et al., 2022b).

3.3 Discussion

If the heuristics or QG are properly designed, NR models trained from their supervision can even match the fully-supervised performance (Wang et al., 2022; Ren et al., 2022). The biggest challenge is the difficulty to pick the most suitable heuristics or QG when we face a new domain. A general solution is to automatically select good pseudo pairs with reinforcement learning (RL) when minimal target-domain annotations are available (Zhang et al., 2020), so as avoiding the need to manually fixing the WS signals, but this would bring significant computational overhead. In practice hyperlink-based approaches often perform the best among the heuristics as they have additional reference information to leverage, which makes them most similar to the actual relevance annotations. However, hyperlink information is not available in most domains and thereby limits its use cases (Sun et al., 2021). QG-based WS signals are often preferred over heuristics-based ones as they can produce naturally-sound questions themselves without relying on the chance to find good pseudo questions in the documents. Nonetheless, obtaining a high-performing QG can also be non-trivial. One big challenge comes from the one-to-many mapping relations between questions and documents. Under this situation, standard supervised learning tends to produce safe questions with less diversity and high lexical overlap with the document. For example, Shinoda et al. (2021) found that QG reinforces the model bias towards high lexical overlap. We will need more sophisticated training techniques such as latent-variable models (Shen and Su, 2018; Xu et al., 2020; Li et al., 2022) and reinforcement learning (Yuan et al., 2017; Zhang and Bansal, 2019; Shen et al., 2019a) to alleviate the

model bias towards safe questions.

4 Resource: Documents + Questions

This section includes WS signals that require additional access to a question set Q . In practice, annotating question-document relations usually requires domain experts to read long documents and careful sampling strategies to ensure enough positive samples, while unlabeled questions are much easier to obtain either through real user-generated content or simulated traffic. Therefore, it is common to have a predominance of unlabeled questions. The crucial point is to establish the missing relevance labels. Suppose a WS method can provide the missing label $WS(q, d)$ for a question-document pair (q, d) , then we can use it to supervise the NR model by:

$$\min_{\mathcal{R}} \mathbb{E}_{q \in Q, d \in D} \mathcal{L}(\mathcal{R}(q, d), WS(q, d)) \quad (4)$$

where \mathcal{L} is the loss function that encourages similarity between $\mathcal{R}(q, d)$ and $WS(q, d)$.

There are three popular types models that can provide such WS signal here: (1) sparse retriever, (2) pre-trained language model and (3) supervised teacher model.

Sparse Retriever (SR) Recent research finds that NR and SR models are complementary. NR models are better at semantic matching while SRs are better at capturing exact match and handling long documents (Chen et al., 2021; Luan et al., 2021). SRs are also more robust across domains (Thakur et al., 2021; Chen et al., 2022). This motivates the use of unsupervised sparse retrievers like BM25 as WS signals. For example, Dehghani et al. (2017); Nie et al. (2018) train a NR model on samples annotated with BM25. Xu et al. (2019) apply four scoring functions to auto-label questions and documents with: (1) BM25 scores, (2) TF-IDF scores, (3) cosine similarity of universal embedding representation (Cer et al., 2018) and (4) cosine similarity of the last hidden layer activation of pre-trained BERT model (Devlin et al., 2019). Both papers observe that the resulting model outperforms BM25 on the test sets. Chen et al. (2021) further show that distilling knowledge from BM25 helps the retriever to better match rare entities and improves zero-shot out-of-domain performance.

Pre-trained Language Model (PLM) As PLMs already encode significant linguistic knowledge, there have also been attempts at using prompt-based PLMs to provide WS signals for question-

document relations (Smith et al., 2022; Zeng et al., 2022). Similar as in question generation, we can use prompts like “Please write a question based on this passage”, concatenate the document and question, then use the probability assigned by the PLM to auto-label question-document pairs. To maximize the chances of finding positive document, normally we first obtain a set of candidate documents by BM25, then apply PLM to auto-label the candidate set (Sachan et al., 2022). This can further exploit the latent knowledge inside PLMs that has been honed through pre-training, so it often shows better performance compared with weak supervision only using BM25 (Nogueira et al., 2020; Singh Sachan et al., 2022).

Supervised Teacher Model A very common choice is using a supervised teacher model to provide WS signals. The teacher model is “supervised” because it is explicitly fine-tuned on annotated question-document pairs. When in-domain annotations are not sufficient, we can leverage out-of-domain (OOD) annotations, if available, to train the teacher model. The teacher model usually employs a more powerful architecture such as with more complex interactions or larger sizes. It may not be directly applicable in downstream tasks due to the latency constraints, but can be useful in providing WS signals for training the NR model. For example, previous research has shown that models with larger sizes or late/cross-interaction structures generalize much better on OOD data (Pradeep et al., 2020; Lu et al., 2021; Ni et al., 2021; Rosa et al., 2022; Muennighoff, 2022; Zhan et al., 2022). After training a teacher model on OOD annotations, applying it to provide WS signals through target-domain question and document collections can significantly improve the in-domain performance of the NR model (Hofstätter et al., 2021; Lin et al., 2021b; Lu et al., 2022). Kim et al. (2022) further show that we can even use the same architecture and capacity to obtain a good teacher model. They expand the question with centroids of word embeddings from top retrieved passages (using BM25), and then use the expanded query for self knowledge distillation. Similar ideas of reusing the same architecture to provide WS signals have also been explored by Yu et al. (2021a); Kulshreshtha et al. (2021); Zhuang and Zuccon (2022).

Discussion The three WS signals listed above work directly on actual questions instead of pseudo

pairs as in §3 so that the NR model can adapt better to the target-domain question distribution. The bottleneck is the quality of the WS signals. SRs and PLMs are unsupervised, which could be more robust when we face a completely different domain (Dai et al., 2022). Otherwise, if we already have certain amounts relevance annotations from the target or similar domains, usually using a supervised teacher model is preferred. Nevertheless, these WS signals inevitably contain noise, and can harm the downstream performance if the noise is significant. There are two main strategies to reduce the noise effects: (1) Apply less strict margin-based loss such as the hinge loss (Dehghani et al., 2017; Xu et al., 2019) and MarginMSE loss (Hofstätter et al., 2020; Wang et al., 2022), then models have fewer chances of overfitting to the exact labels, and (2) Apply noise-resistant training methods such as confidence-based filtering (Mukherjee and Awadallah, 2020; Yu et al., 2021b) and meta-learning-based refinement (Ren et al., 2018; Zhu et al., 2022). Another potential issue is that the amount of training data in this section relies on the amount of questions we have. Unlike the document set which we can obtain for free, the question set takes time to collect and are often orders of magnitudes smaller. If no sufficient questions are available, we can use synthetic questions from question generation, then apply same WS signals in this section, which has been shown to perform on par with using real questions in certain domains (Wang et al., 2020, 2022; Thakur et al., 2022).

5 Resource: Documents + QA Pairs

Many domains have large numbers of already answered questions from customer services, technical support or web forums (Huber et al., 2021). These QA pairs can provide richer information than only unlabeled questions. However, most answers are based on personal knowledge, derived from experience, and do not include a reference to any external document. This prevents their direct use as training data for the NR model. This section introduces three standard methods that exploit QA pairs to provide WS signals despite this difficulty: (1) Answer as document, (2) Answer-document mapping and (3) Latent-variable models.

5.1 Answer as a Document

As a straightforward way to leverage QA pairs, this method directly treats QA pairs as positive samples

and does not distinguish between documents and answers (Lai et al., 2018). These QA pairs can provide direct WS signals to train the NR model:

$$\min_{\mathcal{R}} \mathbb{E}_{q, a^+, a_{1 \sim n}^- \in Q \times A} \mathcal{L}(\mathcal{R}, q, a^+, a_{1 \sim n}^-) \quad (5)$$

where $(q, a^+) \in Q \times A$ are question-answer pairs in the target domain, $a_{1 \sim n}^-$ are sampled n negative answers and \mathcal{L} is the standard ranking loss.

Though simple, this has been a common practice to “warm up” the NR model when no sufficient relevance annotations are available. For large-sized models, this can be crucial to fully leverage the model capacity since we often have orders of magnitude more QA pairs than relevance annotations (Ni et al., 2021; Oğuz et al., 2021). However, the style, structure and format differ between the document and the answer. The answer is a direct response to the question, and so it is easier to predict due to its strong semantic correlation with the question. Whereas the document can be implicit and may contain fewer obvious clues that can imply an answer; deep text understanding is required to predict the relevance between questions and documents (Zhao et al., 2021; Shen et al., 2022b). Therefore, this approach may be insufficient to reach satisfying results as a standalone method.

5.2 Answer-document Mapping

This approach leverages an additional mapping function to automatically link answers to the corresponding documents. The NR model can get WS signals from the linked answers:

$$\min_{\mathcal{R}} \mathbb{E}_{q, a \in (Q, A), d_{1 \sim n}^- \sim D} \mathcal{L}(\mathcal{R}, q, M(a), d_{1 \sim n}^-)$$

where $(q, a) \in (Q, A)$ are question-answer pairs, M is a mapping function from an answer to its corresponding document, and \mathcal{L} is the standard ranking loss. The mapping function is based on hand-crafted heuristics. For long-form descriptive answers, a popular way is to map them to documents with highest ROUGE scores (Lin, 2004) since the answers can be considered as summaries of the original documents (Fan et al., 2019). For short-span answers, a popular way is to map them to top-ranked documents retrieved using BM25 that contain the answer span (Karpukhin et al., 2020; Sachan et al., 2021; Christmann et al., 2022).

Answer-document mapping was widely adopted for constructing large-scale datasets in information retrieval (Joshi et al., 2017; Dunn et al., 2017; Elgohary et al., 2018). This can work well if the

Distribution of $\mathcal{R}(z q)$	Optimization Method
Categorical	Top- k approximation
Multinomial	EM algorithm
	Learning from attention

Table 4: Distribution assumptions made about the neural ranker and corresponding optimization methods, suppose we train the NR model following Equation 6.

mapping has high accuracy, which is often difficult to achieve. Frequent answers or entities might lead to false positive mappings. It is also difficult to find positive documents for boolean and abstractive answers using only heuristics-based mapping functions (Izacard and Grave, 2021). Models can easily overfit to the biases introduced via such mapping function (Du et al., 2022).

5.3 Latent-Variable Model

We can still train the NR model on question-document pairs as in Answer-Document Mapping. However, instead of relying on a heuristic-based mapping function, we can treat this mapping as a “latent variable” within a probabilistic generative process (Lee et al., 2019; Shen, 2022). By this means, the NR model \mathcal{R} gets WS signals from the QA reader \mathcal{G} by maximizing the marginal likelihood:

$$\max_{\mathcal{R}, \mathcal{G}} \mathbb{E}_{q, a \in (Q, A)} \log \sum_{z \sim Z} \mathcal{R}(z|q) \mathcal{G}(a|q, z) \quad (6)$$

where Z indicates all possible document combinations. Directly optimizing over Eq 6 is infeasible as it requires enumerating over all documents. A closed-form solution does not exist due to the deep neural network parameterization of \mathcal{R} and \mathcal{G} . The following section explains popular optimization options. An overview can be seen in Table 4.

Top- k approximation A popular approach is to assume a categorical distribution for $\mathcal{R}(Z|q)$; that is, to assume for each question only a single document is selected and the answer is generated from that one document. Eq 6 can be approximated by enumerating over only the top- k documents, assuming the remaining documents having negligibly small contributions to the likelihood:

$$\max_{\mathcal{R}, \mathcal{G}} \mathbb{D}_{q, a \in (Q, A)} \log \sum_{z \sim E_{topk}} \mathcal{R}(z|q) \mathcal{G}(a|q, z)$$

This has been a popular choice in end-to-end training of text generation models (Lee et al., 2019; Shen et al., 2019b; Guu et al., 2020; Lewis et al.,

2020; Shuster et al., 2021; Ferguson et al., 2022). Despite its simplicity, the top- k approximation has two main drawbacks. (1) The approximation is performed on the top- k documents obtained from the NR model. If the NR model is very weak at the beginning of training, these top- k documents can be a bad approximation to the real joint likelihood and the model might struggle to converge. (2) The assumption that document follow a categorical distribution might be problematic especially if the answer requires evidence from multiple documents (Wang and Pan, 2022).

Expectation–Maximization (EM) algorithm

To address the second drawback of the top- k approximation approach, we can assume a multinomial distribution for $R(Z|q)$ so that an answer can be generated from multiple documents. The cost of this relaxation is the increased difficulty of optimization. Approximating the joint likelihood from top- k samples becomes infeasible due to the combinatorial distribution of document. Singh et al. (2021) propose optimizing it with the EM algorithm under an independent assumption about the posterior distribution of $\mathcal{R}(z|q)$:

$$\begin{aligned} \max_{\mathcal{R}, \mathcal{G}} \mathbb{E}_{q, a \in (Q, A)} [\log \sum_{z \in D_{topk}} \mathcal{R}(z|q) \\ \times SG(\mathcal{G}(a|q, z)) + \log \mathcal{G}(a|q, D_{topk})] \end{aligned} \quad (7)$$

where SG means stop-gradient (gradients are not backpropagated through \mathcal{G}). As can be seen, the training signal for the NR model is essentially the same as in the *Top-k Approximation* case, except that the reader is trained by conditioning on all top- k documents to generate the answer. Singh et al. (2021) also find that Eq 7 is quite robust with respect to parameter initialization. Similarly, Zhao et al. (2021) apply the hard-EM algorithm to train the NR model, which only treats documents with the highest likelihood estimated by the reader as positive. Izacard et al. (2022) further experiment with using the leave-one-out perplexity from the reader to supervise the ranker.

Learning from attention Another way to optimize the NR model in Eq 6 is to leverage attention scores from the reader \mathcal{G} . The assumption is that when training \mathcal{G} to generate the answer, its attention score is a good approximation of question-

document relevance. The training objective is:

$$\begin{aligned} \min_{\mathcal{R}, \mathcal{G}} \mathbb{E}_{q, a \in (Q, A)} \sum_{z \sim E_{topk}} \mathcal{L}(A_z | \mathcal{R}(z|q)) \\ - \log \mathcal{G}(a|q, Z = D_{topk}) \end{aligned} \quad (8)$$

where \mathcal{G} is trained to generate the right answer based on the question and the top- k document, same as in the EM algorithm. A_z is the attention score of \mathcal{G} on the document z . \mathcal{L} is the loss function to encourage the similarity between distributions of the attention scores and retrieving scores.

Izacard and Grave (2021) propose a training process that optimizes \mathcal{R} and \mathcal{G} iteratively. \mathcal{R} is trained to minimize KL divergence between relevance and attention scores. (Lee et al., 2021) jointly optimize \mathcal{R} and \mathcal{G} and apply a stop-gradient operation on \mathcal{G} when updating \mathcal{R} . Sachan et al. (2021) use retriever scores to bias attention scores on the contrary. These can be considered as first-order Taylor series approximations of Eq. 6 by replacing $\mathcal{R}(Z|q)$ with attention scores (Deng et al., 2018).

Discussion Training with latent-variable models can perform close to fully supervised models under certain scenarios (Zhao et al., 2021; Sachan et al., 2021). The main challenge is the training difficulty. In practice, we can often initialize the NR model using the *answer as document* or *answer-document mapping* to make the training more stable. If not enough QA pairs are available, we can use heuristics like masked salient entities (Guu et al., 2020) to form pseudo pairs, then apply the same WS techniques in this section. Combining supervision signals from various various optimization techniques such as learning from attention and EM algorithm can also be beneficial (Izacard et al., 2022). If the independence assumption made by Eq 7 does not hold, we need to resort to more complex optimization algorithms. A potential direction is to apply a Dirichlet prior over $R(z|q_i)$, which is a conjugate distribution to the multinomial distribution (Minka, 2000), with the result that the sampled document are not independent individuals but a combination set. Eq 6 can then be estimated by rejection sampling (Deng et al., 2018) or a Laplace approximation (Srivastava and Sutton, 2017) so as to avoid the independence assumption about the posterior distribution. Nonetheless, this will further increase the training complexity, which is already a key bottleneck for training the NR model.

6 Conclusions

We review standard WS signals used for training NR models in ODQA and provide a structured way of classifying them according to the required resource. For WS signal, we discuss different options and summarize the pros and cons. As a final wrap-up, we list promising directions that we believe worth exploring further: (1) *How to select the most suitable technique for a given scenario?* Despite the wide range of applicable techniques, it is non-trivial to decide how to select the best one except for an empirical experimentation. (2) *To which extent are these techniques complementary?* Existing work compares performance only between similar types of methods but not across the whole range of techniques and resources available. This makes it hard to decide whether different approaches could potentially complement each other and how they should be combined effectively. (3) *Do methods work across languages?* The vast majority of current research is conducted on English datasets. Even though all described methods in this survey have no explicit restrictions on languages they can be applied to, it is likely that their performance will vary across languages, especially for the methods relying on handcrafted heuristics.

Limitations

This survey covers introductions and related work of major WS algorithms used for neural ranking. Due to the space limit, most methods included in this paper are brief. Readers might not have a good understand on all the introduced methods. Interested readers can refer to existing surveys about general knowledge in QA (Zeng et al., 2020; Zhu et al., 2021a; Roy and Anand, 2021; Rogers et al., 2021; Pandya and Bhatt, 2021). Furthermore, we did not provide points to existing ODQA datasets and the performance of recent models. The conclusions in this survey also come from summaries of previous works. The lack of datasets including various resources needed for different WS algorithms prevents a comprehensive, fair comparison across algorithms. We hope future research can work on the creation of more datasets with various availabilities of resources in different domains to enable this comparison. Lastly, we aim to create a big picture from the technology level, so we did not strictly limit our references only to the application of ODQA. The connection to specific ODQA applications might be loose, readers would need

to extract useful information for the specific use cases.

References

- Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173.
- Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848.
- Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.
- Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. *Advances in neural information processing systems*, 6.
- Shuyang Cao and Lu Wang. 2021. Controllable opened question generation with a new question type ontology. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6424–6439.
- Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2021. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174.
- Wei-Cheng Chang, X Yu Felix, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. In *International Conference on Learning Representations*.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.
- Tao Chen, Mingyang Zhang, Jing Lu, Michael Bendersky, and Marc Najork. 2022. Out-of-domain semantics to the rescue! zero-shot hybrid retrieval models. *arXiv preprint arXiv:2201.10582*.

- Xilun Chen, Kushal Lakhota, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2021. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? *arXiv preprint arXiv:2110.06918*.
- Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum. 2022. Conversational question answering on heterogeneous sources. *arXiv preprint arXiv:2204.11677*.
- R Dennis Cook and Sanford Weisberg. 1982. *Residuals and influence in regression*. New York: Chapman and Hall.
- Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.
- Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. In *International Conference on Learning Representations*.
- Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74.
- Marco Del Tredici, Gianni Barlacchi, Xiaoyu Shen, Weiwei Cheng, and Adrià de Gispert. 2021. Question rewriting for open-domain conversational qa: Best practices and limitations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2974–2978.
- Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. 2018. Latent alignment and variational attention. *Advances in Neural Information Processing Systems*, 31.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Luca Di Liello, Siddhant Garg, Luca Soldaini, and Alessandro Moschitti. 2022. Pre-training transformer models with sentence-level objectives for answer sentence selection. *arXiv preprint arXiv:2205.10455*.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.
- Pan Du, Jian-Yun Nie Nie, Yutao Zhu, Hao Jiang, Lixin Zou, and Xiaohui Yan. 2022. Pegan: Answer oriented passage ranking with weakly supervised gan. *arXiv preprint arXiv:2207.01762*.
- Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2067–2073.
- Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.
- Ahmed Elgohary, Chen Zhao, and Jordan Boyd-Graber. 2018. A dataset and baselines for sequential open-domain question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1077–1083.
- Angela Fan, Yacine Jernite, Ethan Perez, David Granger, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.
- James Ferguson, Hannaneh Hajishirzi, Pradeep Dasigi, and Tushar Khot. 2022. **Retrieval data augmentation informed by downstream question answering performance**. In *Proceedings of the Fifth Fact Extraction and VERification Workshop (FEVER)*, pages 1–5, Dublin, Ireland. Association for Computational Linguistics.
- Lingyu Gao, Debanjan Ghosh, and Kevin Gimpel. 2022. "what makes a question inquisitive?" a study on type-controlled inquisitive question generation. *arXiv preprint arXiv:2205.08056*.
- Luyu Gao and Jamie Callan. 2022. **Unsupervised corpus aware language model pre-training for dense passage retrieval**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.
- Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, GP Shrivatsa Bhargav, Dinsh Garg, and Avirup Sil. 2020. Span selection pre-training for question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2782.

- Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–42.
- Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W Bruce Croft, and Xueqi Cheng. 2020. A deep look into neural ranking models for information retrieval. *Information Processing & Management*, 57(6):102067.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.
- Christophe Van Gysel, Maarten De Rijke, and Evangelos Kanoulas. 2018. Neural vector spaces for unsupervised information retrieval. *ACM Transactions on Information Systems (TOIS)*, 36(4):1–25.
- Qiuxiang He, Guoping Huang, Qu Cui, Li Li, and Lemao Liu. 2021. Fast and accurate neural machine translation with translation memory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3170–3180.
- Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666*.
- Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122.
- Patrick Huber, Armen Aghajanyan, Barlas Oğuz, Dmytro Okhonko, Wen-tau Yih, Sonal Gupta, and Xilun Chen. 2021. Ccqa: A new web-scale question answering dataset for model pre-training. *arXiv preprint arXiv:2110.07731*.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.
- Gautier Izacard and Edouard Grave. 2021. Distilling knowledge from reader to retriever for question answering. *ICLR*.
- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.
- Jihyuk Kim, Minsoo Kim, and Seung-won Hwang. 2022. Collective relevance labeling for passage retrieval. *arXiv preprint arXiv:2205.03273*.
- Devang Kulshreshtha, Robert Belfer, Iulian Vlad Serban, and Siva Reddy. 2021. Back-training excels self-training at unsupervised domain adaptation of question generation and passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7064–7078.
- Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi. 2019. Cross-lingual training for automatic question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4863–4872.
- Tuan Manh Lai, Trung Bui, Nedim Lipka, and Sheng Li. 2018. Supervised transfer learning for product information question answering. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1109–1114. IEEE.
- Haejun Lee, Akhil Kedia, Jongwon Lee, Ashwin Paranjape, Christopher D Manning, and Kyoung-Gu Woo. 2021. You only need one model for open-domain question answering. *arXiv preprint arXiv:2112.07381*.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.
- Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

- Jin Li, Peng Qi, and Hong Luo. 2022. Generating consistent and diverse qa pairs from contexts with bn conditional vae. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 944–949. IEEE.
- Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. 2020. Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv:2009.10270*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021a. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325.
- Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021b. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 163–173.
- Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. In *Proceedings of The Web Conference 2020*, pages 2032–2043.
- Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is more: Pretrain a strong siamese encoder for dense text retrieval using a weak decoder. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2791.
- Yuxiang Lu, Yiding Liu, Jiayang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, et al. 2022. Ernie-search: Bridging cross-encoder with dual-encoder via self on-the-fly distillation for dense passage retrieval. *arXiv preprint arXiv:2205.09153*.
- Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.
- Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-train a discriminative text encoder for dense retrieval via contrastive span prediction. *arXiv preprint arXiv:2204.10641*.
- Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021a. Prop: Pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 283–291.
- Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. 2021b. Pre-training for ad-hoc retrieval: Hyperlink is also you need. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1212–1221.
- Sean MacAvaney, Kai Hui, and Andrew Yates. 2017. An approach for weakly-supervised deep information retrieval. *arXiv preprint arXiv:1707.00189*.
- Sean MacAvaney, Andrew Yates, Kai Hui, and Ophir Frieder. 2019. Content-based weak supervision for ad-hoc re-ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 993–996.
- Yosi Mass and Haggai Roitman. 2020. Ad-hoc document retrieval using weak-supervision with bert and gpt2. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4191–4197.
- Shikib Mehri and Mihail Eric. 2021. Example-driven intent prediction with observers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2979–2992.
- Yu Meng, Chenyan Xiong, Payal Bajaj, Paul Bennett, Jiawei Han, Xia Song, et al. 2021. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *Advances in Neural Information Processing Systems*, 34.
- Thomas Minka. 2000. Estimating a dirichlet distribution.
- Bhaskar Mitra, Nick Craswell, et al. 2018. *An introduction to neural information retrieval*. Now Foundations and Trends Boston, MA.
- Iurii Mokrii, Leonid Boytsov, and Pavel Braslavski. 2021. A systematic evaluation of transfer learning and pseudo-labeling with bert-based ranking models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2081–2085.
- Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.
- Subhabrata Mukherjee and Ahmed Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. *Advances in Neural Information Processing Systems*, 33:21199–21212.
- Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

- Yifan Nie, Alessandro Sordani, and Jian-Yun Nie. 2018. Multi-level abstraction convolutional model with weak supervision for information retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 985–988.
- Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718.
- Barlas Oğuz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, et al. 2021. Domain-matched pre-training tasks for dense retrieval. *arXiv preprint arXiv:2107.13602*.
- Shivank Pandey and KC Rajeswari. 2013. Automatic question generation using software agents for technical institutions. *International Journal of Advanced Computer Research*, 3(4):307.
- Hariom A Pandya and Brijesh S Bhatt. 2021. Question answering survey: Directions, challenges, datasets, evaluation matrices. *arXiv preprint arXiv:2112.03572*.
- Ronak Pradeep, Xueguang Ma, Xinyu Zhang, Hang Cui, Ruizhou Xu, Rodrigo Nogueira, and Jimmy Lin. 2020. H2ooloo at trec 2020: When all you got is a hammer... deep learning, health misinformation, and precision medicine. *Corpus*, 5(d3):d2.
- Sheetal Rakangor and YR Ghodasara. 2015. Literature review of automatic question generation systems. *International journal of scientific and research publications*, 5(1):1–5.
- Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. **Few-shot question answering by pretraining span selection**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3066–3079, Online. Association for Computational Linguistics.
- Ori Ram, Gal Shachaf, Omer Levy, Jonathan Berant, and Amir Globerson. 2022. Learning to retrieve passages without supervision. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2687–2700, Seattle, United States.
- Revanth Gangi Reddy, Vikas Yadav, Md Arafat Sultan, Martin Franz, Vittorio Castelli, Heng Ji, and Avirup Sil. 2021. Towards robust neural retrieval models with synthetic pre-training. *arXiv preprint arXiv:2104.07800*.
- Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR.
- Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qifei Wu, Yuchen Ding, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2022. A thorough examination on zero-shot dense retrieval. *arXiv preprint arXiv:2204.12755*.
- Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*, pages 232–241. Springer.
- Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *arXiv preprint arXiv:2107.12708*.
- Guilherme Moraes Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. No parameter left behind: How distillation and model size affect zero-shot retrieval. *arXiv preprint*.
- Rishiraj Saha Roy and Avishek Anand. 2021. Question answering for the curated web: Tasks and methods in qa over knowledge bases and text collections. *Synthesis Lectures on Synthesis Lectures on Information Concepts, Retrieval, and Services*, 13(4):1–194.
- Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6648–6662.
- Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. *arXiv preprint arXiv:2204.07496*.
- Siamak Shakeri, Cicero dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460.
- Xiaoyu Shen. 2022. Deep latent-variable models for text generation. *arXiv preprint arXiv:2203.02055*.
- Xiaoyu Shen, Gianni Barlacchi, Marco Del Tredici, Weiwei Cheng, Bill Byrne, and Adrià de Gispert. 2022a. Product answer generation from heterogeneous sources: A new benchmark and best practices. In *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 99–110.

- Xiaoyu Shen, Gianni Barlacchi, Marco Del Tredici, Weiwei Cheng, and Adrià de Gispert. 2022b. semipqa: A study on product question answering over semi-structured data. In *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 111–120.
- Xiaoyu Shen and Hui Su. 2018. Towards better variational encoder-decoders in seq2seq tasks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 8155–8156.
- Xiaoyu Shen, Jun Suzuki, Kentaro Inui, Hui Su, Dietrich Klakow, and Satoshi Sekine. 2019a. Select and attend: Towards controllable content selection in text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 579–590.
- Xiaoyu Shen, Svitlana Vakulenko, Marco Del Tredici, Gianni Barlacchi, Bill Byrne, and Adrià de Gispert. 2022c. Low-resource dense retrieval for open-domain question answering: A comprehensive survey. *arXiv preprint arXiv:2208.03197*.
- Xiaoyu Shen, Yang Zhao, Hui Su, and Dietrich Klakow. 2019b. Improving latent alignment in text summarization by generalizing the pointer generator. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3762–3773.
- Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. 2021. Can question generation debias question answering models? a case study on question–context lexical overlap. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 63–72.
- Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.
- Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34.
- Devendra Singh Sachan, Mike Lewis, Dani Yogatama, Luke Zettlemoyer, Joelle Pineau, and Manzil Zaheer. 2022. Questions are all you need to train a dense passage retriever. *arXiv e-prints*, pages arXiv–2206.
- Ryan Smith, Jason A Fries, Braden Hancock, and Stephen H Bach. 2022. Language models in the loop: Incorporating prompting into weak supervision. *arXiv preprint arXiv:2205.02318*.
- Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *ICLR*.
- Si Sun, Yingzhuo Qian, Zhenghao Liu, Chenyan Xiong, Kaitao Zhang, Jie Bao, Zhiyuan Liu, and Paul Bennett. 2021. Few-shot text ranking with meta adapted synthetic weak supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5030–5043.
- Nandan Thakur, Nils Reimers, and Jimmy Lin. 2022. Domain adaptation for memory-efficient dense retrieval. *arXiv preprint arXiv:2205.11498*.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. 2021. Adaptable and interpretable neural memoryover symbolic knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3678–3691.
- Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82.
- Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. 2022. Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. *NAACL*.
- Liqiang Wang, Xiaoyu Shen, Gerard de Melo, and Gerhard Weikum. 2020. Cross-domain learning for classifying propaganda in online contents. *arXiv preprint arXiv:2011.06844*.
- Wenya Wang and Sinno Pan. 2022. Deep inductive logic reasoning for multi-hop reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4999–5009.
- Binxia Xu, Siyuan Qiu, Jie Zhang, Yafang Wang, Xiaoyu Shen, and Gerard de Melo. 2020. Data augmentation for multiclass utterance classification—a systematic study. In *Proceedings of the 28th international conference on computational linguistics*, pages 5494–5506.
- Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. Laprador: Unsupervised pretrained dense retriever for zero-shot text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3557–3569.
- Peng Xu, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Passage ranking with weak supervision. *arXiv preprint arXiv:1905.05910*.

- Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for common-sense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025.
- Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021a. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 829–838.
- Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021b. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1063–1077.
- Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordani, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25.
- Xiang Yue, Xiaoman Pan, Wenlin Yao, Dian Yu, Dong Yu, and Jianshu Chen. 2022a. C-more: Pretraining to answer open-domain questions by consulting millions of references. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 371–377.
- Xiang Yue, Ziyu Yao, and Huan Sun. 2022b. Synthetic question value estimation for domain adaptation of question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1340–1351.
- Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. 2020. A survey on machine reading comprehension—tasks, evaluation metrics and benchmark datasets. *Applied Sciences (2076-3417)*, 10(21).
- Ziqian Zeng, Weimin Ni, Tianqing Fang, Xiang Li, Xinran Zhao, and Yangqiu Song. 2022. Weakly supervised text classification using supervision signals from a language model. *arXiv preprint arXiv:2205.06604*.
- Jingtao Zhan, Xiaohui Xie, Jiabin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2022. Evaluating extrapolation performance of dense retrieval. *arXiv preprint arXiv:2204.11447*.
- Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021. Wrench: A comprehensive benchmark for weak supervision. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Kaitao Zhang, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2020. Selective weak supervision for neural information retrieval. In *Proceedings of The Web Conference 2020*, pages 474–485.
- Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509.
- Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. 2021. [Distantly-supervised dense retrieval enables open-domain question answering without evidence annotation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9612–9622, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yang Zhao, Xiaoyu Shen, Wei Bi, and Akiko Aizawa. 2019. Unsupervised rewriter for multi-sentence compression. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2235–2240.
- Jiawei Zhou, Xiaoguang Li, Lifeng Shang, Lan Luo, Ke Zhan, Enrui Hu, Xinyu Zhang, Hao Jiang, Zhao Cao, Fan Yu, et al. 2022. Hyperlink-induced pre-training for passage retrieval in open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7135–7146.
- Dawei Zhu, Xiaoyu Shen, Michael A Hedderich, and Dietrich Klakow. 2022. Meta self-refinement for robust learning with weak supervision. *arXiv preprint arXiv:2205.07290*.
- Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021a. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint*.
- Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021b. Contrastive learning of user behavior sequence for context-aware document ranking. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2780–2791.
- Shengyao Zhuang and Guido Zuccon. 2022. Characterbert and self-teaching for improving the robustness of dense retrievers on queries with typos. *arXiv preprint arXiv:2204.00716*.

Double Retrieval and Ranking for Accurate Question Answering

Zeyu Zhang^{1*}, Thuy Vu², and Alessandro Moschitti²

¹School of Information, The University of Arizona, Tucson, AZ, USA

²Amazon Alexa AI, Manhattan Beach, CA, USA
zeyuzhang@arizona.edu, {thuyvu, amosch}@amazon.com

Abstract

Recent work has shown that an answer verification step introduced in Transformer-based answer selection models can significantly improve the state of the art in Question Answering. This step is performed by aggregating the embeddings of top k answer candidates to support the verification of a target answer. Although the approach is intuitive and sound, it still shows two limitations: (i) the supporting candidates are ranked only according to the relevancy with the question and not with the answer, and (ii) the support provided by the other answer candidates is suboptimal as these are retrieved independently of the target answer. In this paper, we address both drawbacks by proposing (i) a double reranking model, which, for each target answer, selects the best support; and (ii) a second neural retrieval stage designed to encode question and answer pair as the query, which finds more specific verification information. The results on well-known datasets for Answer Sentence Selection show significant improvement over the state of the art.

1 Introduction

In recent years, automated Question Answering (QA) research has received a renewed attention thanks to the diffusion of Virtual Assistants. For example, Google Home, Siri and Alexa provide general information inquiry services, while many other systems serve customer requests in different application domains. Retrieval-based QA is enabled by two main tasks: (i) Answer Sentence Selection (AS2), which, given a question and a set of answer-sentence candidates, consists in selecting sentences (e.g., retrieved by a search engine) that correctly answer the question; and (ii) Machine Reading (MR), e.g., (Chen et al., 2017), which, given a question and a reference text, finds an exact text span that answers the question. Deploying MR

*Work done while the author was an intern at Amazon Alexa

q :	What causes heart disease?
c_1 :	Cardiovascular disease (also called heart disease) is a class of diseases that involve the heart or blood vessels (arteries, capillaries, and veins).
c_2 :	The causes of cardiovascular disease are diverse but atherosclerosis and/or hypertension are the most common.
c_3 :	Cardiovascular disease refers to any disease that affects the cardiovascular system , principally cardiac disease, vascular diseases of the brain and kidney , and peripheral arterial disease.

Table 1: A question with answer candidates.

systems in production is challenging for efficiency reasons, while AS2 models can efficiently target large text databases. Indeed, they originated from TREC QA tracks (Voorhees and Tice, 1999), which dealt with real-world retrieval systems since the first edition. Another limitation of MR is the focus on factoid answers: although it can in principle provide longer answers, the datasets developed for the task mainly contains short answers and in particular named entities. In contrast, as AS2 processes entire sentences, its inference steps always involve sentences/paragraphs, which make the approach agnostic to both factoid and not factoid classes.

Garg et al. (2020) proposed the TANDA approach, which basically uses two stage of fine-tuning on pre-trained Transformer models (using a general dataset, ASNQ, and the target dataset), obtaining impressive improvement over the state of the art for AS2, measured on the two most used datasets, WikiQA (Yang et al., 2015) and TREC-QA (Wang et al., 2007). The approach above, based on pointwise rerankers, was significantly improved by the Answer Support-based Reranker (ASR) (Zhang et al., 2021), which adds an answer verification step similar to the one operated by fact checking systems, e.g., see the FEVER challenge (Thorne et al., 2018).

More specifically, given a question q , and a target answer, t , to be verified, which is taken from a ranked set of answer candidates (c_1, \dots, c_k),

ASR concatenates transformer-based embeddings of (q, c_i) with the max-pooling vector produced by the top k embeddings of (t, c_i) , where the c_i are selected by an initial answer reranking model (e.g., TANDA). For example, Table 1 reports a question, $q = \textit{What causes heart disease?}$, with some candidate answers, c_1, c_2 , and c_3 . Selecting the correct answer c_2 is difficult, without the information: *cardiovascular disease* is also called *heart disease*. This information is provided by c_1 . Thus, to compute the correctness probability of c_1 , they exploit the representation of c_2 , similarly to the way claims are supported in the fact verification.

ASR reduced the error of TANDA by 10% (relative), both on WikiQA and TREC-QA datasets. However, ASR shows two important limitations: first, when attempting the verification step of t , the k candidates, used in the max-pooling operation, are ranked only based on the question, i.e., independently of t . Second, the support for each t is provided by other answer candidates, which again are retrieved independently of t , i.e., t is not part of the query used for searching relevant documents.

In this paper, we provide new answer verification models, which are more efficient and accurate than ASR. We introduce a new architecture, Double Answer Reranking (DAR), which uses two models for reranking target answers and supporting candidates, respectively. Given t , the first, support reranker (SR), sorts (q, t, c_i) triplets with respect to i , in order to find the best support for t , i.e., $s_t = c_i$, while the second, answer reranker (AR), orders (q, t, s_t) triplets with respect to t , thus ranking all target answers.

Additionally, we improve the quality of supports using a second retrieval stage that searches for passages relevant to (q, t) . This is important as standard answer candidates provide only information relevant to q , thus they not necessarily provide useful context for assessing t . As formulating an effective query for retrieving a question/answer pair is a new problem, and can be challenging, we exploit deep passage retrieval (DPR) (Karpukhin et al., 2020). This enables us to automatically produce embeddings for (q, t) as the target query of a neural retrieval model. As DAR is efficient, it can process many candidates from DPR, making Double Retrieval (DR) effective.

The results derived on three well-known AS2 datasets, WikiQA (Yang et al., 2015), TREC (Wang et al., 2007), and SelQA (Jurczyk et al., 2016) and

a popular multi-hop QA dataset, HotpotQA (Yang et al., 2018), show consistent and significant improvement over the state of the art. For example, DAR improves TANDA by 13.6% (relative error reduction), achieving the same accuracy of the computational expensive ASR verification approach (84.36%) while DAR-DR improves the AS2 state of the art, reducing the error by an additional 8%.

We will release the datasets augmented with DPR retrieval (support candidates) for each (q, a) of each of the datasets above.

2 Related work

We focus our research on QA systems based on Information Retrieval. Since early versions, e.g., TREC QA tracks (Voorhees and Tice, 1999), these systems have been based on a search engine, which retrieves documents relevant to the asked questions, followed efficient and accurate passage rerankers to select text that most likely contains the answer. This research was revived introducing the task of answer sentence reranking (Wang et al., 2007).

In recent work, the probability, $p(q, c_i)$, for a passage/sentence, c_i , to be correct for q is estimated using neural networks, e.g., encoding q and c_i text, separately with a CNN (Severyn and Moschitti, 2015). Also designing attention mechanisms, e.g., Compare-Aggregate (Yoon et al., 2019), inter-weighted alignment networks (Shen et al., 2017). The state of the art is achieved with pre-trained Transformers, e.g., (Garg et al., 2020).

A number of researchers has proposed more than one candidate for the inference stage, e.g., using pairwise model, i.e., binary classifiers of the form $\chi(q, c_i, c_j)$, which determine the partial rank between c_i and c_j . For example, (Laskar et al., 2020; Tayyar Madabushi et al., 2018; Rao et al., 2016) use a pairwise loss and encoding. However, these methods have been largely outperformed by the pointwise models based on Transformers.

Bonadiman and Moschitti (2020) designed several joint models that improved early neural models for AS2 but failed to improve Transformer-based models. Jin et al. (2020) used the relation between candidates in Multi-task learning approach for AS2 but as they did not exploit transformer models, their results are rather lower than the state of the art. Very recently, Zhang et al. (2021) proposed ASR, a model based on a pointwise reranker fed with the embeddings refined by a pairwise approach. This significantly improved the state of the art, there-

fore, we analyzed ASR and specifically compare our models with it.

Very different approaches to QA systems than above use MR to extract answers from entire documents. As they have been mainly developed to find answers in a paragraph or in a text of limited size, they are rather inefficient at processing hundreds of documents, while AS2 methods can do this with high efficiency. [Chen et al. \(2017\)](#); [Hu et al. \(2019\)](#); [Kratzwald and Feuerriegel \(2018\)](#) proposed solutions for reliably performing inference with MR models on multiple documents. Still, the efficiency drawback was not solved. Finally, multihop QA uses multiple retrieval stages ([Xiong et al., 2020](#); [Qi et al., 2019](#)) but the answers are just entities.

3 Baseline models for AS2

A general problem formulation for AS2 is the following: given a question q , a subset of its top- k ranked answer candidates, and a target answer $t \in C_k$, train a function, $f : Q \times C^k \rightarrow \mathbb{R}$ such that $f(q, t, c_1, \dots, c_{k-1})$ provides the probability of t to be correct. In this section, we describe our re-implementation of baselines, and the state-of-the-art model for AS2, namely, ASR ([Zhang et al., 2021](#)). More complex models are built on top of simpler ones, thus providing an ablation study.

3.1 Simple binary classifier (SBC)

This approach does not model dependencies between candidates, thus, we simply estimate $p(q, t)$, where $t = c_i, i = 1, \dots, k$ with a transformer-based model. Following ([Garg et al., 2020](#)), we set the input as $q = \text{Tok}_1^q, \dots, \text{Tok}_N^q$ and $t = \text{Tok}_1^t, \dots, \text{Tok}_M^t$, where we start and end the input with [CLS] and [EOS] tags, respectively, and separate sentences with [SEP]. The rest follows the standard transformer logic. We use [CLS] to represent the embedding \mathbf{E} of (q, t) , and we use a softmax to model the probability of the question/candidate pair classification, as $p(q, t) = \text{softmax}(W \times \tanh(E(q, t)) + B)$. We fine-tune this model with log cross-entropy loss: $\mathcal{L} = -\sum_{l \in \{0,1\}} y_l \times \log(\hat{y}_l)$ on pairs of text, where y_l is the correct and incorrect answer label, $\hat{y}_1 = p(q, t)$, and $\hat{y}_0 = 1 - p(q, t)$. We start training from TANDA-RoBERTa (base or large), i.e., RoBERTa fine-tuned on ASNQ ([Garg et al., 2020](#)).

3.2 Pairwise Classifier (PC)

We use the previous TANDA-RoBERTa model similarly to what is done for a multiple-choice QA ([Zellers et al., 2018](#)). We proceed as in the previous section obtaining the CLS representation for each (q, c_i) pairs. Then, for each t , we concatenate the embedding of (q, t) with all the embeddings (q, c_i) , where $c_i \neq t$. This way, (q, t) is always in the first position. We train the model again using binary cross-entropy loss. At classification time, we select one candidate t at a time, set it in the first position, followed by all the others, classify all k target answers, and rerank them based on these scores.

3.3 All Candidate Multi-classifier (ACM)

We concatenate the question text with the text of all k answer candidates, i.e., $(q[\text{SEP}]_{c_1}[\text{SEP}]_{c_2} \dots [\text{SEP}]_{c_k})$, and provide this input to the same TANDA-RoBERTa model used for SBC. We use the final hidden vector E corresponding to the first input token [CLS] in a classification layer with weights $W \in \mathbb{R}^{k \times |E|}$, and train the model using a standard cross-entropy classification loss: $y \times \log(\text{softmax}(EW^T))$, where y is a one-hot vector representing labels for the k candidates, i.e., $|y| = k$. The scores for the candidate answers are calculated as $p(\{c_1, \dots, c_k\}) = \text{softmax}(EW^T)$. Then, we rerank c_i according to their probability.

3.4 Answer Support Reranker (ASR)

The previous models have been shown to be outperformed by ASR ([Zhang et al., 2021](#)), described in Figure 1. ASR consists of five main components: (i) the primary retrieval, which recuperates documents relevant to a question and produces answer sentence candidates, (ii) an SBC, which provides the embedding of the input (q, t) . This is built with the TANDA approach applied to RoBERTa pre-trained transformer ([Garg et al., 2020](#)). (iii) The joint representation of the pairs, $(t, c_i), i = 1, \dots, k, t \neq c_i$, where t and c_i are the top-candidates reranked by SBC, is obtained with a max-pooling operation over the k pairs, (t, c_i) . (iv) The *Answer Support Classifier* (ASC) classifies each (t, c_i) in four classes: (0) both answer correct, (1) t is correct while c_i is not, (2) vice versa, and (3) both incorrect. This multi-classifier is trained end-to-end with the rest of the network in a multi-task learning fashion, using its specific cross-entropy loss, computed with

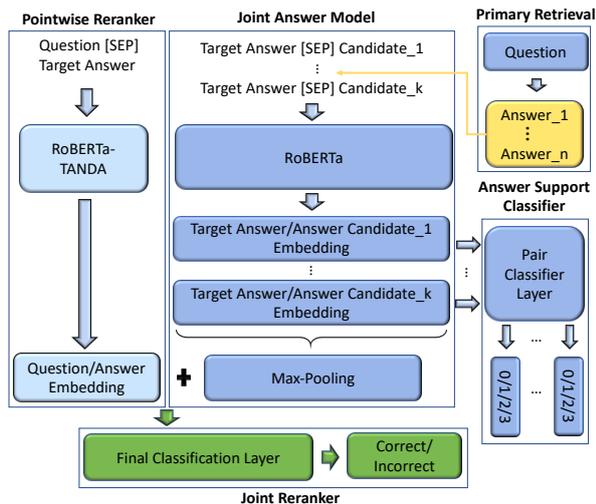


Figure 1: Answer Support-based Reranker (ASR)

the labels above. (v) The *Final Classification Layer* takes in input the concatenation of the SBC embedding with the max-pooling embedding. Thus, the classifier scores t with respect to q , also using the other candidates.

ASC uses pre-trained RoBERTa-base (Liu et al., 2019), to generate $[CLS] \in \mathbb{R}^d$ embedding of $(q, t) = E_t$. \hat{E}_i is the $[CLS]$ output of another RoBERTa-base Transformer applied to answer pairs, i.e., (t, c_i) . Then, E_t is concatenated to the max-pooling tensor from $\hat{E}_1, \dots, \hat{E}_k$, that is, $V = [E_t : \text{Maxpool}([\hat{E}_1, \dots, \hat{E}_k])]$, where $V \in \mathbb{R}^{2d}$ is the final representation of the target answer t . Finally, we apply a binary classification layer: $p(y_i|q, t, c_1, \dots, c_{k-1}) = \text{softmax}(WV + B)$, where $W \in \mathbb{R}^{2d \times 2}$ and B are parameters to transform the representation of the target answer t from dimension $2d$ to dimension 2, which represents correct or incorrect labels.

4 Double Reranking and Retrieval

ASR is the state of the art for joint modeling candidates. However, it suffers from three main limitations: (i) it needs to limit k otherwise the complexity may be too high, this means that it may not be able to process all available supporting candidates, (ii) the top k candidates are the best answer ranked by TANDA, which does not guarantee that these are also the best supports, and (iii) answer candidates may be good supports but they were not retrieved for this purpose. We address the above drawbacks proposing: (i) double reranking functions, which can efficiently rank supports as well as the best target answers, and (ii) a second stage of retrieval that

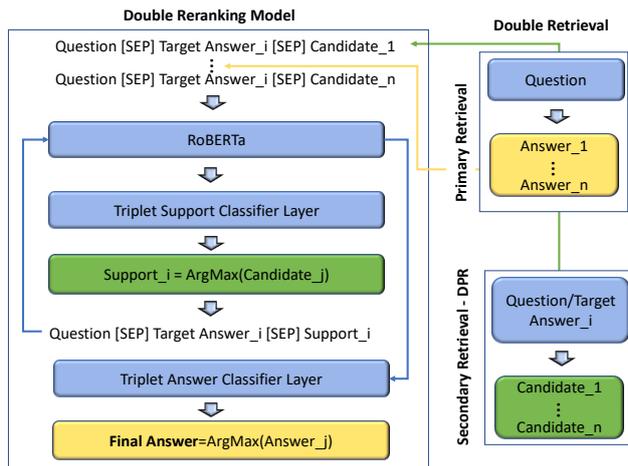


Figure 2: Double Answer Reranker and Retrieval (DAR-DR)

performs support retrieval using a representation of the target answer and question pairs.

4.1 Double Answer Reranking (DAR)

The architecture, shown in Fig. 2, is much simpler than ASR: it just uses one RoBERTa transformer to encode triplets, question, target answer, candidate, i.e., (q, t, c_i) , rather than encoding (q, t) and (t, c_i) with two separate transformer models. Then two classification layers operate two different types of ranking of the same triplets: the first, Support Ranker (SR), given t , learns to rank the best support, c_i higher. The second, Answer Ranker (AR), given the best support, i.e., $s_t = \arg\max_{i:c_i \neq t} SR(q, t, c_i)$, learns to rank the best answer producing, $f = \arg\max_{t \in C_k} AR(q, t, s_t)$, as the final output.

Training DAR Training SR and AR is challenging as, for the former, labels are typically not available in standard datasets. Additionally, defining a support, i.e., a piece of knowledge improving the accuracy of another classifier is not a well-understood problem. Thus, we use feedback from AR directly, i.e., a high relevant support is the one that produces the highest score in AR, if the answer is correct, and the lowest score, otherwise. We train SR and AR, at the same time, in a multi-task learning fashion, also considering that the triplets ranked by SR and AR are essentially the same: learning the different roles of SR and AR boils down from selecting a subset of triplets for their training, along with the appropriate loss function.

SR learns to rank the best supports higher. This can be enforced by requiring that s_t produces

the highest score, $AR(q, t, s_t)$, among c_i scores, $\{AR(q, t, c_i)\}_i$, if t is correct, and the lowest score, otherwise. We enforce this property with a loss function: given a training example, (q, C_k) , $C_k = \{c_1, \dots, c_k\}$, where c_i are associated with training label $l_i \in \{+1, -1\}$, we (i) select the best support, according to the current AR model, $s_t = \arg\text{-max}_{i:c_i \neq t} l_i \times AR(q, t, c_i)$, and (ii) use the following ranking loss function to train SR:

$$L(q, c_1, \dots, c_n) = -\log \frac{e^{\text{sim}(q, s_t)}}{\sum_{i=1}^n e^{\text{sim}(q, c_i)}}. \quad (1)$$

This pushes the support that provides the highest confidence score for AR in the top of the rank.

In contrast, we train AR as a standard binary classifier with the cross-entropy loss using all triplets, i.e., $(q, t, c_i) \forall t, c_i, t \neq c_i$.

4.2 Double Retrieval (DR) with DPR

The right side of Fig. 2 shows two retrieval steps: the first one is the traditional retrieval stage which, given an initial q , recuperates relevant documents, and splits them in answer sentence candidates. This step is typically carried out to build all AS2 datasets. However, if the objective is to retrieve items supporting a target t , the appropriate query should be built with the whole pair (q, t) . For this reason, we propose a secondary retrieval step using (q, t) . We note that (i) DAR approach does not limit the number of initial support to a fixed k as ASR does, either in training or in testing. This makes it suitable to work with more supporting items than those available from the first retrieval step. (ii) Since the semantics of (q, t) is difficult to define, neural retrieval fed with the embedding of the pair above is a promising choice.

Embeddings for support retrieval We adapted the Dense Passage Retrieval (DPR) by Karpukhin et al. (2020) for our task of support retrieval. We built two encoders $E_Q(\cdot)$ for the pairs (q, t) , and $E_P(\cdot)$ for text passages p (typically they are larger than a single sentence). The encoders map the input to a d -dimensional real-valued representation, while an indexing process computes representations for all text using $E_P(\cdot)$. The retrieval of relevant content for (q, t) is done in two steps: (i) we compute the (q, t) representation using $E_Q(\cdot)$; and (ii) we then retrieve M passages that have vector representations the most similar to the pair representation, in terms of dot product:

$$\text{sim}(q, p) = E_Q(q, t)^\top E_P(p). \quad (2)$$

The encoder is trained to make the dot-product similarity corresponding to the expected ranking. Thus, for training our DPR, we use again the ranking loss in Eq. 1, where the label of p is positive if a support is part of the paragraph, i.e., $s_t \in p$.

4.3 Double Ranking and Retrieval

The combination DAR-DR needs to consider the fact that AS2 datasets do not have annotated supports. For standard datasets, we consider candidates as potential supports, where the candidates are also annotated as correct or incorrect answers. In contrast, when we retrieve new support using the (q, t) query, no label is available. However, our DAR approach does not require support labels, thus we can still train our entire DAR-DR model, by simply considering two sets: initial candidates C , on which we can train AR, and a set S containing new supports retrieved by DPR. SR can be trained on $C \cup S$, using the ranking loss (Eq. 1), which only need to estimate the best support. Again, we find it with $s_t = \arg\text{-max}_i AR(q, t, c_i)$, where $t \in C$ and $c_i \in C \cup S \setminus t$.

5 Experiments

We compare our models with several baselines we implemented from previous work, and ASR, which is the current state of the art for AS2. For the evaluation, we used three different datasets traditionally used for AS2. Finally, we provide error analysis and model discussion.

5.1 Datasets

WikiQA is a QA dataset (Yang et al., 2015) containing a sample of questions and answer-sentence candidates from Bing query logs over Wikipedia. The answers are manually labeled. Some questions have no correct answers (*all-*), or only correct answers (*all+*). Table 2 reports the corpus statistics without *all-* questions, and without both *all-* and *all+* questions (*clean*). We follow the most used setting: training with the *noall-* mode and then answer candidate sentences per question in testing with the *clean* mode.

TREC-QA is another popular QA benchmark by Wang et al. (2007). Since the original test set only contain 68 questions and previous method already achieved ceiling performance (Zhang et al., 2021), we combined train., dev. and test sets, removed questions without answers, and randomly re-split into new train., dev. and test sets, which

	Train		Dev		Test	
	#Q	#A	#Q	#A	#Q	#A
no all-	873	8,672	126	1,130	243	2,351
clean	857	8,651	121	1,126	237	2,341

Table 2: WikiQA dataset statistics

contains 816, 204 and 340 questions, and 32,965, 9,591, and 13,417 question-answer pairs for the train., dev. and test sets, respectively.

SelQA is another benchmark for Selection-Based QA (Jurczyk et al., 2016), which composes about 8K factoid questions for the top-10 most prevalent topics among Wikipedia articles. We used the original splits for answer selection filed, which contain 5529 questions for train set, 785 questions for dev. set and 1590 questions for test set. SelQA is a large-scale dataset and it is more than 6 times larger than WikiQA in number of questions.

HotpotQA is a popular benchmark for multi-hop QA (Yang et al., 2018), which contains about 100,000 crowd-sourced questions that require reasoning over separate Wikipedia paragraphs. Each question not only has gold answer phrase but also has two supporting documents that contain the necessary evidence to infer the answer. To make it suitable for the AS2 task, we split paragraph into sentences, and label the sentences containing the gold answer phrase as correct answer, while considering the others as incorrect. For evaluation, we use the official dev-set-distractor as our test set.

5.2 Training and testing details

Metrics The performance of QA systems is typically measured with Accuracy in providing correct answers, i.e., the percentage of correct responses, which also refers to Precision-at-1 (P@1) in the context of reranking. We also use Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) evaluated on the test set, using the entire set of candidates for each question (this varies according to the dataset), to have a direct comparison with the state of the art.

Models We use the pre-trained RoBERTa-Base (12 layer) and RoBERTa-Large-MNLI (24 layer) models, which were released as checkpoints for use in downstream tasks¹.

Reranker training We adopt Adam optimizer (Kingma and Ba, 2014) with a learning rate of 2e-5

¹<https://github.com/pytorch/fairseq>

for the transfer step on the ASNQ dataset (Garg et al., 2020), and a learning rate of 1e-6 for the adapt step on the target dataset. We apply early stopping on the development set of the target corpus for both fine-tuning steps based on the highest MAP score. We set the max number of epochs equal to 3 and 9 for the adapt and transfer steps, respectively. We set the maximum sequence length for RoBERTa to 128 tokens.

ASR training Again, we use the Adam optimizer with a learning rate of 2e-6 for training the ASR model on the target dataset. We utilize one Tesla V100 GPU with 32GB memory and a train batch size of eight. We use two transformer models for ASR: a RoBERTa Base/Large for PR, and one for the joint model (see Fig. 1). We set the maximum sequence length for RoBERTa to 128 tokens and the number of epochs as 20. We select the best k chosen in (Zhang et al., 2021).

DAR implementation and training For training the DAR model, we also use the Adam optimizer but with a different learning rate, 5e-6. We utilize two Tesla A100 GPUs with 40GB memory and a train batch size of 128. DAR only needs one transformer model: a RoBERTa Base/Large (see Fig. 2). The maximum sequence length and the number of epochs are the same with ASR training, which are 128 and 20 separately.

DPR implementation and training We utilize the same training configuration of the original DPR in Karpukhin et al. (2020). Then, we used it to build a large index having up to 130MM passages extracted from 54MM documents of Common-Crawl². We selected English Web documents of 5,000 most popular domains, including Wikipedia, from the recent releases of Common Crawl of 2019 and 2020. We then filtered pages that are too short or without proper HTML structures, i.e., having title and content. To retrieve to N candidates, we input our DPR with (q, t) pairs as query to retrieve top 1000 passages.

DAR-DR implementation and training The training configuration is similar to DAR training with the different steps highlighted in Sec. 4.2. For each (q, c_i) of our datasets, we used our DPR for retrieving 1000 supporting paragraphs, which are then split into sentences, s . We rank s according to a $E_Q(q, t) \cdot E_P(s)$, where $E_P(s)$ provides the

²commoncrawl.org

RoBERTa Base	WikiQA				TREC-QA				SelQA				HotpotQA			
	P@1	RER	MAP	MRR	P@1	RER	MAP	MRR	P@1	RER	MAP	MRR	P@1	RER	MAP	MRR
TANDA (Garg et al.)	–	–	0.8890	0.9010	–	–	–	–	–	–	–	–	–	–	–	–
ASR (Zhang et al.)	0.8436	13.64%	0.9014	0.9123	–	–	–	–	–	–	–	–	–	–	–	–
SBC	0.8189	0.00%	0.8860	0.8983	0.8824	0.00%	0.8979	0.9277	0.9302	0.00%	0.9512	0.9587	0.6598	0.00%	0.7576	0.7685
ACM	0.7819	-20.43%	0.8542	0.8684	0.8824	0.00%	0.8942	0.9272	0.9308	0.86%	0.9511	0.9589	0.6597	-0.03%	0.7574	0.7681
PC	0.8272	4.58%	0.8927	0.9045	0.8882	4.93%	0.9000	0.9319	0.9302	0.00%	0.9514	0.9587	0.6718	3.53%	0.7644	0.7750
ASR (ours)	0.8436	13.64%	0.9014	0.9123	0.9088	22.45%	0.9036	0.9420	0.9314	1.72%	0.9519	0.9592	0.6795	5.79%	0.7724	0.7812
ASR-Rank	0.8436	13.64%	0.9012	0.9108	0.9088	22.45%	0.9181	0.9445	0.9296	-0.86%	0.9503	0.9580	0.6768	5.00%	0.7742	0.7824
DAR	0.8519	18.22%	0.9011	0.9136	0.9118	25.00%	0.9181	0.9446	0.9415	16.19%	0.9592	0.9653	0.6844	7.23%	0.7754	0.7854
DAR-DR	0.8560	20.49%	0.9051	0.9164	0.9176	29.93%	0.9233	0.9493	0.9484	26.07%	0.9616	0.9687	0.6832	6.88%	0.7729	0.7832

Table 3: Performance of different models using RoBERTa base Transformer on WikiQA, TRECQA, SelQA and HotpotQA. RER is the relative error reduction on P@1. The difference between P@1 of DAR and DAR-DR and P@1 of all the other systems is statistically significant at 95%.

embedding representation of each s , even though we trained $E_P(\cdot)$ for passages. We select the top 10 sentences as support for all the experiments with DAR-DR. It should be noted that all datasets for retrieval-based QA are based on candidates retrieved with an initial search engine, e.g., Bing, Google, TREC systems. This constitutes the first standard retrieval in our DR approach.

5.3 Comparative/ablated results

We design a set of baselines (see Sec. 3), which also constitute the best ablation systems of our most complex architecture DAR-DR. Indeed, **SBC** is our reimplementation of TANDA, which corresponds to the basic system (or basic component) of our architecture, it uses only one reranker and no joint inference. **PC** is the simplest joint model, which still uses only one classifier as SBC but applied to pairs of answers. **ASR** (ours) is our reimplementation of ASR, which uses an SBC model, a PC model, and an internal SR (called ASC) model as in DAR, used just for classification, no ranking. **ASR-Rank** extends ASR using the top 3 candidates re-ranked by ASC category 0 score (see Sec. 3.4), instead of using the standard TANDA rank. We introduced, ASR-Rank to show an approach similar to DAR. **ACM** is a joint model over all k candidates (theoretically more expressive than just joint models over pairs). **DAR** uses two rerankers as ASR-Rank but only one transformer and our approach to train them. Finally, **DAR-DR** adds to DAR new candidates retrieved by DPR.

Main results Table 3 reports P@1, MAP and MRR of models on WikiQA, TREC-QA, SelQA and HotpotQA datasets. TANDA and ASR rows report the results obtained by Garg et al. (2020) and Zhang et al. (2021), respectively, which certify the alignment between our and previous work setting and implementation. We note that:

- (i) P@1, MAP and MRR correlate well, thus, we can focus our analysis on P@1, which typically provides the QA performance. The AS2 model P@1 numbers are in the lower 80s% for all datasets but HotpotQA. This means that absolute improvements are not expected to be large, thus we also report the relative error reduction (RER) for P@1, which better shows model differences.
- (ii) Our SBC and ASR replicate the performance reported in previous work (WikiQA and TREC-QA), which are the previous state of the art.
- (iii) We confirm that ASR, using candidate pairwise information greatly improves on single answer classification models, e.g., we observe a relative error reduction of 13.64% (from 81.89 to 84.36) over TANDA and SBC, which do not use the information from other candidates.
- (iv) Our proposed model DAR significantly reduces the error of QA systems with respect to ASR by 4.58% (from 84.36 to 85.19), 2.55% (from 90.88 to 91.18), 14.47% (from 93.14 to 94.15), and 1.44% (from 67.95 to 68.44) on WikiQA, TREC-QA, SelQA, and HotpotQA, respectively. It is interesting to note that DAR only uses the half of the parameters of ASR (125M vs. 250M). The combination between the two rerankers for answer and support generates more selective information than max-pooling pairwise embeddings.
- (v) To verify that the unique feature of DAR of effectively combining training examples and their losses is a key element, we implemented ASR-Rank, which also selects supporting candidates for ASR, using its internal answer pair classifier, $ASC(t, c_i)$. The results derived on WikiQA and TREC-QA show no difference between ASR and ASR-Rank, while the latter underperforms on SelQA. This shows that the improvement produced by DAR is not about selecting the best support in absolute, but it is about selecting the support that

Roberta Large	WikiQA				
	P@1	RER	MAP	MRR	Param.
SBC	0.8724	0.00%	0.9151	0.9266	355M
ASR	0.8971	19.36%	0.9280	0.9399	710M
DAR	0.8889	12.93%	0.9230	0.9362	355M
DAR-DR	0.8930	16.14%	0.9241	0.9375	355M

Table 4: Results on WikiQA using RoBERTa Large.

can produce the highest confidence in the answer selector (see Sec. 4.1).

(vi) DAR-DR introduces 10 additional supports to DAR processing, retrieved with our modified DPR approach. These new candidates do not have any label indicating if they are good or bad support. They are automatically ranked with the DAR approach. The results show an RAR of 2.27%, 4.93%, and 9.88%, on WikiQA, TREC-QA, and SelQA, respectively. Suggesting that retrieving supporting candidates for (q, t) can be very effective. HotpotQA does not benefit from retrieving candidates external to the dataset as the original candidate set always contains at least one correct support by construction, thus no additional retrieval is needed.

(vii) Finally, we perform randomization test (Yeh, 2000) to verify if the models significantly differ in terms of prediction outcome. Specifically, for each model, we compute the best answer for each question and derive binary output based on the ground truth. We then follow the randomization test to measure the statistical significance between two models. We use 100,000 trials for each calculation. The test show statistical significant difference of DAR and DAR-DR vs. all the other models over all datasets but HotpotQA, with $p < 0.05$, and between DAR and DAR-DR on SelQA.

Results with large models We experimented with SBC, ASR, DAR and DAR-DR models implemented on a larger transformer, i.e., RoBERTa Large, on WikiQA. Table 4 reports the comparative results: SBC and ASR replicate the results by Zhang et al. (2021), i.e., a P@1 of 87.24% and 89.71%, respectively; the latter is the state of the art on WikiQA with a P@1 of 89.71%. Both DAR and DAR-DR improve SBC up to 20% RAR. However, even DAR-DR is behind ASR, by about 3.21% of RER. This different outcome with respect previous results on the RoBERTa base can be explained by looking at the column reporting model parameters. As before, ASR uses the double of parameters of DAR, however, in this case the number of parameters is 710M, which is a large number in absolute:

q : what is the measurements of saturn 's moons?
c_1 : The rings of Saturn are made up of objects ranging in size from microscopic to hundreds of meters, each of which is on its own orbit about the planet.
c_2 : Saturn has 62 moons with confirmed orbits , 53 of which have names and only 13 of which have diameters larger than 50 kilometers.
c_3 : The moons of Saturn (also known as the natural satellites of Saturn) are numerous and diverse ranging from tiny moonlets less than 1 kilometer across to the enormous Titan which is larger than the planet Mercury.
c_4 : Saturn has seven moons that are large enough to be ellipsoidal due to having planetary mass , as well as dense rings with complex orbital motions of their own.

Table 5: A question with answer candidates; c_2 and c_3 are correct.

although DAR is a better model, it can hardly improve a model with 355M parameters more.

5.4 Model discussion and error analysis

Tab. 5 shows a question with the rank provided by SBC. The top-1 answer, c_1 is incorrect, as it refers to objects of Saturn’s rings, instead of targeting its moons. SBC probably got tricked by the phrase *ranging in size*. ASR also selected c_1 using the support of the top 3 candidates selected by SBC, i.e., c_2 , c_3 , and c_4 . These candidates support c_1 as they provide more context, e.g., *moon*, which is not in c_1 but it is required in the question. The main problem of ASR is the fact that correct answers also tend to support imperfect but reasonable answers such as c_1 . In contrast, for each t , DAR learns to select the best support: in the example, it selects the correct answer c_2 using c_4 as support. This probably provides phrases such as *seven moons that are large enough* supporting c_2 phrases such as *have diameters larger than*.

In Tab. 6, we see an example, in which SBC ranks an incorrect answer at the top. It probably prefers c_1 to the correct answer c_2 because it matches the main question entity and verb, i.e., *Family Guy* and *premier*, while c_2 does not contain explicit reference to the main entity. Also ASR and DAR cannot select c_2 , as the available supports, c_1 and c_3 , do not provide any useful information. In contrast, DAR-DR can use new retrieved support, i.e., s_1 , which contains the main entity and reinforces the information in c_2 , i.e., *22 millions*.

See Appendix for more discussion.

6 Conclusion

In this paper, we propose, DAR, a transformer architecture based on two reranking heads: (i) the answer reranker (AS2 model) and the answer support

q:	How many viewers did "Family Guy" premier to?
c₁:	Family Guy officially premiered after Fox's broadcast of Super Bowl XXXIII on January 31, 1999, with "Death Has a Shadow."
c₂:	The show debuted to 22 million viewers, and immediately generated controversy regarding its adult content.
c₃:	At the end of its first season, the show was #33 in the Nielsen ratings, with 12.8 million households tuning in.
s₁:	Family Guy has been around since 1999 with 11 seasons to date, the viewing rates have dropped from over 22 millions to 7 million.

Table 6: Example with c_2 correct.

reranker. We optimize the latter imposing a loss function that penalizes non optimal support for the target answer, thus avoiding the need of defining and manually labeling supporting data. Additionally, we introduce a second retrieval stage based on DPR, where we optimize the score function between answer/question pair and the retrieving passage. The experiments with four well-known datasets show consistent improvement of DAR over the state of the art, and the potential benefit of the secondary retrieval, achieving up to 14.47 of relative error reduction (on SelQA). We will release software, models, and the DPR retrieved data for all datasets for fostering research in this field.

Limitations

We propose a new QA architecture that operate a second retrieval. This can make the approach slower than a standard QA system using only one retrieval but, at the same time, it enables the possibility to retrieve critical information. The latter can be used to verify question/answer pairs or also complement the information need of the user. This is clearly a future direction for QA/personal assistant systems. As we explain in the paper, we designed a DPR model which can specifically retrieve supporting items (no just answer candidates), as we can query DPR with the pair (question, answer to be verified). This is a major novelty with respect to systems that can only retrieve text relevant to the question.

Our new approach uses only one support to verify answer correctness. This may be seen as lack of exploration of the model potential. However, using one support only requires a classifier of the form $SR(q, t, s_i)$. If we use more supports, for example two, we will have a classifier of the type $SR(q, t, s_i, s_j)$. This means that to find the arg-max we would need to iterate over k^2 , where k is the number of candidates (so in general k^n with n the number of supports we want to use). This is much

less efficient than our approach. Although, approximated solutions more efficient than $O(k^n)$ can be surely designed, in this paper, we have focused on a rather efficient version, which has also shown to improve the state of the art.

References

- Daniele Bonadiman and Alessandro Moschitti. 2020. [A study on efficiency, accuracy and document structure for answer sentence selection](#). *CoRR*, abs/2003.02349.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. [Reading wikipedia to answer open-domain questions](#). *CoRR*, abs/1704.00051.
- Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. [TANDA: transfer and adapt pre-trained transformer models for answer sentence selection](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7780–7788. AAAI Press.
- Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. [Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension](#). *CoRR*, abs/1906.04618.
- Zan-Xia Jin, Bo-Wen Zhang, Fang Zhou, Jingyan Qin, and Xu-Cheng Yin. 2020. [Ranking via partial ordering for answer selection](#). *Information Sciences*.
- T. Jurczyk, M. Zhai, and J. D. Choi. 2016. [Selqa: A new benchmark for selection-based question answering](#). In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 820–827.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#). *CoRR*, abs/1412.6980.
- Bernhard Kratzwald and Stefan Feuerriegel. 2018. [Adaptive document retrieval for deep question answering](#). In *EMNLP'18*, pages 576–581.
- Md Tahmid Rahman Laskar, Jimmy Xiangji Huang, and Enamul Hoque. 2020. [Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task](#). In *Proceedings of*

- The 12th Language Resources and Evaluation Conference*, pages 5505–5514, Marseille, France. European Language Resources Association.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. 2019. [Answering complex open-domain questions through iterative query generation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2590–2602, Hong Kong, China. Association for Computational Linguistics.
- Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 1913–1916.
- Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR'15*.
- Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. [Inter-weighted alignment network for sentence pair modeling](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189, Copenhagen, Denmark. Association for Computational Linguistics.
- Harish Tayyar Madabushi, Mark Lee, and John Barnden. 2018. [Integrating question classification and deep learning for improved answer selection](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3283–3294, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. [FEVER: a large-scale dataset for fact extraction and VERification](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.
- E. Voorhees and D. Tice. 1999. *The TREC-8 Question Answering Track Evaluation*, pages 77–82. Department of Commerce, National Institute of Standards and Technology.
- Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. [What is the Jeopardy model? a quasi-synchronous grammar for QA](#). In *EMNLP-CoNLL'07*, pages 22–32, Prague, Czech Republic. Association for Computational Linguistics.
- Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, et al. 2020. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*.
- Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Alexander S. Yeh. 2000. [More accurate tests for the statistical significance of result differences](#). *CoRR*, cs.CL/0008005.
- Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2019. [A compare-aggregate model with latent clustering for answer selection](#). *CoRR*, abs/1905.12897.
- Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. 2021. [Joint models for answer verification in question answering systems](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3252–3262. Association for Computational Linguistics.

A Deeper Discussion

A.1 Double Retrieval

The complete architecture DAR-DR can operate a second retrieval, which can make the approach slower than a standard system using only one retrieval but, at the same time, it enables the possibility to retrieve critical information. The latter can be used to verify question/answer pairs or also complement the information need of the user. This is clearly a future direction for QA/personal assistant systems. As we explain in the paper, we designed a DPR model which can specifically retrieve supporting items (no just answer candidates), as we can query DPR with the pair (question, answer to be verified). This is a major novelty with respect to

systems that can only retrieve text relevant to the question.

It should be noted that we apply two answer sentence retrieval steps: (i) the standard one, which is contained in all datasets for AS2 based QA systems. See our description of WikiQA, TREC, SelQA, and HotpotQA. For example, WikiQA uses Bing to retrieve passages. (ii) Our innovative retrieval based on our new DPR model. This takes (q, a) as query and returns passages that have higher probability to be good support for a with respect to q .

Our DAR-DR aims to be an end-to-end system, AS2 tasks are defined using retrieval systems. We also operate the second retrieval. In other words, a DAR-DR system deployed in production will always performs 2 stages of retrieval to provide answers to users.

A.2 AS2 Tradition

Please note that the AS2 research our paper builds on has been contributed for more than 20 years. It started in TREC competitions (QA track 1999). It has been revived in 2007 with the specialization of passage reranking in answer sentence selection (AS2): see for example the systems based on TREC data [https://aclweb.org/aclwiki/Question_Answering_\(State_of_the_art\)](https://aclweb.org/aclwiki/Question_Answering_(State_of_the_art)). Also note that TANDA by Garg et al, 2020 is simply a transformer fined-tuned in two steps (i) on ASNQ dataset proposed by the same authors, and (ii) on the target dataset.

A.3 Ablation study

The baseline models we implemented and compared to are ablated versions of our systems, sometimes including different alternatives (instead of just excluding some features). Sec. 5.3 explains how the different models we test constitute an excellent ablation study.

A.4 Usefulness of reporting result with Relative Error Reduction

The relative error reduction is suitable for reporting the performance in our setting since we are improving state-of-the-art systems with performance ranging from $\sim 81\%$ to $\sim 97\%$ (depending on the measure and datasets). Reporting absolute (or also relative) improvement does not capture the complexity of the task. For example, improving a system from 30% to 31% (margin of improvement 70%) is completely different than improving a system from 98% to 99%, where the margin of im-

provement is 2%. Relative error reduction, which we use, accounts for such difficulties. In any case, whatever lens one uses, the results are statistical significant, showing that we improve the state of the art.

A.5 Model Effectiveness

We report the number of model parameters on the Table 4, which shows that our solution uses half of the parameters of previous state of the art, ASR (indeed that uses two transformer models: instead of our DAR only uses one).

A.6 Multiple supports

Using one support only requires a classifier of the form $SR(q, t, s_i)$. If we use more supports, for example two, we will have a classifier of the type $SR(q, t, s_i, s_j)$. This means that to find the arg-max we would need to iterate over k^2 , where k is the number of candidates (so in general k^n with n the number of supports we want to use). This is much less efficient than our approach. Although, approximated solutions more efficient than $O(k^n)$ can be surely designed, in this paper, we have focused on a rather efficient version, which has also shown to improve the state of the art.

Moreover, although it may happen that to verify information multiple pieces are required, this situation is rather rare in general open QA, as the web contains the needed information in a redundant fashion. This means we can most times retrieve a compact version of an answer *why should it be available only in a fragmented way?*

For other more specific application scenarios, e.g., deriving answers from several axioms and logic formulas (expressed in text format), combinations of different supports, composing different retrieved pieces may be required. However, this scenario is out of the scope of our paper: it can be an interesting new research.

A.7 Comparison with (Zhang et al., 2021)

(Zhang et al., 2021) is a great work which outperforms the previous state of the art in AS2, i.e., TANDA, which seemed very difficult to improve. Our contributions:

- We defined a new techniques to automatically learn to rank support without using any annotation, which is for example used in hotpotQA.

- Our approach outperforms (Zhang et al., 2021) on base architectures. With respect to LARGE, we did not have the language models comparing with the same number of parameters but using 355M parameters less, our approach provide close results.
- Our RoBERTa-base model only use half of the parameters of (Zhang et al., 2021), which these days of energy crisis is absolutely important results.
- Most importantly, we defined a new paradigm for QA (and answer verification), which uses double retrieval, our support reranker can be used to select support obtained with a second stage of retrieval. This to our knowledge is completely new for answer sentence selection.

Our approach improves all previous techniques in a fair comparison, which means similar number of parameters. In case of RoBERTa based, our approach outperforms models with the double of parameters. Specifically, only using 130M parameters, it outperforms an architecture of 260M parameters, i.e., architectures having 130M parameters more. The fact that our approach does not perform an architecture of 710M parameters, i.e., a model that used 355M more than ours, is not a limitation. We show these unfair experiments (for our approach) because they provide strong evidence about the effectiveness of our approach. The lower performance on HotpotQA is expected as the related task is not answer sentence selection, for which our approach was built. Indeed, HotpotQA focused on entities and annotated data such that two paragraphs complement each other, which is a more restrictive assumption than our approach. For the sake of generality, we showed that our approach can also work well for this rather different setting.

Evaluating the Diversity, Equity and Inclusion of NLP Technology: A Case Study for Indian Languages

Simran Khanuja*
Carnegie Mellon University[†]
skhanuja@andrew.cmu.edu

Sebastian Ruder*
Google Research
ruder@google.com

Partha Talukdar
Google Research
partha@google.com

Abstract

In order for NLP technology to be widely applicable, fair, and useful, it needs to serve a *diverse* set of speakers across the world’s languages, be *equitable*, i.e., not unduly biased towards any particular language, and be *inclusive* of all users, particularly in low-resource settings where compute constraints are common. In this paper, we propose an evaluation paradigm that assesses NLP technologies across all three dimensions. While diversity and inclusion have received attention in recent literature, equity is currently unexplored. We propose to address this gap using the Gini coefficient, a well-established metric used for estimating societal wealth inequality. Using our paradigm, we highlight the distressed state of current technologies for Indian (IN) languages (a linguistically large and diverse set, with a varied speaker population), across all three dimensions. To improve upon these metrics, we demonstrate the importance of region-specific choices in model building and dataset creation, and more importantly, propose a novel, generalisable approach to optimal resource allocation during fine-tuning. Finally, we discuss steps to mitigate these biases and encourage the community to employ multi-faceted evaluation when building linguistically diverse and equitable technologies.

1 Introduction

NLP has seen large advances in recent years driven by the rapid progress in transfer learning (Ruder et al., 2019; Devlin et al., 2019). The benefits of these advances, however, are not equally distributed across the world’s languages (Joshi et al., 2020) and users. While linguistic diversity and inclusion have evolved to be a pressing concern today, measures to quantify these are still lacking. The progress of any field is tightly coupled with

its evaluation paradigm and the community is incentivized to work on highly visible metrics and benchmarks. In order for users around the world to reap the benefits of NLP technology, we must move from an evaluation that focuses on optimizing raw performance on available test data to a more holistic user-centric evaluation (Ethayarajh and Jurafsky, 2020; Ruder et al., 2021). In this paper, we attempt to do so by defining an evaluation framework along three dimensions: diversity, equity, and inclusion.¹

Diversity is important as NLP technology should be available to speakers of any language (European Language Resources Association, 2019). To this end, recent work (Blasi et al., 2022) quantifies diversity of NLP technology across the world’s languages by weighing normalized task performance for each language based on its speaker population.

Equity is key as we should aim to develop technology that does not discriminate against speakers of any particular language (Kaneko and Bollegala, 2019). State-of-the-art multilingual models in fact have been shown to perform much better in languages with access to many pre-training resources (Hu et al., 2020). To measure such performance inequity across languages, we propose to use the Gini coefficient (Dorfman, 1979), a measure that has been used to represent the income inequality within social groups.

Finally, inclusion is a concern as the fact that NLP technology is performant in a given task and language does not mean that it is usable by all. State-of-the-art models are becoming larger and larger (Fedus et al., 2021) and the low-resource settings of many languages often coincide with constraints on computational resources (Ahia et al., 2021). The value a technology provides to a user thus also needs to consider how easily such technol-

*Equal contribution.

[†]Work done at Google Research.

¹We focus on assessing these dimensions on the *language level*. Prior work on equity focuses mainly on subpopulations *within* a language (Katell et al., 2020).

ogy can be deployed in practice. [Ma et al. \(2021\)](#) quantify this based on a model’s runtime efficiency, considering factors like throughput and memory.

Our proposed paradigm is language and model agnostic making it applicable to an arbitrary set of languages and models. We apply our paradigm to highlight the distressing state of current technologies for Indian (IN) languages. India is a multilingual society with 1369 rationalized languages and dialects being spoken across the country ([Chandramouli, 2011](#)). Of these, 22 scheduled languages² spoken by almost 97% of the population hold an official recognition and 121 languages have more than 10,000 speakers. Additionally, 21.92% of its population lives below the poverty line ([RBI, 2021](#)). Serving this large varied population justly requires a multi-faceted effort and basing our case study on IN languages directs the way forward.

We evaluate a range of state-of-the-art models and transfer settings ([Hu et al., 2020](#)) across four standard downstream tasks: *Named Entity Recognition* (NER), *Part-of-Speech Tagging* (POS), *Natural Language Inference* (NLI) and *Question Answering* (QA). We observe that region-specific choices, i.e., a) region-specific pre-trained models ([Kakwani et al., 2020](#); [Khanuja et al., 2021](#)) and b) Hindi as the transfer language during fine-tuning, generally yield the best results. In terms of efficiency, we find that smaller models are preferable for easier, syntactic tasks while larger models have the edge on more complex, semantic tasks.

Our findings, however, also highlight that we are still a long way from building perfectly inclusive and equitable NLP technology. Towards bridging this gap, we explore how we can most effectively annotate data for the remaining languages. Past work ([Lin et al., 2019](#); [Ahuja et al., 2022](#)) has relied on heuristic and feature-based approaches to source language selection. In our work, we propose a novel, fully computational approach to model the space of source and target languages, and derive the optimal allocation of a fixed annotation budget to maximize performance on our proposed metrics in a multi-source setting.

Our contributions are the following: **1)** We propose a holistic evaluation paradigm that assesses NLP technology based on their diversity, equity,

and inclusion. **2)** Using this paradigm, we evaluate model capabilities for IN languages and quantify their shortcomings. **3)** We propose a novel approach to select data for fine-tuning these models with the objective of maximizing performance on the proposed metrics. **4)** We discuss steps that must be taken to mitigate these biases and call upon the community to incorporate our evaluation paradigm when building models to track progress towards building linguistically inclusive and diverse technologies.

2 Background and Related Work

Multilingual Models Transformer-based language models (LMs) ([Vaswani et al., 2017](#)) trained on massive amounts of text from multiple languages have enabled the inclusion of an unprecedented number of languages in NLP technologies ([Conneau et al., 2020](#); [Devlin et al., 2018](#)). However, previous research has shown that these models do not serve all languages equally, with resource-poor languages in the long tail suffering the most ([Hu et al., 2020](#); [Lauscher et al., 2020](#)). These models go through a critical step of fine-tuning for the downstream task before being deployed. Several recent works focus on optimal fine-tuning strategies that mitigate transfer gaps and improve overall performance across target languages. [Lin et al. \(2019\)](#) propose a tool that chooses optimal transfer languages based on linguistic features. [Lauscher et al. \(2020\)](#) demonstrate the effectiveness of investing in few-shot in-language training examples. Recently, [Debnath et al. \(2021\)](#) show that investing in an equal number of fine-tuning instances across target languages performs best. These past approaches however, have all been heuristically designed based on the knowledge and intuition of the experimenter.

User-centric Evaluation At its core, the need for language diversity in technologies is tied to the people it serves. Previous work ([Ethayarajh and Jurafsky, 2020](#); [Ma et al., 2021](#)) has highlighted the need for transparent and user-centric leaderboard evaluation, reporting practically relevant statistics such as model size, energy efficiency, and inference latency. It is common for speaker populations of under-represented languages to operate in resource-constrained settings. Therefore, in addition to evaluating *linguistic* diversity, we follow [Ma et al. \(2021\)](#) in computing model efficiency, which serves to assess the *inclusivity* of these technologies. With regards to linguistic di-

²Assamese, Bengali, Bodo, Dogri, Gujarati, Hindi, Kashmiri, Kannada, Konkani, Maithili, Malayalam, Manipuri, Marathi, Nepali, Oriya, Punjabi, Tamil, Telugu, Sanskrit, Santali, Sindhi, Urdu

iversity, Ruder et al. (2021) highlight the need for more fine-grained evaluation across languages and introduce language-specific leaderboards. Blasi et al. (2022) quantify the value of NLP technology weighed by speaker population and determine utilities of several technologies across the world’s languages. Choudhury and Deshpande (2021) propose strategies for fair and efficient model selection depending on one’s application, based on the principles of fairness in economics and social choice theory.

Indian Languages The research community has actively been contributing to the advancement of IN NLP by collecting and open-sourcing data (Kakwani et al., 2020; Ramesh et al., 2021; Abraham et al., 2020; Roark et al., 2020; Kunchukuttan et al., 2017; Khanuja et al., 2020a), building region-specific multilingual models (Khanuja et al., 2021; Kakwani et al., 2020; Ramesh et al., 2021) and creating evaluation benchmarks (Kakwani et al., 2020; Khanuja et al., 2020b)³ Several of these efforts have been undertaken by AI4Bharat⁴, a non-profit open-source community that has additionally been working on developing resources for IN signed languages (Sridhar et al., 2020) and creating key-boards for IN scripts. Recently, Google Research India launched a question answering (QA) challenge named ChAII⁵. Microsoft Research India has also made significant contributions to IN NLP with several efforts directed towards code-mixed language processing⁶ and building tools and datasets for under-represented languages in India⁷.

3 Diversity, Equity and Inclusion (DEI)

There is increasing awareness in society to promote diversity, equity and inclusion in our workforce, wherein such measures have recently been enforced by law (Constitution, 2021). In the social construct, *diversity* is defined as “the practice of including the many communities, identities, races, ethnicities, backgrounds, abilities, cultures, and beliefs of the people, including underserved communities”, *equity* refers to “the consistent and systematic fair, just, and impartial treatment of all individuals” and *inclusion* means “the recognition, and use of the

³https://github.com/AI4Bharat/indicnlp_catalog maintains a list of resources for Indian NLP.

⁴<https://ai4bharat.org/>

⁵<https://www.kaggle.com/c/chaii-hindi-and-tamil-question-answering>

⁶<https://www.microsoft.com/en-us/research/project/melange>

⁷<https://www.microsoft.com/en-us/research/project/ellora>

talents and skills of employees of all backgrounds” (Constitution, 2021). Given the ubiquitous use of technology in our daily lives, we as technology makers hold the responsibility of making sure all voices are heard and equally represented in the technology we serve. Given that our research community is incentivized to work on highly visible metrics and benchmarks, an important first step is to encourage evaluation along these dimensions. Previous work mainly focused on average performance (as measured by accuracy or F1 for NLU tasks), which is not indicative of differences in DEI. Hence, while models claim state of the art based on an increase in average performance, this increase may only be due to making the “rich richer” (see Table 4).

We propose an evaluation paradigm for current NLP technology that operationalizes the well-established diversity, equity and inclusion pillars on a language level: we quantify diversity based on the value diverse speaker populations derive from a technology, equity based on egalitarian performance across speaker populations, and inclusion based on a technology’s accessibility. We employ metrics of Blasi et al. (2022) and Ma et al. (2021) to measure diversity and inclusion respectively and propose a new metric to quantify equity. We describe the metrics in more detail below:

3.1 Diversity: Utility, Demand and the Global Metric

The global metric introduced by Blasi et al. (2022) helps quantify linguistic diversity. Formally, this metric is composed of the utility of a technology weighed by its demand. The utility u_1 of a system for a task and language is its performance normalized by the best possible performance (typically, human-level performance) afforded by the task:

$$u_1 = \frac{\text{performance}_1}{\text{theoretical max performance}}$$

Demand d_1 is characterized by taking into consideration demographic and linguistic perspectives. Under the demographic perspective, the demand for a given technology in a language is estimated to be proportional to the number of speakers of the language itself n_1 ($d_1 \propto n_1$). Under the linguistic perspective, the demand across languages is identical ($d_1 \propto 1$). These two alternatives, as well as any intermediate combination of them, are parameter-

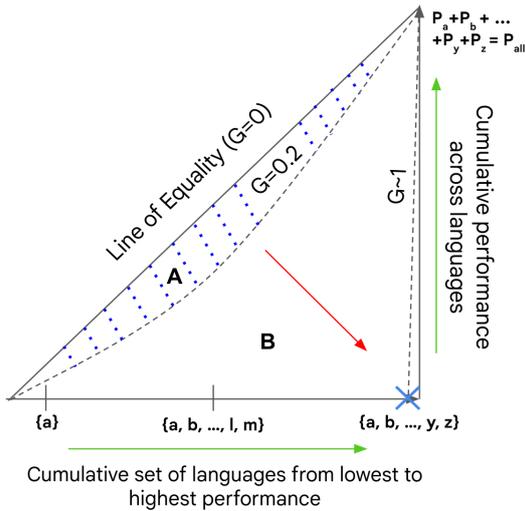


Figure 1: *Graphical Representation* of the Gini coefficient (G), given by $A/(A + B)$ when $G = 0.2$. Each point on the graph depicts the proportion of the total (cumulative) performance P_{all} (e.g., accuracy, F1, etc) that is achieved by the bottom $n\%$ of languages combined. Assume we have a language set $\{a, b, \dots, y, z\}$ with performances $P_a \leq P_b \leq \dots \leq P_y \leq P_z$ and $P_a + P_b + \dots + P_y + P_z = P_{all}$. When all languages perform the same, i.e., $P_a = P_b = \dots = P_y = P_z$, $G = 0$, as represented by the line of equality, i.e., the bottom $n\%$ of languages also account for $n\%$ of the total performance. The value of G increases as the disparity in performance between all languages increases, and approaches unity in the case of perfect inequality (here, this would mean $P_a = P_b = \dots = P_y = 0$ and $P_z = P_{all}$), i.e., the model / application supports only one language. See §3.2 for details.

ized through a single exponent τ :

$$d_1^{(\tau)} = \frac{n_1^\tau}{\sum_{l' \in L} n_{l'}^\tau}$$

where $\tau = 1$ corresponds to a demographic notion of demand and $\tau = 0$ to a linguistic one. The global metric can now be defined as:

$$M_\tau = \sum_{l \in L} d_1^{(\tau)} \cdot u_l$$

In essence, $M_\tau = 0$ means that no user benefits from language technology and $M_\tau = 1$ corresponds to each language user enjoying perfect technology. Given our people-centric aim to measure benefit for all speakers, we employ the demographic notion of demand ($M_{\tau=1}$).

3.2 Equity: Gini Coefficient

While diversity accounts for a language’s speaker population, it does not take into account inequal-

ities in the performance across languages. While several past works have highlighted transfer gaps in performance across languages (Hu et al., 2020), none have quantified this dispersion.⁸ Traditionally used measures of statistical dispersion like standard deviation or calculating range are sub-optimal choices as they are scale-dependant, unbounded and highly sensitive to outliers, which makes them unsuitable for data that does not approach a normal distribution (De Maio, 2007).

Beyond these measures, several nuanced metrics have been introduced to quantify disparity in income distributions. The choice of income inequality indicator is of significant importance since it has implications in measuring health, state-level mortality, etc. (De Maio, 2007). The Gini coefficient (Dorfman, 1979) has been most commonly used for this purpose (De Maio, 2007).

Hurley and Rickard (2009) lay out six desirable attributes of a measure of sparsity, drawing from past literature (Dalton, 1920; Rickard and Fallon, 2004) and prove the Gini coefficient to be the only measure having all six, among a varied set of alternatives. Briefly, these properties and their relevance in measuring linguistic disparity across tasks include: i) *Robin Hood*: a drop in high-performant and gain in low-performant languages should lead to higher equity; ii) *Scale Invariance*: no change in relative performance should lead to no change in equity, regardless of changes in absolute values; iii) *Rising Tide*: adding a constant value to each language’s performance should increase equity; iv) *Cloning*: equity must remain invariant under cloning, i.e., if two identical distributions are combined, the equity remains unchanged; v) *Bill Gates*: if one language hypothetically gains infinite performance, equity should tend to zero; vi) *Babies*: adding languages with zero performance in the distribution should decrease equity.

Given that the Gini coefficient satisfies all of these attributes (Hurley and Rickard, 2009), we propose to use the same in pursuit of quantifying the inequalities amongst languages with regard to downstream tasks in NLP. A pictorial representation of the Gini coefficient for this setting can be found in Figure 1. Downstream task performance closely follows the highly skewed data distributions on which massively multilingual models are pre-trained. By including the Gini coefficient mea-

⁸Hu et al. (2020) only considered the difference between English and other languages as cross-lingual transfer gap.

sure in our evaluation, we aim to incentivize model builders to invest in equitable performance, despite data differences. Details on how the Gini coefficient is calculated are given in Appendix A.3.

3.3 Inclusion: Efficiency Score

Language technology is only beneficial if it can be deployed and accessed by users in a region. We employ efficiency to quantify inclusion as user devices are resource-constrained in many low-resource settings. Following work on user-centric evaluation (Ethayarajh and Jurafsky, 2020; Ma et al., 2021), we propose to incorporate efficiency into model performance based on throughput and memory, each of which are defined below.⁹

Throughput Number of instances the model can process per second on a CPU, assuming that GPUs are rarely used for deployment at scale in resource-constrained environments.

Memory Saved The size of the model is considered to be a measure of how expensive a model is to use in practice. Since we wish to minimize this metric, *memory used* is transformed into *memory saved* by subtracting it from a maximum available memory of 16 GB (Ma et al., 2021). We show the memory and throughput values for our models in Appendix A.1.

Following Ma et al. (2021), to calculate the efficiency score we first convert each metric into units of performance, by calculating the average marginal rate of substitution (AMRS) for each metric M (i.e., throughput and memory). $\text{AMRS}(M, \text{perf})$ tells us the rate at which model creators, as a group, are trading off M for a one-point increase in perf while keeping utility constant. For example, if AMRS of “memory saved” with respect to accuracy were 0.5 GB, then each GB of memory saved would on average be worth 2 points of accuracy. Dividing M by $\text{AMRS}(M, \text{perf})$ converts it to units of performance. Details on how one can calculate $\text{AMRS}(M, \text{perf})$ can be found in Appendix A.1. For a model x_i , $\text{Efficiency}(x_i)$ is then defined as :

$$\text{Efficiency}(x_i) = \sum_M w_M * \frac{M(x_i)}{\text{AMRS}(M, \text{perf})}$$

⁹Ma et al. (2021) additionally consider fairness and robustness, both of which are highly contextual and difficult to define in the context of multilingual models at present. Hence, we focus on model aspects that are objectively measurable.

where we choose $w_{\text{perf}} = 0.5$, $w_{\text{throughput}} = 0.25$ and $w_{\text{memory}} = 0.25$ as default weights. In practice, these weights can be adjusted based on user requirements and existing constraints.

4 Egalitarian Annotation Budget Allocation

Model development involves not just the design of an architecture or training but also data annotation. The proposed dimensions thus cannot only be used to assess models but can also inform how data should be annotated across many languages. As fine-tuning on a few labeled examples in the target language has shown to improve zero-shot transfer performance, we study how to allocate an annotation budget across a number of source languages S in order to optimize for inclusion and equity across a set of target languages T . Previous work employs a feature-based approach to select a single source language to maximize performance on a target language (Lin et al., 2019) or labels examples across all source languages equally (Debnath et al., 2021). We propose a fully computational approach for modeling the space of source and target languages for a multi-source multi-target language setting. This is done by empirically estimating performance of language $t \in T$ on a held-out set when fine-tuned on x labeled instances of language $s \in S$, $\forall (s, t)$ pairs, which follows a power-law distribution (Rosenfeld et al., 2019). We now seek to find the optimal allocation $\{x_s : s \in S\}$ subject to $\sum_{s \in S} x_s \leq X$ (details in Appendix A.5).

We follow a simple greedy approach to solve this constrained optimization problem as shown in Table 10. Specifically, at each step we allocate a sample to the source language conferring the highest marginal gain to all target languages, which is quantified by the summation of the increase in the global metric and the reduction in Gini.¹⁰ At present, we assign equal weight to each metric but this can be changed according to user preferences.

5 Experiments

5.1 Experimental setup

Languages We base our case study on the 22 scheduled languages of India spoken by 97% of its population. We also include English, since it has a sizeable population of 128.5M speakers (Table 1).

¹⁰Future work may consider more complex approaches that consider language relatedness based on work on transfer relationship learning (Zamir et al., 2018; Song et al., 2019).

Language	as	bn	brx	doi	en	gu	hi	kn	kok	ks	mai	ml
Speakers (in M)	23.6	107.4	1.6	2.8	128.5	60.3	691.6	58.8	2.6	7	14.3	35.6
Language	mni	mr	ne	or	pa	sa	sat	sd	ta	te	ur	-
Speakers (in M)	2.2	99.1	3.4	42.6	36.1	3.1	7.7	3.1	76.6	94.5	63.2	-

Table 1: The number of speakers (in millions) for each of the 22 scheduled languages and English. We take the sum total of first, second and third language speakers for each language.

Task	Dataset	Test Langs.	HP
NER	WikiAnn (Pan et al., 2017; Rahimi et al., 2019)	bn, en, gu,	97.6
		hi, ml, mr, pa, ta, te, ur	
POS	Universal Dependencies v2.6 (Nivre et al., 2018)	en, hi, mr, ta, te, ur	97
NLI	XNLI (Conneau et al., 2018)	en, hi, ur	92.8
QA	XQuAD (Artetxe et al., 2019); TyDiQA-GoldP (Clark et al., 2020)	bn, en, hi, te	91.2;
			90.1

Table 2: *Finetuning Tasks and Datasets*. HP denotes the human performance for each task. For QA, HP is 91.2 F1 for XQuAD and 90.1 F1 for TyDiQA.

Language	NER	POS	NLI	QA
English	20,000	21,261	392,702	88,602
Hindi	5,000	13,305	392,702 (-tran)	88,602 (-tran)

Table 3: *Number of training instances for English and Hindi*. (-tran) denotes that the English fine-tuning set has been translated to Hindi.

Tasks We select tasks from the XTREME (Hu et al., 2020) benchmark. Dataset details and the human performance (HP) for each task can be found in Table 2. For each task, we only evaluate on IN language test sets.

Models Model selection is motivated by two key factors that we wish to explore in our study: **i)** general vs region-specific choices; and **ii)** model efficiency. We choose IndicBERT (Kakwani et al., 2020), MuRIL (Khanuja et al., 2021) and XLM-R (Conneau et al., 2020), the first two being region-specific models and the third being a state-of-the-art model trained on 100+ languages. We consider both the base and large versions for MuRIL and XLM-R. IndicBERT follows the ALBERT architecture (Lan et al., 2019) and is hence much smaller than the base versions of both models. IndicBERT is trained on 11, MuRIL on 16, and XLM-R on 15 IN languages (details in Appendix A.2).

Task	Model	Baseline F1/Accuracy \uparrow	Diversity $M_{\tau=1} \uparrow$	Equity Gini Coeff. \downarrow	Inclusion Efficiency \uparrow
NER	MuRIL _{base}	77.6	69.6	0.59	69.1
	XLM-R _{large}	68.0	61.2	0.60	44.4
	MuRIL _{large}	77.7	68.2	0.59	63.1
POS	MuRIL _{base}	75.0	54.7	0.76	52.5
	XLM-R _{large}	79.2	60.3	0.75	48.0
	MuRIL _{large}	77.3	58.6	0.76	51.8
NLI	MuRIL _{base}	74.1	45.5	0.88	58.7
	XLM-R _{large}	78.7	46.6	0.88	57.3
	MuRIL _{large}	78.6	47.4	0.88	57.8
QA	MuRIL _{base}	76.1	53.8	0.83	77.8
	XLM-R _{large}	75.7	56.6	0.83	76.3
	MuRIL _{large}	77.7	57.9	0.83	75.7

Table 4: *DEI Results compared to baseline F1/accuracy performance*. Here, we compare models’ accuracy/F1 performances (usually reported as the evaluation metric) to their DEI metrics. We observe that while performances may significantly vary, DEI metrics (especially equity) don’t change as much, indicating that multilingual models make the rich "richer" to increase average performance but may not be moving towards being truly multilingual and equitable across languages. More discussions in Section 5.2.

Fine-tuning We initially fine-tune the selected models using training data in English (EN) given the availability of labeled data across tasks. However, past works highlight that this choice is sub-optimal and one can obtain better performance by transferring from closely related languages (Lauscher et al., 2020; Cotterell and Heigold, 2017; Dong et al., 2015; Turc et al., 2021). To examine this effect in our case study, we additionally fine-tune models on Hindi (HI) because **i)** 15 out of 22 languages belong to the same language family as HI (Indo-Aryan); **ii)** we have training data available for all tasks in HI¹¹; and **iii)** HI has the highest speaker population, which may lead to higher demographic utility and is a future-safe choice to obtain annotations for any task. Table 3 summarizes training data statistics for EN and HI.

¹¹Training sets for NLI and QA have been machine-translated from English, which has been shown to perform similar to human-generated train sets (Turc et al., 2021).

Metric	Train Lang.	Model	NER	POS	NLI	QA	Average
$M_{\tau=1} \uparrow$ (Diversity)	English	MuRIL _{base}	69.6	54.7	45.5	53.8	55.9
		XL _M -R _{large}	61.2	60.3	46.6	56.6	56.2
		MuRIL _{large}	68.2	58.6	47.4	57.9	58.0
	Hindi	MuRIL _{base}	75.1	67.3	46.8	54.7	61.0
		XL _M -R _{large}	74.4	66.8	49.4	53.2	60.9
		MuRIL _{large}	74.8	66.5	49.2	54.6	61.3
Gini Coeff. \downarrow (Equity)	English	MuRIL _{base}	0.59	0.76	0.88	0.83	0.76
		XL _M -R _{large}	0.6	0.75	0.88	0.83	0.77
		MuRIL _{large}	0.59	0.76	0.88	0.83	0.77
	Hindi	MuRIL _{base}	0.59	0.75	0.87	0.83	0.76
		XL _M -R _{large}	0.59	0.76	0.88	0.83	0.77
		MuRIL _{large}	0.59	0.75	0.87	0.83	0.76
Efficiency \uparrow (Inclusion)	English	MuRIL _{base}	69.1	52.5	58.7	77.8	64.5
		XL _M -R _{large}	44.4	48	57.3	76.3	56.5
		MuRIL _{large}	63.1	51.8	57.8	75.7	62.1
	Hindi	MuRIL _{base}	69.8	56.2	59.8	77.3	65.8
		XL _M -R _{large}	49.2	49.0	59.1	75.1	58.1
		MuRIL _{large}	65.2	53.7	58.8	75.0	63.2

Table 5: *Region-specific fine-tuning results.* Note that the metrics are computed considering all 23 languages as detailed in Section 5.1. Region-specific fine-tuning helps, but disparities along DEI axes persist. More discussions in Section 5.2.

5.2 Zero-shot transfer results

How do DEI metrics compare to baseline standard performance metrics (F1/Accuracy)?

We report results of the best-performing models in Table 4. While average performance is similar across tasks, there are stark differences in DEI metrics. The diversity metric helps discern whether the change in performance is more skewed towards languages with a relatively high or low speaker population. For example, for POS, MuRIL_{base} and XL_M-R_{large} have a 4.2% difference in performance but a 5.9% difference in $M_{\tau=1}$. This indicates that the difference is more pronounced for languages with large speaker populations. Similarly, for NLI, the difference in performance and $M_{\tau=1}$ is 4.6% and 1.1% respectively, which also highlights a lack of test data to quantify larger differences in diversity. With regards to equity, we observe that even though major differences exist compared to average performance, the Gini coefficient remains relatively unchanged, indicating that while overall performance has increased, the disparity in performance amongst languages has not yet been addressed by any model. Regarding efficiency or inclusion, while MuRIL_{large} beats MuRIL_{base} in performance, MuRIL_{base} is more efficient to use, across all tasks.

Where are we today w.r.t DEI of NLP technology? We report results of best-performing models (fine-tuned on EN and HI) in Table 5 (detailed results with XL_M-R_{base} and IndicBERT in Table 13). Overall, the diversity metric is highest

Metric	Budget	Model	Fine-tuning Strategy			
			English	Hindi	Egalitarian	Greedy
$M_{\tau=1} \uparrow$	1,000	XL _M -R _{large}	54.0	66.2	65.4	65.3
		MuRIL _{large}	60.4	71.3	74.1	73.6
		XL _M -R _{large}	59.4	74.4	75.4	75.7
	5,000	MuRIL _{large}	65.4	74.8	78.2	78.3
		XL _M -R _{large}	59.0	-	77.6	77.6
		MuRIL _{large}	70.5	-	79.6	79.9
Gini Coeff. \downarrow	1,000	XL _M -R _{large}	0.6	0.6	0.59	0.59
		MuRIL _{large}	0.6	0.6	0.58	0.58
		XL _M -R _{large}	0.6	0.59	0.59	0.59
	5,000	MuRIL _{large}	0.59	0.59	0.58	0.58
		XL _M -R _{large}	0.61	-	0.59	0.59
		MuRIL _{large}	0.59	-	0.58	0.58

Table 6: *Performance on NER under different annotation budgets.* We observe that the greedy approach (§4) performs best across all metrics. Note that the HI train set has 5,000 examples only. Details in §5.3.

for MuRIL_{large}, when fine-tuned on HI. We also observe that the diversity metric increases with region-specific choices, both in pre-training and fine-tuning. The Gini coefficient remains relatively high at around 0.76 even for the best models, which highlights the disparity in performance even among languages within a single region.¹² With regards to efficiency, averaging across languages and tasks, MuRIL_{base} performs best.

What is the way forward? Overall, the absolute values of the global metric and the Gini coefficient indicate that there lies great potential in both increasing the utility of our models and making them more equitable. Since model performances partially reflect the amount of raw data used in pre-training (Lauscher et al., 2020), creating equitable unlabeled data resources would alleviate these issues. However, this is an ambitious undertaking that is extremely resource intensive and can certainly not be achieved for 6500 languages in the near future. We thus investigate how limited amounts of data can be used to maximally improve utility and equity during fine-tuning.

5.3 Few-shot results

Problem Formulation For few-shot fine-tuning, we focus on NER where sufficient labeled training data for seven IN languages is available. We employ the source languages $S = \{\text{bn, en, hi, ml, mr, ta, ur}\}$ and seek to optimize metrics on the target languages $T = \{\text{bn, en, gu, hi, ml, mr, pa, ta, te, ur}\}$. In

¹²For comparison, for OECD countries from 2008–2009, the Gini coefficient on income for the entire population ranged between 0.34 and 0.53 while the Gini coefficient for the entire world has been estimated to be between 0.61 and 0.68 (Hillebrand et al., 2009; Klugman and Nations, 2010).

each setting, we have a limited annotation budget, which we can divide among the source languages. We compare against several competitive baselines: **i)** using only examples from EN or HI respectively; **ii)** distributing the annotation budget in an egalitarian (uniform) way across all source languages (Debnath et al., 2021); and **iii)** our novel greedy approach proposed in §4. For the greedy approach, we illustrate the best-fit curves for each (s, t) pair in Appendix A.5 (Table 11).

Results We show the results under various annotation budgets in Table 6. Overall, we find that our method yields a higher global metric under most budgets (5 of 6 cases) and also yields a lower Gini coefficient under *all* budget schemes. The optimal allocations for each budget are shown in Table 12. As we can see, the greedy algorithm converges to a solution that is close to uniform. This provides further evidence for the benefits of an egalitarian distribution of annotation budget in order to maximize performance across all languages as the expected marginal gain for languages that have been under-represented during training will be highest. Both the egalitarian and greedy approaches significantly outperform fine-tuning on EN or HI. For instance, our greedy approach outperforms fine-tuning on 10,000 EN examples by 1–3% with a budget of only 1,000 examples.

6 Discussion

Building evaluation datasets Having uncovered the linguistic inequity and exclusivity of current NLP technologies, we seek to identify practical measures we can take in order to mitigate these biases. As a first step, it is paramount to build representative evaluation sets for all languages as they are required to accurately measure diversity and equity. Out of the 23 languages in our case study, most do not have evaluation data across tasks despite holding official recognition and being spoken by 97% of the population. In light of the benefits of an egalitarian data distribution during few-shot learning, we also recommend the collection of small amounts of data across many languages for training, in order to maximize marginal gain. These datasets should be collected at the grass-roots level, involving the community they need to serve to capture culturally relevant phenomenon. A prime example of this is the Masakhane organisation¹³ steering efforts towards data collection in

¹³<https://www.masakhane.io/>

African languages, involving the local community. Incentivizing rural, low-income workers to provide for such data also serves as a viable source of supplementary income, and does not degrade dataset quality (Abraham et al., 2020).

Trading off multilinguality and regionality

From a modeling perspective, multilingual pre-trained models have been instrumental to NLP systems supporting an unprecedented number of languages, because of their zero-shot transfer capabilities. However, while these are a big step towards linguistic inclusion, they are subject to limitations such as highly skewed pre-training distributions and limited transfer to under-represented languages (Hu et al., 2020; Lauscher et al., 2020), a bias towards the source language, and sub-optimal tokenization (Wang et al., 2021). A way to combat these issues is to make region-specific choices, both in pre-training and fine-tuning, as observed in §5.2. Localizing the problem also enables one to incorporate linguistic expertise (Nzeyimana and Niyongabo Rubungo, 2022) and provide support for culturally relevant phenomena like transliteration or code-mixing. Despite this, we must be wary of excessive fragmentation in pre-training as it leads to higher maintenance costs and there is a possibility that these benefits will be overcome with advances in compute and model capacity in the near future. Optimal fine-tuning however, is promising, as evidenced in §5.3 where we observe significant gains in moving away from the zero-shot paradigm.

7 Conclusion

We have proposed a framework for the evaluation of NLP technology based on diversity, equity, and inclusion and proposed the Gini coefficient to quantify equity. We have assessed to what extent several modeling and data choices affect the value NLP technology confers to speakers of Indian languages. We have also proposed an algorithmic method for resource allocation for task-specific fine-tuning, which outperforms a purely egalitarian distribution of data labeling. Finally, we highlight the importance of building representative evaluation sets from the grass-roots level to enable tracking progress, and discuss how even with the best modeling strategies, we have a long road ahead in building inclusive, equitable systems. While region-specific choices help to a certain extent, building a single global multilingual model without compro-

missing on the three metrics is something we should move towards in the future. We sincerely hope our evaluation paradigm aids in tracking the community’s progress in building linguistically diverse technologies.

Limitations

We do not consider the inequalities that may exist within subgroups in a language given the lack of fine-grained evaluation data. In multilingual countries like India, each language is composed of several dialects (Hindi alone is composed of 58 dialects (Chandramouli, 2011)). As disparities exist along multiple axes such as caste, gender, religion and so on (Sambasivan et al., 2021), it is imperative to go beyond the language level. We only consider pre-trained language models for our experiments given their massive language coverage and zero-shot transfer capabilities. There have been efforts to build language-specific, task-specific models which we do not include in our study. Our greedy data allocation method is a strong baseline that outperforms standard approaches such as selecting a single source language or uniform selection. It can be improved by incorporating notions of language similarity, which requires more complex methods (Song et al., 2019).

Acknowledgements

We would like to thank the reviewers for their insightful feedback. We would also like to thank Melvin Johnson and Slav Petrov for helpful feedback on a draft of this post.

References

- Basil Abraham, Danish Goel, Divya Siddarth, Kalika Bali, Manu Chopra, Monojit Choudhury, Pratik Joshi, Preethi Jyoti, Sunayana Sitaram, and Vivek Seshadri. 2020. Crowdsourcing speech data for low-resource languages from low-income workers. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2819–2826.
- Orevaoghene Ahia, Julia Kreutzer, and Sara Hooker. 2021. *The Low-Resource Double Bind: An Empirical Study of Pruning for Low-Resource Machine Translation*. In *Findings of EMNLP 2021*.
- Kabir Ahuja, Shanu Kumar, Sandipan Dandapat, and Monojit Choudhury. 2022. *Multi task learning for zero shot performance prediction of multilingual models*. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5454–5467, Dublin, Ireland. Association for Computational Linguistics.
- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*.
- Damian Blasi, Antonios Anastasopoulos, and Graham Neubig. 2022. *Systematic inequalities in language technology performance across the world’s languages*. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5505, Dublin, Ireland. Association for Computational Linguistics.
- Chandramouli. 2011. Census of india 2011 provisional population totals. *New Delhi: Office of the Registrar General and Census Commissioner*.
- Monojit Choudhury and Amit Deshpande. 2021. How linguistically fair are multilingual pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12710–12718.
- Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. *Unsupervised cross-lingual representation learning at scale*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.
- US Constitution. 2021. US constitution, 2021. <https://www.whitehouse.gov/briefing-room/presidential-actions/2021/06/25/executive-order-on-diversity-equity-inclusion-and-accessibility-in-the-federal-workforce/>.
- Ryan Cotterell and Georg Heigold. 2017. Cross-lingual, character-level neural morphological tagging. *arXiv preprint arXiv:1708.09157*.
- Hugh Dalton. 1920. The measurement of the inequality of incomes. *The Economic Journal*, 30(119):348–361.
- Fernando G De Maio. 2007. Income inequality measures. *Journal of Epidemiology & Community Health*, 61(10):849–852.

- Arnab Debnath, Navid Rajabi, Fardina Fathmiul Alam, and Antonios Anastasopoulos. 2021. Towards more equitable question answering systems: How much more data do you need? *arXiv preprint arXiv:2105.14115*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. In *Proceedings of NAACL 2019*.
- Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732.
- Robert Dorfman. 1979. A formula for the gini coefficient. *The review of economics and statistics*, pages 146–149.
- Kawin Ethayarajh and Dan Jurafsky. 2020. **Utility is in the Eye of the User: A Critique of NLP Leaderboards**. In *Proceedings of EMNLP 2020*.
- European Language Resources Association. 2019. BLT4All: Language Technologies for All. <https://lt4all.elra.info/en/>. [Online; accessed Dec. 2019.].
- William Fedus, Barret Zoph, and Noam Shazeer. 2021. **Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity**. *arXiv preprint arXiv:2101.03961*.
- Evan Hillebrand et al. 2009. Poverty, growth and inequality over the next 50 years. In *Expert Meeting on How to feed the World in*, volume 2050, pages 2012–02.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. **XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization**. In *Proceedings of ICML 2020*.
- Niall Hurley and Scott Rickard. 2009. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. **The State and Fate of Linguistic Diversity and Inclusion in the NLP World**. In *Proceedings of ACL 2020*.
- Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. **IndicNLP Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.
- Masahiro Kaneko and Danushka Bollegala. 2019. **Gender-preserving debiasing for pre-trained word embeddings**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1641–1650, Florence, Italy. Association for Computational Linguistics.
- Michael Katell, Meg Young, Dharma Dailey, Bernease Herman, Vivian Guetler, Aaron Tam, Corinne Binz, Daniella Raz, and P. M. Krafft. 2020. **Toward situated interventions for algorithmic equity: Lessons from the field**. *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 45–55.
- Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. **Muril: Multilingual representations for indian languages**. *arXiv preprint arXiv:2103.10730*.
- Simran Khanuja, Sandipan Dandapat, Sunayana Sitaram, and Monojit Choudhury. 2020a. **A new dataset for natural language inference from code-mixed conversations**. *arXiv preprint arXiv:2004.05051*.
- Simran Khanuja, Sandipan Dandapat, Anirudh Srivasan, Sunayana Sitaram, and Monojit Choudhury. 2020b. **Gluecos: An evaluation benchmark for code-switched nlp**. *arXiv preprint arXiv:2004.12376*.
- Jeni Klugman and Development Programme United Nations. 2010. *The real wealth of nations: pathways to human development*. Palgrave Macmillan.
- Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. **The iit bombay english-hindi parallel corpus**. *arXiv preprint arXiv:1710.02855*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. **Albert: A lite bert for self-supervised learning of language representations**. *arXiv preprint arXiv:1909.11942*.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. **From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers**. *arXiv preprint arXiv:2005.00633*.
- Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. **Choosing Transfer Languages for Cross-Lingual Learning**. In *Proceedings of ACL 2019*.

- Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. 2021. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. *arXiv preprint arXiv:2106.06052*.
- Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, and Antonsen et al. 2018. **Universal Dependencies 2.2**. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. **KinyaBERT: a morphology-aware Kinyarwanda language model**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5347–5363, Dublin, Ireland. Association for Computational Linguistics.
- Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958.
- Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for ner. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164.
- Gowtham Ramesh, Sumanth Doddapaneni, Aravindh Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *arXiv preprint arXiv:2104.05596*.
- Reserve Bank of India RBI. 2021. **Handbook of statistics on indian economy**. RBI, Reserve Bank of India.
- Scott Rickard and Maurice Fallon. 2004. The gini index of speech. In *Proceedings of the 38th Conference on Information Science and Systems (CISS'04)*.
- Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, İşin Demirşahin, and Keith Hall. 2020. **Processing South Asian languages written in the Latin script: the Dakshina dataset**. In *Proceedings of The 12th Language Resources and Evaluation Conference (LREC)*, pages 2413–2423.
- Jonathan S Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. 2019. A constructive prediction of the generalization error across scales. *arXiv preprint arXiv:1909.12673*.
- Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Graham Neubig, and Melvin Johnson. 2021. **XTREME-R: Towards More Challenging and Nuanced Multilingual Evaluation**. In *Proceedings of EMNLP 2021*.
- Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. **Transfer learning in natural language processing**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.
- Nithya Sambasivan, Erin Arnesen, Ben Hutchinson, Tulsee Doshi, and Vinodkumar Prabhakaran. 2021. Re-imagining algorithmic fairness in india and beyond. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 315–328.
- Jie Song, Yixin Chen, Xinchao Wang, Chengchao Shen, and Mingli Song. 2019. **Deep model transferability from attribution maps**. In *Advances in Neural Information Processing Systems*.
- Advaith Sridhar, Rohith Gandhi Ganesan, Pratyush Kumar, and Mitesh Khapra. 2020. Include: A large scale dataset for indian sign language recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1366–1375.
- Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of english in zero-shot cross-lingual transfer. *arXiv preprint arXiv:2106.16171*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021. **Multi-view Subword Regularization**. In *Proceedings of NAACL 2021*.
- Amir R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling Task Transfer Learning. In *Proceedings of CVPR 2018*.

A Appendix

A.1 Efficiency

We report the throughput and memory for each model and task in Table 7. For NLI, POS and NER, the maximum sequence length is 128 and for QA it’s 384.

As detailed in Section 3.3, we need to calculate the AMRS for each metric M (throughput and memory saved), to calculate the efficiency score. As described in (Ma et al., 2021), each model has properties (or *goods*) that inform its utility. Here, these goods are throughput, memory saved, and performance. A model is a point in this space of goods and an indifference curve is a set of points

that provide the same utility (for different values of these properties). These curves are monotonically negatively sloped, i.e., for a model with higher accuracy to be on the same curve as one with a lower accuracy, it will have to use up more memory or have lower throughput. For a given indifference curve, the rate at which this trade-off is made, is called the marginal rate of substitution (MRS).

To calculate MRS, and consequently AMRS, (Ma et al., 2021) make two key assumptions: **i)** All models lie on the same indifference curve; **ii)** if $M(x_i) > M(x_{i+1})$ and $\text{perf}(x_i) > \text{perf}(x_{i+1})$, then there exists a model $(\text{perf}(x_{i+1}), M(x_i) + (M(x_i) - M(x_{i+1})))$ on the same indifference curve as x_i . For our case study, we believe that assuming regional and global models to lie on the same indifference curve would be inaccurate, since models with the same capacity (size and architecture) have been trained on a different set of languages. In the case of (Ma et al., 2021), they only consider models pre-trained on English. Here, we assume that regional models (trained on 15-17 languages) would be strictly better on all dimensions and hence lie on a different indifference curve as compared to global models (trained on 100+ languages). Hence, we assume IndicBERT, MuRIL_{base} and MuRIL_{large} to lie on one indifference curve and XLM-R_{base} and XLM-R_{large} to lie on another. The second assumption holds in our case as well.

For a model x_i , Efficiency(x_i), MRS, and AMRS are given by :

$$\text{Efficiency}(x_i) = \sum_M w_M * \frac{M(x_i)}{\text{AMRS}(M, \text{perf})}$$

$$\text{AMRS}(M, \text{perf}) = \overline{\text{MRS}}$$

$$\text{MRS} = \left\{ \left| \frac{M(x_i) - M(x_{i+1})}{\text{perf}(x_i) - \text{perf}(x_{i+1})} \right| \mid 1 \leq i < n \right\}$$

A.2 Pre-training Languages

In Section 5.1, we choose IndicBERT, MuRIL and XLM-R as pre-trained multilingual models to base our analysis upon. IndicBERT is trained on 11 IN languages that include Assamese (as), Bengali (bn), Gujarati (gu), Hindi (hi), Kannada (kn), Malayalam (ml), Marathi (mr), Oriya (or), Punjabi (pa), Tamil (ta), Telugu (te). XLM-R includes 15 IN languages in training with the addition of Nepali (ne), Sanskrit (sa), Sindhi (sd) and Urdu (ur) over IndicBERT and MuRIL is trained on 16 IN languages, with the addition of Kashmiri (ks) over XLM-R.

Model	Metric	NER	PoS	NLI	QA
IndicBERT	Memory Saved	15.9GB			
	Throughput	22.6	20.2	22.9	10.5
	Perf (EN)	41.3	71.6	69.7	52.4
XLM-R _{base}	Memory Saved	15GB			
	Throughput	24.4	23.2	26.4	14.9
	Perf (EN)	61.7	82.2	77.1	72.1
MuRIL _{base}	Memory Saved	15.1GB			
	Throughput	23.8	23.1	26.2	15.7
	Perf (EN)	74.9	80.3	78.9	77.5
XLM-R _{large}	Memory Saved	13.9GB			
	Throughput	9.4	10.0	10.4	4.1
	Perf (EN)	64.6	83.7	81.8	81.4
MuRIL _{large}	Memory Saved	14.1GB			
	Throughput	9.8	9.9	10.5	4.2
	Perf (EN)	71.8	83.4	82.8	83.0
AMRS (Regional)	Throughput	1.7	3.4	2.2	1.1
	Memory	0.1	0.3	0.2	0.1
AMRS (Global)	Throughput	3.7	7.1	3.3	1.2
	Memory	0.3	0.5	0.2	0.1

Table 7: The throughput is given by the number of instances processed per second by the fine-tuned models on CPU.

A.3 Gini Coefficient

The Gini coefficient is mathematically computed based on the Lorenz curve, which plots the relation between population size and the cumulative income earned by that population as shown in Figure 1. To plot the Lorenz curve, individuals are sorted in increasing order of income (x-axis) and their cumulative wealth is plotted on the y-axis. In essence, a point (x, y) indicates that the bottom x% of the population holds y amount of wealth. The line at 45 degrees represents perfect equality of incomes. The Gini coefficient G is then calculated as the ratio of the area that lies between the line of equality and the Lorenz curve (A in Figure 1), over the total area under the line of equality (A + B in Figure 1). If G = 0, every person in the population receives an equal percentage of income and if G = 1, a single person receives 100% of the income. Since the axes scale from 0 to 1, A + B = 0.5. In essence, if the Lorenz curve is represented by the function Y = L(X) then G can be given as:

$$G = \frac{A}{A + B} = 2A = 1 - 2B = 1 - 2 \int_0^1 L(X) dX$$

For a population with values y_i , $i = 1 \dots n$, that are indexed in non-decreasing order ($y_i \leq y_{i+1}$):

$$G = \frac{1}{n} \left(n + 1 - 2 \frac{\sum_{i=1}^n (n + 1 - i) y_i}{\sum_{i=1}^n y_i} \right)$$

For comparison, for OECD countries from 2008–2009, the Gini coefficient on income for the entire

Metric	Train Lang.	Model	NER	PoS	NLI	QA	Average
Gini Coeff. ↓	English	IndicBERT	0.155	0.107	0.051	0.091	0.101
		XLM-R _{base}	0.095	0.067	0.058	0.048	0.067
		MuRIL _{base}	0.047	0.086	0.048	0.03	0.052
		XLM-R _{large}	0.084	0.06	0.049	0.026	0.055
		MuRIL _{large}	0.051	0.086	0.051	0.027	0.057
	Hindi	IndicBERT	0.173	0.073	0.004	0.041	0.073
		XLM-R _{base}	0.067	0.037	0.039	0.046	0.047
		MuRIL _{base}	0.062	0.032	0.036	0.012	0.035
		XLM-R _{large}	0.057	0.04	0.033	0.029	0.04
		MuRIL _{large}	0.065	0.057	0.033	0.014	0.042

Table 8: *Gini Coefficient* for all models calculated only across languages having evaluation sets for each task.

Task	Batch Size	Learning Rate	No. of Epochs	Warmup Ratio	Max. seq. Length
NER	32	2e-5	10	0.1	128
POS	32	2e-5	10	0.1	128
NLI	64	2e-5	3	0.1	128
QA	32	3e-5	2	0.1	384

Table 9: Hyperparameter details for each fine-tuning task

population ranged between 0.34 and 0.53. The Gini coefficient on income for the entire world has been estimated to be between 0.61 and 0.68 (Hillebrand et al., 2009; Klugman and Nations, 2010). In our experiments on Indian languages, state-of-the-art models achieve an average Gini coefficient of 0.77, which highlights the disparity in performance even among languages within a single region.

As mentioned in Section 5.2, calculating the Gini coefficient across all 23 languages doesn’t reflect the dispersion in performances across languages for which we have test sets. To compare between baselines, we additionally report the Gini coefficient evaluated only across those languages for which we have test sets as shown in Table 8. We observe that region-specific choices (MuRIL_{base} fine-tuned on HI) lead to the lowest value, similar to what we observe with the global metric.

A.4 Fine-tuning Details

We fine-tune all models using the hyperparameters mentioned in Table 9 for each task and model consistently throughout the paper. We make use of the XTREME codebase¹⁴ to finetune these models using a NVIDIA A100 GPU. We make an exception for IndicBERT when fine-tuning on NER, where we fine-tune for 15 epochs instead of 10, to reach convergence.

A.5 Budget Allocation

In Section 4, we describe an empirical budget allocation scheme for fine-tuning of pre-trained models

that can jointly optimize on our proposed metrics. We follow a greedy approach to solve this problem, as shown in Table 10. In this paper, we solve this for one task, namely NER, but the methodology proposed is generally extensible to any task and combination of languages since it is purely empirical. We select seven source languages for which we have enough training data and fine-tune MuRIL_{large} and XLM-R_{large} for each of these source languages independently, for two epochs. During fine-tuning, we evaluate on each of our target languages after every 10 steps of training. Given our batch-size is 32, we gather data-points at a step size of 320 training instances. Consequently, say we have 5000 training instances for a source language, we gather approximately 30 sample points for that source language and any target language. Using these, we plot best-fit curves for $\forall(s, t)$ pairs using the *scipy.optimize.curve_fit* package. Given a function, $f(x)$, *curve_fit* uses non-linear least squares to fit $f(x)$ to the observed data-points. We define $f(x)_{s,t} = a_{s,t} + b_{s,t} * x^{-c_{s,t}}$, because the relation between model performance and training data follows a power-law distribution (Rosenfeld et al., 2019). The best-fit curves for each source and target pair are shown in Table 11. The visualizations of the best-fit curves for a sample training language (Tamil) are shown in Figures 2, 3. Having determined constant values $\{a_{s,t}, b_{s,t}, c_{s,t}\} \forall(s, t)$ independently, we proceed with finding the optimal allocation using the algorithm described in Table 10. We solve this for three different budgets, i.e., 1,000; 5,000 and 10,000 and the optimal allocations for each budget are shown in Table 12.

¹⁴<https://github.com/google-research/xtreme>

Greedy Algorithm

- 1: **Input:** Fine-tuning labeled data $\forall s \in S$. A fixed budget of labeled data instances X
 - 2: **Initialize:** Set the total number of allocated instances to zero, i.e., $\text{allocated} = 0$, the number of allocated samples for each source language to zero, i.e. $\text{samples}[s] = 0 \forall s \in S$, the current global metric for each source language to $-\text{inf}$, i.e. $\text{current_gm}[s] = -\text{inf} \forall s \in S$ and the current gini coefficient for each source language to 1, i.e. $\text{current_gini}[s] = 1 \forall s \in S$
 - 3: **while** $\text{allocated} < X$ **do**
 - 4: $\text{highest_marginal_gain} = 0$
 - 5: **for** s in S **do**
 - 6: $\text{gm}_s = \sum_{t \in T} d_t^{(r)} * (a_{s,t} + b_{s,t} * (\text{samples}[s] + 1)^{-c_{s,t}})$
 - 7: $\text{gini}_s = F[\text{abs}(\text{performance}_{s,t}(\text{samples}[s] + 1)) \forall t \in T]$
 - 8: $\Delta \text{gm}_s = \text{gm}_s - \text{current_gm}[s]$
 - 9: $\Delta \text{gini}_s = \text{current_gini}[s] - \text{gini}_s$
 - 10: $\text{marginal_gain} = \alpha * \Delta \text{gm}_s + \beta * \Delta \text{gini}_s$
 - 11: **if** $\text{marginal_gain} > \text{highest_marginal_gain}$ **do**
 - 12: $\text{highest_marginal_gain} = \text{marginal_gain}$
 - 13: $\text{best_language} = s$
 - 14: $\text{best_gm} = \text{gm}_s$
 - 15: $\text{best_gini} = \text{gini}_s$
 - 16: **end if**
 - 17: **end for**
 - 18: $\text{samples}[\text{best_language}] = \text{samples}[\text{best_language}] + 1$
 - 19: $\text{allocated} = \text{allocated} + 1$
 - 20: $\text{current_gm}[\text{best_language}] = \text{best_gm}$
 - 21: $\text{current_gini}[\text{best_language}] = \text{best_gini}$
 - 22: **end while**
-

Table 10: A greedy approach to solve the constrained optimization for the budget allocation problem as described in Appendix A.5.

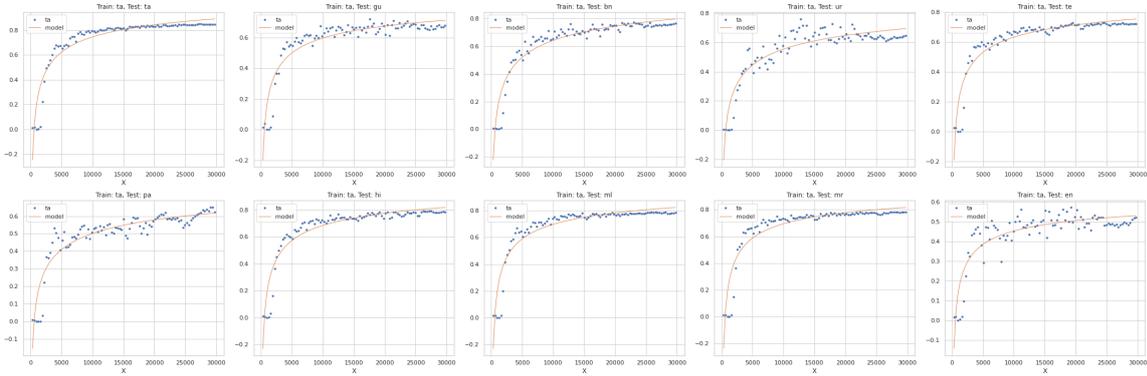


Figure 2: Best-fit curves for XLM-R when fine-tuned on Tamil for each of the target languages.

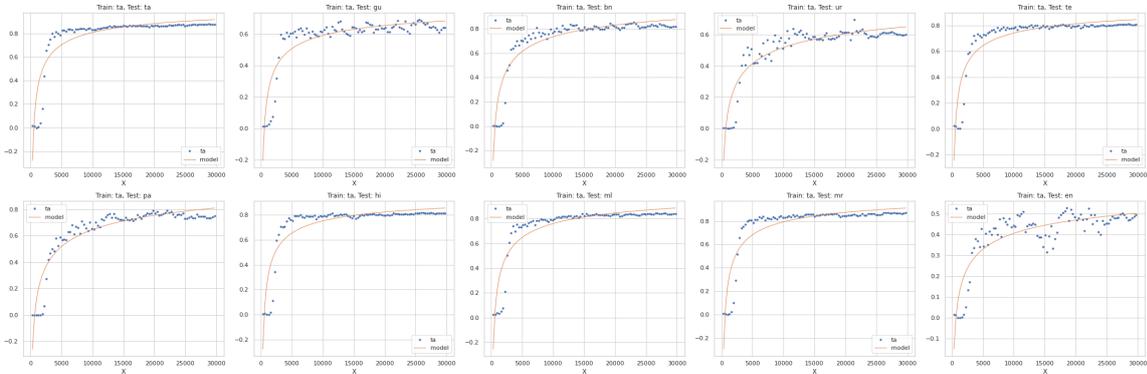


Figure 3: Best-fit curves for MuRIL when fine-tuned on Tamil for each of the target languages.

Test	Train	MuRIL		XLM-R	
		Edge Weight	R-squared	Edge Weight	R-squared
bn	bn	$1.2 - 29.0 * x^{-0.5}$	0.88	$1.3 - 11.5 * x^{-0.4}$	0.93
	en	$1.2 - 11.4 * x^{-0.4}$	0.78	$1.1 - 8.1 * x^{-0.3}$	0.89
	hi	$1.4 - 9.4 * x^{-0.3}$	0.85	$1.1 - 8.1 * x^{-0.3}$	0.92
	ml	$1.2 - 10.7 * x^{-0.3}$	0.86	$2.3 - 4.8 * x^{-0.1}$	0.92
	mr	$1.9 - 6.5 * x^{-0.2}$	0.88	$1.9 - 4.6 * x^{-0.1}$	0.93
	ta	$1.2 - 10.5 * x^{-0.3}$	0.83	$1.3 - 6.1 * x^{-0.2}$	0.90
	ur	$1.0 - 13.5 * x^{-0.4}$	0.88	$1.0 - 6.5 * x^{-0.3}$	0.91
en	bn	$0.9 - 4.4 * x^{-0.3}$	0.86	$1.0 - 5.2 * x^{-0.3}$	0.90
	en	$1.1 - 16.4 * x^{-0.4}$	0.82	$1.1 - 14.6 * x^{-0.4}$	0.85
	hi	$1.0 - 5.6 * x^{-0.3}$	0.88	$1.0 - 7.6 * x^{-0.3}$	0.90
	ml	$1.9 - 3.5 * x^{-0.1}$	0.88	$1.0 - 6.1 * x^{-0.3}$	0.86
	mr	$1.2 - 3.2 * x^{-0.2}$	0.84	$1.2 - 4.8 * x^{-0.2}$	0.91
	ta	$0.8 - 4.2 * x^{-0.3}$	0.76	$0.7 - 6.9 * x^{-0.4}$	0.76
	ur	-	0.88	$1.0 - 3.9 * x^{-0.2}$	0.90
gu	bn	$2.6 - 4.3 * x^{-0.1}$	0.93	$1.0 - 4.8 * x^{-0.3}$	0.88
	en	$0.9 - 5.5 * x^{-0.3}$	0.80	$0.7 - 11.3 * x^{-0.5}$	0.78
	hi	$1.2 - 5.4 * x^{-0.2}$	0.87	$0.7 - 13.3 * x^{-0.5}$	0.86
	ml	$1.2 - 7.8 * x^{-0.3}$	0.85	$1.6 - 4.3 * x^{-0.2}$	0.90
	mr	$1.1 - 6.4 * x^{-0.3}$	0.87	$1.1 - 6.0 * x^{-0.3}$	0.85
	ta	$0.8 - 11.3 * x^{-0.4}$	0.78	$1.0 - 7.6 * x^{-0.3}$	0.84
	ur	$1.4 - 3.4 * x^{-0.1}$	0.91	$1.6 - 3.2 * x^{-0.1}$	0.89
hi	bn	$0.9 - 17.2 * x^{-0.5}$	0.88	$1.2 - 4.8 * x^{-0.2}$	0.94
	en	$1.0 - 11.7 * x^{-0.4}$	0.83	$0.9 - 8.6 * x^{-0.4}$	0.88
	hi	$1.1 - 23.9 * x^{-0.5}$	0.90	$1.3 - 9.5 * x^{-0.3}$	0.92
	ml	$1.1 - 12.4 * x^{-0.4}$	0.85	$1.4 - 5.7 * x^{-0.2}$	0.90
	mr	$1.1 - 17.6 * x^{-0.5}$	0.85	$2.0 - 5.3 * x^{-0.2}$	0.93
	ta	$1.0 - 19.1 * x^{-0.5}$	0.78	$1.1 - 8.7 * x^{-0.3}$	0.88
	ur	$1.0 - 8.0 * x^{-0.3}$	0.92	$1.2 - 4.6 * x^{-0.2}$	0.94
ml	bn	$1.1 - 5.9 * x^{-0.3}$	0.88	$1.3 - 4.3 * x^{-0.2}$	0.92
	en	$1.2 - 5.0 * x^{-0.2}$	0.85	$0.8 - 7.6 * x^{-0.3}$	0.85
	hi	$2.1 - 5.0 * x^{-0.1}$	0.86	$1.0 - 10.8 * x^{-0.4}$	0.90
	ml	$1.1 - 21.4 * x^{-0.5}$	0.83	$1.3 - 7.8 * x^{-0.3}$	0.90
	mr	$1.5 - 7.3 * x^{-0.3}$	0.86	$1.4 - 6.4 * x^{-0.3}$	0.91
	ta	$1.1 - 12.8 * x^{-0.4}$	0.81	$1.1 - 10.0 * x^{-0.4}$	0.86
	ur	$1.1 - 5.1 * x^{-0.2}$	0.89	$1.1 - 4.7 * x^{-0.2}$	0.89
mr	bn	$1.0 - 9.3 * x^{-0.4}$	0.88	$1.1 - 5.1 * x^{-0.2}$	0.89
	en	$0.9 - 8.8 * x^{-0.3}$	0.81	$0.9 - 7.6 * x^{-0.3}$	0.87
	hi	$1.3 - 10.3 * x^{-0.3}$	0.86	$1.1 - 9.6 * x^{-0.4}$	0.91
	ml	$1.1 - 15.2 * x^{-0.4}$	0.83	$1.3 - 6.4 * x^{-0.3}$	0.90
	mr	$1.2 - 21.9 * x^{-0.5}$	0.86	$1.6 - 7.6 * x^{-0.3}$	0.92
	ta	$1.1 - 17.3 * x^{-0.4}$	0.79	$1.1 - 10.7 * x^{-0.4}$	0.85
	ur	$1.2 - 5.2 * x^{-0.2}$	0.92	$1.3 - 4.2 * x^{-0.2}$	0.91
pa	bn	$1.0 - 6.4 * x^{-0.3}$	0.86	$0.9 - 4.2 * x^{-0.3}$	0.82
	en	$1.2 - 4.0 * x^{-0.2}$	0.84	$1.1 - 2.9 * x^{-0.2}$	0.85
	hi	$1.9 - 5.9 * x^{-0.2}$	0.84	$1.8 - 4.0 * x^{-0.1}$	0.93
	ml	$1.0 - 9.7 * x^{-0.4}$	0.83	$1.2 - 3.6 * x^{-0.2}$	0.87
	mr	$2.7 - 5.3 * x^{-0.1}$	0.88	$1.3 - 3.9 * x^{-0.2}$	0.84
	ta	$1.4 - 6.3 * x^{-0.2}$	0.86	$1.0 - 4.7 * x^{-0.2}$	0.84
	ur	$1.2 - 4.5 * x^{-0.2}$	0.92	$0.8 - 4.2 * x^{-0.3}$	0.87
ta	bn	$1.1 - 7.2 * x^{-0.3}$	0.89	$1.0 - 4.6 * x^{-0.2}$	0.93
	en	$1.0 - 7.9 * x^{-0.3}$	0.83	$0.8 - 6.7 * x^{-0.3}$	0.86
	hi	$1.4 - 6.7 * x^{-0.3}$	0.90	$1.2 - 5.9 * x^{-0.3}$	0.92
	ml	$1.0 - 14.1 * x^{-0.4}$	0.83	$1.3 - 4.7 * x^{-0.2}$	0.92
	mr	$1.3 - 9.7 * x^{-0.3}$	0.86	$2.7 - 5.0 * x^{-0.1}$	0.94
	ta	$1.1 - 19.7 * x^{-0.5}$	0.79	$1.2 - 9.4 * x^{-0.3}$	0.88
	ur	$1.2 - 5.0 * x^{-0.2}$	0.92	$1.5 - 3.4 * x^{-0.1}$	0.92
te	bn	$1.1 - 5.4 * x^{-0.3}$	0.90	$0.8 - 4.7 * x^{-0.3}$	0.88
	en	$0.8 - 10.1 * x^{-0.4}$	0.79	$0.7 - 6.7 * x^{-0.4}$	0.83
	hi	$1.0 - 9.7 * x^{-0.4}$	0.91	$0.9 - 7.3 * x^{-0.3}$	0.86
	ml	$1.0 - 15.3 * x^{-0.4}$	0.83	$1.1 - 5.6 * x^{-0.3}$	0.88
	mr	$1.0 - 12.5 * x^{-0.4}$	0.87	$1.7 - 4.5 * x^{-0.2}$	0.93
	ta	$1.0 - 16.5 * x^{-0.4}$	0.81	$1.0 - 7.7 * x^{-0.3}$	0.87
	ur	$1.4 - 4.2 * x^{-0.2}$	0.91	$1.4 - 3.1 * x^{-0.1}$	0.90
ur	bn	$0.6 - 11.3 * x^{-0.5}$	0.86	-	0.76
	en	$1.1 - 5.5 * x^{-0.2}$	0.83	$1.0 - 5.9 * x^{-0.3}$	0.81
	hi	$2.6 - 4.8 * x^{-0.1}$	0.85	-	0.96
	ml	$1.1 - 8.2 * x^{-0.3}$	0.80	$2.5 - 5.0 * x^{-0.1}$	0.85
	mr	$4.3 - 6.1 * x^{-0.1}$	0.87	$5.2 - 7.0 * x^{-0.0}$	0.91
	ta	$1.1 - 5.0 * x^{-0.2}$	0.83	$1.2 - 5.2 * x^{-0.2}$	0.83
	ur	$1.0 - 42.2 * x^{-0.6}$	0.87	$1.1 - 20.9 * x^{-0.5}$	0.90

Table 11: *Power-law equations* empirically determined for each source and target pair. Please refer to Section A.5 for more details

Metric	Budget	Model	bn	en	hi	ml	mr	ta	ur
GM _{r=0}	1,000	XLM-R _{large}	128	157	145	134	133	163	140
		MuRIL _{large}	137	135	134	158	142	159	135
		XLM-R _{large}	704	792	693	794	696	628	693
	5,000	XLM-R _{large}	743	644	749	783	745	852	484
		MuRIL _{large}	1322	1349	1400	1481	1457	1479	1512
		XLM-R _{large}	1302	1468	1379	1421	1425	1448	1557
GM _{r=1}	1,000	XLM-R _{large}	126	160	159	134	129	163	129
		MuRIL _{large}	142	136	152	143	148	157	122
		XLM-R _{large}	710	805	713	803	707	639	623
	5,000	XLM-R _{large}	744	644	761	772	747	848	484
		MuRIL _{large}	1308	1363	1456	1465	1459	1471	1478
		XLM-R _{large}	1308	1488	1396	1406	1416	1441	1545

Table 12: *Optimal allocations under different budgets.* Please refer to Section A.5 for more details

Metric	Train Lang.	Model	NER	POS	NLI	QA	Average
M _{r=0} ↑ (Linguistic)	English	IndicBERT	16.5	16.1	6.5	5.3	11.1
		XLM-R _{base}	27.0	21.4	10.3	13.8	18.1
		MuRIL _{base}	33.4	20.7	10.5	14.9	19.9
		XLM-R _{large}	28.7	21.9	11.0	15.6	19.3
		MuRIL _{large}	31.5	21.3	11.1	15.9	20.0
	Hindi	IndicBERT	23.7	17.6	6.6	4.8	13.2
		XLM-R _{base}	30.4	22.4	10.6	13.5	19.2
		MuRIL _{base}	34.0	22.7	10.8	14.7	20.6
		XLM-R _{large}	33.0	22.4	11.5	15.2	20.5
		MuRIL _{large}	33.4	22.4	11.4	15.7	20.7
M _{r=1} ↑ (Demographic)	English	IndicBERT	39.2	44.2	36.6	28.4	37.1
		XLM-R _{base}	59.2	58.1	43.6	49.9	52.7
		MuRIL _{base}	69.6	54.7	45.5	53.8	55.9
		XLM-R _{large}	61.2	60.3	46.6	56.6	56.2
		MuRIL _{large}	68.2	58.6	47.4	57.9	58.0
	Hindi	IndicBERT	61.0	61.6	39.8	29.9	48.1
		XLM-R _{base}	70.3	66.7	45.8	50.6	58.3
		MuRIL _{base}	75.1	67.3	46.8	54.7	61.0
		XLM-R _{large}	74.4	66.8	49.4	53.2	60.9
		MuRIL _{large}	74.8	66.5	49.2	54.6	61.3
Gini Coeff. ↓	English	IndicBERT	0.67	0.81	0.92	0.84	0.81
		XLM-R _{base}	0.61	0.76	0.88	0.83	0.77
		MuRIL _{base}	0.59	0.76	0.88	0.83	0.76
		XLM-R _{large}	0.6	0.75	0.88	0.83	0.77
		MuRIL _{large}	0.59	0.76	0.88	0.83	0.77
	Hindi	IndicBERT	0.68	0.8	0.91	0.83	0.81
		XLM-R _{base}	0.59	0.75	0.87	0.83	0.76
		MuRIL _{base}	0.59	0.75	0.87	0.83	0.76
		XLM-R _{large}	0.59	0.76	0.88	0.83	0.77
		MuRIL _{large}	0.59	0.75	0.87	0.83	0.76
Efficiency ↑	English	IndicBERT	53.6	50.2	56.7	66.0	56.6
		XLM-R _{base}	44.4	48.1	57.4	76.7	56.7
		MuRIL _{base}	69.1	52.5	58.7	77.8	64.5
		XLM-R _{large}	44.4	48	57.3	76.3	56.5
		MuRIL _{large}	63.1	51.8	57.8	75.7	62.1
	Hindi	IndicBERT	62.5	53.7	56.9	64.5	59.4
		XLM-R _{base}	48.3	50.0	58.5	75.9	58.2
		MuRIL _{base}	69.8	56.2	59.8	77.3	65.8
		XLM-R _{large}	49.2	49.0	59.1	75.1	58.1
		MuRIL _{large}	65.2	53.7	58.8	75.0	63.2

Table 13: *Zero-shot fine-tuning results.* Overall, MuRIL_{large} scores highest on the utility metrics, the Gini coefficient is relatively high across all models and both MuRIL_{base} and MuRIL_{large} are, on average, equal with regards to efficiency. Note that the metrics are computed considering all 23 languages as detailed in Section 5.1. More discussions in Section 5.2.

Joint Reasoning on Hybrid-knowledge sources for Task-Oriented Dialog

Mayank Mishra
IBM Research
mayank.mishra1@ibm.com

Danish Contractor
IBM Research
danish.contractor@ibm.com

Dinesh Raghu
IBM Research
diraghu1@in.ibm.com

Abstract

Traditional systems designed for task oriented dialog utilize knowledge present only in structured knowledge sources to generate responses. However, relevant information required to generate responses may also reside in unstructured sources, such as documents. Recent state of the art models such as HyKnow (Gao et al., 2021b) and SEKNOW (Gao et al., 2021a) aimed at overcoming these challenges make limiting assumptions about the knowledge sources. For instance, these systems assume that certain types of information, such as a phone number, is *always* present in a structured knowledge base (KB) while information about aspects such as entrance ticket prices, would always be available in documents.

In this paper, we create a modified version of the MutliWOZ-based dataset prepared by (Gao et al., 2021a) to demonstrate how current methods have significant degradation in performance when strict assumptions about the source of information are removed. Then, in line with recent work exploiting pre-trained language models, we fine-tune a BART (Lewis et al., 2020) based model using prompts (Brown et al., 2020; Sun et al., 2021) for the tasks of querying knowledge sources, as well as, for response generation, without making assumptions about the information present in each knowledge source. Through a series of experiments, we demonstrate that our model is robust to perturbations to knowledge modality (source of information), and that it can fuse information from structured as well as unstructured knowledge to generate responses.

1 Introduction

Most existing work on task-oriented dialog systems assumes that the knowledge required for completing a task (eg: booking a restaurant reservation), resides in structured knowledge sources. Thus, typical task-oriented dialog systems require generating a *belief state*, that can be used to query a knowledge

base to fetch entity results; these results are then used to generate responses. Recognizing that information is not always present in structured resources, recently methods that can additionally use unstructured knowledge (eg: document collections), have also been developed (Kim et al., 2020; Gao et al., 2021a). However, current state-of-the-art models designed for such tasks make limiting assumptions about the nature of knowledge sources, that make them unsuitable for use in real-world settings.

Limitations of existing methods: First, current task-oriented dialog systems designed to reason over hybrid knowledge sources assume that a knowledge base and the unstructured knowledge source encode separate pieces of information about entities (eg: the zip-code is always in structured knowledge, ticket prices are always available in unstructured text) (Kim et al., 2020; Zhang et al., 2021). This is not reflective of real-world knowledge, where independent information systems are often fused to enable applications.

Second, existing systems are trained to learn the source of different pieces of information, thus, making them unsuitable for situations where any field that was previously in a structured knowledge source is now available in an unstructured knowledge source (and vice versa). In effect, a simple change in the modality of information can result in a failure of the model to utilize the information present in knowledge, as existing models memorize the source of every piece of information.

Third, such systems assume that each knowledge grounded response can contain information from only *one* source type (Kim et al., 2020; Gao et al., 2021a; Zhang et al., 2021) – either structured or unstructured knowledge. This is an artificial constraint imposed to make modelling easier, but real-world conversations can routinely require systems to fuse information from more than one knowledge type (eg: See Dialog turn 4 in Figure 1).

Contributions: In this paper, we present our work

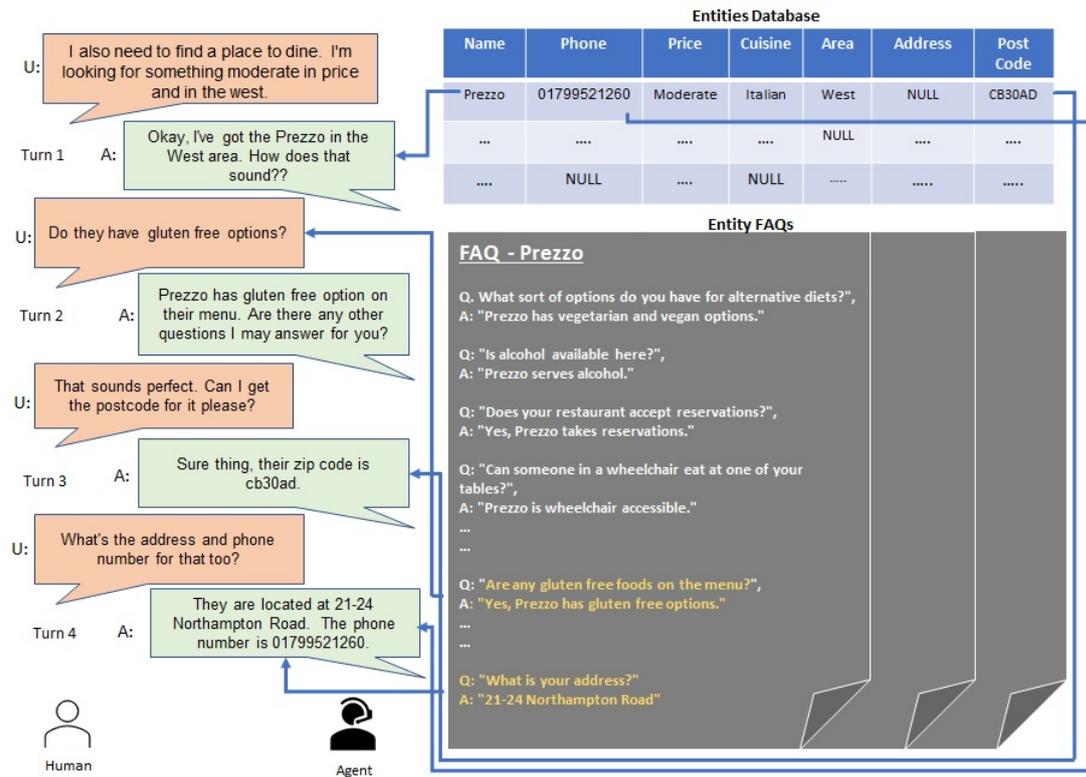


Figure 1: Example of a dialog requiring the use of data from two different sources. Agent Turn 4, requires incorporating information from both structured (DB Table) and unstructured data (a document consisting of FAQs for the entity).

aimed at removing each of these strict assumptions from task-oriented dialog systems. Current methods for joint-reasoning in task oriented dialogs have been developed using an augmented version of MultiWOZ 2.1 which contains additional dialog turns based on new unstructured information (Gao et al., 2021a). Unfortunately, no attempt has been made to distribute information across knowledge sources. We therefore create a modified version of this dataset (called HYBRIDTOD) that optimally redistributes information across structured and unstructured knowledge so that most dialogs in the train dataset are affected by this change.

A trivial method of redistributing information across structured and unstructured knowledge sources would be to arbitrarily move structured fields for some entities to the unstructured knowledge source. However, since the universe of entities in the dataset is very large and not all entities are directly referred to in the dialogs, such a method of redistributing information may not be as effective if the dialogs do not use the slot-values that have been redistributed. We therefore, develop an automated graph based approach which uses the max-cut of the graph to optimally redistribute infor-

mation from structured to unstructured knowledge sources.

Lastly, in line with recent work exploiting pre-trained language models, we fine-tune BART (Lewis et al., 2020) using prompts for the tasks of querying knowledge as well as response generation without making assumptions about the information present in each knowledge source. Specifically, we do Prompt+LM finetuning (Liu et al., 2021a) in which both the prompt and model parameters are trainable (Ben-David et al., 2021; Liu et al., 2021b; Han et al., 2021). Through a series of experiments, we demonstrate that our model is robust to perturbations to knowledge modality (source of information), and it can fuse information from structured as well as unstructured knowledge to generate responses.

In summary we make the following contributions¹: (1) We prepare a new version of the MultiWOZ-DSTC9 combined dataset (Kim et al., 2020; Gao et al., 2021a) called HYBRIDTOD to study the reasoning on hybrid knowledge sources for task oriented dialog systems. (2) We demon-

¹The constructed dataset and code used is available at <https://github.com/mayank31398/HybridToD>

Slot Type	Slot Values	Question Template	Answer Template
price	cheap	What is the price range?	It has \${price} pricing.
	expensive	How costly is \${restaurant name}?	\${restaurant name} is \${price}
cuisine	Italian	What is the cuisine?	\${restaurant name} caters for \${cuisine} cuisine.
	Thai	What type of food is served here?	You can find \${cuisine} food here

Table 1: Examples of templates used for moving slot values from the structured to the unstructured knowledge source.

strate that our model (referred to as JOINTLM) is also able to fuse information from both knowledge modalities and beats existing state-of-the-art systems on standardized metrics. (3) We present detailed ablation studies demonstrating the value of our modelling choices.

2 Related Work

Modeling Task Oriented Dialogs: Multiple flavours of this problem have been defined to address different aspects of modeling - eg: belief state tracking to assess whether a model is able to correctly decode the query needed given a current conversational context (Dey and Desarkar, 2021; Li et al., 2021; Yang et al., 2021), generating responses given belief states to assess whether a model is able to correctly predict the knowledge attributes to be used in a response (Yang et al., 2021; Chen et al., 2019; Gao et al., 2020; Mohapatra et al., 2021), end-to-end modeling of dialog systems where models are assessed on the correctness of the response generated including the values used from the knowledge base (Bordes et al., 2017; Raghu et al., 2021b), etc. Recent work that assumes that belief state annotations are latent and not available for training have also been developed (Raghu et al., 2021a).

Knowledge Grounded Dialog: Dialog systems that generate responses on information grounded in external knowledge have also been developed. Unlike, work on task oriented dialogs, which primarily focuses on using structured knowledge to complete a ‘goal’ or accomplish a ‘task’ (eg: POI recommendation for in car navigation (Eric et al., 2017), restaurant, hotel or flight booking (El Asri et al., 2017), etc), most existing knowledge grounded systems are designed to address informational needs of users (eg: answering queries based on collections of documents, making response recommendations to contact center agents). Finally, contemporaneous to our work, knowledge grounded response generation tasks that combine information from hybrid knowledge sources have also been proposed

(Nakamura et al., 2022). Here, unlike task oriented dialog systems, which require the retrieval of an entity to make recommendations or accomplish a task, in such tasks, the goal is to answer an informational seeking query in a chat-conversation. Models are required to use the dialog context to fetch related tables (often flattened and encoded as independent table cells), along with documents to generate a response.

3 The HYBRIDTOD Dataset

The dataset prepared by (Gao et al., 2021b) (referred to as the SEKNOW-MULTIWOZ dataset in this paper) is the only publicly available task-oriented dialog dataset in which the dialogs are grounded on two types of knowledge sources: structured and unstructured (FAQs). However, SEKNOW-MULTIWOZ is not indicative of a real-world setting due to two major limitations: (1) It has a strict, slot-type to knowledge-source type mapping. For example, the slot-type ‘cuisine’ is always in the structured source while ‘timings’ of operation would always be mentioned in unstructured documents, and (2) an agent response contains information from only one source (i.e., either from structured or unstructured). To alleviate these limitations, we systematically modify the knowledge sources in SEKNOW-MULTIWOZ to construct a new dataset that we refer to as HYBRIDTOD.

Dataset Construction: We first create an undirected graph $G = (V, E)$ where each vertex $v \in V$ is a unique slot-value and an edge $e \in E$ exists between two vertices, if the slot values represented by these vertices occur together in a training dialog utterance. For instance, in Figure 1 nodes associated with slot-values “21-24 Northampton Road” and phone number “01799521660” would have an edge between them due to Turn 4. Similarly, vertices corresponding to the values for slot-type ‘cuisine’ *Italian* and the slot-type ‘address’ *21-24 Northampton Road* would have had an edge between them if the utterance at Turn 4 was instead, “*It is an Italian restaurant located at 21-24 Northampton*

Domain	Context-Response pairs			Number of entities
	train	validation	test	train/validation/test
hotel	19370	2316	2295	33
restaurant	19716	2162	2188	110
attraction	8192	1226	1246	79
total	47278	5704	5729	222

Table 2: Number of context-response pairs in the dataset

Domain	SEKNOW-MULTIWOZ	HYBRIDTOD
hotel	10.97	6.79
restaurant	8.12	5.25
attraction	9	6.38

Table 3: Average number of slot values by domain in the structured knowledge source for each dataset.

Road". Our goal is to move some of the slot-values (vertices) in G that are originally in the structured knowledge source to the unstructured knowledge source so as to alleviate some of the limitations of the original dataset.

In order to identify which vertices to move, we create a maxcut of the graph G using the Max-CutBM algorithm (Boumal et al., 2016). A maxcut results in a graph in which the most number of edges from the original graph are ‘cut’. After the application of MaxCut, all slot values in one graph partition are retained in the structured knowledge source, while the others are converted to text QA pairs using templates and included as part of the unstructured document associated with that entity. The templates for the restaurant slot types ‘price’ and ‘cuisine’ are shown in Table 1 for illustration. Since the edges of the graph are based on slot-value mentions in dialog utterances, applying a maxcut modifies the knowledge source in a way that it affects most dialog turns in the dataset; in other words, the max-cut ensures the maximum possible utterances in the dataset have information fused from both knowledge sources.

Since we move slot values from one partition of the graph to the unstructured knowledge source, a slot type can now have some values in structured knowledge and some in unstructured knowledge (as an FAQ). We find that our approach ends up modifying each entity referred to in the dataset, and that slot-values of the same type are now distributed across different types of knowledge.

Domain	SEKNOW-MULTIWOZ	HYBRIDTOD	UNSTRUCTURED TOD
hotel	36.52	40.58	46.48
restaurant	14.96	17.83	22.7
attraction	0	2.62	8

Table 4: Average number of FAQs for each domain in the unstructured knowledge source.

For experimentation, we also create a version of the dataset with all slot values² moved from the structured to the unstructured knowledge source. We refer to this dataset as UNSTRUCTURED TOD. To construct HYBRIDTOD and UNSTRUCTURED TOD dataset, we only consider dialogs from 3 domains: hotel, restaurant and attraction. We omit dialogs from other domains as they do not have associated knowledge. For example, the taxi domain only contains the information that the slot-type *phone* should match the regular expression $[\"^[0-9]{10}\$\"]$, but does not contain any instance of phone numbers present in the train dialogs.

Dataset Statistics: The number of context-response pairs (spread across the 3 domains: hotel, restaurant and attractions) for HYBRIDTOD are shown in Table 2. We also show the entity distribution by domain-type. The restaurant domain dominates the knowledge sources, occupying almost half of the total entities and the other half is constituted by hotel and attraction domains. Tables 3 and 4 show the distribution of entity slot-values in structured knowledge sources and FAQs in the unstructured knowledge source for each domain in the datasets. As can be seen, the average number of slot-values presented in structured knowledge are lesser in HYBRIDTOD as compared to SEKNOW-MULTIWOZ and correspondingly the number of FAQs in HYBRIDTOD are higher as compared to SEKNOW-MULTIWOZ. We present the detailed slot-type distribution of SEKNOW-MULTIWOZ and HYBRIDTOD in the appendix. We find that approximately 50% slot-values are moved to unstructured knowledge from the structured sources for each slot-type.

Limitations of the Dataset: Information about entities is only redistributed from the structured knowledge source to the unstructured knowledge source. In effect, information that was previously in unstructured knowledge sources continues to remain there. Redistributing information from unstructured documents to structured documents would require annotations to be able to extract facets to be converted to slot-types.

We describe our model, JOINTLM in the next section.

²The entity name is also a slot type but we always retain it in both knowledge sources.

4 JOINTLM

The problem of utilizing information and responding to users in task-oriented dialogs can be broken down into parts: (i) Querying Knowledge Source (structured and/or unstructured) to return entities (ii) Generating Responses (eg:sharing information about entities, requesting for more details from the user, etc).

We represent the dialog context as $c = (u_1, r_1, \dots, u_n)$, where (u_i, r_i) represent the user and the system response utterance at i^{th} turn respectively. We represent the entity e required for generating the response as the concatenation of its slot-values (from structured KB), represented as e^s , and FAQs from the unstructured knowledge source, represented as e^{us} :

$$\begin{aligned} [e^s] &= \langle struct \rangle \langle slot \rangle slot_1 \langle val \rangle value_1 \\ &\quad \langle slot \rangle slot_2 \langle val \rangle value_2 \dots \\ [e^{us}] &= \langle unstruct \rangle \langle doc \rangle document_1 \\ &\quad \langle doc \rangle document_2 \dots \\ [e] &= [e^s] [e^{us}] \end{aligned}$$

where $\langle struct \rangle$, $\langle unstruct \rangle$ are special tokens to demarcate the start of structured knowledge and unstructured knowledge of an entity respectively. $\langle slot \rangle$, $\langle val \rangle$ demarcate the slot-type and its value and $\langle doc \rangle$ denotes the start of a document from unstructured knowledge. We train JOINTLM to jointly model two tasks: entity retrieval and response generation. We use a hyperparameter α to weigh the two tasks during training, where α denotes the number of training samples used for entity retrieval task. Note that $\alpha = 0.5$ denotes equal number of examples for both the tasks.

4.1 Entity Retrieval

As discussed, prior to generating a response, we need to retrieve the relevant entity required to generate the response. We represent the inputs to the language model (LM) for this task as:

$$\begin{aligned} &\langle entity_retrieval_task \rangle \langle u \rangle u_1 \langle r \rangle r_1 \dots \\ &\langle u \rangle u_n \langle entity \rangle [e_j] \end{aligned}$$

where, $e_j \in \mathcal{E}$, the set of all entities, $\langle entity_retrieval_task \rangle$ and $\langle entity \rangle$ are special tokens for task prompting and demarcating the start of an entity. We train the model to generate the special tokens $z_j = \langle relevant \rangle$ or $z_j = \langle irrelevant \rangle$

for each entity e_j given the context c . We choose the best entity e as:

$$e = \underset{e_j}{\operatorname{argmax}} p(z_j = \langle relevant \rangle | c, e_j) \quad (1)$$

During training we use a subset of the entities in \mathcal{E} for creating the positive and negative set of entities. However, at inference time, we evaluate on all the entities in \mathcal{E} .

4.2 Response Generation

After scoring all entities, we use the context and the best entity e (the entity with the highest score for the $\langle relevant \rangle$ token) and generate response using the same LM. We represent the inputs for this task as:

$$\begin{aligned} &\langle response_task \rangle \langle u \rangle u_1 \langle r \rangle r_1 \dots \\ &\langle u \rangle u_n \langle entity \rangle [e] \end{aligned}$$

where $\langle response_task \rangle$ is a special token to prompt this task. We train the model to generate the response token-by-token.

4.3 Training details

We train our model to minimize $\sum_{(c,r)} \mathcal{L}(\theta, c, r)$, where

$$\begin{aligned} \mathcal{L}(\theta, c, r) &= -\alpha \log p_\theta(z_j | c, e_j) \\ &\quad - (1 - \alpha) \log p_\theta(r | c, e_j) \end{aligned}$$

The first term in the above objective represents the log-likelihood of retrieving the relevant entity and the second term is the log-likelihood of generating the response. Note that the term α (percentage of samples for each task) can be adjusted by changing the number of examples for the two tasks in a given batch of fixed size.

To train our model, we use early stopping with $patience = 5$ for the above objective on the validation set to prevent overfitting of our model. The loss was optimized using AdamW optimizer (Loshchilov and Hutter, 2017). We use a batch-size of 8 examples, with 4 examples for entity retrieval and 4 for response generation per batch. For the 4 examples for entity retrieval, 2 are positive and 2 are negative examples (effectively our batch is $2 + 2 + 4$). We use equation 1 during inference to pick the highest scored relevant entity.

Train Dataset	Test Dataset	Model	Bleu-1	Bleu-4	slot-values		
					prec.	recall	F1
HYBRIDTOD	SEKNOW-MULTIWOZ	JOINTLM	30.63	8.60	50.48	45.37	47.79
		SEKNOW	29.20	7.83	43.16	28.65	33.14
HYBRIDTOD	HYBRIDTOD	JOINTLM	30.59	8.67	50.56	45.83	48.08
		SEKNOW	29.05	7.70	44.29	29.12	35.14
HYBRIDTOD	UNSTRUCTURED-TOD	JOINTLM	30.30	8.44	51.05	45.37	48.04
		SEKNOW	27.43	6.68	42.96	19.62	27.11

Table 5: All models trained on HYBRIDTOD and evaluated on the rest of the datasets

Train Dataset	Test Dataset	Model	Bleu-1	Bleu-4	slot-values		
					prec.	recall	F1
SEKNOW-MULTIWOZ	SEKNOW-MULTIWOZ	JOINTLM	29.07	8.06	49.74	41.31	45.13
		SEKNOW	31.00	9.14	52.17	44.98	48.31
SEKNOW-MULTIWOZ	HYBRIDTOD	JOINTLM	27.77	7.54	44.48	36.39	40.03
		SEKNOW	26.61	7.32	42.19	26.70	33.31
SEKNOW-MULTIWOZ	UNSTRUCTURED-TOD	JOINTLM	27.03	7.17	46.29	34.93	39.82
		SEKNOW	26.19	6.42	41.96	19.48	26.53

Table 6: All models trained on HYBRIDTOD and evaluated on the rest of the datasets

5 Experiments

Our experiments are aimed at answering the following questions: (1) How does JOINTLM perform compared to the baseline when trained and tested on HYBRIDTOD? (2) How does the change in slot-value distribution across structured and unstructured sources affect the performance of the models? (3) Is joint training of PromptLM for the two tasks of entity retrieval and response generation helpful? (4) How does JOINTLM compare with natural baselines for entity retrieval?

Experimental Setup: Task oriented dialog systems have to identify relevant entities (e.g. restaurants) from associated knowledge sources needed to generate a response. In order to identify these relevant entities, existing datasets provide the belief state annotations during training. Additionally, in our work for each dialog context, we associate a set of (positive) entities that exactly match the requirements present in the dialog context and a set of (negative) entities that do not match by an automated method. Note that the text snippets in the unstructured corpus do not have any annotations.

For all of our experiments, we use BART (Lewis et al., 2020) encoder-decoder based language model and finetune the pretrained model on the three datasets i.e, SEKNOW-MULTIWOZ (Gao et al., 2021a), HYBRIDTOD and UNSTRUCTURED-TOD datasets.

Baseline: We use the current state-of-the-art model for joint reasoning, SEKNOW (Gao et al., 2021a) model as our baseline. SEKNOW is designed to use belief state annotations – specifically, SEKNOW is trained to generate the belief state given the dialog context. These belief states are then used to query

the knowledge sources and generate a delexicalised response using the context and the generated belief state. The slot-values in the delexicalised response are then populated using an unordered set of entities returned by the belief state query on the structured knowledge source.

5.1 Evaluation Metrics

We report BLEU scores for assessing response generation performance and slot-value precision, recall and $F1$ for comparing the slot-value filling performance against the baseline. As described previously, since no new slot types were created from unstructured documents, the slot-value metrics are computed only using the slot-types that were originally present in the structured knowledge source.

We also report success@k for entity retrieval baselines to assess the performance of systems on the entity selection task. We define success@k as 1 if the top-k scored entities contain a relevant entity for response generation and 0 otherwise. However, note that it is not possible to measure success@k on SEKNOW since it generates the response using an unordered set of entities returned by the belief state query. We thus compare the two models only based on their performance on response generation.

5.2 Results

Knowledge-Source Memorization: We train and test both JOINTLM and the baseline model, SEKNOW on HYBRIDTOD and observe that JOINTLM outperforms SEKNOW by 13 points on slot-value F1 score (Row 1, Table 5). Also, the performance of SEKNOW drops from 48.31 (Row

Train Dataset	Test Dataset	Model	success@1	success@5	Bleu-1	Bleu-4	prec.	recall	F1
HYBRIDTOD	HYBRIDTOD	JOINTLM	84.50	86.57	30.59	8.67	50.56	45.83	48.08
		SEPLM	79.79	85.64	29.96	8.66	47.08	42.53	44.69
		TF-IDF	28.31	34.49	-	-	-	-	-

Table 7: Performance of models on the entity retrieval task.

1, Table 6) when trained/tested on the SEKNOW-MULTIWOZ dataset to 35.14 (Row 1, Table 5) when trained/tested on HYBRIDTOD dataset. This severe drop in performance is indicative of the fact that SEKNOW learns the source of slot-values and is unable to use information when the source of the particular slot-value can be varying (structured/unstructured) across entities.

Generalization of JOINTLM: To assess the generalization performance of the models, we train all the models on HYBRIDTOD and test on other datasets which have different slot-value distributions. As can be seen from Table 5, when trained on HYBRIDTOD, JOINTLM outperforms SEKNOW on all three dataset settings, SEKNOW-MULTIWOZ, HYBRIDTOD and UNSTRUCTURED-TOD across all response generation metrics. We also notice that JOINTLM trained on HYBRIDTOD is robust to change in the knowledge modality during inference (slot-value F1 stays at approx. 48). This is not the case for SEKNOW which exhibits large drop (31% from SEKNOW-MULTIWOZ to HYBRIDTOD and 45% from SEKNOW-MULTIWOZ to UNSTRUCTURED-TOD) in slot-value F1, as the distribution of slot-types changes in different datasets (Table 5).

We also train the models on SEKNOW-MULTIWOZ, and test on the other datasets and notice that JOINTLM outperforms SEKNOW on both HYBRIDTOD and UNSTRUCTURED-TOD (Table 6). However, SEKNOW has better slot-value F1 than JOINTLM on HYBRIDTOD. We hypothesize that this is because the belief state labels are more informative and provide a very strong signal for SEKNOW on SEKNOW-MULTIWOZ and this has the effect of SEKNOW learning the knowledge modality which is not the case for JOINTLM. This suggests that JOINTLM has better generalization performance.

5.3 Model Ablation Study

To study the importance of joint-training of our model, we also train a model without prompts using entity annotations, where two different BART (Lewis et al., 2020) models are trained for retrieval and generation. We call this model SEPLM. This

Dialog context

User: Suggest me some *Turkish* restaurants in Cambridge.

Agent: Yes there is a *Turkish* restaurant in Cambridge with two different locations, would you like the addresses?

User: I would like if its located in the *center* of the city price one located in the center of city and I don't care about the pricing.

Response

Ground truth response: I have two *Turkish* restaurants, both in the *center* and both *expensive*. May I recommend *Meze Bar*?

	SEKNOW	JOINTLM
SeKnow-MultiWOZ	I found a restaurant at <i>196 Mill Road City Centre</i> . Their phone number is <i>01223362372</i> .	<i>Meze Bar</i> is a <i>Turkish</i> restaurant in the <i>expensive</i>
HybridToD	I found a restaurant at <i>196 Mill Road City Centre</i> . Their phone number is {null}.	price range. Their address is <i>196 Mill Road City</i>
UnstructuredToD	I found a restaurant at {null}. Their phone number is {null}.	<i>Centre</i> .

Table 8: Response generated by SEKNOW and JOINTLM on the three datasets.

model is trained on HYBRIDTOD and is compared against JOINTLM on both entity retrieval and response generation (Table 7). We observe that JOINTLM outperforms SEPLM in both the tasks with a 5 points difference in success@1 and a 3 points difference in slot-value F1. This confirms that the joint modeling of the 2 tasks using prompting yields a better model than learning a separate model for the 2 tasks at hand. For a detailed evaluation on all other dataset combinations, please refer to the Appendix. For comparison with a non-neural entity retrieval baseline, we also report the success scores BM25 based TF-IDF retriever which are significantly worse than the neural retrievers used for JOINTLM and SEPLM. These experiments highlight the benefit of joint modeling of the two tasks.

5.4 Qualitative Study

In Table 8, we show the responses generated for a sample dialog by JOINTLM and SEKNOW on the three datasets used for our experiments. It should be noted that JOINTLM generates the same response for all the three datasets. However, SEKNOW is not able to populate the required slot-values for this entity (*Meze Bar*) in the response in HYBRIDTOD and UNSTRUCTURED-TOD when those slot-values are no longer available in the structured source.

6 Conclusion

In this paper we presented a new dataset, HYBRID-TOD that requires reasoning over both structured and unstructured knowledge sources to generate responses to dialogs. Unlike existing task-oriented dialog datasets, it does not restrict slot-types to specific knowledge sources. Through our experiments we demonstrated how existing methods do not adapt well to changing distributions of slot-type sources and that our model JOINTLM (trained using entity annotations rather than belief state), not only generates better responses by reasoning over both knowledge sources, it also learns a better retriever for entities. In future work, we also plan to train our models without using any annotations i.e without any supervision on entity label information.

7 Limitations

Our dataset and model are not intended to be directly used in a real-world system as they have some inherent limitations. As mentioned in Section 3, we only redistribute slot types from structured knowledge sources to unstructured knowledge sources. Due to a lack of resources we are unable to annotate unstructured documents – our dataset has a bias that certain information will always appear in unstructured information. In addition, we rely on a pre-trained language model, BART, to generate responses. We have not assessed to what extent the generated responses could exhibit any form of social bias or toxic language (when prompted). We do not recommend that our system be used in a real-world deployed chatbot without further study. Lastly, this work has been assessed only on English language data using a pretrained language model developed for English.

8 Acknowledgement

The authors would like to thank Raunak Sinha for his contributions on the scripts for the development of the dataset.

References

- Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. [PADA: A prompt-based autoregressive approach for adaptation to unseen domains](#). *CoRR*, abs/2102.12206.
- Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. [Learning end-to-end goal-oriented dialog](#). In

International Conference on Learning Representations.

- Nicolas Boumal, Vladislav Voroninski, and Afonso S. Bandeira. 2016. The non-convex burer–monteiro approach works on smooth semidefinite programs. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16*, page 2765–2773, Red Hook, NY, USA. Curran Associates Inc.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Xiuyi Chen, Jiaming Xu, and Bo Xu. 2019. A working memory model for task-oriented dialog response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2687–2693.
- Suvodip Dey and Maunendra Sankar Desarkar. 2021. [Hi-dst: A hierarchical approach for scalable and extensible dialogue state tracking](#). In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 218–227.
- Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. [Frames: a corpus for adding memory to goal-oriented dialogue systems](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany. Association for Computational Linguistics.
- Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. [Key-value retrieval networks for task-oriented dialogue](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.
- Silin Gao, Ryuichi Takanobu, and Minlie Huang. 2021a. [End-to-end task-oriented dialog modeling with semi-structured knowledge management](#). *CoRR*, abs/2106.11796.
- Silin Gao, Ryuichi Takanobu, Wei Peng, Qun Liu, and Minlie Huang. 2021b. [HyKnow: End-to-end task-oriented dialog modeling with hybrid knowledge management](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1591–1602, Online. Association for Computational Linguistics.

- Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. [Paraphrase augmented task-oriented dialog generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 639–649, Online. Association for Computational Linguistics.
- Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. [PTR: prompt tuning with rules for text classification](#). *CoRR*, abs/2105.11259.
- Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tur. 2020. [Beyond domain APIs: Task-oriented conversational modeling with unstructured knowledge access](#). In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 278–289, 1st virtual meeting. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021. [Zero-shot generalization in dialog state tracking through generative question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1063–1074.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *CoRR*, abs/2107.13586.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. [GPT understands, too](#). *CoRR*, abs/2103.10385.
- Ilya Loshchilov and Frank Hutter. 2017. [Decoupled weight decay regularization](#). *arXiv preprint arXiv:1711.05101*.
- Biswesh Mohapatra, Gaurav Pandey, Danish Contractor, and Sachindra Joshi. 2021. [Simulated chats for building dialog systems: Learning to generate conversations from instructions](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1190–1203, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Kai Nakamura, Sharon Levy, Yi-Lin Tuan, Wenhua Chen, and William Yang Wang. 2022. [HybridDialogue: An information-seeking dialogue dataset grounded on tabular and textual data](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 481–492, Dublin, Ireland. Association for Computational Linguistics.
- Dinesh Raghu, Nikhil Gupta, and Mausam. 2021a. [Un-supervised Learning of KB Queries in Task-Oriented Dialogs](#). *Transactions of the Association for Computational Linguistics*, 9:374–390.
- Dinesh Raghu, Atishya Jain, Mausam, and Sachindra Joshi. 2021b. [Constraint based knowledge base distillation in end-to-end task oriented dialogs](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5051–5061, Online. Association for Computational Linguistics.
- Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. [ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation](#). *CoRR*, abs/2107.02137.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. [Ubar: Towards fully end-to-end task-oriented dialog system with gpt-2](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14230–14238.
- Weijie Zhang, Jiaoxuan Chen, Haipang Wu, Sanhui Wan, and Gongfeng Li. 2021. [A knowledge-grounded dialog system based on pre-trained language models](#). *CoRR*, abs/2106.14444.

A Appendix

A.1 Additional Results

We present additional results for the comparison of JOINTLM, SEPLM and SEKNOW (Gao et al., 2021a) when trained on HYBRIDTOD and tested on the other datasets (Table 9). We see that JOINTLM outperforms SEPLM and SEKNOW on all the datasets demonstrating the importance of joint modeling.

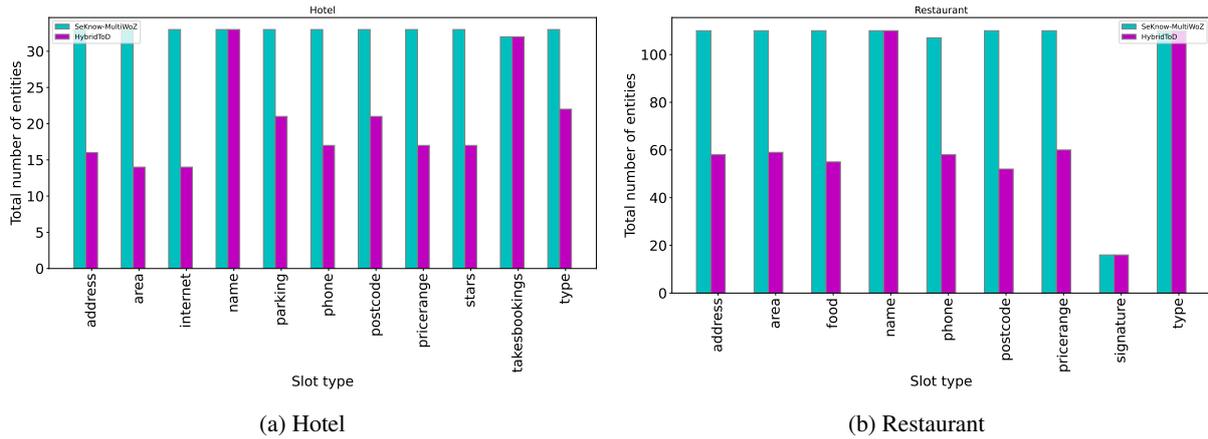


Figure 2: Figures 2a and 2b show the slot-value distribution by slot-types in the hotel and restaurant domains the three datasets.

Train Dataset	Test Dataset	Model	Bleu-1	Bleu-4	slot-values		
					prec.	recall	F1
HYBRIDTOD	SEKNOW-MULTIWOZ	JOINTLM	30.63	8.60	50.48	45.37	47.79
		SEPLM	30.03	8.63	47.26	42.76	44.89
		SEKNOW	29.20	7.83	43.16	28.65	33.14
HYBRIDTOD	HYBRIDTOD	JOINTLM	30.59	8.67	50.56	45.83	48.08
		SEPLM	29.96	8.66	47.08	42.53	44.69
		SEKNOW	29.05	7.70	44.29	29.12	35.14
HYBRIDTOD	UNSTRUCTURED TOD	JOINTLM	30.30	8.44	51.05	45.37	48.04
		SEPLM	29.78	8.41	47.08	41.63	44.19
		SEKNOW	27.43	6.68	42.96	19.62	27.11

Table 9: All models trained on HYBRIDTOD and evaluated on the rest of the datasets

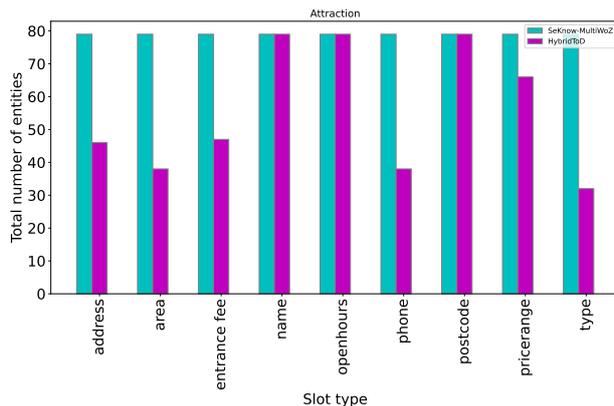


Figure 3: This figure shows the slot-value distribution by slot-types in the attraction domains in the three datasets.

A.2 Additional Dataset Statistics

We present the detailed slot-type distribution of SEKNOW-MULTIWOZ and HYBRIDTOD in Figure 2 and 3. We find that approximately 50% slot-values are moved to unstructured knowledge from the structured sources for each slot-type. The bar-graphs show the number of entities with a particular slot-type.

A.3 Hyperparameters and Training Details

For all our experiments, we use BART (Lewis et al., 2020) model from the HuggingFace Transformers library (Wolf et al., 2020). To train the BART model, we use early stopping with *patience* = 5 on the validation set to prevent overfitting of both the entity retriever and the response generator. We use learning rate = 10^{-5} with AdamW optimizer (Loshchilov and Hutter, 2017). We use a batch-size of 8 examples, with 4 examples for entity retrieval and 4 for response generation per batch. For the 4 examples for entity retrieval, 2 are positive and 2 are negative examples (effectively our batch is 2 + 2 + 4). All the experiments are conducted on a single A100 80GB GPU.

Revisiting Offline Compression: Going Beyond Factorization-based Methods for Transformer Language Models

Mohammadreza Banaei^{*1}, Klaudia Bałazy^{*2}, Artur Kasymov²
Rémi Lebret¹, Jacek Tabor², Karl Aberer¹

¹EPFL, ²Jagiellonian University

Abstract

Recent transformer language models achieve outstanding results in many natural language processing (NLP) tasks. However, their enormous size often makes them impractical on memory-constrained devices, requiring practitioners to compress them to smaller networks. In this paper, we explore offline compression methods, meaning computationally-cheap approaches that do not require further fine-tuning of the compressed model. We challenge the classical matrix factorization methods by proposing a novel, better-performing autoencoder-based framework. We perform a comprehensive ablation study of our approach, examining its different aspects over a diverse set of evaluation settings. Moreover, we show that enabling collaboration between modules across layers by compressing certain modules together positively impacts the final model performance. Experiments on various NLP tasks demonstrate that our approach significantly outperforms commonly used factorization-based offline compression methods.¹

1 Introduction

The recent trend of pre-training Transformer (Vaswani et al., 2017) language models on enormous unsupervised corpus has led to outstanding performances on many downstream tasks. For downstream tasks, these pre-trained models are then either fine-tuned (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019) or the *prompting* paradigm (Brown et al., 2020) is used (especially in the so-called Large Language Models), which avoids having a different model per task (Lieber et al., 2021; Rae et al., 2021; Smith et al., 2022; Thoppilan et al., 2022). In each of the two paradigms, it has been shown

that increasing the scale of language models generally leads to better performance on a range of downstream tasks (Devlin et al., 2019; Brown et al., 2020). Indeed, for autoregressive language models, Kaplan et al. (2020) demonstrated a power-law relationship between the number of parameters and the respective performance. Wei et al. (2022) further showed that certain abilities of language models emerge only when the number of its parameters passes certain thresholds, providing an incentive to scale these models further.

Although scaling up these language models make them empirically powerful across many diverse tasks, it makes them infeasible to train for many NLP practitioners due to huge pre-training costs. More importantly, even using the available pre-trained models for inference is becoming more challenging (especially for memory-constrained applications like edge devices), with recent models having hundreds of billions of parameters (Zhang et al., 2022).

With the rise of NLP model sizes, there have been many efforts to compress transformer language models without compromising their performance. Although being inherently different, many of these efforts rely on knowledge distillation (Hinton et al., 2015) to help the compressed model (i.e., the student model) better imitate the parent model (i.e., the teacher model). However, these approaches often need costly distillation on upstream or downstream tasks (Sanh et al., 2019) as well as expensive data augmentation techniques (Jiao et al., 2019) to help improve the compressed model performance. These approaches become even less feasible when enormous language models are being distilled.

Another line of research focuses on computationally-cheap methods (i.e., offline compression) where a smaller model can be achieved from a pre-trained model without it being necessarily fine-tuned over a downstream

^{*}Equal contribution

Correspondence to: mohammadreza.banaei@epfl.ch, klaudia.balazy@doctoral.uj.edu.pl

¹Our code is public: github.com/MohammadrezaBanaei/auto-encoder-based-transformer-compression

or upstream task. These offline methods include weight pruning (Li et al., 2016; Han et al., 2015), weight quantization (Zhou et al., 2016; Hubara et al., 2016), tensor factorization (Lan et al., 2019; Winata et al., 2019; Bałazy et al., 2021; Cordonnier et al., 2020) and hybrid approaches (Wang et al., 2019; Mao et al., 2020).

This paper proposes a novel offline factorization-based method for compressing transformer language models. The paper’s main goal is to propose an offline method that produces a competitive language model (compared to the original model perplexity) before any fine-tuning is performed. Similar to Bałazy et al. (2021), we use an autoencoder model (see Figure 1a) to compress different modules’ weights. However, unlike the previous work, our approach is not limited to the token embeddings (a.k.a. word embeddings) and can be applied to other transformer modules as well. We also propose and thoroughly investigate the impact of various enhancements for the approach of obtaining the compressed model. It is worth noting that although the experiments and ablation studies are only done BERT_{BASE} model, our

In Section 4 we demonstrate that applying small changes to the autoencoder architecture (e.g., introducing non-linearity to the decoder) and its loss objective results in superior performance to the Singular Value Decomposition (SVD) baseline as measured by model perplexity and its performance on the downstream tasks. Moreover, inspired by the redundancies present across self-attention heads (Cordonnier et al., 2020), in Section 4.2 we show that the compressed models perform in general better when compressing certain modules from different layers together.

Additionally, in Section 4.6 we investigate the effectiveness of a (parameter) sensitivity-based² compression by incorporating fisher information (Pascanu and Bengio, 2013) in the loss objective. We later show that incorporating these weights significantly improve the compressed language model performance (i.e., perplexity).

Finally, in Section 4.7 and in Section 4.8 we discuss the performance of our approach in comparison to various offline-compression baselines and demonstrate that our method provides the best or competitive quality of the compressed model.

Our main contribution can be summarized as

²We call it sensitivity as it measures how sensitive the model performance is to the reconstruction error of a certain parameter.

follows:

- We propose a novel autoencoder-based framework for low-cost compression of transformer language models and conduct an extensive ablation study on its different aspects.
- We show that enabling collaboration across layers by compressing different layers modules together boosts the performance.
- We demonstrate that our approach significantly outperforms other commonly used offline-compression methods on various NLP downstream tasks.³

2 Related work

Deep transformer language models have gained increasing attention in recent years since the seminal work of Devlin et al. (2019). Many recent efforts demonstrate that scaling up these language models’ parameters generally results in better performance on a range of downstream tasks (Devlin et al., 2019; Brown et al., 2020; Kaplan et al., 2020). This empirical observation resulted in recent language models having over a thousand times more parameters (Lieber et al., 2021; Rae et al., 2021; Smith et al., 2022) than the BERT_{BASE} model (Devlin et al., 2019). Although empirically powerful, these models are becoming harder to use for memory-constrained applications, which led to many efforts toward language model compression in recent literature.

Although many recent efforts for language model compression take advantage of distillation (Hinton et al., 2015) techniques to better imitate the uncompressed model (i.e., the teacher model) behavior, this paper focuses mainly on *offline* compression methods. By *offline*, we refer to approaches that do not need fine-tuning the whole model on a downstream/upstream dataset. In the case of language model compression, these methods aim to output a compressed model without losing too much performance (measured by perplexity) that can then be fine-tuned (with or without distillation) or prompted for a certain downstream task. It is worth noting that these methods can still be combined with distillation techniques, but starting the finetuning from a better compressed

³It is worth noting that although the experiments and ablation studies in this paper are only done on the BERT_{BASE} model, our proposed approach can potentially be used for any transformer-based architecture.

language model would generally reduce its costs of training (e.g., by improving convergence time).

Offline compression methods, while being diverse, can be roughly categorized into few paradigms, namely weight pruning (See et al., 2016; Li et al., 2016; Han et al., 2015; Fan et al., 2019; Michel et al., 2019; Voita et al., 2019), quantization (Gong et al., 2014; Hubara et al., 2016; Zhou et al., 2016), tensor factorization (Lan et al., 2019; Winata et al., 2019; Bałazy et al., 2021; Cordonnier et al., 2020; Panahi et al., 2021; Ren et al., 2022) and hybrid approaches (Wang et al., 2019; Mao et al., 2020).

This paper primarily focuses on the effectiveness of low-rank factorization-based approaches in recent literature (Lan et al., 2019; Panahi et al., 2021; Ren et al., 2022) as an *offline* compression method. Lan et al. (2019) proposed a SVD-based (Halko et al., 2011) technique to compress the token embedding module. Later works have shown that more complex architectures like autoencoders can result in better compression quality than SVD methods (Lioutas et al., 2020; Bałazy et al., 2021). By taking advantage of the autoencoder, we are able to more easily enforce different properties by either changing its training objective or architecture (e.g., preserving l_2 norm in reconstructed embeddings). Bałazy et al. (2021) emphasized on the importance of *direction* in token embedding compression, and in this work we demonstrate its potential importance for other transformer modules as well in different compression ratios.

Moreover, Cordonnier et al. (2020) showed the significance of redundant information in self-attention heads and compressed different heads (in a certain layer) together to improve compression performance. Following a similar idea, we later show that compressing heads from different layers together would generally further boost the compression quality.

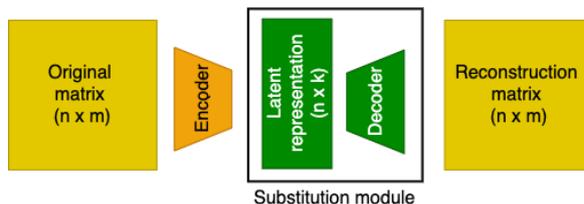
Hsu et al. (2022) proposed a weighted SVD (using Fisher Information) (Pascanu and Bengio, 2013) to outperform the classical SVD. We further investigate the benefits of a non-uniform compression (i.e., a weighted reconstruction loss in the autoencoder loss objective) in Section 4.6 by analyzing different weighting schemes for parameters.

Moreover, Ren et al. (2022) proposed using tensor decomposition techniques to compress language models to relatively high compression ratios while using a two-stage distillation technique.

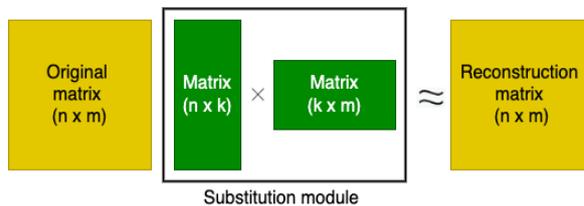
Moreover, Panahi et al. (2021) proposes using the Kronecker product as an alternative for the factorization of transformer modules. Appendix A.7 discusses using Tucker (De Lathauwer et al., 2000) or Kronecker-based methods as an offline approach. It is worth noting that models with relatively high compression ratios become highly dependent on distillation techniques to perform reasonably on downstream tasks. For instance, Ren et al. (2022) claims that even randomly initializing the compressed BERT nearly achieves identical performance compared to tensor decomposition from a pre-trained model.

3 Model

Our offline compression approach is based on the an autoencoder neural network architecture, similar to Lioutas et al. (2020) and Bałazy et al. (2021). However, in this work, we focus on compressing all the transformer weight matrices rather than just the token embedding matrix. Furthermore, we are exploring many more compression improvements using autoencoder as well as investigating architecture-independent techniques.



(a) Autoencoder-based compression with customizable objective function. Our approach minimizes the root mean square error (RMSE) and cosine distance between the original and reconstruction matrix. In this setting, the original matrix’s latent representation and the decoder form the substitution module.



(b) Classical matrix-factorisation-based compression with root mean square error (RMSE) minimization objective. Two smaller matrices, approximating the target matrix after multiplication, form the substitution module.

Figure 1: A high-level view of the matrix compression approaches using classical matrix-factorization and autoencoder model that is leveraged in this work. The purpose of the compression is to provide a parameter-efficient substitution module to replace the original matrix in the considered model.

The autoencoder architecture consists of encoder function $h(\cdot)$ that maps model input $x \in \mathbb{R}^m$ to some latent representation $l \in \mathbb{R}^k$. The second element of the architecture is the decoder function $g(\cdot)$ responsible for mapping $l \in \mathbb{R}^k$ into the approximation $\tilde{x} \in \mathbb{R}^m$ of the input x .

Let us assume that we want to compress the matrix $A \in \mathbb{R}^{(n \times m)}$. Using the gradient-descent algorithm, we train the autoencoder model to produce the appropriate approximation \tilde{A} of the original matrix. As a compressed module, we understand the hidden representation $h_\Psi(A) \in \mathbb{R}^{(n \times k)}$ together with the decoding module $g_\Phi(\cdot)$. In this setting, the formula for the *compression ratio* of the original module can be expressed as:

$$\frac{n \cdot m}{(n \cdot k) + |\Phi|}, \quad (1)$$

which is the ratio of the original matrix size to the hidden representation size and the number of parameters in the decoder module $|\Phi|$. We illustrated the approach of compressing a matrix using the autoencoder model in Figure 1a.

Our approach to offline compression based on the autoencoder offers flexibility in performing the ablation study as we are able to easily modify its elements, for example, decoder module complexity level or the loss function components. By using an autoencoder architecture with a linear decoder and the RMSE cost function, we can obtain the equivalent approximation as provided by a simple matrix factorization. An illustrative comparison of our compression method with the classical matrix factorization approach is shown in Figure 1.

Following Bałazy et al. (2021), we train our autoencoder with the multi-objective cost function consisting of l_2 norm loss and cosine distance loss:

$$\Psi_\beta(X, \tilde{X}) = (1 - \beta) \cdot L_2(X, \tilde{X}) + \beta \cdot CD(X, \tilde{X}), \quad (2)$$

where X represents the original matrix, \tilde{X} is the reconstructed matrix, $L_2(X, \tilde{X})$ represents the root mean square error (RMSE) loss function, and $CD(X, \tilde{X})$ is the mean cosine distance loss for all pairs of vectors (rows) of the original and reconstructed matrices. The β hyperparameter ($0 \leq \beta \leq 1$) is responsible for determining the weight we would like to assign to the different components of the loss function.

4 Experiments

This section describes our motivations and the results of various analyses and experiments that we

conducted to investigate the topic of offline compression thoroughly.

We performed our experiments for different weight matrices in the transformer architecture, as each type of weight matrix may have different characteristics, and a given compression method may or may not be appropriate. The analyses described below are performed for token embedding, self-attention (keys, queries, values), and output-dense weight matrices.

We focus our study on the BERT_{BASE} model (Devlin et al., 2019), but the same methods could be applied to other transformer architectures. All experiments are conducted for three compression ratios (3, 10, and 25) to investigate the differences given the different number of available parameters. All experimental settings of the various studies presented in the following sections are included in Appendix A.

We evaluate the quality of our compressed models on the masked (Devlin et al., 2019) language modeling task (using the WikiText-103 test dataset (Merity et al., 2016)) and multiple datasets from the GLUE benchmark (Wang et al., 2018).

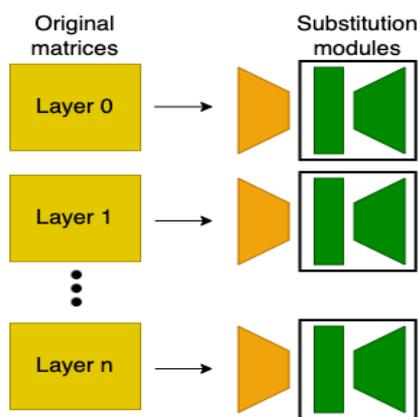
4.1 Cosine distance objective

First, we investigate whether including the direction component in the compression objective has a positive effect on the compression of weight matrices other than token embeddings in the transformer model. Bałazy et al. (2021) demonstrated that supplementing the loss function with the cosine distance between pairs of rows of the original and reconstructed matrix produces noticeably better compression results for the token embeddings matrix. Unfortunately, their study does not examine other matrices in the transformer, whereas because of the different nature of these matrices, we believe it is worth investigating.

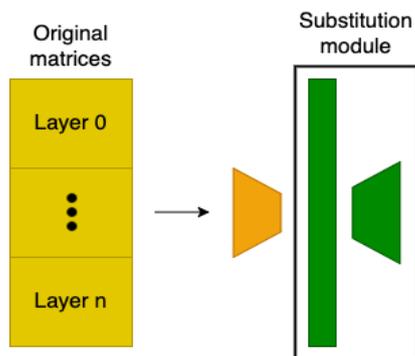
Results In Table 3 and Table 4 (in Appendix A.1), we present the effect of adding the cosine distance component to the cost function for the keys and output-dense matrices from the BERT_{BASE} model. It seems that for matrices other than token embeddings, considering the direction of vectors (rows in the matrix) in most cases may have a positive impact on the final results of the compressed model. However, the benefits of using this component are not as significant as in the case of the token embeddings matrix. Indeed, there are some examples where minimizing only Euclidean distance or

adding only a small proportion of cosine distance provides the best results. We suppose this behavior is the consequence of the token embeddings matrix nature, where the rows represent specific tokens used to construct the words. It seems that the representation of the part or entire word is largely encoded in the vector direction. This characterization does not necessarily apply to other matrices though often there is a subtle benefit from adding a cosine distance component to the reconstruction objective.

4.2 Concatenated and separated weight matrices



(a) Separated matrices compression mode. All substitution modules have separate decoder.



(b) Concatenated matrices compression mode. All substitution modules share the decoder module that enables collaboration between layers.

Figure 2: Separated and concatenated weight matrices compression. We demonstrate that compressing concatenated matrices from all layers provides a better-performing substitution module. In the concatenated setting, the substitution modules share a decoder that allows for cross-layer collaboration and provides the potential to further eliminate redundant information.

This section investigates whether compressing separately each weight matrix from the considered model is the best possible strategy. Our in-

tuition is that compressing together the same type of weight matrices from different layers may bring certain advantages. First, it could allow for minimizing redundant information in the model weights, and second, it could enable collaboration between compressed modules across different layers. Suppose that the neural network model consists of n layers (l_0, l_1, \dots, l_{n-1}). Each layer l_i encapsulates a particular weight matrix W_{l_i} . Conventionally, each W_{l_i} matrix is considered separately during the compression process. In the concatenated mode, we propose compressing a single matrix $W = [W_{l_0}, W_{l_1}, \dots, W_{l_{n-1}}]$ resulted from concatenating all W_{l_i} matrices. Given the proposed compression process, the compressed weight matrices share a common decoder as illustrated in Figure 2.

Results Experiments discussed in this section demonstrate that compressing concatenated weight matrices performs better than compressing each matrix separately in terms of the compressed model performance as well as the compression process time. Figure 3 presents the performance achieved by models with compressed output-dense matrices in separated and concatenated modes (similar experiments are presented for key, query, and value matrices in Figure 7 in Appendix A.2). We report the initial perplexity and the final score achieved on the MRPC and SST2 downstream tasks for different compression ratios. We observe the apparent dominance of the concatenated mode over the separated mode for both perplexity and downstream task performance. This may indicate that sharing the decoder helps to reduce redundant information and saves parameters for further knowledge encoding.

Furthermore, the separated compression mode is more computationally expensive at the initial stage as we must compress each matrix individually. In the concatenated mode, we perform only a single compression process on the matrices' concatenation.

Considering the better performance and faster training process, we only analyze the concatenated weight matrices compression in the following sections.

4.3 Initial perplexity vs downstream tasks performance

Perplexity is a popular measure determining how well the language model predicts a particular se-

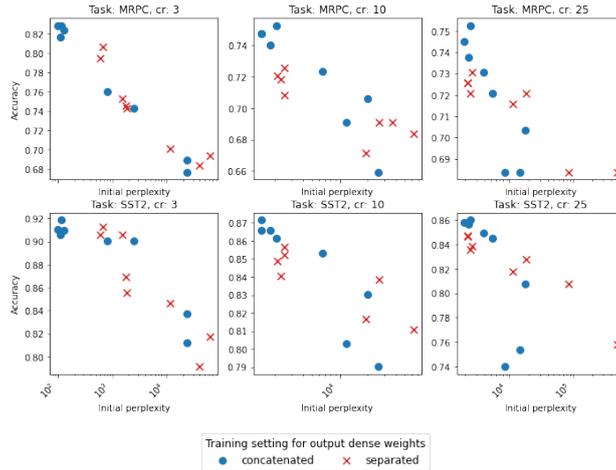


Figure 3: Initial perplexity and downstream tasks performance of output-dense matrices’ compression in separated/concatenated compression modes⁴. Using the concatenated mode generally results in better performance for this module. We observe similar pattern for key, query, and value weight matrices (see Figure 7 in Appendix A.2).

quence of tokens (the lower the perplexity the better). This section introduces that the masked language model perplexity metric may be considered as a low-cost yet effective way to evaluate a compressed module. We show that in most cases the compressed models with the lowest initial perplexity yields the best performance when fine-tuned on a downstream tasks.

Results We examined the relation between the initial $BERT_{BASE}$ perplexity after applying compressed weight matrices and its final performance after fine-tuning on a downstream task. In Figure 8 (in Appendix A.3) we report results for token embeddings matrix, self-attention keys, queries, values and final output-dense matrices. We observe that in most cases models with the lowest initial perplexity result in the best performance on the downstream task (MRPC and SST2). Therefore, we consider the masked language model perplexity metric to be a good low-cost method to preliminarily evaluate the quality of a compressed module.

4.4 Linear and non-linear decoder module

In this section we investigate the effect of using different decoder module in the autoencoder model on the final model’s performance. We experiment with a simple linear layer decoder and two non-linear decoder versions.

Results Figure 4 presents initial perplexity and final downstream tasks performance achieved when using linear and non-linear decoder in the autoen-

coder model while compressing token embeddings matrix. We may observe that for the token embeddings better final results are produced when using non-linear decoder. However, as demonstrated in Figure 9 (in Appendix A.4), a different pattern is apparent for key matrices where linear models considerably outperform the non-linear versions in most cases.

4.5 Preserving vector norm

Furthermore, we examine whether preserving the original l_2 vector norms of the vectors representing rows in the reconstructed matrix to be the same as in the original vectors is beneficial for the compression.

Results Figure 5 presents initial perplexity and downstream tasks performance when enabling or disabling the preserving vector norm technique for the token embeddings matrix. We may see that in most cases the version with enabled preserving vector norm achieves better results. In addition to token embeddings, in Figure 10 (in Appendix A.5) we also demonstrate the effect of preserving l_2 vector norm during compression of the output-dense matrix.

4.6 Sensitivity

Most offline compression methods focus only on the raw weight matrices taken from the considered pre-trained model. However, we could also leverage the unsupervised upstream dataset to improve the compression quality. Hsu et al. (2022) proposed using additional weights computed on the

⁴Each point represents one hyperparameter setting.

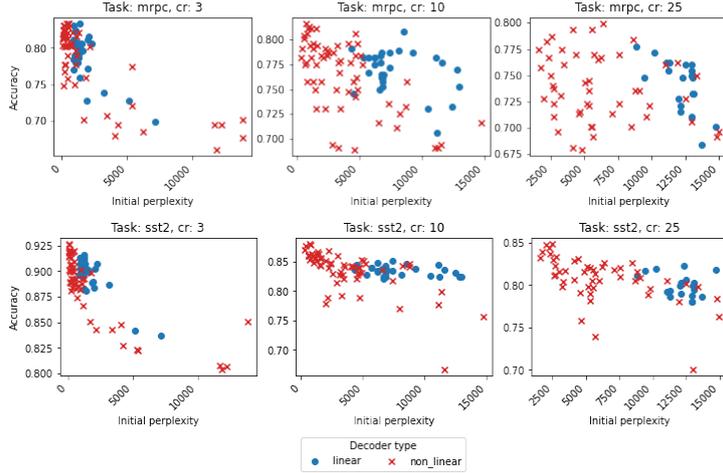


Figure 4: Initial perplexity and downstream tasks performance for the compressed token embeddings matrix when using either a linear or a non-linear decoder module in the autoencoder model. Here we present results for different hyperparameters settings (see Appendix A.4 for further details). A non-linear decoder seems to be a better choice for compressing this module.

entire upstream dataset to enhance the low-rank factorization method. They used the Fisher Information weights I that measure the amount of the observable information in dataset D about a single model parameter w . A feasible approximation \hat{I}_w of the Fisher Information I_w for parameter w may be expressed as:

$$I_w = E[(\frac{\partial}{\partial w} \log P(D|w))^2] \approx \frac{1}{|D|} \sum_{d \in D} (\frac{\partial}{\partial w} L(d; w))^2 = \hat{I}_w \quad (3)$$

where L is the target pre-training task objective (e.g., cross-entropy or MSE).

For the entire weights matrix W , Hsu et al. (2022) presented even more simplified and computationally effective row-wise diagonal Fisher Information matrix \hat{I} , where each diagonal value is the sum of the corresponding row of the Fisher Information approximation matrix \hat{I}_W :

$$\hat{I} = \text{diag}(\sqrt{\sum_{j=1} \hat{I}_{W_{1j}}}, \dots, \sqrt{\sum_{j=1} \hat{I}_{W_{nj}}}). \quad (4)$$

We present the distributions of the row-wise Fisher Information for the upstream dataset (i.e., masked language modeling on the WikiText-103 dataset) in Figure 6. We notice that each distribution contains some outliers which point to the potentially irrelevant weights in the considered weight matrix. In this section, we demonstrate that the weights’ relevance information may be leveraged in the compression process to improve the quality of compressed modules.

Hsu et al. (2022) used the Fisher Information directly on the original model weights in their Fisher-Weighted SVD (FWSVD) approach:

$$W \approx FWSVD(W) = \hat{I}^{-1} SVD(\hat{I}W). \quad (5)$$

In contrast, our method does not modify the original weight matrices but rather uses the Fisher Information in the loss function to help the model focus more on important weights. Moreover, we apply different transformations on the original Fisher Information values to modify the relative importance of the module weights to reduce the undesirable influence of outliers (Appendix A.6 discusses various transformations we experimented with for different modules).

CR	Method	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)
3	AE	118.41	91.97	88.53 / 84.31
	AE+Fisher	33.27	92.55	88.36 / 83.33
10	AE	712.98	88.07	85.87 / 80.88
	AE+Fisher	250.59	89.33	87.27 / 81.62
25	AE	4926.08	82.80	84.19 / 77.45
	AE+Fisher	2728.41	83.83	84.35 / 77.70

Table 1: The effect of adding the Fisher Information to the autoencoder-based (AE) compression of token embeddings. We report the compressed BERT_{BASE} upstream task perplexity (on the WikiText-103 dataset) and the downstream performance over two GLUE tasks. Each AE result represents a median from 3 runs with different seeds.

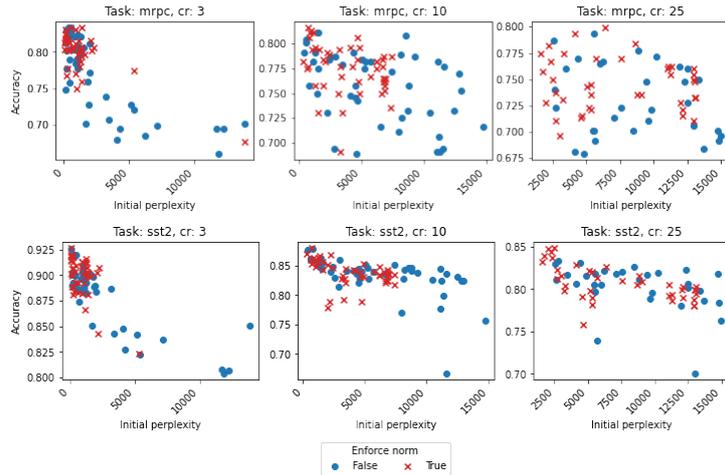


Figure 5: The effect of preserving l_2 vector norm on the perplexity and downstream tasks performance while compressing the token embeddings matrix for different hyperparameters settings. Preserving norm seems to generally improve the compression performance for this module.

CR	Architecture	MRPC (F1/Acc)	SST-2 (Acc)	RTE (Acc)	QNLI (Acc)	QQP (F1/Acc)
1	BERT _{BASE}	88.85/84.07	92.32	65.70	90.66	87.49/90.71
3	SVD	84.35/77.45	86.70	62.09	85.61	84.64/88.38
	Our	85.25/77.77	90.25	62.45	88.68	86.07/89.71
10	SVD	78.46/68.38	82.00	52.71	77.48	79.36/83.45
	Our	81.45/71.08	83.94	57.76	81.48	81.57/85.75
25	SVD	77.10/66.42	78.33	53.43	62.66	74.21/79.10
	Our	81.60/71.32	80.05	55.23	72.96	78.45/83.11

Table 2: Final BERT_{BASE} model compression (token embedding matrix, all key matrices, and all output-dense matrices). The baseline SVD algorithm compresses each matrix separately. Our autoencoder-based approach, incorporates mechanisms developed in the ablation study presented in this work (see Table 8 for the detailed AE design choices). For each setting, we present the median score from experiments with three different seeds. Our approach consistently outperforms the classical factorization method.

Results Table 1 presents the benefits of incorporating sensitivity for compression of token embeddings where both upstream perplexity and downstream task performance is improved. We further demonstrate the positive influence of Fisher information for three transformer modules in Table 5 (in Appendix A.6) for both autoencoder and SVD methods. Additionally, Table 6 (in Appendix A.6) presents the compression performance provided by AE using different Fisher Information transformations. We observe that incorporating the Fisher Information with batch normalization into the compression process considerably improves the model perplexity as well as the downstream task performance.

4.7 Comparison with other offline compression approaches

In Table 5, we compare our approach with the most popular matrix factorization method, namely Singular Value Decomposition (SVD), for the compression of three different transformer modules. We may see that our approach outperforms or is competitive with SVD in most settings. Additionally, in Appendix A.7, we discuss the poor performance of Kronecker Product and Tucker Decomposition (as two other factorization-based methods) in the offline compression setting. We also compare our solution to a non-factorization baseline, namely pruning, and show that our autoencoder-based method also outperforms it in most studied settings (see Table 7 in Appendix A.7).

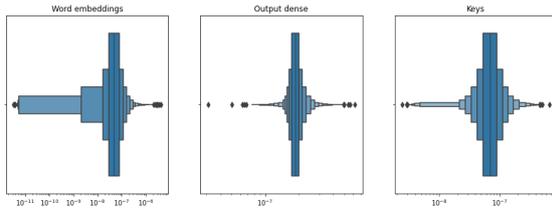


Figure 6: Row-wise Fisher Information distribution for three different modules in the $BERT_{BASE}$ model. The Fisher information values are passed as importance weights in the autoencoder loss function to help the compression model focus more on the module’s important weights. Appendix A.6 discusses different transformations applied to Fisher information to help the compression model handle outlier Fisher information values.

4.8 Compressing multiple types of transformer modules

For the final experiment, we took into consideration all the analysis insides presented in this work and prepared the concluding experiment on offline compression methods. In this experiment, we compressed multiple $BERT_{BASE}$ weight matrices at the same time (token embedding matrix, all key matrices and all output-dense matrices). We compared the offline compression quality produced by the autoencoder approach and the baseline SVD factorization matrix method.

For the compression with SVD we classically compressed each matrix separately. We tested three different seeds and various number of iterations for SVD algorithm. For the autoencoder approach, we compressed considered matrices by selecting appropriate mechanisms based on our ablation study. For both SVD and our approach, we report the median of the final scores from the experiments with three different seeds to exclude potential outliers. The compressed matrices with the lowest perplexity were applied into $BERT_{BASE}$ that was then fine-tuned on various NLP tasks (MRPC, SST-2, RTE, QNLI and QQP). In the resulting table we reported the median of the final scores to exclude potential outliers.

Final experiment results are presented in Table 2. Our approach consistently outperforms the SVD baseline on all tested downstream tasks.⁵

4.9 Compression time

Generally, compressing modules using autoencoder and SVD takes a comparable amount of time. How-

⁵The hyperparameter setting for autoencoder is provided in Table 8 in Appendix A.8

ever, using concatenated mode (as proposed in our paper) speeds up this process significantly. In Appendix A.9, we report compression times for different modules (Table 9) and the compressed models’ inference and fine-tuning time.

5 Conclusions

This work comprehensively studies various methods for the offline compression of transformer language models. We analyze various changes in the proposed architecture and its optimization function. We test different input modifications and evaluate the compressed language model performance in each scenario. By analyzing various compression settings, we show that our autoencoder-based approach outperforms classical matrix factorization on various NLP downstream tasks. Furthermore, we believe the techniques analyzed in this study might also be useful for low-cost compression of different weight matrices unrelated to language models.

Limitations

A limitation of our approach that we may identify is the need to analyze each module type to determine the best mechanisms for its compression. Our module-specific findings could be reflected in corresponding modules in other language models, but this would require further investigation. Additionally, A (reasonably-sized) unsupervised corpus must also be used for computing the Fisher Information for the compression procedure, which is more computationally demanding than other offline approaches suggested in this study.

Acknowledgements

The work of Klaudia Bałazy was supported by the National Centre of Science (Poland) Grant No. 2020/39/D/ST6/01332. Klaudia Bałazy is affiliated with Doctoral School of Exact and Natural Sciences at the Jagiellonian University. The research of Jacek Tabor was carried out within the research project "Bio-inspired artificial neural network" (grant no. POIR.04.04.00-00-14DE/18-00) within the Team-Net program of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund. Artur Kasymov work was supported by the National Centre of Science (Poland) Grant No. 2019/33/B/ST6/00894.

References

- Klaudia Balaży, Mohammadreza Banaei, Rémi Lebret, Jacek Tabor, and Karl Aberer. 2021. Direction is what you need: Improving word embedding compression in large language models. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 322–330.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. 2020. Multi-head attention: Collaborate instead of concatenate. *arXiv preprint arXiv:2006.16362*.
- Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. 2000. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*.
- Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*.
- Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288.
- Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Yen-Chang Hsu, Ting Hua, Sungeon Chang, Qian Lou, Yilin Shen, and Hongxia Jin. 2022. [Language model compression with weighted low-rank factorization](#). In *International Conference on Learning Representations*.
- Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks. *Advances in neural information processing systems*, 29.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- Diederik P. Kingma and Jimmy Ba. 2014. [Adam: A method for stochastic optimization](#). Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.
- Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. Jurassic-1: Technical details and evaluation. *White Paper. AI21 Labs*.
- Vasileios Lioutas, Ahmad Rashid, Krtin Kumar, Md Akmal Haidar, and Mehdi Rezagholizadeh. 2020. Improving word embedding factorization for compression using distilled nonlinear neural decomposition. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2774–2784.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, Georgia, USA.
- Yihuan Mao, Yujing Wang, Chufan Wu, Chen Zhang, Yang Wang, Yaming Yang, Quanlu Zhang, Yunhai Tong, and Jing Bai. 2020. Ladabert: Lightweight adaptation of bert through hybrid model compression. *arXiv preprint arXiv:2004.04124*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

- Aliakbar Panahi, Seyran Saeedi, and Tom Arodz. 2021. [Shapeshifter: a parameter-efficient transformer using factorized reshaped matrices](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 1337–1350. Curran Associates, Inc.
- Razvan Pascanu and Yoshua Bengio. 2013. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.
- Yuxin Ren, Benyou Wang, Lifeng Shang, Xin Jiang, and Qun Liu. 2022. Exploring extreme parameter compression for pre-trained language models. *arXiv preprint arXiv:2205.10036*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Abigail See, Minh-Thang Luong, and Christopher D Manning. 2016. Compression of neural machine translation models via pruning. *arXiv preprint arXiv:1606.09274*.
- Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deep-speed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.
- Elena Voita, David Talbot, Fedor Moiseev, Rico Senrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2019. Structured pruning of large language models. *arXiv preprint arXiv:1910.04732*.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.
- Genta Indra Winata, Andrea Madotto, Jamin Shin, Elham J Barezi, and Pascale Fung. 2019. On the effectiveness of low-rank matrix factorization for lstm model compression. *arXiv preprint arXiv:1908.09982*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.
- Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2016. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*.

A Experiments

In this section, we describe the general assumptions for the experiments and the specific setting of the hyperparameters for each individual experiment.

We train the autoencoder model using gradient descent procedure and Adam optimizer (Kingma and Ba, 2014). In most of our experiments, resulting compressed modules are inserted into a pre-trained language model and fine-tuned on two different downstream tasks from GLUE benchmark (Wang et al., 2018), MRPC and SST2, with

a default learning rate $\lambda = 2 \cdot 10^{-5}$ proposed by Hugging Face Transformers⁶ (Wolf et al., 2019).

A.1 Cosine distance objective

In Table 3 and Table 4, we report the effect adding the cosine distance component to the compression objective for the keys and output-dense matrices from the BERT_{BASE} model.

Experimental setup We study the effect of the cosine distance component in the loss function formulated in Equation (2) on the compression quality of self-attention keys matrices and the fully-connected output-dense weight matrices. We compress each of these matrices from each layer separately and then apply all the compressed matrices from certain type (keys or output-denses) to the transformer model to evaluate the compression quality. We inspect the effect of various ratios between Euclidean distance component (L_2) and the cosine distance component (CD) in Equation (2), namely 1:0, 10:1, 1:1, 1:10 which corresponds to $\beta \in \{0.0, 0.0909, 0.5, 0.909\}$. For each model we test different learning rates $\lambda \in \{5 \cdot 10^{-3}, 10^{-3}, 5 \cdot 10^{-4}, 10^{-4}\}$.

Compression ratio	Cosine coefficient	Initial perplexity	MRPC (Acc)	SST-2 (Acc)
3	0.0	22.9	74.5	91.0
	0.0909	22.2	76.2	92.1
	0.5	25.5	77.2	92.1
	0.9091	27.0	76.2	91.4
10	0.0	45.8	72.5	91.5
	0.0909	32.5	72.0	90.7
	0.5	34.4	70.3	91.5
	0.9091	43.2	71.3	91.6
25	0.0	116.9	70.3	90.6
	0.0909	167.6	69.6	90.2
	0.5	115.3	69.1	90.9
	0.9091	507.7	69.3	90.1

Table 3: The impact of adding a component responsible for preserving the direction in the compression of the key matrices from the self-attention block in BERT_{BASE} model. We report the best (across different learning rates) initial model’s perplexity after compression as well as it’s performance on a downstream tasks for the different cosine coefficients (β in Equation (2)). Adding a component to the compression objective that aims to preserve the direction of the vectors in reconstructed matrix may bring a slight improvement on the quality of the result.

⁶<https://github.com/huggingface/transformers>

Compression ratio	Cosine coefficient	Initial perplexity	MRPC (Acc)	SST-2 (Acc)
3	0.0	327.4	83.1	91.0
	0.0909	332.8	82.3	91.5
	0.5	599.2	79.4	90.5
	0.9091	1510.4	75.2	90.5
10	0.0	1396.5	74.5	88.1
	0.0909	1917.8	74.5	85.9
	0.5	2099.3	72.0	84.9
	0.9091	2326.1	71.8	84.0
25	0.0	1988.0	73.8	84.5
	0.0909	2099.0	73.8	85.3
	0.5	2144.7	72.5	84.7
	0.9091	2148.6	72.5	84.6

Table 4: The impact of adding a component responsible for preserving the direction in the compression of the output-dense matrices in BERT_{BASE} model. Different cosine coefficients refer to β in Equation (2). We report the score for the model with the best initial perplexity across different learning rates as well as it’s performance on MRPC and SST2 downstream tasks.

A.2 Concatenated and separated weight matrices

In Figure 7 we present the performance achieved by models with compressed key, query and value matrices when using separated and concatenated mode in the compression process.

Experimental setup With the objective of comparing the compression of separate and concatenated matrices, we analyze the various matrices in the transformer architecture: the key, query, and value matrices from the self-attention module and output-dense matrix (one of the fully connected end matrices). We optimize loss function described previously in Equation (2) with 1:1 and 1:10 ratios for the l_2 norm loss coefficient and cosine distance coefficient, respectively. The decoder in the autoencoder model is a single fully connected layer. The model is trained with different learning rates $\lambda \in \{5 \cdot 10^{-3}, 10^{-3}, 5 \cdot 10^{-4}, 10^{-4}\}$.

A.3 Initial perplexity vs downstream tasks performance

In Figure 8 we present the initial perplexity and the performance on MRPC and SST2 downstream tasks for the language model with various compressed modules.

Experimental setup We examine the initial masked language model perplexity and downstream tasks performance relation for token embed-

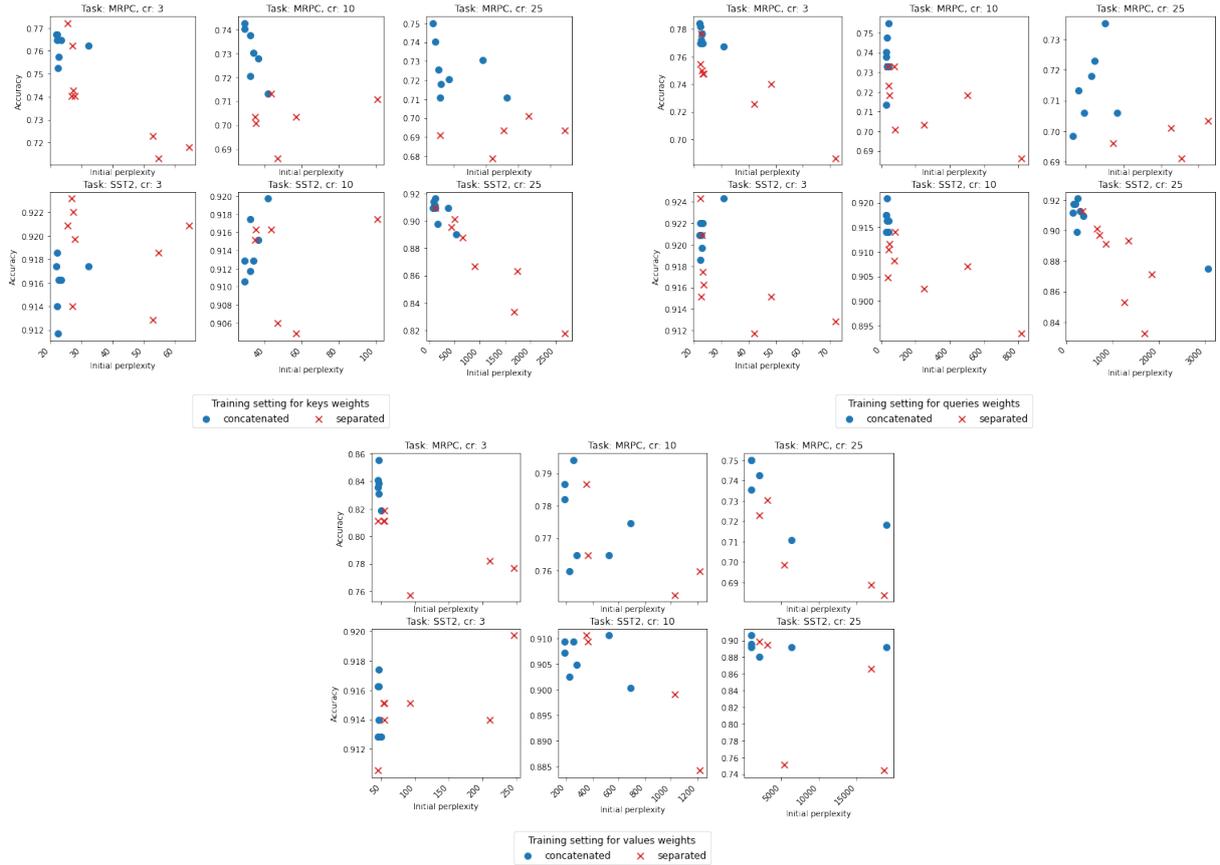


Figure 7: Initial perplexity and downstream tasks performance with separated and concatenated compression mode for output-dense weights matrices, queries weights matrices and values weights matrices in the BERT model.

ding matrix, self-attention matrices: keys, queries, values and final output-dense matrices. For simplicity of the experiment, we use the linear decoder in the autoencoder model. The loss and learning rate combinations for the models’ training are the same as in experiment in previous section. We conducted all experiments with two different seeds to obtain more reliable correlation information.

A.4 Linear and non-linear decoder module

In Figure 9 we present the initial perplexity and final downstream tasks performance achieved when using linear and non-linear decoder in the autoencoder model for key matrices compression. Unlike token embeddings for key matrices, the linear models outperform non-linear ones in most scenarios. Similar set of experiments on output-dense matrices also showed that linear models outperform non-linear ones.

Experimental setup We conduct experiments with the following settings: linear encoder/decoder and non-linear encoder/decoder. For non-linear encoder/decoder case, we examine architectures

with 1 and 2 hidden layers. As non-linear activation functions we use LeakyReLU (Maas et al., 2013) and Tanh. We investigate the loss configurations (Equation (2)) with 1:0, 1:1, 1:10 and 1:100 ratios of the l_2 norm loss to the cosine distance loss.

A.5 Preserving vector norm

In Figure 10 we show the potential benefits of preserving l_2 vector norm during compression of the output-dense matrices for two different downstream tasks .

Experimental setup We repeat the same set of experiments as in the previous section (Section 4.4), but each experiment is executed with and without the preserving vector norm option enabled.

A.6 Sensitivity

In Table 5 and Table 6 we report the positive impact of adding Fisher Information (with different coefficient transformations) for weight matrix compression for both autoencoder and SVD approaches.

Experimental setup We precompute the Fisher Information coefficients for token embeddings, self-

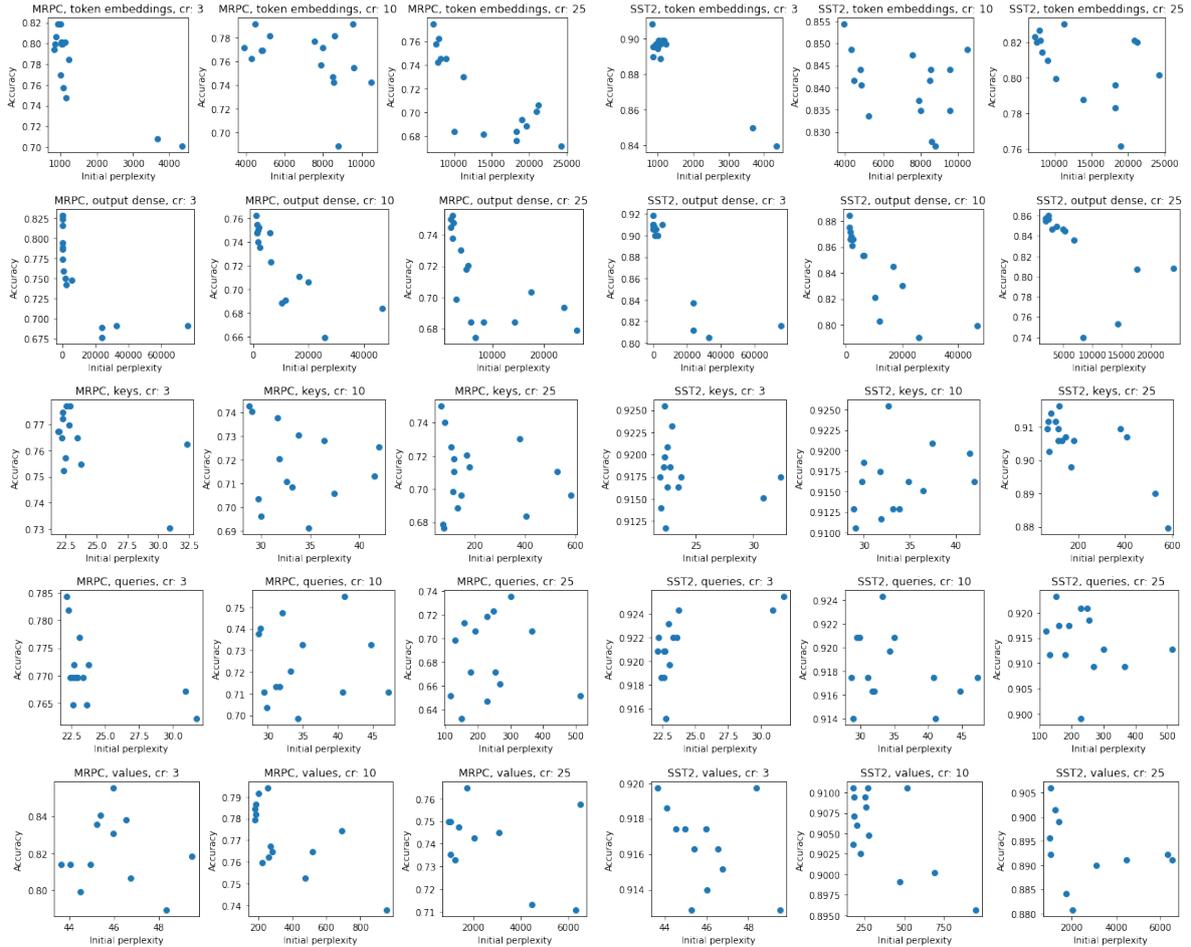


Figure 8: Initial perplexity and final accuracy on MRPC and SST2 downstream tasks for token embeddings, output-dense, keys, queries and values BERT_{BASE} matrices compression.

attentions keys and output-dense weight matrices. We apply them the best hyperparameter setting obtained from the previous experiments. We use various transformations for Fisher Information coefficients: exponential transformation x^a with $a = 0.1, 0.5, 0.9, 2.0$, logarithmic transformation $\log_e(x) + C$ with a C equals to minimum value so that all x are positive, logarithmic transformation $\log_e(x) + C + 10$, and raw Fisher Information coefficients without any transformation (*Vanilla*). For some transformations, we also add a batch sum normalization (+BN). We also report results without using Fisher Information (*No Fisher*).

A.7 Comparison to other offline compression methods

Kronecker Product offline compression Inspired by a promising results achieved by using a Kronecker product for training the transformer model from scratch (Panahi et al., 2021) we have attempted to produce a compression of the orig-

inal transformer matrices by using a Kronecker product of two matrices approximating the original matrix. We have trained these matrices using the gradient descent algorithm. Unfortunately, the results were unsatisfactory for each of the tested settings. For example, for concatenated key matrices and a compression ratio of 10 the perplexity for the Kronecker product was around 1500, while for the autoencoder perplexity below 50 is achieved in many different settings.

Tucker decomposition offline compression

Moreover, we also experimented with the Tucker decomposition (De Lathauwer et al., 2000) as an offline compression method. For token embeddings compression, we observed that the compressed language model starts having high perplexities even in low compression ratios. For instance, for CR=3, the model perplexity becomes almost 1500, while the autoencoder model can achieve perplexities below 40 for the same compression ratio. This finding is consistent with the observation of (Ren et al.,

CR	Method	Word embeddings			Output dense			Keys		
		Perplexity	SST-2 (Acc)	MRPC (F1/Acc)	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)
3	SVD	1842.38	89.68	83.71 / 75.49	99.98	91.74	88.77 / 84.07	22.04	92.32	85.30 / 77.45
	SVD+Fisher	20.20	91.86	88.62 / 83.82	71.05	91.74	88.74 / 83.82	20.67	91.74	86.22 / 78.92
	AE	118.41	91.97	88.53 / 84.31	70.69	91.74	86.17 / 78.92	22.35	92.43	84.48 / 76.23
	AE+Fisher	33.27	92.55	88.36 / 83.33	64.67	91.40	85.58 / 77.94	22.27	92.32	84.44 / 75.98
10	SVD	13196.30	83.37	82.85 / 74.02	989.18	89.56	83.92 / 75.49	30.75	91.97	81.54 / 73.04
	SVD+Fisher	65.17	87.73	85.01 / 77.70	723.61	90.14	84.76 / 76.47	41.49	91.74	81.00 / 72.06
	AE	712.98	88.07	85.87 / 80.88	1197.19	88.42	83.97 / 75.49	29.01	91.40	80.42 / 72.30
	AE+Fisher	250.59	89.33	87.27 / 81.62	1249.62	89.45	84.64 / 75.98	29.30	91.40	81.34 / 72.79
25	SVD	20178.74	77.64	82.33 / 71.81	1603.45	88.19	84.09 / 75.25	52.50	90.71	78.19 / 69.36
	SVD+Fisher	913.23	75.23	82.02 / 73.77	1205.34	87.27	83.88 / 74.75	80.14	91.17	74.36 / 65.69
	AE	4926.08	82.80	84.19 / 77.45	1462.41	85.61	82.02 / 72.06	69.24	91.74	78.78 / 69.36
	AE+Fisher	2728.41	83.83	84.35 / 77.70	1453.44	87.56	84.54 / 75.98	73.28	90.83	78.40 / 69.61

Table 5: The effect of adding the Fisher Information to the SVD-based and autoencoder-based (AE) compression. We report the $BERT_{BASE}$ upstream task perplexity and the downstream tasks final scores. Each autoencoder result represents a median from 3 runs with different seeds and each SVD score is a result of the best iteration from run with one seed. For autoencoder model compression we selected the Fisher Information transformation for each of the compressed modules based on the results from Table 6 ($x^{0.5} + BN$ for word embeddings; $x^{2.0} + BN$ for output-dense matrices; $\log_e(x) + C + 10$ for key matrices).

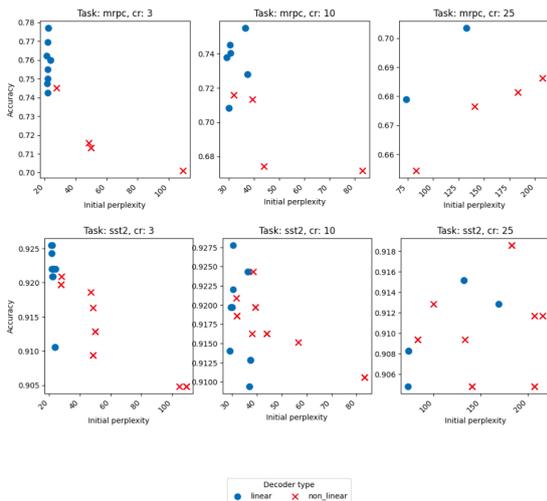


Figure 9: Initial perplexity and downstream tasks performance when using linear and non-linear decoder module in the autoencoder model for the compression of the keys matrix. Using a linear decoder generally appears to be a better choice for this matrix.

2022) that even randomly initializing the factorized tensors perform very close to the models initialized by e.g., tucker decomposition. Therefore, we also do not find Tucker decomposition an efficient method in the context of offline compression.

Pruning We also compare our proposed autoencoder framework with a pruning baseline as another offline compression baseline. The pruning algorithm here is based on PyTorch unstructured L1 pruning (Paszke et al., 2019). For this experiment,

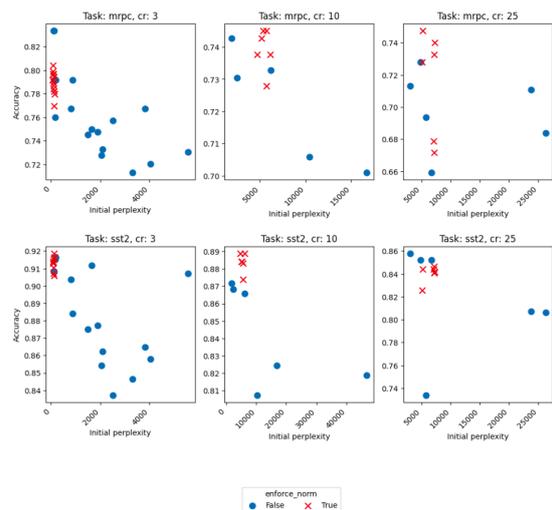


Figure 10: The effect of preserving l_2 vector norm on the perplexity and downstream tasks performance while compressing the output-dense matrices. We can see that enforcing norm for compression of this module generally improves the result.

we compress token embedding, keys, and output-dense matrices using either autoencoder or pruning approaches. The models are evaluated on four GLUE tasks as presented in Table 7. We can see that autoencoder-based compression outperforms pruning baseline in most studies settings, especially when higher compression ratios are studied.

CR=3	Word embeddings			Output-dense			Keys		
	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)
$x^{2.0} + \text{BN}$	30.88	92.32	87.97 / 82.84	64.67	91.40	85.58 / 77.94	22.44	92.20	83.94 / 75.25
$x^{0.9} + \text{BN}$	24.97	92.55	88.32 / 83.09	62.93	91.51	86.13 / 78.92	22.18	92.20	84.04 / 75.49
$x^{0.5} + \text{BN}$	33.27	92.55	88.36 / 83.33	65.07	91.40	85.81 / 78.19	22.32	91.97	83.89 / 75.25
$x^{0.1} + \text{BN}$	58.08	91.86	86.47 / 81.13	68.19	91.40	86.17 / 78.92	22.29	91.97	83.99 / 75.25
Vanilla+BN	24.68	92.55	87.25 / 82.11	62.60	91.63	86.08 / 78.43	22.39	92.20	83.70 / 74.51
Vanilla	564.37	87.96	87.32 / 82.35	66.91	90.77	85.99 / 78.68	22.73	92.55	83.62 / 74.75
$\log_e(x) + C + 10$	68.46	92.43	86.91 / 81.62	68.61	91.86	86.04 / 78.68	22.27	92.32	84.44 / 75.98
$\log_e(x) + C$	236.24	92.09	86.75 / 81.13	68.17	91.74	86.17 / 78.92	22.26	91.97	84.36 / 75.74
No Fisher	118.41	91.97	88.53 / 84.31	70.69	91.74	86.17 / 78.92	22.35	92.43	84.48 / 76.23

CR=10	Word embeddings			Output-dense			Keys		
	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)
$x^{2.0} + \text{BN}$	147.29	86.93	84.85 / 77.94	1249.62	89.45	84.64 / 75.98	30.78	91.40	79.73 / 70.10
$x^{0.9} + \text{BN}$	95.32	88.30	85.22 / 78.92	1251.53	88.19	84.04 / 75.25	30.47	91.28	79.46 / 70.10
$x^{0.5} + \text{BN}$	250.59	89.33	87.27 / 81.62	1195.41	88.53	83.81 / 75.00	29.81	91.40	80.60 / 71.57
$x^{0.1} + \text{BN}$	567.88	87.16	84.73 / 77.21	1148.47	88.42	83.68 / 75.25	29.26	91.17	80.74 / 72.06
Vanilla+BN	136.84	88.99	85.86 / 79.90	1251.65	88.65	83.79 / 75.00	30.53	91.40	78.85 / 69.36
Vanilla	15994.88	79.70	81.22 / 68.38	1547.14	87.16	83.20 / 74.26	31.65	91.63	77.82 / 68.14
$\log_e(x) + C + 10$	806.71	88.30	86.06 / 80.15	1145.25	88.42	83.71 / 75.00	29.30	91.40	81.34 / 72.79
$\log_e(x) + C$	854.19	88.07	86.25 / 80.39	1148.46	88.99	84.09 / 75.25	29.25	91.17	80.14 / 72.06
No Fisher	712.98	88.07	85.87 / 80.88	1197.19	88.42	83.97 / 75.49	29.01	91.40	80.42 / 72.30

CR=25	Word embeddings			Output-dense			Keys		
	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)	Perplexity	SST-2 (Acc)	MRPC (F1/Acc)
$x^{2.0} + \text{BN}$	2779.96	80.73	83.99 / 75.98	1453.44	87.56	84.54 / 75.98	77.04	91.06	77.95 / 68.63
$x^{0.9} + \text{BN}$	1983.19	83.49	82.08 / 73.04	1371.93	87.96	83.31 / 74.26	71.90	90.60	79.66 / 70.34
$x^{0.5} + \text{BN}$	2728.41	83.83	84.35 / 77.70	1438.66	87.73	83.20 / 74.26	74.69	90.71	79.73 / 70.59
$x^{0.1} + \text{BN}$	5502.34	82.45	84.12 / 76.96	1447.31	85.89	82.58 / 73.28	74.34	91.17	78.68 / 69.85
Vanilla+BN	2167.99	80.50	82.60 / 73.77	1384.83	87.10	83.87 / 75.49	71.37	90.71	78.97 / 69.85
Vanilla	14880.66	77.29	80.19 / 70.10	1748.90	86.35	82.26 / 73.28	66.53	91.17	76.92 / 67.65
$\log_e(x) + C + 10$	5426.84	82.45	83.90 / 76.47	1449.46	86.93	82.89 / 73.53	73.28	90.83	78.40 / 69.61
$\log_e(x) + C$	4359.66	82.91	83.89 / 76.47	1446.55	86.58	81.79 / 72.06	76.81	91.06	77.78 / 68.63
No Fisher	4926.08	82.80	84.19 / 77.45	1462.41	85.61	82.02 / 72.06	69.24	91.74	78.78 / 69.36

Table 6: Incorporating Fisher Information coefficients in the autoencoder-based compression process for different compression ratios (CR) on the SST2 and the MRPC tasks. The first column demonstrates the transformation(s) applied to Fisher information before being passed to autoencoder loss function (more details in Appendix A.6). We also report the perplexity of the compressed model on the upstream task. Each score is the median of experiment results with three different seeds.

CR	Architecture	MRPC (F1/Acc)	SST-2 (Acc)	RTE (Acc)	QNLI (Acc)
1	BERT _{BASE}	88.85/84.07	92.32	65.70	90.66
3	Pruning	86.25/80.15	90.37	56.47	88.68
	Our	85.25/77.77	90.25	62.45	88.68
10	Pruning	79.61/69.61	79.93	50.21	70.65
	Our	81.45/71.08	83.94	57.76	81.48
25	Pruning	80.86/69.36	75.11	51.97	60.57
	Our	81.6/71.32	80.05	55.23	72.96

Table 7: Comparison of BERT_{BASE} model compression using autoencoder (AE) and pruning approaches (compressing token embedding matrix, all key matrices and all output-dense matrices). For AE we present the median score from experiments with 3 different seeds.

A.8 Hyperparameter setting for the final experiment

This section presents the hyperparameter setting used for the Section 4.8 where multiple modules (token embedding, keys and output-dense) are compressed together. The autoencoder hyperparameters setting for this experiment can be found in Table 8.

A.9 Compression time

In Table 9 we report compression times for keys, output-denses and token embeddings matrices from BERT_{BASE} model using autoencoder and SVD compression approaches. For autoencoder we present times for both separated and concatenated matrices compression (as proposed in our paper) showing the advantage of using the latter approach.

We additionally performed experiments to compare the inference time of uncompressed BERT_{BASE} with the extreme case of CR=25 using our approach (AE) and the SVD baseline. Compressed modules are token embeddings, key, and output-dense modules. We observe that inference times are very similar, with a slight increase in inference time when our model is used. In particular, the evaluation times (in seconds) for BERT/SVD/AE were respectively 0.474/0.477/0.493 (for the MRPC dataset) and 0.987/0.997/1.034 (for the SST2 dataset). Moreover, fine-tuning the BERT_{BASE} model using compressed modules from autoencoder for the bigger datasets in GLUE, namely QNLI and QQP, takes at most 15% and 25% longer than the SVD baseline, respectively. It is also worth noting that when a linear autoencoder is incorporated, the inference time is the same as the SVD baseline.

Module	CR	Learning rate	Cosine:L ₂ coefficients	Decoder	Enforce norm	Fisher transformation
Token embeddings	3	$5 \cdot 10^{-4}$	10:1	Non-linear (1 hidden layer)	Yes	$x^{0.5} + BN$
	10	$5 \cdot 10^{-4}$	10:1	Non-linear (2 hidden layers)	Yes	$x^{0.5} + BN$
	25	10^{-3}	10:1	Non-linear (1 hidden layer)	Yes	$x^{0.5} + BN$
Output-denses	3	10^{-3}	1:10	Linear	Yes	$x^{2.0} + BN$
	10	10^{-4}	1:10	Linear	No	$x^{2.0} + BN$
	25	$5 \cdot 10^{-4}$	0:1	Linear	No	$x^{2.0} + BN$
Keys	3	$5 \cdot 10^{-4}$	0:1	Linear	Yes	$\log_e(x) + C + 10$
	10	$5 \cdot 10^{-4}$	1:1	Linear	No	$\log_e(x) + C + 10$
	25	10^{-3}	0:1	Linear	No	$\log_e(x) + C + 10$

Table 8: The best hyperparameters for the BERT_{BASE} model compression described in Section 4.8. In this experiment, token embedding matrix, all keys and all output-dense matrices are compressed using our proposed autoencoder-based framework.

Method	Mode	CR	Token embeddings	Keys	Output-denses
SVD	separated	3	~9.5min	(~5.5*12)min	(~6.0*12)min
SVD	separated	10	~8.0min	(~5.5*12)min	(~6.0*12)min
SVD	separated	25	~7.5min	(~5.5*12)min	(~6.0*12)min
AE	separated/concatenated	3	~7.7min	(~5.5*12)min/~6.1min	(~6.0*12)min/~6.5min
AE	separated/concatenated	10	~7.5min	(~5.5*12)min/~6.0min	(~6.0*12)min/~6.1min
AE	separated/concatenated	25	~7.5min	(~5.5*12)min/~5.9min	(~6.0*12)min/~6.1min

Table 9: Training time to retrieve compressed modules (key, output-dense, and token embeddings) of BERT_{BASE} model using autoencoder (AE) and SVD approach. For the AE, we provide training times for the separated and concatenated modes to demonstrate another benefit of using the concatenated version, given its much better training time.

PriMeSRL-Eval: A Practical Quality Metric for Semantic Role Labeling Systems Evaluation

Ishan Jindal^a, Alexandre Rademaker^a, Khoi-Nguyen Tran^a, Huaiyu Zhu^a,
Hiroshi Kanayama^a, Marina Danilevsky^a, Yunyao Li^{b*}

^aIBM Research, ^bApple

ishan.jindal@ibm.com, alexrad@br.ibm.com, kndtran@ibm.com,
huaiyu@us.ibm.com, hkana@jp.ibm.com, mdanile@us.ibm.com,
yunyaoli@apple.com

Abstract

Semantic role labeling (SRL) identifies predicate-argument structures in a sentence. This task is usually accomplished in four steps: predicate identification, predicate sense disambiguation, argument identification, and argument classification. Errors introduced at one step propagate to later steps. Unfortunately, the existing SRL evaluation scripts do not consider the full effect of this error propagation aspect. They either evaluate arguments independent of predicate sense (CoNLL09) or do not evaluate predicate sense at all (CoNLL05), yielding an inaccurate SRL model performance on the argument classification task. In this paper, we address key practical issues with existing evaluation scripts and propose a more strict SRL evaluation metric, *PriMeSRL*. We observe that by employing *PriMeSRL*, the quality evaluation of all SoTA SRL models drops significantly, and their relative rankings also change. We also show that *PriMeSRL* successfully penalizes actual failures in SoTA SRL models.

1 Introduction

Semantic Role Labeling (SRL) extracts predicate-argument structures from a sentence, where predicates represent relations (verbs, adjectives, or nouns) and arguments are the spans attached to the predicate demonstrating “who did what to whom, when, where, and how.” As one of the fundamental natural language processing (NLP) tasks, SRL has been shown to help a wide range of NLP downstream applications such as natural language inference (Zhang et al., 2020b; Liu et al., 2022), question answering (Maqsood et al., 2014; Yih et al., 2016; Zhang et al., 2020b; Dryjański et al., 2022), machine translation (Shi et al., 2016; Rapp, 2022), content moderation and verification (Calvo Figueras et al., 2022; Fharook et al., 2022), information extraction (Niklaus et al., 2018; Zhang

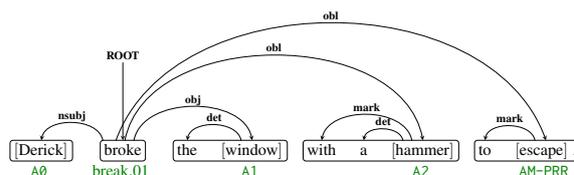


Figure 1: An SRL example with head-based semantic roles on top of Universal Dependencies annotation.

et al., 2020a). In all of these applications, the quality of the underlying SRL models has a significant impact on the downstream tasks. Despite this, few studies exist on how to properly evaluate the quality of SRL systems in practice.

Given a sentence, a typical SRL system obtains predicate-argument structure by following a series of four steps: 1) predicate identification; 2) predicate sense disambiguation; 3) argument identification; and 4) argument classification. The predicate senses and their argument labels are taken from inventories of frame definitions such as Proposition Bank (PropBank) (Palmer et al., 2005), FrameNet (Baker et al., 1998), and VerbNet (Schuler, 2005).

The accuracy of SRL extraction is affected by the correctness of each of these steps. Consider the example in Figure 1 using PropBank¹ annotations:

The SRL system must:

1. Identify the verb ‘break’ as a predicate
2. Disambiguate its particular sense as ‘break.01’,² which has four associated arguments: A0 (the breaker), A1 (thing broken), A2 (the instrument), A3 (the number of pieces), and A4 (from what A1 is broken away).³

¹In this paper we discuss SRL based on PropBank frames.

²<https://verbs.colorado.edu/propbank/framesets-english-aliases/break.html>

³Note that in PropBank each verb sense has a specific set

^b Work done while at IBM Research

3. Identify each argument as it occurs (‘Derick’, ‘the window’, etc.)
4. Classify the arguments (‘Derick’ : A0)

Finally, this example has one additional modifier: the AM-PRP (the purpose). Figure 1 illustrates the same analysis on top of the universal dependencies annotations where only the tokens’ head of phrases are annotated with the proper argument.

To obtain a completely correct predicate-argument structure both the predicate sense and all of its associated arguments need to be correctly extracted. Mistakes introduced at one step may propagate to later steps, leading to further errors.

For instance, in the above example, a wrong predicate sense ‘break.02’ (*break in or gain entry*) has not only a different meaning from ‘break.01’ (*break*) but also a different set of arguments. In many cases, even if an argument for a wrong predicate sense is labeled with the same numerical roles (A1, A2, etc), their meanings can be very different. Therefore, in general, the labels for argument roles should be considered to be incorrect when the predicate sense itself is incorrect. However, existing SRL evaluation metrics (e.g. (Hajič et al., 2009)) do not penalize argument labels in such cases.

The currently used evaluation metrics also do not evaluate *discontinuous arguments* accurately. Some arguments in the PropBank original corpora have discontinuous spans that all refer to the same argument. This can happen for a number of reasons such as in verb-particle constructions. In a dependency-based analysis, these arguments end up being attached to distinct syntactic heads (Surdeanu et al., 2008). Take as an example the sentence, “I know your answer will be that those people should be allowed to live where they please as long as they pay their full locational costs.” For the predicate “allow.01,” the A1 (action allowed) is the discontinuous span “those people” (A1) and “to live where they please as long as they pay their full locational costs” (C-A1). Existing evaluation metrics treat these as two independent labels.

A similar problem exists for the evaluation of reference arguments (R-X). For example, in the sentence “This is exactly a road that leads nowhere”, for the predicate “lead.01”, the A0 “road” is referenced by C-A0 “that”. If A0 is not correctly identified, the reference C-A0 is meaningless.

of underspecified roles, given by numbers: A0, A1, A2, and so on. This is because of the well-known difficulty of defining a universal set of thematic roles (Jurafsky and Martin, 2021).

In this paper, we conduct a systematic analysis of the pros and cons of different evaluation metrics for SRL, including:

- Proper evaluation of predicate sense disambiguation task;
- Argument label evaluation in conjunction with predicate sense;
- Proper evaluation for discontinuous arguments and reference arguments; and
- Unified evaluation of argument head and span.

We then propose a new metric for evaluating SRL systems in a more accurate and intuitive manner in Section 3, and compare it with currently used methods in Section 4. PriMeSRL is available at <https://github.com/UniversalPropositions/PriMeSRL-Eval>.

2 Existing Evaluation Metrics for SRL

Most of the existing evaluation metrics came from shared tasks for the development of systems capable of extracting predicates and arguments from natural language sentences. In this section, we summarize the approaches to SRL evaluation in the shared tasks from SemEval and CoNLL.

2.1 Senseval and SemEval

SemEval (Semantic Evaluation) is a series of evaluations of computational semantic analysis systems that evolved from the Senseval (word sense evaluation) series.

SENSEVAL-3 (Litkowski, 2004) addressed the task of automatic labeling of semantic roles and was designed to encourage research into and use of the FrameNet dataset. The system would receive as input a target word and its frame, and was required to identify and label the frame elements (arguments). The evaluation metric counted the number of arguments correctly identified (complete match of span) and labeled, but did not penalize those spuriously identified. An overlap score was generated as the average of proportion of partial matches.

SemEval-2007 contained three tasks that evaluate SRL. Task 17 and 18 identified arguments for given predicates using two different role label sets: PropBank and VerbNet (Pradhan et al., 2007). They used the `srl-eval.pl` script from the CoNLL-2005 scoring package (Carreras and Màrquez, 2005) (see below). Task 19 consists of

recognizing words and phrases that evoke semantic frames from FrameNet and their semantic dependents, which are usually, but not always, their syntactic dependents. The evaluation measured precision and recall for frames and frame elements, with partial credit for incorrect but closely related frames. Two types of evaluation were carried out. The first is the label matching evaluation. The participant’s labeled data were compared directly with the gold standard labeled using the same evaluation procedure used in the previous SRL tasks at SemEval. The second is the semantic dependency evaluation, in which both the gold standard and the submitted data were first converted to semantic dependency graphs and compared.

SemEval-2012 (Kordjamshidi et al., 2012) and **SemEval-2013** (Kolomiyets et al., 2013) introduced the ‘Spatial Role Labeling’ task, but this is somewhat different from the standard SRL task and will not be discussed in this paper. Since **SemEval-2014** (Marelli et al., 2014), a deeper semantic representation of sentences in a single graph-based structure via semantic parsing has superseded the previous ‘shallow’ SRL tasks.

2.2 CoNLL

The **CoNLL-2004** shared task (Carreras and Màrquez, 2004) was based on the PropBank corpus, comprising six sections of the Wall Street Journal part of the Penn Treebank (Kingsbury and Palmer, 2002) enriched with predicate–argument structures. The task was to identify and label the arguments of each marked verb. The precision, recall, and F1 of arguments were evaluated using the `sr1-eval.pl` program. For an argument to be correctly recognized, the words spanning the argument as well as its semantic role have to be correct. The verb argument is the lexicalization of the predicate of the proposition. Most of the time, the verb corresponds to the target verb of the proposition, which is provided as input, and only in a few cases the verb participant spans more words than the target verb. This situation makes the verb easy to identify and, since there is one verb with each proposition, evaluating its recognition overestimates the overall performance of a system. For this reason, the verb argument is excluded from evaluation. The shared task proceedings do not detail how non-continuous arguments are evaluated. In **CoNLL-2005** (Carreras and Màrquez, 2005) a system had to recognize and label the arguments of each target

verb. The evaluation method remained the same as CoNLL-2004, using the same evaluation code.

The **CoNLL 2008** shared task (Surdeanu et al., 2008) was dedicated to the joint parsing of syntactic and semantic dependencies. The shared task was divided into three subtasks: (i) parsing of syntactic dependencies, (ii) identification and disambiguation of semantic predicates, and (iii) identification of arguments and assignment of semantic roles for each predicate. SRL was performed and evaluated using a dependency-based representation for both syntactic and semantic dependencies.

The official evaluation measures consist of three different scores: (i) syntactic dependencies are scored using the labeled attachment score (LAS), (ii) semantic dependencies are evaluated using a labeled F1 score, and (iii) the overall task is scored with a macro average of the two previous scores. The semantic propositions are evaluated by converting them to semantic dependencies, i.e., a semantic dependency from every predicate to all its individual arguments were created. These dependencies are labeled with the labels of the corresponding arguments. Additionally, a semantic dependency from each predicate to a virtual ROOT node was created. The latter dependencies are labeled with the predicate senses. This approach guarantees that the semantic dependency structure conceptually forms a single-rooted, connected (not necessarily acyclic) graph. More importantly, this scoring strategy implies that if a system assigns the incorrect predicate sense, it still receives some points for the arguments correctly assigned. Several additional evaluation measures were applied to further analyze the performance of the participating systems. The *Exact Match* reports the percentage of sentences that are completely correct, i.e., all the generated syntactic dependencies are correct and all the semantic propositions are present and correct. The *Perfect Proposition F1* score entire semantic frames or propositions. The ratio between labeled F1 score for semantic dependencies and the LAS for syntactic dependencies.

As in CoNLL-2008, the CoNLL-2009 shared task (Hajič et al., 2009) combined syntactic dependency parsing and the task of identifying and labeling semantic arguments of verbs or nouns for six more languages in addition to the original English from CoNLL-2008. Predicate disambiguation was still part of the task, whereas the identification of argument-bearing words was not. This deci-

sion was made to compensate for the significant differences between languages and between the annotation schemes used. The evaluation of SRL was done similar to CoNLL-2008.

3 The Proposed Approach

We propose PriMeSRL, a new metric for evaluating SRL systems, based on the following high-level rules that aim to overcome the drawbacks in existing metrics:

1. Predicate senses are considered correct only when the full predicate.sense is correct, not just the sense number. (Table 1)
2. Core arguments are considered correct only when the predicate sense has been correctly identified. (Table 1)
3. An argument of the form C-X is considered together with its associated X argument to cover the full region of the argument. (Table 2)
4. An argument of the form R-X is considered as a reference, so its correctness depends on the correctness of the referenced X. (Table 3)

3.1 Predicate sense disambiguation evaluation

Current evaluation metrics either do not evaluate the predicate sense disambiguation task (e.g. CoNLL05), or evaluate only the sense number of the predicate (e.g. CoNLL09). In this section, we only contrast with the CoNLL09 evaluation script.

To begin with, what is the predicate sense number? Similar to word senses in Wordnet (Fellbaum, 2010), the predicate senses in PropBank inside a predicate frame file are generally ordered from most to least frequently used, with the most common sense numbered 01 (Pradhan et al., 2022). The sense numbers (01, 02, 03, ...) do not have any associated semantic meaning, and merely convey that one particular meaning of the predicate is more common than another.

Therefore, predicting and evaluating only the sense number is not sensible. It can be a reasonable goal to a certain extent, such as when predicate location is given and the task is only to disambiguate the sense of the predicate (as proposed in the CoNLL09 shared task.) But the consequence of this approach is that a sense number classifier could predict a sense number that does not even exist in the associated frame file. Of course, an unknown sense number for a predicate does not

text = Yesterday, John bought a car.

ID	FORM	FLAG	PRED SENSE	Predicate-argument prediction			
				Gold	P1	P2	P3
1	Yesterday	-	-	TMP	TMP	TMP	TMP
2	,	-	-	-	-	-	-
3	John	-	-	A0	A0	A0	A0
4	bought	Y	buy.01	buy.01	<i>buy_out.03</i>	<i>buy.05</i>	<i>sell.01</i>
5	a	-	-	-	-	-	-
6	car	-	-	A1	A1	A1	A1
7	.	-	-	-	-	-	-

Predicate Evaluation	R	CoNLL05	do not evaluate			
		CoNLL09	1/1	0/1	0/1	1/1
		PriMeSRL	1/1	0/1	0/1	0/1
	P	CoNLL05	do not evaluate			
		CoNLL09	1/1	0/1	0/1	1/1
		PriMeSRL	1/1	0/1	0/1	0/1
Argument Evaluation	R	CoNLL05	3/3	3/3	3/3	0/3
		CoNLL09	3/3	3/3	3/3	3/3
		PriMeSRL	3/3	1/3	1/3	1/3
	P	CoNLL05	3/3	3/3	3/3	0/3
		CoNLL09	3/3	3/3	3/3	3/3
		PriMeSRL	3/3	1/3	1/3	1/3

Table 1: Comparing evaluation metrics on 4 examples, showing the effect of wrong predicate sense on argument label evaluation. *RED-italic* shows a wrong prediction by a hypothetical model. **GREEN** cell highlights where PriMeSRL differs from existing metrics.

have a semantic meaning, making it unsuitable for practical use cases. For a practical end-to-end SRL system, the sense number classifier should predict both the predicate location and associated sense number together (i.e. predicate.sense) so that the contextual meaning of the predicate is correctly captured, as performed in (Roth and Lapata, 2016; Li et al., 2018; Conia et al., 2021; Conia and Navigli, 2022).

Evaluating the predicate sense disambiguation task of such practical systems using existing evaluation metrics is not optimal. Consider the example in Figure 1, where the gold predicate.sense is ‘break.01’ (*break, cause to not be whole*⁴). Suppose an SRL system predicts⁵ the predicate.sense label ‘pull.01’ (*causing motion*⁶). The existing CoNLL09 evaluation script will give a fully correct score because the predicted sense number **01** exactly matches the gold sense number, despite the different semantic meanings. In contrast, PriMeSRL evaluates the predicate.sense as a whole instead of only the sense number.

⁴<https://verbs.colorado.edu/propbank/framesets-english-aliases/break.html>

⁵See Table 6 for several examples of such mistakes actually made by a SoTA SRL system on CoNLL09 data.

⁶<https://verbs.colorado.edu/propbank/framesets-english-aliases/pull.html>

text = Many confusing questions have been taxing my mind for years about Egypt and its people.

ID	FORM	F	PRED SENSE	Gold	P1	P2	P3	P4	P5	P6	P7
1	Many	-	-	-	-	-	-	-	-	-	-
2	confusing	-	-	-	-	-	-	-	-	-	-
3	questions	-	-	A0	A0	A1	A1	C-A0	A1	C-A0	C-A0
4	have	Y	-	-	-	-	-	-	-	-	-
5	been	Y	-	-	-	-	-	-	-	-	-
6	taxing	Y	tax.01	-	-	-	-	-	-	-	-
7	my	-	-	-	-	-	-	-	-	-	-
8	mind	-	-	A2							
9	for	-	-	-	-	-	-	-	-	-	-
10	years	-	-	TMP							
11	about	-	-	-	-	-	-	-	-	-	-
12	Egypt	-	-	C-A0	C-A1	C-A0	C-A1	A0	C-A2	C-A0	-
13	and	-	-	-	-	-	-	-	-	-	-
14	its	-	-	-	-	-	-	-	-	-	-
15	people	-	-	-	-	-	-	-	-	-	-
16	.	-	-	-	-	-	-	-	-	-	-

Argument HEAD Evaluation	R	Conll09	4/4	3/4	3/4	2/4	2/4	2/4	3/4	2/4
		PriMeSRL	3/3	2/3	2/3	2/3	3/3	1/3	3/3	2/3
Argument SPAN Evaluation	P	Conll09	4/4	3/4	3/4	2/4	2/4	2/4	3/4	2/3
		PriMeSRL	3/3	2/4	2/4	2/3	3/3	1/3	3/3	2/3

Argument HEAD Evaluation	R	Conll05	3/3	2/3	2/3	2/3	2/3	1/3	2/3	2/3
		PriMeSRL	3/3	2/3	2/3	2/3	3/3	1/3	3/3	2/3
Argument SPAN Evaluation	P	Conll05	3/3	2/4	2/4	2/3	2/4	1/3	2/3	2/3
		PriMeSRL	3/3	2/4	2/4	2/3	3/3	1/3	3/3	2/3

Table 2: Comparing evaluation metrics on 7 examples, showing the effect of C-X labels. *RED-italic* and **GREEN** cell are used in the same manner as Table 1.

3.2 Argument evaluation with incorrect predicate sense

Current metrics evaluate the arguments independent of the predicate sense. That is, they evaluate arguments as if the predicate location and sense are both correct. In practice, the predicates predicted by models can of course be wrong, and in such cases, the corresponding core argument labels (A0, A1, etc.) generally do not refer to the correct argument - even if the label itself matches the gold label - and should be penalized. Contextual arguments, or adjunct arguments, such as AM-LOC, AM-TMP, etc, remain the same across different predicates and do not need to be penalized for predicate errors.

Table 1 illustrates the difference between PriMeSRL and existing evaluation metrics, CoNLL09 and CoNLL05. For predicate sense evaluation, PriMeSRL is often equal to CoNLL09 (CoNLL05 does not measure this aspect.) PriMeSRL explicitly penalizes the cases where the lemma is wrongly identified (Example P3): The CoNLL09 script considers the label as correct as long as the “predicate sense number” is correct. It is unlikely for a model to predict "sell.01" in P3 for the gold predicate "buy.01". We choose this example to smoothly motivate the need for more strict evaluation metrics for SRL. In fact, SoTA SRL systems made this type of error. Table 6 provides some incorrect model predictions from a SoTA SRL model, where a model confuses "overheat" with "soothe" as an example.

text = This is exactly a road that leads nowhere.

ID	FORM	F	PRED SENSE	Gold	P1	P2	P3	P4	P5	P6
1	This	-	-	-	-	-	-	-	-	-
2	is	-	-	-	-	-	-	-	-	-
3	exactly	-	-	-	-	-	-	-	-	-
4	a	-	-	-	-	-	-	A0	-	-
5	road	-	-	A0	A1	A0	A1	-	R-A0	R-A0
6	that	-	-	R-A0	R-A0	R-A1	R-A1	R-A0	R-A0	A0
7	leads	Y	lead.01	-	-	-	-	-	-	-
8	nowhere	-	-	A4						
9	.	-	-	-	-	-	-	-	-	-

Argument HEAD Evaluation	R	Conll09	3/3	2/3	2/3	1/3	2/3	2/3	1/3
		PriMeSRL	3/3	1/3	2/3	1/3	1/3	1/3	1/3
Argument SPAN Evaluation	P	Conll09	3/3	2/3	2/3	1/3	2/3	2/3	1/3
		PriMeSRL	3/3	1/3	2/3	1/3	1/3	1/3	1/3

Argument HEAD Evaluation	R	Conll05	3/3	2/3	2/3	1/3	2/3	1/3	1/3
		PriMeSRL	3/3	1/3	2/3	1/3	1/3	1/3	1/3
Argument SPAN Evaluation	P	Conll05	3/3	2/3	2/3	1/3	2/3	1/3	1/3
		PriMeSRL	3/3	1/3	2/3	1/3	1/3	1/3	1/3

Table 3: Comparing evaluation metrics on 6 examples, showing the effect of R-X labels. *RED-italic* and **GREEN** cell are used in the same manner as Table 1.

For argument evaluation, both CoNLL09 head evaluation and CoNLL05 span evaluation wrongly mark all the arguments in examples P1, P2, and P3 as correct, despite the predicate sense being wrong. This is corrected by PriMeSRL.

3.3 Evaluation of C-X arguments

An argument label with prefix C- is used in situations where an argument consists of multiple non-adjacent parts (Surdeanu et al., 2008). If conceptually the whole argument should be labeled X, then operationally one part will get label X and the other parts get label C-X. The existing evaluation metrics treat all these labels as independent, which is incorrect as it increases the weight of these arguments and assigns partial credit when an exact match is required. We now describe PriMeSRL for span-based and head-based evaluations.

Span-based evaluation: For an argument split into multipart spans with labels X and C-X, the complete span can be represented by the set of all tokens identified by these labels. The full set of tokens produced by the model should be compared to the set in the gold data, and a single credit should be assigned if these sets are equal.

Head-based evaluation: An argument with X and C-X parts has these as separate heads. A model prediction is considered correct if and only if all heads for this argument are correct, in which case it is given one whole credit. This evaluation does not distinguish between X and C-X and will penalize an argument if it has extra or missing parts.

Table 2 compares PriMeSRL with CoNLL05

Model	Evaluation script	In-domain					Out-of-domain				
		PSD		Argument Classification			PSD		Argument Classification		
		F1	P	R	F1	(r)	F1	P	R	F1	(r)
(Conia et al., 2021)	CoNLL09	96.9	89.5	89.5	89.5	(3)	87.8	82.0	81.9	81.9	(3)
	PriMeSRL	95.5(↓1.4)	86.6	86.6	86.6(↓2.9)	(2)	80.9(↓6.9)	72.4	72.6	72.5(↓9.4)	(4)
(Biloshmi et al., 2021) _{nested}	CoNLL09	97.1	89.3	81.9	85.4	(4)	89.7	82.8	75.7	79.1	(4)
	PriMeSRL	96.4(↓0.7)	86.8	79.8	83.1(↓2.3)	(4)	86.7(↓3.0)	75.7	69.9	72.7(↓6.4)	(3)
(Biloshmi et al., 2021) _{flat}	CoNLL09	97.4	90.9	89.6	90.2	(1)	90.1	83.9	82.1	83.0	(2)
	PriMeSRL	96.9(↓0.5)	88.6	87.4	88.0 (↓2.2)	(1)	87.8(↓2.3)	77.6	76.3	76.9 (↓6.1)	(1)
(Jindal et al., 2022)	CoNLL09	96.8	89.9	89.3	89.6	(2)	89.8	82.9	83.1	83.02	(1)
	PriMeSRL	95.5(↓1.3)	86.8	86.3	86.55(↓3.0)	(3)	83.4(↓6.4)	73.9	74.3	74.1(↓8.9)	(2)

Table 4: Comparison of SoTA SRL models with PriMeSRL and CoNLL09 evaluation metrics on CoNLL09 dataset. (r) denotes the ranking of SRL models corresponding to the evaluation metric. **BOLD** shows the best model with CoNLL09 evaluation script and **BOLD** shows the best SRL model with PriMeSRL.

Dataset	Args	Train	Dev	Test	ood
CoNLL09	C-X	0.77	1.05	0.88	1.15
	R-X	1.98	2.03	2.07	2.24
CoNLL05	C-X	1.22	1.24	1.71	0.91
	R-X	3.26	3.36	3.38	2.91

Table 5: Representation of C-X and R-X arguments in each split of different SRL datasets.

and CoNLL09 on seven examples. For span evaluation, the variances among labels with and without C- do not penalize the result, as long as the whole span is correct. That is our proposal for counting continuation arguments is the same as the CoNLL05 evaluation script, which provides one full credit if all heads for continuation arguments are identified and labeled correctly. We only differ that we do not distinguish between A0 and C-A0 labels. In this manner, we are not as strict as the CoNLL05 script. For head evaluation, note that the denominators reflect the number of arguments rather than the number of split parts, and numerators count correct whole arguments.

3.4 Evaluation of R-X arguments

An argument label with prefix R- indicates a reference argument; thus, R-X is a reference to the argument X. For R-X to be correct, X must also be correct, but apart from this requirement, PriMeSRL treats them as separate arguments.

Table 3 compares evaluating R-X arguments using PriMeSRL with the metric used in CoNLL09 on 6 examples P1 through P6. For P1 in Table 3 (Head Evaluation), CoNLL09 gives credit for cor-

rectly identified R-A0 for which no/incorrect A0 is predicted, which is meaningless. The same is true for the Span evaluation script CoNLL05. However, we do not penalize the correctly labeled main argument for incorrect R-X.

4 Comparisons with Existing Metrics

In this section, we discuss the effectiveness of existing SRL evaluation metrics and demonstrate how PriMeSRL differs in various use cases, using SoTA neural SRL models as test models.

4.1 General settings

For simplicity of comparison with existing results, we assume the gold predicate location is given for all the experiments following Shi and Lin (2019); Jindal et al. (2020); Conia and Navigli (2022). However, PriMeSRL is able to handle missing or spurious predicates. We use Conia et al. (2021); Biloshmi et al. (2021); Jindal et al. (2022) as SoTA SRL models.

4.2 Datasets

We show the impact of evaluating with PriMeSRL on the CoNLL09 and CoNLL05 datasets. Table 5 shows the percentage of C-X and R-X arguments in each split of the different datasets. Note that these arguments make up < 3% of the total arguments; 5.09% total of the arguments in CoNLL05 test, and 2.95% in CoNLL09 test. Therefore, we expect to observe an F1 drop of at most about 3 and 5 points on the argument classification subtask due to mishandling C-X and R-X arguments for CoNLL09 and CoNLL05 datasets, respectively.

Id	Sentence	Gold	Predicted
1	He was able, now, to sit for hours in a chair in the living room and stare out at the bleak yard without moving.	stare.01	look.01
2	She greeted her husband’s colleagues with smiling politeness , offering nothing.	politeness.01	minimalism.01
3	It was a Negro section of peeling row houses, store-front churches and ragged children.	peel.01	peer.01
4	He was calm, drugged , and lazy.	drug.01	dropper.01
5	The walk and his fears had served to overheat him and his sweaty armpits cooled at the touch of the night air.	overheat.01	soothe.01
6	He did not resent their supervision or Virginia’s sometimes tiring sympathy.	tire.01	hiring.01

Table 6: Conia et al. (2021) model predictions on examples from CoNLL09 OOD set. All of these predicate senses are marked correct by the CoNLL09 evaluation script. PriMeSRL correctly penalizes all of these senses.

Model	Evaluation script	In-domain					Out-of-domain				
		PSD		Argument Classification			PSD		Argument Classification		
		F1	P	R	F1	(r)	F1	P	R	F1	(r)
(Zhang et al., 2021) _{crf}	CoNLL05	100	86.5	88.3	87.4	(3)	100	79.0	81.1	80.0	(3)
	PriMeSRL	100	86.1	87.8	87.03(↓0.4)	(2)	100	78.7	80.8	79.7(↓0.3)	(2)
(Zhang et al., 2021) _{crf2o}	CoNLL05	100	86.9	88.6	87.7	(2)	100	78.9	81.2	80.03	(2)
	PriMeSRL	100	86.5	88.1	87.3 (↓0.4)	(1)	100	78.5	80.8	79.6(↓0.4)	(3)
(Jindal et al., 2022)	CoNLL05	100	87.4	88.0	87.74	(1)	100	80.4	81.4	80.9	(1)
	PriMeSRL	100	86.8	87.1	87.0(↓0.7)	(3)	100	79.7	80.5	80.1 (↓0.8)	(1)

Table 7: Comparison of SoTA SRL models with PriMeSRL and CoNLL05 evaluation metrics on CoNLL05 dataset. (r) denotes the ranking of SRL models corresponding to the evaluation metric. **BOLD** shows the best model with CoNLL05 evaluation script and **BOLD** shows the best SRL model with PriMeSRL.

4.3 Evaluation

4.3.1 Predicate sense disambiguation

The PSD column in Table 4 compares the impact of PriMeSRL w.r.t. the existing evaluation script on the EN subset of the CoNLL09 dataset using SoTA SRL models. We observe a consistent quality drop in predicate sense disambiguation (PSD) both for in-domain and out-of-domain (OOD) sets. Surprisingly, we observe a significant quality drop on the OOD set of an average of ~ 5 F1 points for all the SRL models, which significantly lowers the SoTA performance on the OOD set. This shows that existing SRL models still have a lot of room for improvement.

Continuing the PSD analysis, Table 6 shows example instances from the CoNLL09 dataset that have correct sense numbers (01) but wrong predicate.sense - yet all of which are marked correct by the CoNLL09 evaluation script. For example, the first row shows how the difference between ‘stare.01’ (*looking intently*⁷) and “look.01” (*causal*

*look*⁸) is ignored. While these two at least share the same underlying meaning (*look*), in row 5 the model’s prediction of ‘soothe.01’ means the opposite of the gold label ‘overheat.01’ (once again, the existing CoNLL09 evaluation script marks this as correct.) Clearly, predicate sense should be evaluated by including the actual value predicate.sense instead of only relying on the sense number.

4.3.2 Argument head evaluation

Argument classification column in Table 4 compares the impact of PriMeSRL w.r.t the existing evaluation script on the EN subset of the CoNLL09 dataset using SoTA SRL models. We observe a quality drop in the argument classification task both for in-domain and OOD sets, with a significant quality drop of an average of ~ 8 F1 points on the OOD set. This drop in the argument classification task is expected because part of this error is propagated from the predicate sense disambiguation task which itself is significant. It is interesting to note that, although the major contribution of argument classification drop is due to error propagation from

⁷<https://verbs.colorado.edu/propbank/framesets-english-aliases/stare.html>

⁸<https://verbs.colorado.edu/propbank/framesets-english-aliases/look.html>

Input Sentence	SRL			Downstream Application		
	SRL model prediction	Existing eval score	PriMeSRL score	Application prompt	Prediction	Expected
[S1] XYZ company bought \$2.4 billion in Fannie Mae bonds.	[XYZ company]A0 [bought] sell.01 [\$2.4 billion in Fannie Mae bonds]A1	3/3	0/3	<u>QA</u> Who bought Fannie Mae bonds?	None	XYZ company
	[XYZ company]A0 [bought] buy_out.03 [\$2.4 billion in Fannie Mae bonds]A1	2/3	0/3	<u>QA</u> Who bought Fannie Mae bonds completely?	XYZ company	None
[S2] XYZ company bought out \$2.4 billion in Fannie Mae bonds.	[XYZ company]A0 [bought]buy_out.03 out [\$2.4 billion in Fannie Mae bonds]A1	3/3	3/3	<u>NLI</u> Does S2 entails S3?	Yes	No
[S3] XYZ company bought \$2.4 billion in Fannie Mae bonds.	[XYZ company]A0 [bought] buy_out.03 [\$2.4 billion in Fannie Mae bonds]A1	2/3	0/3			

Table 8: Example illustrations of the how impact of SRL errors on downstream applications is captured by the new evaluation method, where **Red color** represents the wrong prediction by an SRL model, leading to incorrect predictions by a downstream application (QA: Question Answering; NLI: Natural Language Inference.)

the earlier stage, there is also a consistent drop due to penalizing correct arguments with wrong predicate sense, of ~ 1.5 and ~ 3 F1 points for in-domain and OOD sets, respectively.

Since the performance drop is not uniform, we observe a change in the relative ranking of the SRL models. As an example, the CoNLL09 evaluation script scores the SRL models [Blloshmi et al. \(2021\)](#) and [Jindal et al. \(2022\)](#) similarly (83.0 F1) on OOD set whereas PriMeSRL clearly shows a difference in performance. Further, PriMeSRL makes clear that the quality of existing SRL systems is not as high as previously thought, especially on OOD data.

4.3.3 Argument span evaluation

Similar to argument head evaluation, we compare the impact of PriMeSRL w.r.t to the existing evaluation script on the SRL span dataset (CoNLL05 dataset) using SoTA SRL models in Table 7. Since CoNLL05 does not evaluate predicate sense, we do not observe the impact of incorrect PSD on argument classification. Therefore, the only drop of argument classification is due to incorrect handling of C-X and R-X arguments. Although Table 5 shows that the total number of C-X and R-X in the CoNLL05 dataset is $\sim 5\%$ of the total number of arguments, we only observe a slight drop in quality evaluation ($< 1\%$) with PriMeSRL. This is because

on argument span evaluation, PriMeSRL is similar to CoNLL05 (except in a few cases as described in the last row-block of Tables 2 and 3.) As in the comparison with the CoNLL09 dataset, we again observe a change in the relative ranking of the SRL models.

4.4 Discussion

The existing evaluation metrics for SRL are disconnected from the actual practical performance of the SRL models. This makes it difficult to choose the best quality SRL model for the required downstream application. Current evaluation metrics do not pay sufficient attention to the error propagation aspect of the four-staged SRL task; instead, they evaluate the steps independently and linearly combine them to compute the overall SRL system score. However, the analyses in Tables 4 and 7 clearly show that the linear combination of the independent performance of individual steps is not equivalent to the true overall quality.

This does not negate the usefulness of the existing evaluation metrics. Indeed, these metrics provide an evaluation of each individual step, serving as an important guide for improving the quality of individual steps and hence the overall quality of the SRL system. However, whenever a real-world NLP system utilizes an SRL system as one

of its components, it is important to understand the quality of semantic roles in relation with, and conditional on their predicate sense disambiguation. Table 8 illustrates the impact of such SRL errors on two downstream applications (question answering and natural language inference). Existing evaluation scripts overlook such SRL errors and treat them as correct, despite the fact that the predicted predicate-argument structure is meaningless and leads to incorrect outputs for the downstream application.

5 Conclusion

In this paper, we highlighted key issues with existing SRL evaluation metrics and showed that the proposed evaluation metric, PriMeSRL, scores SoTA SRL models in a more accurate and intuitive manner. By releasing our evaluation code, we plan to promote these metrics in the community in order to improve the evaluation quality for SRL systems that contribute to downstream applications.

Limitations

We have shown the impact of our proposed new evaluation metrics in the current SoTA SRL models ranking. To further validate the impact of this work, we plan to conduct an in-depth study on how downstream applications' performance relates to the evaluation metrics in future work.

We acknowledge that the problems we have pointed out for previous evaluation metrics are not bugs, but rather design decisions given the timing of the shared tasks and the limitations on datasets and methods. Consider, for instance, that a unified syntactic dependency annotation schema like Universal Dependencies (Nivre et al., 2016) was unavailable before October 2014. Given that, in this paper, we didn't present a deep discussion on the impact of UD compared to previously used syntactic dependencies schemas.

Acknowledgements

We would like to thank our anonymous reviewers for their constructive comments and feedback. Further, the authors acknowledge the IBM Research Cognitive Computing Cluster service for providing resources that have contributed to the research results reported within this paper.

References

- Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.
- Rexhina Blloshmi, Simone Conia, Rocco Tripodi, and Roberto Navigli. 2021. Generating senses and roles: An end-to-end model for dependency-and span-based semantic role labeling. In *IJCAI*, pages 3786–3793.
- Blanca Calvo Figueras, Montse Oller, and Rodrigo Agerri. 2022. A semantics-aware approach to automated claim verification. In *Proceedings of the Fifth Fact Extraction and VERification Workshop (FEVER)*, pages 37–48, Dublin, Ireland. Association for Computational Linguistics.
- Xavier Carreras and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 89–97, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.
- Simone Conia, Andrea Bacciu, and Roberto Navigli. 2021. Unifying cross-lingual semantic role labeling with heterogeneous linguistic resources. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–351, Online. Association for Computational Linguistics.
- Simone Conia and Roberto Navigli. 2022. Probing for predicate argument structures in pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4622–4632.
- Tomasz Dryjański, Monika Zaleska, Bartek Kuźma, Artur Błażejowski, Zuzanna Bordzicka, Paweł Bujnowski, Klaudia Firlag, Christian Goltz, Maciej Grabowski, Jakub Jończyk, Grzegorz Kłosiński, Bartłomiej Paziewski, Natalia Paszkiewicz, Jarosław Piersa, and Piotr Andruszkiewicz. 2022. Samsung research Poland (SRPOL) at SemEval-2022 task 9: Hybrid question answering using semantic roles. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1263–1273, Seattle, United States. Association for Computational Linguistics.
- Christiane Fellbaum. 2010. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

- Shaik Fharook, Syed Sufyan Ahmed, Gurram Rithika, Sumith Sai Budde, Sunil Saumya, and Shankar Bilaradar. 2022. *Are you a hero or a villain? a semantic role labelling approach for detecting harmful memes*. In *Proceedings of the Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations*, pages 19–23, Dublin, Ireland. Association for Computational Linguistics.
- Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. *The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages*. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.
- Ishan Jindal, Ranit Aharonov, Siddhartha Brahma, Huaiyu Zhu, and Yunyao Li. 2020. Improved semantic role labeling using parameterized neighborhood memory adaptation. *arXiv preprint arXiv:2011.14459*.
- Ishan Jindal, Alexandre Rademaker, Michał Ulewicz, Ha Linh, Huyen Nguyen, Khoi-Nguyen Tran, Huaiyu Zhu, and Yunyao Li. 2022. *Universal proposition bank 2.0*. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1700–1711, Marseille, France. European Language Resources Association.
- Daniel Jurafsky and James H. Martin. 2021. Speech and language processing. <https://web.stanford.edu/~jurafsky/slp3/>.
- Paul Kingsbury and Martha Palmer. 2002. *From TreeBank to PropBank*. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).
- Oleksandr Kolomiyets, Parisa Kordjamshidi, Marie-Francine Moens, and Steven Bethard. 2013. *SemEval-2013 task 3: Spatial role labeling*. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 255–262, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Parisa Kordjamshidi, Steven Bethard, and Marie-Francine Moens. 2012. *SemEval-2012 task 3: Spatial role labeling*. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 365–373, Montréal, Canada. Association for Computational Linguistics.
- Zuchao Li, Shexia He, Jiaxun Cai, Zhuosheng Zhang, Hai Zhao, Gongshen Liu, Linlin Li, and Luo Si. 2018. *A unified syntax-aware framework for semantic role labeling*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2411, Brussels, Belgium. Association for Computational Linguistics.
- Ken Litkowski. 2004. *Senseval-3 task: Automatic labeling of semantic roles*. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 9–12, Barcelona, Spain. Association for Computational Linguistics.
- Ling Liu, Ishan Jindal, and Yunyao Li. 2022. Is semantic-aware bert more linguistically aware? a case study on natural language inference. In *Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI)*, Seattle, USA. Association for Computational Linguistics.
- Edward Loper, Szu-Ting Yi, and Martha Palmer. 2007. Combining lexical resources: mapping between propbank and verbnet. In *Proceedings of the 7th International Workshop on Computational Linguistics, Tilburg, the Netherlands*.
- Umar Maqsood, Sebastian Arnold, Michael Hülfenhaus, and Alan Akbik. 2014. Nerdle: Topic-specific question answering using wikia seeds. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 81–85.
- Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. *SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment*. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics.
- Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. *A survey on open information extraction*. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. *Universal Dependencies v1: A multilingual treebank collection*. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of

- semantic roles. *Computational linguistics*, 31(1):71–106.
- Sameer Pradhan, Julia Bonn, Skatje Myers, Kathryn Conger, Tim O’Gorman, James Gung, Kristin Wright-Bettner, and Martha Palmer. 2022. Propbank comes of age—larger, smarter, and more diverse. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288.
- Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. [SemEval-2007 task-17: English lexical sample, SRL and all words](#). In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic. Association for Computational Linguistics.
- Reinhard Rapp. 2022. [Using semantic role labeling to improve neural machine translation](#). In *Proceedings of the Language Resources and Evaluation Conference*, pages 3079–3083, Marseille, France. European Language Resources Association.
- Michael Roth and Mirella Lapata. 2016. [Neural semantic role labeling with dependency path embeddings](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1202, Berlin, Germany. Association for Computational Linguistics.
- Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.
- Chen Shi, Shujie Liu, Shuo Ren, Shi Feng, Mu Li, Ming Zhou, Xu Sun, and Houfeng Wang. 2016. Knowledge-based semantic embedding for machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2245–2254.
- Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.
- Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. [The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies](#). In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England. Coling 2008 Organizing Committee.
- Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206.
- Hongming Zhang, Haoyu Wang, and Dan Roth. 2020a. [Unsupervised label-aware event trigger and argument classification](#). *CoRR*, abs/2012.15243.
- Yu Zhang, Qingrong Xia, Shilin Zhou, Yong Jiang, Zhenghua Li, Guohong Fu, and Min Zhang. 2021. Semantic role labeling as dependency parsing: Exploring latent tree structures inside arguments. *arXiv preprint arXiv:2110.06865*.
- Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020b. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.

A Some historical background to existing evaluation metrics for SRL

Shared tasks have boosted the development of systems capable of extracting predicates and arguments from natural language sentences. Two regular academic events have promoted SRL shared tasks: SemEval and CoNLL. In this section, we summarize the approaches to SRL evaluation in the shared tasks and categorize their shortcomings.

A.1 Senseval and SemEval

SemEval (Semantic Evaluation) is a series of evaluations of computational semantic analysis systems that evolved from the Senseval (word sense evaluation) series. The SENSEVAL-3 (Litkowski, 2004) was about the automatic labeling of semantic roles and was designed to encourage research into and use of the FrameNet dataset. The systems receive as input unsegmented sentences (the constituents are not identified) a target word and its frame. They have to identify the frame elements within that sentence and tag them with the appropriate frame element name. In general, FrameNet frames contain many frame elements (an average of 10), most of which are not instantiated in a given sentence. Systems were not penalized if they returned more frame elements than those identified in the gold data. In scoring, each frame element returned by a system was counted as an item attempted. If the frame element had been identified in the gold data, the answer was scored as correct. In addition, the scoring program required that the frame boundaries identified by the system’s answer overlap with the gold annotation. An additional measure of system performance was the degree of overlap. If a system’s answer coincided precisely with the start and end position in the gold data, the system received an overlap score of 1.0. If not, the overlap score was the number of characters overlapping divided by the length of the gold annotation. The number attempted was the number of non-null frame elements generated by a system. Precision was com-

puted as the number of correct answers divided by the number attempted. The recall was computed as the number of correct answers divided by the number of frame elements in the test set. Overlap was the average overlap of all correct answers. The percent Attempted was the number of frame elements generated divided by the number of frame elements in the test set, multiplied by 100.

At SemEval-2007, three tasks evaluate SRL. In task 17 (subtask 2), the goal of the systems was to locate the constituents, which are the arguments of a given verb, and assign them appropriate semantic roles. Systems have to annotate the corpus using two different role label sets: the PropBank and the VerbNet. SemLink mapping (Loper et al., 2007) was used to generate the VerbNet roles. The precision, recall, and F-measure for both role label sets were calculated for each system output using the `sr1-eval.pl` script from the CoNLL-2005 scoring package (Carreras and Màrquez, 2005) (see below). Task 18 focused on Arabic and also used the same CoNLL-2005 scoring package. Task 19 consists of recognizing words and phrases that evoke semantic frames from FrameNet and their semantic dependents, which are usually, but not always, their syntactic dependents. The evaluation measured precision and recall for frames and frame elements, with partial credit for incorrect but closely related frames. Two types of evaluation were carried out. The first is the label matching evaluation. The participant’s labeled data were compared directly with the gold standard labeled using the same evaluation procedure used in the previous SRL tasks at SemEval. The second is the semantic dependency evaluation, in which both the gold standard and the submitted data were first converted to semantic dependency graphs and compared.

SemEval-2012 and SemEval-2013 introduced the ‘Spatial Role Labeling’ task. It concerns the identification of trajectors, landmarks, spatial indicators, the links between them, and the type of spatial relationships, including region, direction, and distance. Although similar to the standard SRL task, we will not discuss Spatial Role Labeling and its evaluation in this paper. Starting from SemEval-2014, a deeper semantic representation of sentences in a single graph-based structure via semantic parsing substituted the ‘shallow’ SRL tasks.

A.2 CoNLL

The Conference on Computational Natural Language Learning (CoNLL) is a yearly conference organized by the ACL’s Special Interest Group on Natural Language Learning (SIGNLL), focusing on theoretically, cognitively and scientifically motivated approaches to computational linguistics since 1999. The 2004 and 2005 shared tasks of the CoNLL were dedicated to SRL monolingual setting (English). The CoNLL-2008 shared task proposes a unified dependency-based formalism, which models both syntactic dependencies and semantic roles. The CoNLL-2009 builds on the CoNLL-2008 task and extends it to multiple languages.

The CoNLL-2004 shared task (Carreras and Màrquez, 2004) was based on the PropBank corpus, six sections of the Wall Street Journal part of the Penn Treebank (Kingsbury and Palmer, 2002) enriched with predicate–argument structures. The participants need to come up with machine learning strategies to SRL on the basis of only partial syntactic information, avoiding the use of full parsers and external lexico-semantic knowledge bases. The annotations provided for the development of systems include, apart from the argument boundaries and role labels, the levels of processing treated in the previous editions of the CoNLL shared task, i.e., words, PoS tags, base chunks, clauses, and named entities. In practice, number of target verbs are marked in a sentence, each governing one proposition. A system has to recognize and label the arguments of each target verb. The systems were evaluated with respect to precision, recall and the F1 measure using the `sr1-eval.pl` program. For an argument to be correctly recognized, the words spanning the argument as well as its semantic role have to be correct. The verb argument is the lexicalization of the predicate of the proposition. Most of the time, the verb corresponds to the target verb of the proposition, which is provided as input, and only in few cases the verb participant spans more words than the target verb. This situation makes the verb easy to identify and, since there is one verb with each proposition, evaluating its recognition overestimates the overall performance of a system. For this reason, the verb argument is excluded from evaluation. The shared task proceedings does not details how non-continuous arguments are evaluated.

Compared to the shared task of CoNLL-2004, three novelties were introduced in the 2005 edition

(Carreras and Màrquez, 2005): 1) the complete syntactic trees, with information of the lexical head for each syntactic constituent, given by two alternative parsers have been provided as input; 2) the training corpus has been substantially enlarged; 3) a cross-corpora evaluation is performed using a fresh test set from the Brown corpus. Evaluation didn't changed compared to CoNLL-2004 and it was reported to use the same evaluation code, a system has to recognize and label the arguments of each target verb. To support the role labeling task, sentences contain input annotations, that consist of syntactic information and named entities. Evaluation is performed on a collection of unseen test sentences, that are marked with target verbs and contain only predicted input annotations.

The CoNLL 2008 shared task (Surdeanu et al., 2008) was dedicated to the joint parsing of syntactic and semantic dependencies. The shared task was divided into three subtasks: (i) parsing of syntactic dependencies, (ii) identification and disambiguation of semantic predicates, and (iii) identification of arguments and assignment of semantic roles for each predicate. SRL was performed and evaluated using a dependency-based representation for both syntactic and semantic dependencies.

The task addressed propositions centered around both verbal and nominal predicates. The data was composed by the Penn Treebank, BBN's named entity corpus, PropBank and NomBank. The dependency-annotated data was obtain from a conversion algorithm from the constituent analyses. convert the underlying constituent analysis of PropBank and NomBank into a dependency analysis, the head of a semantic argument was identified with a straightforward heuristic. But there are cases that require special treatment, some arguments ended up with several syntactic heads and some arguments that were initially discontinuous in PropBank or NomBank where merged.

The official evaluation measures consist of three different scores: (i) syntactic dependencies are scored using the labeled attachment score (LAS), (ii) semantic dependencies are evaluated using a labeled F1 score, and (iii) the overall task is scored with a macro average of the two previous scores. The semantic propositions are evaluated by converting them to semantic dependencies, i.e., a semantic dependency from every predicate to all its individual arguments were created. These dependencies are labeled with the labels of the corresponding

arguments. Additionally, a semantic dependency from each predicate to a virtual ROOT node was created. The latter dependencies are labeled with the predicate senses. This approach guarantees that the semantic dependency structure conceptually forms a single-rooted, connected (not necessarily acyclic) graph. More importantly, this scoring strategy implies that if a system assigns the incorrect predicate sense, it still receives some points for the arguments correctly assigned. Several additional evaluation measures were applied to further analyze the performance of the participating systems. The *Exact Match* reports the percentage of sentences that are completely correct, i.e., all the generated syntactic dependencies are correct and all the semantic propositions are present and correct. The *Perfect Proposition F1* score entire semantic frames or propositions. The ratio between labeled F1 score for semantic dependencies and the LAS for syntactic dependencies.

As in CoNLL-2008, the CoNLL-2009 shared task (Hajič et al., 2009) combined syntactic dependency parsing and the task of identifying and labeling semantic arguments of verbs or nouns for six more languages (Catalan, Chinese, Czech, German, Japanese and Spanish) in addition to the original English from CoNLL-2008. Participants can choose the joint task (syntactic dependency parsing and SRL), or SRL-only (syntactic dependency provided). The novelty is that the evaluation data indicated which words were to be dealt with (for the SRL task). Predicate disambiguation was still part of the task, whereas the identification of argument-bearing words was not. This decision was made to compensate for the significant differences between languages and between the annotation schemes used. The evaluation of SRL was done similar to CoNLL-2008.

Prompt-based Learning for Text Readability Assessment

Bruce W. Lee^{1,2,‡}, Jason Hyung-Jong Lee²

¹University of Pennsylvania - PA, USA

²LXPER AI Research - Seoul, South Korea

brucelws@seas.upenn.edu

jasonlee@lxper.com

Abstract

We propose the novel adaptation of a pre-trained seq2seq model for readability assessment. We prove that a seq2seq model – T5 or BART – can be adapted to discern which text is more difficult from two given texts (pairwise). As an exploratory study to prompt-learn a neural network for text readability in a *text-to-text* manner, we report useful tips for future work in seq2seq training and ranking-based approach to readability assessment. Specifically, we test nine input-output formats/prefixes and show that they can significantly influence the final model performance.

Also, we argue that the combination of text-to-text training and pairwise ranking setup 1) enables leveraging *multiple* parallel text simplification data for teaching readability and 2) trains a neural model for the general concept of readability (therefore, better cross-domain generalization). At last, we report a 99.6% pairwise classification accuracy on Newsela and a 98.7% for OneStopEnglish, through a joint training approach. Our code is available at github.com/brucewlee/prompt-learning-readability.

1 Introduction

Readability assessment evaluates the reading difficulty of a given piece of text (Vajjala, 2021). The early traditional readability assessment methods like Flesch-Kincaid Grade Level (Kincaid et al., 1975) utilized a linear regression formula fitted to data from large-scale reading experiments on human subjects. More recently, readability assessment has often been viewed as a classification task (Feng et al., 2010). Under this classification-based task formulation, models using various handcrafted features (Xia et al., 2016; Vajjala and Meurers, 2012), computer-generated features (Martinc et al., 2021; Imperial, 2021), or both (Lee et al., 2021)

have been reported. Showing the potential that neural modeling is more suitable than handcrafted features in holistically capturing the inherent linguistic properties that affect readability.

Among the varying approaches to readability assessment, fine-tuning deep transformer models (Vaswani et al., 2017), that are pre-trained with language modeling objectives (e.g. *Masked Language Modelling*, *Next Sentence Prediction*), has proven highly effective in multiple reports (Lee and Vajjala, 2022; Lee et al., 2021). So far, encoder-only transformer architectures like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have been the go-to approach, and few reports discuss the applicability of other architecture types. Further, there is no report on how readability assessment can be cast in a text-to-text task formulation (§2). But recent reports (Raffel et al., 2020) show that a text-to-text is promising for multiple downstream tasks.

The main contribution of this paper is that we fine-tune full encoder-decoder transformer architectures (also referred to as *sequence-to-sequence*) and check if they can learn about text readability. A sequence-to-sequence model has been previously fine-tuned for downstream tasks like document ranking (Nogueira et al., 2020), but few reports discuss whether the architecture can learn about readability.

We fine-tune BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) on the popular OneStopEnglish (Vajjala and Lučić, 2018) and Newsela (Xu et al., 2015) data. Then, we measure their performance on two other datasets with readability annotations for US and CEFR curricula, respectively.

We also conduct methodological explorations on how a sequence-to-sequence model can be well-trained to learn text readability. This includes how input and output format should be structured, considering that the fine-tuning for a sequence-to-sequence model is naturally cast in a text-to-text

[‡]Woong Sung (Bruce) Lee was on leave from the University of Pennsylvania during the research period.

format (Nogueira et al., 2020). We summarize our research questions in two:

- 1) Can a sequence-to-sequence model be fine-tuned for text readability – with a parallel text simplification dataset?
- 2) If so, can the performance generalize across domains? In other words, is the model learning the *dataset* or the concept of *readability*?

These research questions and our task approach are intentionally formulated in reference to what previous literature (Vajjala, 2021; Lee et al., 2021) have proposed as future directions. As we elaborate further in the following sections, our approach simplifies some inherent problems that we had in the readability assessment. But our study has limitations and requires further explorations (§5).

2 Background Knowledge

2.1 Sequence-to-Sequence Transformers

Pre-trained sequence-to-sequence transformers essentially incorporate both encoder and decoder parts from the original Transformer (Vaswani et al., 2017) architecture. The most notable examples are T5 and BART, pre-trained using document denoising strategies (i.e. during pre-training, input is an intentionally corrupted document, and output is a recovered document or corrupted portions).

Though BART allows some flexibility in altering the model architecture for downstream tasks (Lewis et al., 2020), T5 is built with the intention of unifying all NLP tasks into a text-to-text-format (Raffel et al., 2020). Here, text-to-text means that both input and output are always text strings, unlike encoder-only, BERT-style models that only output either a class label ([CLS] token) or an input span.

2.2 Existing Downstream Tasks

Pre-trained sequence-to-sequence models can be fine-tuned to most NLP downstream tasks, including neural machine translation (Wang et al., 2022) and abstractive summarization (Saito et al., 2020). Then, a training instance is formatted in a way that is much like *telling* the model what to do by adding task-specific prefix (Raffel et al., 2020).

If a model were to be fine-tuned for translation, the input format could be “*translate English to German: That is good.*” and the target output is “*Das ist gut.*” When applied to document ranking, Nogueira et al. (2020) proposed a slightly different format. The input format was written “*Query: ...*

Document: ... Relevant:” so that the target output tokens – “*true*” or “*false*” – can naturally come after the input format. Our work is the most influenced by this formatting approach.

3 Experimental Setup

3.1 Methods

All our experiments are based on T5 and BART, both obtained from the respective online repositories through Huggingface (Wolf et al., 2019). Since the text-to-text formulation has never been used to fine-tune text readability¹, here we limit to the simple task of comparing the readability of two given texts. Following our input format (Table 1), we fed two text snippets of varying difficulties to the model every instance. Then, the model was trained to give the corresponding target output. We tested nine input-output formats, as shown in Table 1.

3.2 Datasets and Preparation

3.2.1 Data Type and Permutation Methods

The datasets that we use in this study are of two types. *Parallel* type contains multiple reading-level versions of a text (mostly through human expert paraphrasing). We call a grouping of text in multiple reading levels a *slug*. On the other hand, there is *distinct* type of datasets. *Distinct* type is a more common format where each text is given a readability level with no multiple versions of the same content. Our naming and permutation strategies are inspired by existing work on pairwise ranking for readability (Lee and Vajjala, 2022).

A parallel dataset $D_{parallel}$ can be expressed as a row-wise collection of i slugs $D_{parallel} = [S_1, \dots, S_i]$, where a slug is a column-wise collection of j pairs of a text and a reading level $S_i = [(x_{i,1}, y_{i,1}), \dots, (x_{i,j}, y_{i,j})]$. For parallel dataset, we perform permutation jP_2 on the slug level, creating $S'_i = [((x_{i,1}, x_{i,2}), (y_{i,1}, y_{i,2})), \dots, ((x_{i,j-1}, x_{i,j}), (y_{i,j-1}, y_{i,j}))]$. A pair like $((x_{i,1}, x_{i,2}), (y_{i,1}, y_{i,2}))$ is considered an instance for train/dev/test. Then, $D'_{parallel} = [S'_1, \dots, S'_i]$, where all S' are flattened to make $D'_{parallel}$ an iterable collection of tuples. This setup is intended to be robust to future implementations of paraphrase-based text simplification datasets where the standards for readability annotation/level are only consistent within a slug.

A distinct dataset $D_{distinct}$ can be expressed as a collection of i pairs of a text and a reading

¹Lee et al. (2021) trains BART for readability but uses class labels (BERT-style) as target output, instead of text tokens

Type	Input Format	Target Output
Question	"Which Text is more difficult? Text 1: ... Text 2: ..."	"Text 1" or "Text 2"
Statement	"Text 1 is more difficult than Text 2. Text 1: ... Text 2: ..."	"True" or "False"
Follow-up	"Text 1: ... Text2: ... More difficult:"	"Text 1" or "Text 2"
Reverse-Question	"Which Text is easier? Text 1: ... Text 2: ..."	"Text 2" or "Text 1"
Reverse-Statement	"Text 1 is easier than Text 2. Text 1: ... Text 2: ..."	"False" or "True"
Reverse-Follow-up	"Text 1: ... Text2: ... Easier:"	"Text 2" or "Text 1"
Alternate-Question	"Which Text is harder? Text 1: ... Text 2: ..."	"Text 1" or "Text 2"
Alternate-Statement	"Text 1 is harder than Text 2. Text 1: ... Text 2: ..."	"True" or "False"
Alternate-Follow-up	"Text 1: ... Text2: ... Harder:"	"Text 1" or "Text 2"

Table 1: Input-Output format candidates we tested. The text-to-text formulation is intuitive internally (model) and externally (human) because the model input and output are both representations of some semantic concept.

level $D_{distinct} = [(x_1, y_1), \dots, (x_i, y_i)]$. Then, we can perform permutation iP_2 to create $D'_{distinct} = [((x_1, x_2), (y_1, y_2)), \dots, ((x_{i-1}, x_i), (y_{i-1}, y_i))]$. This setup is under the postulation that readability annotation is consistent throughout the dataset. Pairwise instances of two same levels are removed.

3.2.2 Datasets

Two *parallel* and two *distinct* datasets are used. Also, we agree with Vajjala (2021)'s concern that "(in some datasets) articles tagged with different reading levels don't share the same topical content (...) question what the readability assessment models learn - is it a notion of text complexity, or topical differences among texts?". Hence, we only use parallel data – NEWS and OSEN – for training.

Our dataset processing strategy (§3.2.1) and pairwise comparison approach force the model to learn label-agnostic, the global concept of relative difficulties of texts. That is, the model learns that a text annotated level 5 should be harder than a level 3 (or a level 2 or a level 1) within a slug or a dataset. Such a setup is inherently robust against cross-domain usage (Table 3). Further, it enables combining multiple datasets of various slug sizes or readability annotation standards for joint training (§4).

Newsela(NEWS) is a *parallel* text simplification dataset introduced by Xu et al. (2015). It consists of 2,154 slugs, each item re-written 4 or 5 times for children at different grade levels. Hence, a total of 10,786 texts are contained, and 43,316 pairwise instances are created after data permutation (§3.2.1). Random shuffling split these instances into 6:2:2 for train/test splits. We provide reproducible scripts for all datasets through code.

OneStopEnglish(OSEN) is a *parallel* dataset intended for both text simplification and readability assessment research (Vajjala and Lučić, 2018). It

Format Type	OSEN		NEWS	
	T5	BART	T5	BART
Question	0.815(30)	0.965(18)	0.981(3)	-
Statement	0.639(29)	0.978(30)	-	-
Follow-up	0.784(25)	0.960(27)	-	-
Reverse-Q	0.793(30)	0.991(30)	-	0.993(3)
Reverse-S	0.524(28)	0.978(26)	-	-
Reverse-F	0.828(30)	0.991(30)	-	0.993(5)
Alternate-Q	0.811(30)	0.960(25)	-	-
Alternate-S	0.617(29)	0.978(30)	-	-
Alternate-F	0.789(28)	0.938(29)	-	-

Table 2: Validation set accuracy reports on NEWS and OSEN. The best epoch is reported in brackets. NEWS is only reported for the best format type due to data size.

consists of 189 slugs, each item in 3 paraphrases at different reading levels. A total of 567 texts are contained, and 1,134 pairwise instances are created. We use a 6:2:2 split ratio through random shuffling.

Common Core Standards(CCSB) is a *distinct* collection of exemplary official texts with readability annotations in U.S. Common Core Standards. We scraped data from the source ourselves. We used 69 story-type texts in 6 reading levels. After permutation, 3,846 pairwise instances are created.

Cambridge English Readability(CAMB) is a *distinct* dataset of reading passages from main suite Cambridge English Exams (Xia et al., 2016). All 331 texts are labeled A2, B1, B2, C1, or C2 reading levels, following the CEFR standards. After permutation, 87,574 pairwise instances are created.

3.3 Training

The batch size is fixed at 8, both for training and inference. The learning rate is fixed at 1e-5 for T5 and BART. We fine-tune OSEN for 30 epochs and NEWS for 3 epochs. We report the best epoch performance based on the validation set in Table 2. For joint training (Table 3), we take an OSEN-trained model and then fine-tune further using NEWS for 3 more epochs.

Model / Fine-Tune Data	Test Data			
	OSEN	NEWS	CCSB	CAMB
Flesch-Kincaid / None	0.978	0.986	0.798	0.808
T5 / OSEN	<u>0.784</u>	<u>0.518</u>	<u>0.509</u>	<u>0.492</u>
BART / OSEN	<u>0.978</u>	<u>0.871</u>	<u>0.639</u>	<u>0.629</u>
T5 / NEWS	<u>0.907</u>	<u>0.967</u>	<u>0.747</u>	<u>0.764</u>
BART / NEWS	0.987	<u>0.993</u>	<u>0.793</u>	0.883
T5 / OSEN + NEWS	<u>0.992</u>	<u>0.987</u>	<u>0.771</u>	<u>0.778</u>
BART / OSEN + NEWS	<u>0.983</u>	0.996	<u>0.790</u>	<u>0.865</u>

Table 3: In-domain and *cross-domain* accuracies across datasets. For OSEN and NEWS, test sets (§3.2.2) are used. The best result per dataset is in **bold**. T5 is trained with `Question` format, whereas BART is trained with `Reverse-F` format. Flesch-Kincaid refers to the popular Flesch-Kincaid Grade Level formula published in Kincaid et al. (1975). We use the implementation in `github.com/textstat/textstat`.

4 Results

1. **A pretrained sequence-to-sequence model could be fine-tuned for text readability, in a text-to-text style.** Table 2 shows that the concept of readability could be fine-tuned in a text-to-text task formulation, some setups with decent accuracies of > 0.9 . For a smaller dataset (OSEN), BART significantly outperformed T5, but their performance deviation was little on a larger dataset (NEWS). We believe this is caused due to difference in pre-training methodologies that caused T5 to require more training steps to learn about our downstream task. Also, BART always generalized better than T5 across unseen datasets in Table 3.

2. **Input-output format significantly affected the final performance, especially when fine-tuning T5 with lesser training steps (OSEN).** Among the nine input-output formats we tested, T5 and BART performed best under `Question` and `Reverse-Q/F` types, respectively. Performance deviations caused by input-output format changes were larger than we expected. Further, no certain format generally ensured good results across models. This raises the need for additional "format-tuning" processes when exploring new models. However, it must be noted that several observations point to T5 being under-trained for the general concept of readability at the data size of OSEN (see Table 2 and Table 3). The input-output format's influence is lesser for setups where models learned better about readability.

3. **Joint training has the potential to help both in-domain and cross-domain performances.** Joint training of multiple datasets for a single model is an under-explored concept in readabil-

ity assessment. This is because human experts annotate existing datasets with varying standards. Dataset construction can also differ (e.g. different number of classes or too difficult to map classes). Hence, it was unknown if combining datasets of varying labeling standards improves performance.

This work solves the problem by re-casting the task into a simple, universal question of comparing two texts' difficulties (§3.2.2). Table 3 shows that in- and cross-domain performances can improve through joint training. For example, in-domain accuracies for OSEN increased to 0.208 when the model was further fine-tuned with a larger extra data, NEWS. However, a NEWS-only model generally performed better than the OSEN+NEWS model in Table 3. We expect that OSEN, which is almost 40 times smaller, only confused the model.

4. **Exposing the model to more texts with a wider range of readability helped fine-grained readability comparison.** Importantly, we showed that readability assessment models fine-tuned with *parallel* datasets could be generalized across *distinct* datasets (e.g. OSEN \rightarrow CCSB). But model performances varied depending on label distance. Models performed better when the two compared texts' readability labels were larger apart (i.e. the model is more likely to guess level 1 vs level 4 correctly than level 1 vs level 2). This problem worsened when the model was trained using OSEN. Using NEWS as training data or extra data helped. We want to point out that a slug size in NEWS is 4 or 5, exposing the model to more permutations.

5. **Text-to-text style fine-tuning required more training steps than expected.** The majority of our OSEN fine-tuning experiments showed that the model's validation set performance continues to increase up to epoch 30. This is contrastive to how usual classification approaches, using encoder-only models, only fine-tune up to epoch 3~5 even on smaller datasets like OSEN or CAMB (Lee et al., 2021). Intuitively speaking, there is potential that better performance can be achieved if fine-tuned further. We will explore this concept in the future.

6. **Though often overlooked, traditional readability formulas provide challenging baselines.** The traditional readability formulas are criticized for their low performances in multi-class ranking or regression-based readability task formulation (Lee and Lee, 2023). However, they provide surprisingly strong baselines for pairwise difficulty comparisons, as seen in Table 3.

5 Conclusion

So far, we have reported our exploratory work on casting readability assessment tasks in a text-to-text formulation for BART and T5. We summarized our observations into five categories in §4, which can serve as base guidelines for future work. Our experimental setup and data permutation methods allow the joint training of more than one dataset, regardless of whether the dataset construction is *parallel* or *distinct* (§3). Using NEWS as extra training data further to fine-tune an OSEN-trained model greatly improved model performance. However, we did not train the other way around (NEWS → OSEN), which should be proved in the future.

6 Limitations

Our limitations are in input text length and output labels. Though our novel task formulation allows the application of essential concepts like *joint training* or *cross-domain evaluation* in the field of readability assessment, it is based on a pairwise classification method. Since the pairwise approach only allows the readability ranking of two texts (e.g. which is easier?), it lacks practicality compared to regression or multi-label classification-based models. Though we achieve an almost perfect accuracy of 99.6% in Newsela data, knowing which is easier out of texts has little use as a real-world system. Hence, further research must be conducted to generate **more useful output labels** and process longer sequences. Like [Nogueira et al. \(2020\)](#), we are looking into using a sliding window to generate output labels for longer input sequences. We are also researching neural models pre-trained specifically for readability assessment using the prompt-based learning method introduced in the paper. Such a model can be leveraged for multi-class classification.

References

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. 2010. A comparison of features for automatic readability assessment. In *Coling 2010: Posters*, pages 276–284.
- Joseph Marvin Imperial. 2021. Bert embeddings for automatic readability assessment. *arXiv preprint arXiv:2106.07935*.
- J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.
- Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. [Pushing on text readability assessment: A transformer meets handcrafted linguistic features](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Bruce W Lee and Jason Hyung-Jong Lee. 2023. Traditional readability formulas compared for english. *arXiv preprint arXiv:2301.02975*.
- Justin Lee and Sowmya Vajjala. 2022. A neural pairwise ranking model for readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3802–3813.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. [Supervised and Unsupervised Neural Approaches to Text Readability](#). *Computational Linguistics*, pages 1–39.
- Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

- Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, and Junji Tomita. 2020. Abstractive summarization with combination of pre-trained sequence-to-sequence and saliency models. *arXiv preprint arXiv:2003.13028*.
- Sowmya Vajjala. 2021. Trends, limitations and open challenges in automatic readability assessment research. *arXiv preprint arXiv:2105.00973*.
- Sowmya Vajjala and Ivana Lučić. 2018. Onestopenglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 297–304.
- Sowmya Vajjala and Detmar Meurers. 2012. [On improving the accuracy of readability classification using insights from second language acquisition](#). In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 163–173, Montréal, Canada. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Wenxuan Wang, Wenxiang Jiao, Yongchang Hao, Xing Wang, Shuming Shi, Zhaopeng Tu, and Michael Lyu. 2022. Understanding and improving sequence-to-sequence pretraining for neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2591–2600.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. [Text readability assessment for second language learners](#). In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, San Diego, CA. Association for Computational Linguistics.
- Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

A Obtaining Dataset

We obtained Newsela by requesting academic access at newsela.com/data.

OneStopEnglish dataset is freely available at github.com/nishkalavallabhi/OneStopEnglishCorpus.

Cambridge English Readability dataset is freely available at ilexir.co.uk/datasets/index.html.

We crawled Common Core Appendix B from www.corestandards.org/assets/Appendix_B.pdf.

B Training Details

B.1 Models, GPU, Train Time

All pre-trained models are retrieved from Huggingface. Fine-tuning code is written in PyTorch. We used a single NVIDIA RTX 2080 GPU for all our training.

T5

- huggingface.co/t5-base
- 1,020 seconds for 30 epochs on permuted OneStopEnglish, in `Question` format
- 6,994 seconds for 3 epochs on permuted Newsela, in `Question` format

BART

- huggingface.co/facebook/bart-base
- 532 seconds for 30 epochs on permuted OneStopEnglish, in `Reverse-F` format
- 4,084 seconds for 3 epochs on permuted Newsela, in `Reverse-F` format

B.2 More on Input Sequence Length

The current experimental setup is inherently weak against long texts. This is because we input a set of two texts with format prefixes (Table 1). Hence, for a model with 512 max token length our actual token length limitation per text is ≤ 256 , with the exact number depending on chosen format type. Though both BART and T5 support longer input sequences, in comparison to other popular models like BERT or RoBERTa, we must empirically confirm if good performance can be achieved when BART and T5 are used with longer max sequence length. In this paper, the max sequence length was set to 512 (which means that ≤ 256 token limit existed per text), and most texts from OSEN had to be truncated before training.

Best Practices in the Creation and Use of Emotion Lexicons

Saif M. Mohammad

National Research Council Canada

saif.mohammad@nrc-cnrc.gc.ca

Abstract

Words play a central role in how we express ourselves. Lexicons of word–emotion associations are widely used in research and real-world applications for sentiment analysis, tracking emotions associated with products and policies, studying health disorders, tracking emotional arcs of stories, and so on. However, inappropriate and incorrect use of these lexicons can lead to not just sub-optimal results, but also inferences that are directly harmful to people. This paper brings together ideas from Affective Computing and AI Ethics to present, some of the practical and ethical considerations involved in the creation and use of emotion lexicons — *best practices*. The goal is to provide a comprehensive set of relevant considerations, so that readers (especially those new to work with emotions) can find relevant information in one place. We hope this work will facilitate more thoughtfulness when one is deciding on what emotions to work on, how to create an emotion lexicon, how to use an emotion lexicon, how to draw meaningful inferences, and how to judge success.

1 Introduction

Words often convey affect (emotions, sentiment, feelings, and attitudes); either explicitly through their core meaning (denotation) or implicitly through connotation. For example, *dejected* denotes sadness. On the other hand, *failure* simply connotes sadness. Either through denotation or connotation, both words are associated with sadness. A compilation of such associations is referred to as a *word–affect association lexicon* (aka *emotion lexicon*).¹ An entry in a lexicon usually includes a word, an emotion category or affect dimension (e.g., joy, fear, valence, arousal, etc.), and a score indicating association (or strength of association).

¹This includes *sentiment lexicons* that capture valence (association with the positive–negative dimension) and other lexica that capture affect-related phenomena.

Examples of emotion lexicons include the General Inquirer (Stone et al., 1966), ANEW (Nielsen, 2011; Bradley and Lang, 1999), LIWC (Pennebaker et al., 2001), Pittsburgh Subjectivity Lexicon (Wilson et al., 2005), *NRC Emotion Lexicon* (Mohammad and Turney, 2010, 2013), and the *NRC Valence, Arousal, and Dominance (VAD) Lexicon* (Mohammad, 2018). These were all created by manual annotation (either by experts or crowd-sourced). There also exist lexicons that were generated automatically from large text corpora using statistical and/or machine learning algorithms; e.g., WordNet Affect (Strapparava et al., 2004), SentiWordNet (SWN) (Baccianella et al., 2010).

Emotion lexicons have a wide range of applications in commerce, public health, and research (in NLP, Psychology, Social Sciences, Digital Humanities, etc.). Some notable examples include: tracking brand and product perception via social media posts, tracking support for controversial issues and policies, tracking buy-in for non-pharmaceutical health measures such as social distancing during a pandemic, literary analysis, and developing more natural dialogue systems. The lexicons can be used on their own or in support of neural machine learning (ML) algorithms for emotion recognition.

Lexicon-based emotion analyses are especially popular in real-world applications and research outside of computer science because they are interpretable, have a low carbon footprint, and do not require significant programming expertise. Further, since outputs of ML models are highly dependent on training data, use of a model often requires retraining, and there may not exist labeled data from the target domain. Further, Teodorescu and Mohammad (2022) show that when determining broad trends (emotion arcs) and aggregating information from hundreds (if not more) instances for every time step, simple lexicon-based methods are extremely accurate (correlations above 0.95 with ground truth arcs).

However, inappropriate and incorrect use of these lexicons, can lead to not just sub-optimal results, but also inferences that are directly harmful. For example, using lexicons to infer emotions from limited amount of data to make judgments about refugee applications, to make judgments about which groups of people are shown certain advertisements and which groups are not, marking businesses owned by some groups of people as less liked than that of others, etc.

Emotions are deeply personal, private, and complex. Even the best natural language systems largely only employ pattern matching based on huge amounts of historical data, and thus often do not really understand what the user is trying to convey, let alone how they are feeling. In fact, some recent commercial and governmental uses of emotion recognition have garnered considerable criticism, including: infringing on one's privacy, exploiting vulnerable sub-populations, and even allegations of pseudo-science (Mohammad, 2022b; Wakefield, 2021; ARTICLE19, 2021; Woensel and Nevil, 2019).

This paper brings together ideas from Affective Computing and AI Ethics to present, in one place, some of the practical and ethical considerations involved in the creation and use of emotion lexicons — *best practices*.² We hope this work will facilitate more thoughtfulness when one is deciding on what emotions to work on, how to create an emotion lexicon, how to use an emotion lexicon, and how to judge success. Additional benefits of such a document include:

1. Presents the trade-offs of relevant choices so that stakeholders can make informed decisions appropriate for their context.
2. Has citations and pointers; acts as a jumping off point for further reading.
3. Helps engage the various stakeholders of an emotion task with each other. Helps stakeholders challenge assumptions made by researchers and developers.
4. Helps develop harm mitigation strategies.
5. Acts as a useful introductory document on emotion lexicons (complements survey articles).

Note that even though this article is focused on emotion lexicons, many of the ethical consid-

²This paper is a reframed and expanded avatar of an earlier datasheet paper for emotion lexicons (Mohammad, 2020).

erations apply broadly to natural language lexicons/resources in general. Also, see Mohammad (2022b) for a broader discussion on the ethical considerations associated with automatic emotion recognition (AER).

This work is in the same spirit as other recent innovations in exercising responsible research such as datasheets for datasets (Gebru et al., 2018), model cards for systems (Mitchell et al., 2019), and ethics sheets for AI tasks (Mohammad, 2022a). However, unlike datasheets and model cards which are designed for individual datasets and systems and that are published after the work is done, the goal of this work is to provide a more general-purpose relevant resource, accessible at the very beginning of one's project. Also, unlike an ethics sheet for a automatic emotion recognition that may cover all kinds of ethical considerations associated with the task of interest, this document has a focus on the creation of emotion lexicons and their use in AI tasks.

Ethics considerations are not about objective metrics or simple checklists. They involve engaging with issues that impact stake holders, especially those that are already disadvantaged. Thus, a big component of this work is to raise awareness of relevant issues, to underscore how often there are no easy solutions, and that meaningful change requires painstaking, slow, and deliberate engagement with the stakeholders. Additionally, such documents are useful for those that are impacted to question and challenge assumptions made by unfair decisions of automated systems.

2 Best Practices

Below we present various best practices (practical and ethical considerations) pertaining to 22 aspects of emotion lexicon creation and use. The 22 aspects are grouped under the coarser categories pertaining to a lexicon's life cycle: A. Lexicon Design, B. Annotation, C. Entries in the Lexicon, and D. Applying the Lexicon. Note that while many considerations are presented from the perspective of lexicon creation, they are also relevant to the users of a lexicon — knowing what decisions were made during the creation of a lexicon help one to assess appropriateness of using the lexicon.

The goal is to provide a comprehensive set of relevant considerations, so that readers (especially those new to research or new to work with emotions) can find the information in one place. Thus,

we include both the considerations that are especially specific to emotions, as well as others that apply more broadly (even if they are somewhat well known). Also, the points listed below are not meant to be the final word, but rather jumping off points for further thought and discussion.

2.1 Overview

An overview of the 22 aspects is presented below; followed by the detailed descriptions.

A. LEXICON DESIGN

1. Purpose or Objective
2. Emotion Category or Dimension
3. Word Senses and Dominant Sense Priors
4. Discrete or Continuous Value Labels

B. ANNOTATION

5. Questionnaire
6. Comparative Annotations
7. Annotators
8. Quality Control

C. ENTRIES IN THE LEXICON

9. Annotation Aggregation
10. Relative (not Absolute)
11. Coverage
12. Not Immutable
13. Perceptions (not “truth”)
14. Socio-Cultural Biases
15. Inappropriate Biases
16. Errors
16. Mechanism to Report and Fix Errors

D. APPLYING THE LEXICON

18. Fit of the Lexicon to One’s Data
19. Rescaling the Lexicon for One’s Task
20. Metrics & Features Drawn from the Lexicon
21. Removing Neutral Words
22. Inferences

2.2 Detailed Descriptions

A. LEXICON DESIGN

#1. Purpose or Objective: Consider and document the objective(s) of building the emotion lexicon. There can be more than one objective. The objectives guide various design choices involved in the creation of the lexicon. See [Selbst et al. \(2019\)](#) for common pitfalls in designing and framing socio-technical systems; and [Mohammad \(2022b\)](#) for common pitfalls in designing and framing automatic emotion recognition tasks. Users of emotion lexicons can study the purpose of each lexicon to determine which is most suitable for their use case.

Broadly speaking, the objectives tend to be around the study of word–emotion associations (exploring various research questions at the intersection of language and emotions) and aiding automatic emotion detection from utterances. However, individual projects often have specific goals, for example, to study specific phenomenon such as loneliness and empathy, to study inappropriate biases, to detect what emotions people perceive from utterances, to study how automatic systems should perceive the emotions in utterances, how automatic systems should use words to convey emotions, etc. It is important to recognize that some of these objectives are very related, but they have important differences. For example, while a general-purpose emotion lexicon will capture a number of benign associations, it will also capture inappropriate societal biases. If one wants to use a lexicon in a text generation system, then they should either use a lexicon designed specifically for that purpose, or address the biases in a general purpose lexicon, before using it.

Work using emotion lexicons should not claim that using it one can determine one’s emotional state from their utterance. At best, recognition systems (whether they use emotion lexicons or not) capture what one is trying to convey or what is perceived by the listener/viewer; and even there, given the complexity of human expression, they are often inaccurate. Several studies have shown that it is difficult to fully measure psychological states of people ([Stark, 2018](#); [Barrett, 2017b](#)).

In contrast, statistical analyses with features drawn from emotion lexicons can be used to accurately determine broad trends in the emotional state of a population over time ([Teodorescu and Mohammad, 2022](#)). Here, inferences are drawn at aggregate level from much larger amounts of data. Studies on public health, such as those on loneliness ([Guntuku et al., 2019](#); [Kiritchenko et al., 2020](#)), depression ([De Choudhury et al., 2013](#); [Resnik et al., 2015](#)), suicidality prediction ([MacAvaney et al., 2021](#)), bipolar disorder ([Karam et al., 2014](#)), stress ([Eichstaedt et al., 2015](#)), emotions during a pandemic ([Vishnubhotla and Mohammad, 2022](#)), and general well-being ([Schwartz et al., 2013](#)) fall in this category. Here too, however, it is best to be cautious in making claims about mental state, and use emotion recognition as one source of evidence amongst many (and involve expertise from public health and psychology).

#2. Emotion Category or Dimension: A key decision in the creation of an emotion lexicon is which conceptualization or facet of emotion to use. For example, should it capture emotion categories such as joy, sadness, fear, optimism, etc., or will it capture dimensions such as valence, arousal, and dominance. Psychologists and neuro-scientists have identified several theories of emotion that can inform the choice of categories and dimensions, including: the Basic Emotions Theory (BET) (Ekman, 1992; Ekman and Davidson, 1994), the Dimensional Theory (Osgood et al., 1957; Russell, 1980; Russell and Mehrabian, 1977; Russell, 2003), Cognitive Appraisal Theory (Scherer, 1999; Lazarus, 1991), and the Theory of Constructed Emotions (Barrett, 2017b).

Since ML approaches rely on human-annotated data (which can be hard to obtain in large quantities), emotion recognition research has often gravitated to the Basic Emotions Theory, as that work allows one to focus on a small number of emotions. This attraction has been even stronger in the vision research because of BET's suggested mapping between facial expressions and emotions. However, many of the tenets of BET, such as the universality of some emotions and their fixed mapping to facial expressions, stand discredited or are in question (Barrett, 2017a; Barrett et al., 2019).

Carefully consider which emotion formulation you wish to capture in your lexicon, or is appropriate for your task/project. For example, one may choose to work with the dimensional model or the model of constructed emotions if the goal is to infer behavioural or health outcome predictions. Despite criticisms of BET, it makes sense for some NLP work to focus on *categorical emotions* such as joy, sadness, guilt, pride, fear, etc. (including what some refer to as basic emotions) because people often talk about their emotions in terms of these concepts. Many human languages have words for these concepts (even if our individual mental representations for these concepts vary to some extent) (Wierzbicka, 1999). However, note that work on categorical emotions by itself is not an endorsement of the BET. Do not refer to some emotions as basic emotions, unless you mean to convey your belief in the BET. Careless endorsement of theories can lead to the perpetuation of ideas that are actively harmful (such as suggesting we can determine internal state from outward appearance—physiognomy).

#3. Word Senses and Dominant Sense Priors: Words when used in different senses and contexts may be associated with different emotions. The entries in the emotion lexicons are mostly indicative of the emotions associated with the predominant senses of the words. This is usually not too problematic because most words have a highly dominant main sense (which occurs much more frequently than the other senses). In specialized domains, some terms might have a different dominant sense than in general usage. Entries in the lexicon for such terms should be appropriately updated or removed. However, if the goal of the project is to create a lexicon for a specialized domain, then one should guide the annotation process accordingly.

#4. Discrete or Continuous Value Labels: Many emotion lexicons have discrete binary labels for words (positive–negative, joy–no joy, fear–no fear, and so on). Lexicons such as ANEW and the NRC VAD Lexicon have real-valued scores between 0 and 1, -1 and 1, 0 to 5, 0 to 100, etc. Real-valued scores allows one to make finer distinctions in the degree of emotion. They allow one to determine the intensity of emotion. Binary-labeled lexicons are used primarily to determine density of emotion word usage; for example, to explore whether there is a higher percentage of tweets with loneliness words during the Covid-19 pandemic, than in the years before the pandemic. Determine which type of lexicon is more aligned with your objectives.

B. ANNOTATION

#5. Questionnaire: Arguably the most crucial aspect in the creation of an emotion lexicon is the questionnaire. What is asked and how it is asked determines the outcome. Below are key recommendations in the design of questionnaires:

- a. Where appropriate, break the task/question into simpler sub-tasks/sub-questions.
- b. It is better to have separate tasks for different questions and emotion dimensions. Asking for responses about more than one emotion dimension requires the annotator to switch contexts and leads to more cognitive load.
- c. Keep the instructions clear and easy to follow.
- d. Examples are more important than definitions. People tend to learn faster and better through examples. It is still good to include simple definitions of relevant concepts.

- e. Refer to the theories for emotions work in psychology on to how to collect emotional information from respondents. Especially useful are the terms used to define emotion dimensions: e.g., as per the dimensional model of emotions (Russell, 1980) *arousal* is defined as the active–sluggish dimension, in the stereotype content model of social perception (Cuddy et al., 2008), *warmth* is defined as the trustworthiness, friendliness, kindness dimension. These words should be used when eliciting annotation responses.
- f. Keep the instructions brief. This is respectful of annotator time, and one can only keep track of a limited number of instructions at a time.
- g. Explain the purpose of the annotation task. This is respectful of annotators. People have a right to know (in appropriate detail) what research they are contributing their time for. This may also lead to more engaged annotators.
- h. Include an optional comment box that gives annotators a way to provide feedback, raise issues, and to be heard.
- i. Make the questionnaire and instructions freely available. This helps others to build on your work. It allows users to see exactly how the questions were phrased, and thus how to interpret the resulting emotion lexicon.

See also other data curation and questionnaire development tips from non-NLP fields such as psychology (Aguinis et al., 2021).

#6. Comparative Annotations: Real-valued scores provide fine-grained emotion information; however, it is difficult for humans to provide direct scores at this granularity. A popular approach to obtain real-valued scores is by providing the annotators with numeric rating scales.³ These scales have numbers (usually 1 to 5 or 1 to 7) and the annotator has to select which number is most indicative of the degree of association with the property of interest for the given word; given that the lowest number on the scale indicates least association and the highest number indicates the most association.⁴ The scores for an item from multiple annotators is averaged to obtain a real-valued score that is assigned to the word–emotion pair.

³<https://www.questionpro.com/blog/rating-scale/>

⁴It is good practice to anchor the numeric values with labels such as maximum/moderate/low association.

A common problem of annotation by rating scales is inconsistencies in annotations among different annotators. One annotator might assign a score of 87 to one word, while another annotator may assign a score of 81 to the same word. It is also common that the same annotator might assign different scores to the same word, if asked to annotate again after a period of time. Further, annotators often have a bias towards selecting scores in the middle of the scale, known as *scale region bias* (Presser and Schuman, 1996; Baumgartner and Steenkamp, 2001).

Paired Comparisons (Thurstone, 1927; David, 1963) is a comparative annotation method, where respondents are presented with pairs of items and asked which item has more of the property of interest (for example, which is more positive). The annotations can then be converted into a ranking of items by the property of interest, and one can even obtain real-valued scores indicating the degree to which an item is associated with the property of interest. The paired comparison method does not suffer from the problems discussed above for the rating scale, but it requires a large number of annotations—order N^2 , where N is the number of items to be annotated.

Best–worst scaling (BWS) (Louviere, 1991) is a form of comparative annotation, like paired comparison, but it requires much fewer annotations. Annotators are given n items (an n -tuple, where $n > 1$ and commonly $n = 4$).⁵ They are asked which item is the *best* (highest in terms of the property of interest) and which is the *worst* (least in terms of the property of interest). When working on 4-tuples, best–worst annotations are particularly efficient because each best and worst annotation will reveal the order of five of the six item pairs (e.g., for a 4-tuple with items w , x , y , and z , if w is the best, and z is the worst, then $w > x$, $w > y$, $w > z$, $x > z$, and $y > z$). Real-valued scores of association between the items and the property of interest can be determined using simple arithmetic on the number of times an item was chosen best and number of times it was chosen worst (Orme, 2009; Flynn and Marley, 2014). It has been empirically shown that three annotations each for $2N$ 4-tuples is sufficient for obtaining reliable scores

⁵At its limit, when $n = 2$, best–worst scaling reduces to a *paired comparison* (Thurstone, 1927; David, 1963); However, then a much larger set of tuples need to be annotated (closer to N^2).

(where N is the number of items) (Louviere, 1991; Kiritchenko and Mohammad, 2016). Kiritchenko and Mohammad (2016; 2017) showed through empirical experiments on emotion lexicons that BWS produces more reliable and more discriminating scores than those obtained using rating scales.

Within the NLP community, BWS has been used for creating datasets for relational similarity (Jurgens et al., 2012), word-sense disambiguation (Jurgens, 2013), word-sentiment intensity (Kiritchenko and Mohammad, 2016), sentence-sentence semantic relatedness (Abdalla et al., 2023), etc.

#7. Annotators: Who is recruited to annotate the data also impacts the lexicon that is generated.

- a. *Experts or Crowd:* If a task has clear correct and wrong answers and knowing the answers requires some training/qualifications, then one can employ domain experts to annotate the data. However, emotion annotations largely do not fall in this category. People are the best judges of their emotions and how they use words to communicate them. If the goal is to determine how people use language or we want to know how people perceive words, phrases, and sentences then we might want to employ a large number of annotators (crowdsourcing). Note that this is also a scenario where there can be more than one appropriate answer.
- b. *Diversity:* Emotion lexicons are a function of their annotators. Consider who all should be represented in the annotator pool, and actively recruit people from under-represented groups. Seek appropriate demographic information (respectfully and ethically). Document annotator demographics at an aggregate level.
- c. *Informed Consent, Privacy, and Potential for Harms:* Provide a clear and easy-to-understand description of what the task will involve, potential risks, and what information will be collected, before obtaining consent from the annotators. Note that if the terms included for annotation or the chosen dimension of annotation is particularly negative, then there may be significant risk of adversely impacting the annotator's mental health. In such cases, suitable avenues for recourse must be provided.
- d. *Remuneration:* Determine fair compensation for the task. Inform the annotators of the pay and the time commitment expected.
- e. *Miscellaneous:* There are several other ethical considerations also involved with such work such as: worker invisibility, lack of learning trajectory, humans-as-a-service paradigm, worker well-being, and worker rights (Dolmaya, 2011; Fort et al., 2011; Standing and Standing, 2018; Irani and Silberman, 2013).
- f. *Ethics Approval:* Obtain approval of the project and annotation plan from your institution's research ethics board before conducting the annotation. The ethics boards are also a great source of feedback for improving the ethical standards of the annotation process. If unsure whether some work requires ethics approval, reach out to the ethics board. Many institutions provide expedited review in cases of low risk.

Document these considerations so that the users can judge suitability of the lexicon for their work.

#8. Quality Control: Good quality control strategies can make a large difference for any scenario of annotations, but are especially important when the annotations are done via crowdsourcing. Quality control strategies can be of three kinds:

Type 1: applied before data annotation begins

Type 2: applied during data annotation, and

Type 3: applied after data annotation.

It is recommended to apply measures of all three kinds. Examples of Type 1 include: careful questionnaire design and setting up training or qualification annotations to screen annotators.

A particularly powerful example of a Type 2 measure is to intersperse the instances with small number of hidden gold instances (~5%) — instances for which the appropriate label(s) are pre-determined (by, say, the authors). If a crowd worker responds with an answer not already marked as appropriate, then they are immediately notified, the annotation is discarded. If an annotator's accuracy on the gold questions falls below a pre-chosen threshold (say, 80%), then they are refused further annotation, and all of their annotations are discarded. This way the gold instances serve as a mechanism to avoid malicious annotations, as well as a way to further train the annotators. This also avoids scenarios where an annotator provides responses to a large number of questions, only to later learn that they misinterpreted something, rendering all of their annotations useless. The use of gold questions was popularized by the crowdsourcing platform CrowdFlower (now, Figure8).

Examples of Type 3 quality control measures include: removal of responses from people who answer questions too quickly, or whose responses are more than two standard deviations away from the responses of others. There also exist approaches that identify which annotators to trust using machine learning algorithms (Raykar and Yu, 2012; Hovy et al., 2013).

C. ENTRIES IN THE LEXICON

#9. Annotation Aggregation: Each instance in a lexicon (usually a word) is often annotated by a number of annotators. Standard practice in aggregating the responses from multiple annotators is to take the most frequent response. However, it should be noted that sometimes other responses are also appropriate. Further, different socio-cultural groups can perceive language differently, and taking the majority vote can have the effect of only considering the perceptions of the majority group. When these views are crystallized in the form of a lexicon, it can lead to the false perception that the norms so captured are “standard” or “correct”, whereas other associations are “non-standard” or “incorrect”. Thus, it is worth explicitly disavowing that view and stating that the lexicon simply captures the perceptions of the majority group among the annotators. Thus, it is recommended to also make available disaggregated annotations (annotations in their raw form – without aggregation). Note that it is also problematic to consider all annotator responses as valid because sometimes annotators make mistakes, and some may have inappropriate biases (see #15).

#10. Relative (not Absolute): The absolute values of the association scores themselves usually have no meaning. The scores help order the words relative to each other. For example, a term with a high valence score is associated with more positiveness than a term with a lower score.

#11. Coverage: Some lexicons have a few hundred terms, and some have tens of thousands of terms. However, even the largest lexicons do not include all the terms in a language. Mostly, they include entries for the canonical forms (lemmas), but some also include morphological variants. The high-coverage lexicons, such as the NRC Emotion Lexicon, have tens of thousands of terms. However, when using the lexicons in specialized domains, one may find that a number of common terms in the domain are not listed in the lexicons.

#12. Not Immutable: The associations do not indicate an inherent unchangeable attribute. Emotion associations can change with time, but these lexicon entries are largely fixed. They pertain to the time they are created or the time associated with the corpus from which they are created.

#13. Perceptions (not “truth”): Emotion lexicons largely capture how speakers of a language perceive the emotion associations of words. As mentioned in the previous bullet, this can change with time. Further, it can also be different for different people. Mohammad and Turney (2013) found that when the annotators are asked to judge emotion associations in terms of ‘how speakers of a language perceive the word’, the results have lower variance than when asked ‘the emotions evoked in the annotator’. Consider your objective when deciding which of the two framings (or some other) is more appropriate for your use case.

#14. Socio-Cultural Biases: Since the emotion lexicons have been created by people (directly through crowdsourcing or indirectly through the texts written by people) they capture various human biases. These biases may be systematically different for different socio-cultural groups. Document who produced the data (people from which countries, what is the gender distribution, age distribution, etc.) in the paper describing the dataset or in the associated datasheet. An advantage of crowdsourcing is that the annotations are from a wider pool of annotators; however, crowd annotators are systematically different from, and not representative of, the general population.

#15. Inappropriate Biases: Some of the human biases that have percolated into the lexicons may be rather inappropriate. For example, entries with low valence scores for certain demographic groups or social categories. Studying such biases in the lexicon can be useful to show and address some of the historical inequities that have plagued humankind. Nonetheless, when these lexicons are used in specific tasks, care must be taken to remove such entries from the lexicons where necessary.

#16. Errors: Even though the researchers take several measures to ensure high-quality and reliable data annotation (e.g., multiple annotators, clear and concise questionnaires, framing tasks as comparative annotations, interspersed check questions, etc.), human-error can never be fully eliminated in large-scale annotations. Expect a

small number of clearly wrong entries. Automatically generated lexicons also can have erroneous entries. They are often built on the assumption that the tendency of a word to co-occur with emotion-associated seed terms is proportional to its association with that emotion. However, in any corpus, there will always be some amount of chance high co-occurrences that are not accurate reflections of the true associations.

#17. Mechanism to Report and Fix Errors: Provide a mechanism for users to report issues and errors. Fix errors and where appropriate issue warnings for how some types of entries can be mis-interpreted or misused. Periodically assess whether certain types of entries need to be proactively checked. For example, there has been growing recognition that emotion associations associated with identity groups are particularly sensitive, affected by historical bias, and so one must be careful in how they interpret the associations captured in lexicons.

D. APPLYING THE LEXICON

#18. Examining the Fit of the Lexicon: Manually examine the emotion associations of the most frequent terms in your data. Remove entries from the lexicon that are not suitable (due to mismatch of sense, inappropriate human bias, etc.).

#19. Rescaling the Lexicon for One's Task: Depending on your specific use case, you may choose to re-scale the scores from 0 to 1, -1 to 1, 1 to 10, etc. Note that if using the lexicon entries as features in machine learning experiments, the scale (0 to 1 or -1 to 1) can make a difference—e.g. if the score is used as a weight for features.

#20 Metrics and Features Drawn from the Lexicon: For text analysis, one can calculate various metrics such as the percentage of emotion words (when the lexicons provides a list of words associated with a category) or average emotion intensity (for real-valued associations). When determining the scores, a further choice is how to handle words that are not in the lexicon. Two common approaches include: 1. Treat words that are not in the lexicon as neutral; 2. Ignore these words in the calculation of the scores. The latter approach does not make assumptions of neutrality, and is not impacted by the number of such out of lexicon words in a piece of text. See [Teodorescu and Mohammad \(2022\)](#) for a systematic

analysis of the impact of various lexicon features on the quality of emotion arcs generated with them.

#21. Creating Subsets of the Lexicon: Sometimes it is better to use a subset of the emotion lexicon, rather than the whole lexicon.

Removing Neutral Words: One can use the whole lexicon to calculate metrics such as average valence of the words in a text; however, one can also choose to disregard terms with close to 0 valence scores. when calculating the same metric. Removal of such neutral terms from the analysis will show greater variations in the average scores when comparing across different sets of data of interest or across time. For example, when looking at the average tweet happiness over time of day, using full or neutral-removed lexicon is expected to get roughly similar curves, but the neutral-removed lexicon will show a greater amplitude (divergence of scores from the peaks to troughs). ([Dodds et al., 2011](#)) describes this as turning up the magnifier knob in a microscope. Note, however, that just having larger score differences between the target and control does not mean that the emotion word usage is substantially different or significant; and conversely, just because the score difference for a metric is small in value does not mean that the differences in emotion word usages are not substantial. (More on this in #22).

Removing Low-Association Words: Use of low-association terms from a lexicon may not be beneficial for some downstream applications. These entries may also include a greater percentage of annotation errors. See [Teodorescu and Mohammad \(2022\)](#) for experiments on multiple datasets and multiple emotion dimensions that examine usefulness of removing low-association terms from a lexicon when generating emotion arcs.

Removing Highly Polysemous and Certain Domain Words: For some applications, it is beneficial to discard highly ambiguous words. Entries for highly ambiguous words are more likely to include emotion associations for a sense that is not common in one's data. As stated in #3, it is also recommended to remove entries not appropriate for the target domain; e.g., the word *harry* has a negative meaning, but it should not be used when analyzing text where a person has the name *Harry*.

#22. Inferences: When drawing inferences from texts using counts of emotion words:

- a. It is more appropriate to make claims about emotion word usage rather than emotions of the speakers. For example, *‘the use of anger words grew by 20%’* rather than *‘anger grew by 20%’*. A marked increase in anger words is likely an indication that anger increased, but there is no evidence that anger increased by 20%. Further, it is important to understand the emotion metrics and to interpret them accordingly. For example, many off-the-shelf tools provide a “sentiment score” for the input textual instances, without providing adequate details about what this score means. As discussed in #21, the scores themselves can have large or small values, and just knowing that the score difference between a target and control is large (or small) is not enough to draw meaningful inference. On the other hand, grounded metrics that tie the score to attributes such as percentage of positive words tend to be less open to misinterpretation.
- b. Comparative analysis is your friend. Often, emotion word counts on their own are not useful. For example, *‘the use of anger words grew by 20% when compared to [data from last year, data from a different person, etc.]’* is more useful than saying *‘on average, 5 anger words were used in every 100 words’*.
- c. Lexicon features (or any other automatically drawn features) are *not* well suited to draw meaningful emotional inferences from individual utterances. Human language and behaviour are highly variable and complex. However, with careful design, they can be useful to draw inferences about broad trends at an aggregate level (Teodorescu and Mohammad, 2022).
- d. Inferences drawn from large amounts of text are more reliable than those drawn from small amounts of text. Teodorescu and Mohammad (2022) show that this is the single most important feature in determining the fidelity of the predicted emotion trends with the true emotion trends, among a host of features they explored. For many emotion dimensions and dataset domains, it is advisable to determine aggregate emotion scores using at least 100 instances. For example, if there are at least 100 tweets per day about a product of interest, the average valence scores of all the words in the tweets every day is expected to produce a fairly accurate valence arc (x-axis is day, y-axis is average valence score for the corresponding day).

3 Limitations

This paper does not present a new NLP model or dataset. Thus, there are no corresponding limitations to discuss. However, the paper itself can be viewed as a document discussing limitations of existing approaches to do sentiment and emotion analysis using emotion lexica. The 22 best practises presented in the paper discuss approaches to engage with and counter these limitations.

While this document was a result of engaging a larger community through blog posts, talks, and discussions, we had relatively low access to developers of commercial sentiment analysis systems. Thus the list presented here may have missed some important considerations. We encourage readers and impacted stakeholders to challenge the assumptions latent in the document, and identify new ethical considerations not included here or not gaining adequate attention in the research community.

4 Concluding Remarks

Emotion lexicons are simple yet powerful tools to analyze text. However, use of the lexicons (even for tasks that it is suited for) can lead to inappropriate bias. Applying a lexicon to any new data should only be done after first investigating its suitability, and requires careful analysis to minimize unintentional harm. In this paper, we presented 22 best practises that include considerations that can help mitigate such unwanted outcomes, as well as strategies to make the best use of emotion lexicons towards drawing meaningful and accurate inferences. The best practises are organized as per a lexicon’s life cycle: A. Lexicon Design, B. Annotation, C. Entries in the Lexicon, and D. Applying the Lexicon. We also provide pointers to relevant literature to explore the best practises in more detail. It should be noted that these practises are not meant to be the final word, but rather jumping off points for further thought, discussion, and additional measures towards the responsible use of emotion lexicons.

Acknowledgments

Many thanks to Emiel van Miltenburg, Annika Schoene, Mallory Feldman, Tara Small, Roman Klinger, and Peter Turney for thoughtful comments and discussions.

References

- Mohamed Abdalla, Krishnapriya Vishnubhotla, and Saif M. Mohammad. 2023. What makes sentences semantically related: A textual relatedness dataset and empirical study. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Dubrovnik, Croatia. Association for Computational Linguistics.
- Herman Aguinis, N. Sharon Hill, and James R. Bailey. 2021. Best practices in data collection and preparation: Recommendations for reviewers, editors, and authors. *Organizational Research Methods*, 24(4):678–693.
- ARTICLE19. 2021. Emotional entanglement: China’s emotion recognition market and its implications for human rights. <https://www.article19.org/wp-content/uploads/2021/01/ER-Tech-China-Report.pdf>.
- Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Proceeding of the 7th International Conference on Language Resources and Evaluation*, volume 10 of *LREC '10*, pages 2200–2204.
- Lisa Feldman Barrett. 2017a. *How emotions are made: The secret life of the brain*. Houghton Mifflin Harcourt.
- Lisa Feldman Barrett. 2017b. The theory of constructed emotion: an active inference account of interoception and categorization. *Social cognitive and affective neuroscience*, 12(1):1–23.
- Lisa Feldman Barrett, Ralph Adolphs, Stacy Marsella, Aleix M Martinez, and Seth D Pollak. 2019. Emotional expressions reconsidered: Challenges to inferring emotion from human facial movements. *Psychological science in the public interest*, 20(1):1–68.
- Hans Baumgartner and Jan-Benedict E.M. Steenkamp. 2001. Response Styles in Marketing Research: A Cross-National Investigation. *Journal of Marketing Research*, 38(2):143–156.
- Margaret M Bradley and Peter J Lang. 1999. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical report, The Center for Research in Psychophysiology, University of Florida.
- Amy JC Cuddy, Susan T Fiske, and Peter Glick. 2008. Warmth and competence as universal dimensions of social perception: The stereotype content model and the bias map. *Advances in experimental social psychology*, 40:61–149.
- Herbert Aron David. 1963. *The method of paired comparisons*. Hafner Publishing Company, New York.
- Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013. Predicting depression via social media. In *Seventh international AAAI conference on weblogs and social media*, pages 128–137.
- Peter Sheridan Dodds, Kameron Decker Harris, Isabel M Kloumann, Catherine A Bliss, and Christopher M Danforth. 2011. Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. *PLoS one*, 6(12):e26752.
- Julie McDonough Dolmaya. 2011. The ethics of crowdsourcing. *Linguistica Antverpiensia, New Series—Themes in Translation Studies*, (10).
- Johannes C Eichstaedt, Hansen Andrew Schwartz, Margaret L Kern, Gregory Park, Darwin R Labarthe, Raina M Merchant, Sneha Jha, Megha Agrawal, Lukasz A Dziurzynski, Maarten Sap, et al. 2015. Psychological language on Twitter predicts county-level heart disease mortality. *Psychological science*, 26(2):159–169.
- Paul Ekman. 1992. Are there basic emotions? *Psychological Review*, 99(3):550–553.
- Paul Ed Ekman and Richard J Davidson. 1994. *The nature of emotion: Fundamental questions*. Oxford University Press.
- T. N. Flynn and A. A. J. Marley. 2014. Best-worst scaling: theory and methods. In Stephane Hess and Andrew Daly, editors, *Handbook of Choice Modelling*, pages 178–201. Edward Elgar Publishing.
- Karën Fort et al. 2011. Amazon Mechanical Turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413–420.
- Timnit Gebru, Jamie H. Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, H. Wallach, Hal Daumé, and Kate Crawford. 2018. Datasheets for datasets. In *Proceedings of the conference on Fairness, Accountability, and Transparency in Machine Learning*, Stockholm, Sweden.
- Sharath Chandra Guntuku, Rachele Schneider, Arthur Pelullo, Jami Young, Vivien Wong, Lyle Ungar, Daniel Polsky, Kevin G Volpp, and Raina Merchant. 2019. Studying expressions of loneliness in individuals using Twitter: an observational study. *BMJ open*, 9(11):e030355.
- Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.
- Lilly C Irani and M Six Silberman. 2013. Turkopticon: Interrupting worker invisibility in Amazon Mechanical Turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 611–620.

- David Jurgens. 2013. Embracing ambiguity: A comparison of annotation methodologies for crowdsourcing word sense labels. In *NAACL*.
- David Jurgens, Saif M. Mohammad, Peter Turney, and Keith Holyoak. 2012. [Semeval-2012 task 2: Measuring degrees of relational similarity](#). In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval)*, pages 356–364, Montréal, Canada.
- Zahi N Karam, Emily Mower Provost, Satinder Singh, Jennifer Montgomery, Christopher Archer, Gloria Harrington, and Melvin G Mcinnis. 2014. Ecologically valid long-term mood monitoring of individuals with bipolar disorder using speech. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4858–4862. IEEE.
- Svetlana Kiritchenko, Will Hipson, Robert Coplan, and Saif M. Mohammad. 2020. [SOLO: A corpus of tweets for examining the state of being alone](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1567–1577, Marseille, France.
- Svetlana Kiritchenko and Saif Mohammad. 2017. Best-Worst Scaling More Reliable than Rating Scales: A Case Study on Sentiment Intensity Annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470.
- Svetlana Kiritchenko and Saif M. Mohammad. 2016. Capturing reliable fine-grained sentiment associations by crowdsourcing and best–worst scaling. In *Proceedings of The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, San Diego, California.
- Richard S Lazarus. 1991. Progress on a cognitive-motivational-relational theory of emotion. *American psychologist*, 46(8):819.
- Jordan J. Louviere. 1991. Best-worst scaling: A model for the largest difference judgments. Working Paper.
- Sean MacAvaney, Anjali Mittu, Glen Coppersmith, Jeff Leintz, and Philip Resnik. 2021. [Community-level research on suicidality prediction in a secure environment: Overview of the CLPsych 2021 shared task](#). In *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology: Improving Access*, pages 70–80, Online. Association for Computational Linguistics.
- Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229.
- Saif Mohammad. 2022a. [Ethics sheets for AI tasks](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8368–8379, Dublin, Ireland. Association for Computational Linguistics.
- Saif M. Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.
- Saif M. Mohammad. 2020. [Practical and ethical considerations in the effective use of emotion and sentiment lexicons](#).
- Saif M. Mohammad. 2022b. Ethics sheet for automatic emotion recognition and sentiment analysis. *Computational Linguistics*, 48(2):239–278.
- Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, LA, California.
- Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.
- Finn Årup Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC Workshop on ‘Making Sense of Microposts’: Big things come in small packages*, pages 93–98, Heraklion, Crete.
- Bryan Orme. 2009. Maxdiff analysis: Simple counting, individual-level logit, and HB. Sawtooth Software, Inc.
- Charles Egerton Osgood, George J Suci, and Percy H Tannenbaum. 1957. *The measurement of meaning*. 47. University of Illinois press.
- James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.
- Stanley Presser and Howard Schuman. 1996. *Questions and Answers in Attitude Surveys: Experiments on Question Form, Wording, and Context*. SAGE Publications, Inc.
- Vikas C Raykar and Shipeng Yu. 2012. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *The Journal of Machine Learning Research*, 13(1):491–518.
- Philip Resnik, William Armstrong, Leonardo Claudino, Thang Nguyen, Viet-An Nguyen, and Jordan Boyd-Graber. 2015. [Beyond LDA: Exploring supervised topic modeling for depression-related language in Twitter](#). In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology*:

- From Linguistic Signal to Clinical Reality*, pages 99–107, Denver, Colorado.
- James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.
- James A Russell. 2003. Core affect and the psychological construction of emotion. *Psychological review*, 110(1):145.
- James A Russell and Albert Mehrabian. 1977. Evidence for a three-factor theory of emotions. *Journal of research in Personality*, 11(3):273–294.
- Klaus R Scherer. 1999. *Appraisal theory*. John Wiley & Sons Ltd.
- Hansen Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Richard E Lucas, Megha Agrawal, Gregory J Park, Shrinidhi K Lakshminanth, Sneha Jha, Martin EP Seligman, et al. 2013. Characterizing geographic variation in well-being using tweets. In *Seventh International AAAI Conference on Weblogs and Social Media*, pages 583–591.
- Andrew D Selbst, Danah Boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and abstraction in sociotechnical systems. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 59–68.
- Susan Standing and Craig Standing. 2018. The ethical use of crowdsourcing. *Business Ethics: A European Review*, 27(1):72–80.
- Luke Stark. 2018. Algorithmic psychometrics and the scalable subject. *Social Studies of Science*, 48(2):204–231.
- Philip Stone, Dexter Dunphy, Marshall Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press.
- Carlo Strapparava, Alessandro Valitutti, et al. 2004. Wordnet affect: an affective extension of wordnet. In *Lrec*, volume 4, page 40. Lisbon.
- Daniela Teodorescu and Saif M. Mohammad. 2022. [Evaluating automatically generated emotion arcs: A case for simple methods using emotion lexicons](#). arXiv.
- Louis L. Thurstone. 1927. A law of comparative judgment. *Psychological review*, 34(4):273.
- Krishnapriya Vishnubhotla and Saif M. Mohammad. 2022. Tweet emotion dynamics: Emotion word usage in tweets from us and canada. In *Proceedings of the Thirteenth International Conference on Language Resources and Evaluation (LREC 2022)*, Marseille, France.
- Jane Wakefield. 2021. AI emotion-detection software tested on Uyghurs. BBC. <https://www.bbc.com/news/technology-57101248>.
- Anna Wierzbicka. 1999. *Emotions across languages and cultures: Diversity and universals*. Cambridge university press.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 347–354.
- Lieve Van Woensel and Nissy Nevil. 2019. What if your emotions were tracked to spy on you? European Parliamentary Research Service, PE 634.415. [https://www.europarl.europa.eu/RegData/etudes/ATAG/2019/634415/EPRS_ATA\(2019\)634415_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/ATAG/2019/634415/EPRS_ATA(2019)634415_EN.pdf).

The Role of Semantic Parsing in Understanding Procedural Text

Hossein Rajaby Faghihi¹, Parisa Kordjamshidi¹, Choh Man Teng², and James Allen²

¹ Michigan State University, ² Florida Institute for Human and Machine Cognition
{rajabyfa, kordjams}@msu.edu, {jallen, cmteng}@ihmc.org

Abstract

In this paper, we investigate whether symbolic semantic representations, extracted from deep semantic parsers, can help reasoning over the states of involved entities in a procedural text. We consider a deep semantic parser (TRIPS) and semantic role labeling as two sources of semantic parsing knowledge. First, we propose PROPOLIS, a symbolic parsing-based procedural reasoning framework. Second, we integrate semantic parsing information into state-of-the-art neural models to conduct procedural reasoning. Our experiments indicate that explicitly incorporating such semantic knowledge improves procedural understanding. This paper presents new metrics for evaluating procedural reasoning tasks that clarify the challenges and identify differences among neural, symbolic, and integrated models.

1 Introduction

Procedural reasoning is the ability to track entities and understand their evolution given a sequence of actions (Tandon et al., 2020). This kind of reasoning is crucial in understanding recipes (Bosselut et al., 2018; Yagcioglu et al., 2018), manuals and tutorials (Tandon et al., 2020; Wu et al., 2022), cybersecurity text (Pal et al., 2021), natural events (Tandon et al., 2020), and even stories (Storks et al., 2021). An example of a procedural text in the natural event domain, its entities of interest, and their state changes are shown in Figure 1.

Inferring actions and their impact on entities involved in a procedural text can be challenging in various aspects. **First**, there are dependencies between steps to be considered in predicting a plausible action set. For instance, an entity destroyed at step t of the process cannot be moved again at step $t + 1$. **Second**, some sentences contain ambiguous local signals by including multiple action verbs. For example, "The oxygen is consumed in the process of forming carbon dioxide.", where the oxygen

Process	Participants			
	plant	animal	bone	oil
Before the process begins	?	?	-	-
1. Plants and animals die in a watery environment	watery environment	watery environment	-	-
2. Over time, sediments build over	sediment	sediment	-	-
3. The body decomposes	sediment	-	sediment	-
4. Gradually buried material becomes oil	-	-	-	sediment

Figure 1: An example of procedural text and its annotation (location of objects). ‘-’ means the entity does not exist; ‘?’ means the entity’s location is unknown.

is being destroyed, and the carbon dioxide is being created. **Third**, the sentences are incomplete in some steps. For instance, a step of the process might only indicate "is buried in mud", which cannot be understood without context. **Fourth**, finding the properties of some entities may require reasoning over both the global context and local relations. For instance, in the sentences "1. Magma rises to the surface. 2. Magma cools to form lava", the location of ‘Lava’ after step 2 should be inferred from the prior location of Magma, which is indicated in its previous step. **Fifth**, common sense is required to understand some consequences. For example, in Figure 1, step 3, one should use common sense to realize that ‘decomposing body’ would expose the ‘bones’, which will be left behind in the ‘sediment’. **Sixth**, understanding some relations requires an advanced co-reference resolution. In Figure 1, step 4, a complex co-reference resolution is required to understand that the ‘buried material’ refers to both ‘plants and animals bones’ and that they are transforming into the ‘oil’.

Except for the common-sense (Zhang et al., 2021) and the ability to make consistent global decisions actions (Gupta and Durrett, 2019), the other challenges might have only been indirectly tackled in the recent research (Huang et al., 2021; Faghihi and Kordjamshidi, 2021), but have neither been addressed explicitly nor properly evalu-

ated to measure their success on resolving these challenges. In this paper, we evaluate whether semantic parsers can alleviate some of these challenges. Semantic parsers provide semantic frames that identify predicates and their arguments in a sentence. For instance, in the sentence ‘Move bag to the yard’, “Move” is the predicate, “bag” and “the yard” are the arguments with types “affected”¹ and “location” respectively. Such semantic information can help disambiguate multi-verb local connections between predicates and arguments (Huang et al., 2021). They can also provide meaningful local relations, making it easier to connect global information to infer entities’ states. For instance, in the same sentence, “Magma cools to form lava”, “Magma” is noted as the ‘affected’ and ‘lava’ is the result of the predicate ‘form’. This makes it easier to infer that the location of ‘lava’ should match the last location of ‘magma’.

For our study, we consider both the classic semantic role labeling (SRL)², based on (Shi and Lin, 2019), which is a relatively shallow semantic parsing model, as well as the deep semantic parser TRIPS³ (Ferguson et al., 1998; Allen and Teng, 2017). To investigate the effect of semantic parsing on procedural reasoning, we analyze its effect as a standalone symbolic model as well as its integration in a neuro-symbolic model that combines semantic parsing with state-of-the-art neural models to solve the procedural reasoning task.

First, we design a set of heuristics to extract a symbolic abstraction from the TRIPS parser, called PROPOLIS. We use this baseline to further showcase the effectiveness of semantic parsing information in solving the procedural task. Next, we integrate the semantic parsers with two well-established procedural reasoning neural backbones, namely NCET (Gupta and Durrett, 2019) and TSLM (Faghihi and Kordjamshidi, 2021) (and its extension CGLI (Ma et al., 2022)), through encoding the semantic relations as a graph attention neural network (GAT) (Shi et al., 2020).

For our experiments, we use Propara dataset (Tandon et al., 2020) that introduces the procedural reasoning task over natural events that are described in English. We realized the existing evaluation metrics of this dataset do not reflect the actual performance of the models and

fail to identify the challenges and shortcomings of the models. Consequently, we propose new evaluation criteria to shed light on the differences between the models, even when they perform similarly based on the prior metrics.

In summary, our contributions are (1) Proposing a symbolic model (Propolis) to solve the procedural reasoning task based on semantic parsing, (2) Proposing a set of new evaluation metrics which can identify the strength and weaknesses of the models, and (3) Showcase the benefits of integrating semantic parsing into the neural models. The code and models proposed in this work are all available in GitHub⁴.

2 Related Research

Procedural text understanding has been investigated in many benchmarks such as ScoNe (Long et al., 2016), bAbI (Weston et al., 2015), and ProcessBank (Berant et al., 2014). Recent research has focused on procedural reasoning as tracking entities throughout a procedural text. Datasets such as Propara (Tandon et al., 2020), Recipes (Bosselut et al., 2018), Procedural Cyber-Security text (Pal et al., 2021), and OpenPI (Tandon et al., 2020) are in the same direction. Procedural reasoning can also be influential in addressing causal reasoning (WIQA) (Tandon et al., 2019), story understanding (Trip) (Storks et al., 2021), and abstractive multi-modal question answering (RecipeQA) (Yagcioglu et al., 2018).

This paper primarily focuses on tracking entities’ states and properties throughout a procedural text. Recent research has addressed this problem by predicting actions and properties on local context (Prolocal) (Dalvi et al., 2018), autoregressive global predictions based on distance vectors (Proglobal) (Dalvi et al., 2018), integrating structural common-sense knowledge built over VerbNet (ProStruct) (Tandon et al., 2018), building dynamic knowledge graphs over entities (KG-MRC) (Das et al., 2018), explicitly encoding the model to explain dependencies between actions (XPAD) (Dalvi et al., 2019), formulating local predictions and global sequential information flow and sequential constraints (NCET) (Gupta and Durrett, 2019), formulating the task in a QA setting (DynaPro, TSLM) (Amini et al., 2020; Faghihi and Kordjamshidi, 2021), inte-

¹referred to as ‘Patient’ in some other parsing formalisms.

²<https://demo.allennlp.org/semantic-role-labeling>

³<http://trips.ihmc.us/parser/cgi/parse>

⁴<https://github.com/HLR/ProceduralSemanticParsing>

grating common-sense knowledge from ConceptNet (KOALA) (Zhang et al., 2021), utilizing large generative language models (LEMON) (Shi et al., 2022), or using both the question answering setting and sequential structural constraints at the same time (CGLI) (Ma et al., 2022). All the models mentioned above investigate different neural architectures to tackle the task, while we are more interested in augmenting them with additional knowledge from semantic parsers. Recent research has also investigated the integration of semantic role labeling into the procedural reasoning task (REAL) (Huang et al., 2021), which is very close to our goal in this paper. However, in this work, we propose and investigate a variety of combinations, a deeper semantic representation (TRIPS) and named relations, in addition to a symbolic approach for solving the procedural reasoning task solely based on semantic parsing.

3 Technical Approach

Problem definition The procedural reasoning task can be formally defined by a procedural text including m steps, $S = \{s^1, s^2, \dots, s^m\}$, a set of entities $E = \{e_1, e_2, \dots, e_n\}$, where n is the number of entities, and a set of properties. Specifically, in the Propara dataset, the property of interest is only the location of the entities $P_L = \{p_{L_1}^0, p_{L_1}^1, \dots, p_{L_n}^m\}$, where $p_{L_i}^t$ denotes the j th entity at step t . In Propara, the location prediction starts at step 0, which indicates the entity’s location before the process begins. The location of an entity can either be known (represented by a string) or unknown (represented by "?"). Similar to prior research (Tandon et al., 2020), the location property is used to infer a set of actions $A = \{a_1^1, a_2^1, \dots, a_n^m\}$, where a_t^j denotes the action type applied to entity j at step t . Following the prior research (Dalvi et al., 2018), we extract all the noun phrases from the sentences and only consider those as location candidates.

We investigate two different modeling approaches to solve this problem. First, we use a symbolic and parsing-based model, and second, we integrate semantic parsing with neural models. We use two different sources for semantic extraction: SRL and TRIPS. In general, SRL is coarse-grained and shallow compared to TRIPS. The connections in TRIPS are not limited to the pairwise connections between predicates and arguments but are extended to the semantic connections between any two words. Since TRIPS relies on a general pur-

pose ontology, it also augments the arguments and predicates with additional information about a set of possible features (mobility, container, negation) and mapping of the words to hierarchical ontology classes (i.e., mapping “water” to “beverage”). SRL is centered around the semantic frames of the verbs (predicates) and identifies each predicate’s main and adjunct (mainly time and location) arguments in the sentence. Figure 2 and 3 show examples of the SRL and TRIPS parses, respectively.

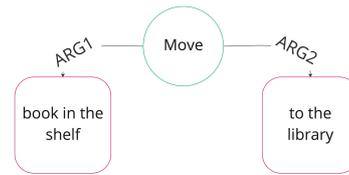


Figure 2: The SRL annotation for the sentence “Move the book in the shelf to the library”.

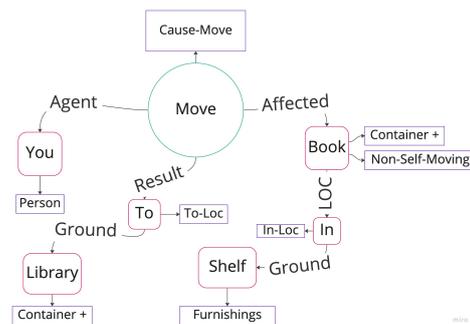


Figure 3: The TRIPS parse for the sentence “Move the book in the shelf to the library”.

The symbolic model only uses the TRIPS parser as it provides more extended extractions and meaningful relations, while both SRL and TRIPS are used for integration with the neural baselines.

3.1 PROPOLIS: Symbolic Procedural Reasoning

We propose the PROPOLIS model, which solves the procedural reasoning task merely by symbolic semantic parsing. PROPOLIS operates on the TRIPS parser in three steps. First, it makes an abstraction over the original parse to summarize the information in the graph and include a smaller set of actions and changes in objects and their locations. Second, it uses a set of rules to transform the abstracted parses into clear actions and identifies the affected objects by the actions, using the semantic roles, while extracting an ending location or starting location. Lastly, it performs global reasoning to connect the local decisions and produce

a consistent sequential set of actions/locations for each entity of interest. More details about the steps are also available in Appendix B.

3.1.1 Graph Abstraction

The original TRIPS parse includes many concepts and edges that do not directly affect entities' location or existence. Therefore, we make a more concise graph abstraction to facilitate processing the entities, actions, and locations. To obtain a more informative abstraction, firstly, the relevant classes of the TRIPS ontology are mapped to action classes defined in the Propara dataset (Create, move, or destroy). For instance, the verb 'flow' is first mapped to the 'fluidic motion' class in the TRIPS ontology, which is a child of the 'motion' class, and the 'motion' class is mapped to the 'move' action in the Propara dataset. This will help distinguish the predicates that signal a change in the location or existence of objects. Second, the important arguments are identified in the parse, and the locations are extracted. The graph is decomposed to include a set of events with their arguments. Each event may contain different roles such as "agent", "affected", "result", "to_location", "from_location", or other roles required by its semantic frame.

3.1.2 Rule-based Local Decisions

We use a set of heuristic rules to map the abstracted graph onto actual actions over the entities of interest. The rules are written according to the semantic frames and the type of predicates and arguments in each parse. For instance, if a semantic frame is mapped to 'Move' and has both the 'agent' and 'affected' arguments, then the 'affected' argument specifies the object being moved. The same frame with only an 'agent' argument indicates a move for the object in the 'agent' role. Table 1 shows the most frequent templates we used to transform the local parses into actual decisions over the entities.

3.1.3 Global Reasoning

The two first steps are merely based on the local sentence-level actions of each step. We need additional global reasoning over the whole procedure to predict the outputs. Global reasoning ensures that local decisions form a valid global sequence of actions for a given entity. For instance, if an entity is predicted to be destroyed at step 2 and moved at step 3, we consider the 'destroyed' action a wrong local decision since a destroyed object cannot move later in the process. The graph also contains pas-

sive indications of object location in phrases such as "the book on the shelf" or even indications of prior locations in terms of a 'from_location' argument. These phrases do not generate actions but provide information that should be used in previous steps. For example, if step t has a local prediction 'Move' for entity e with no target location and step $t + 1$ has a 'from_location' for entity e , then the 'from_location' should be used as the target location of the 'Move' action in the previous step.

3.2 Integration with Neural Models

Here, we investigate whether explicitly incorporating semantic parsers with neural models can help better understand the procedural text. We choose two of the recently proposed and most commonly used backbone architectures for procedural reasoning tasks, namely NCET (Gupta and Durrett, 2019) and TSLM (Faghihi and Kordjamshidi, 2021) (and its extension CGLI (Ma et al., 2022)). Similar to (Huang et al., 2021), we rely on a graph attention network (GAT) to integrate the information from the semantic parsers into the neural baselines.

Following (Huang et al., 2021), the nodes in this graph are either (1) predicates in the semantic frames, (2) mentions of entities of interest (Exact match or Co-reference), or (3) noun phrases in the sentence. An edge in the SRL graph exists between two nodes if they have a (predicate, argument) connection or they are both parts of the same verb semantic frame (argument to argument) (Huang et al., 2021). It is relatively straightforward to build a semantic graph with the TRIPS parser because it outputs the parse as a graph.

An edge is created between any pairs of nodes (phrases) in the graph if any subsets of these two phrases are connected in the original parse. The edge types are preserved. Since not all the nodes in the original parse are present in the new simplified graph, we may lose some key connections. To fix this, if two nodes (phrases) are not connected in the new graph but have been connected in the original one, we find the shortest path between them in the original parse and connect them with a new edge with the type being the concatenation of all the edge types in the path. Lastly, nodes are connected across sentences based on either an exact match or co-reference resolution.

Both NCET and TSLM models are trained based on Cross Entropy to compute the loss for both actions and locations. The final loss of the model

Main Predicate	Roles	Decisions
Move	Affected, Agent	The “Affected” is being moved.
Move	Agent	The “Agent” is being moved.
Destroy	Affected	The “Affected” is being destroyed
Create	Affected_Result, Affected	The “Affected_Result” is being created
Create	Affected	The “Affected” is being created
Change	Affected, Res	The “Affected” is being destroyed, and the “Res” is being created

Table 1: The rules used to evaluate the effect of actions on various roles of the semantic frame

is calculated by $L_{total} = L_{action} + \lambda * L_{location}$, where λ is a balancing hyper-parameter.

3.2.1 Integration with NCET as Backbone

The NCET model uses a language model to encode the context of the procedure and compute representations for mentions of entities, verbs, and locations. These representations are used in two sub-modules for predicting the actions and the locations. To integrate the semantic parsers with the NCET architecture, we use the output of the language model to initialize the semantic graph representations. Then multiple layers of graph attention network (based on TransformerConv (Shi et al., 2020)) are applied to encode the graph structure. We combine the updated graph representations with the initial mention representations. These combined representations are later used in subsequent prediction modules.

More formally, we start by using a language model to encode the context of the process $h' = LM(S)$, where S is the procedure, and h' is the embedding output from the language model. The representations are further encoded by a BiLSTM $h = BiLSTM(h')$.

Graph Attention Network Since each node in the semantic graph corresponds to a subset of tokens in the original paragraph, we use the mean average of these tokens’ representation to initialize the nodes’ embedding denoted as v_i^0 . If the graph contains edge types, the edges between each two nodes i and j are denoted by e_{ij} and is represented by the average token embedding through the same LM model used for encoding the story, $e_{ij} = Mean(LM(e_{ij}^{text}))$. Lastly, we use C layers of TransformerConv (Shi et al., 2020) to encode the graph structure. More details on the graph encoder is available in Appendix C.

Representing Mentions To integrate the semantic parses with the baseline model, we use both representations obtained from the language model and the graph encoder to represent entities, verbs, and

Tag	Description
O_D	Entity does not exist after getting destroyed
O_C	Entity does not exist before getting created
E	Entity exists and does not change
C	Entity is created
D	Entity is destroyed

Table 2: The list of output tags/actions

locations in the process. Mention representations are denoted by $r_t^m = [M(h_t^m); M(h_{g_t}^m)]$, where t is one step of the process, h_t^m is the average representation of tokens in the story corresponding to the mention m in step t , $h_{g_t}^m$ is the average embedding of nodes corresponding to mention m in step t , and the function M replaces the representations with zero if there is no mention of m in step t .

Location Prediction We first encode the pairwise representation of an entity e and location candidate lc at each step t , denoted by $\mathbf{x}_t^{(e,lc)} = [\mathbf{r}_t^e; \mathbf{r}_t^{lc}]$. Next, we use an LSTM to encode the step-wise flow of the pair representation to get $\bar{h}_t^{(e,lc)} = LSTM([\mathbf{x}_t^{(e,lc)}])$. Finally, the probability of each location candidate lc to be the location of entity e at step t is calculated by a *softmax* over the potential candidates, $p^{(e,lc)_t} = Softmax(W_{loc}^t \bar{h}_t^{(e,lc)})$, where W_{loc}^t is the learning parameters of a single multi-layer perceptron.

Action Prediction To predict the action for entity e at step t , we create a new representation for the entity based on its mention and the sentence verbs, denoted by $x_t^e = [r_t^e; Mean_{v \in np(e_t)}(r_t^v)]$, where $np(e_t)$ is the set of verbs whose corresponding node in the graph has a path to any of the nodes representing entity e in step t . The final representations are then produced using a BiLSTM over the steps, $h_t^e = LSTM([x_t^e])$. Lastly, a neural CRF layer is used to consider the sequential structure of the actions by learning transition scores during the training of the model (Gupta and Durrett, 2019). The set of possible actions is shown in Table 2.

	Local	Global Loc	Global Ent		Global Loc and Ent	Ambiguous
	Both	Both	Actions	Locations	Both	Actions
Train	885	367	438	340	114	593
Dev	116	44	66	3	9	76
Tests	105	61	98	71	18	110

Table 3: The number of decisions per category of evaluation with the new decision-level metric. ‘‘Both’’ refers to both location and action decisions and is used since the number of those decisions is the same in most cases. The number of decisions in the ‘Global Ent’ case can be different for the actions and the locations because this category also considers ‘destroy’ events that have no corresponding locations.

3.2.2 Integration with TSLM as Backbone

The TSLM (Faghihi and Kordjamshidi, 2021) model reformulates the procedural reasoning task as a question-answering problem. The model simply asks the question, ‘Where is entity e ?’ at each step of the process. To include the context of the whole process when asking the same question at different steps, TSLM further introduces a time-aware language model that can encode additional information about the time of events. Given the new encoding, each step of the process is mapped to either past, present, or future. TSLM uses the answer to the question at each step to form a sequence of decisions over the location of entity e . To integrate the semantic graph with this model, we first extend the graph by adding a question node. The graph is then initialized using the time-aware language model. The encoded representations of the graph, after applying multiple layers of GAT, are combined with the original token representations and used for extracting the answer to the question.

Initial Representation For each entity e and timestamp t , the string ‘‘where is e ? s_1 $\langle/s\rangle$ s_2 $\langle/s\rangle$... s_m $\langle/s\rangle$ ’’ is fed into the time-aware language model. Accordingly, the tokens’ representations for timestamp t are $(h_e)_t^i = LM(S, t)$.

Graph Attention Module Inspired by (Zheng and Kordjamshidi, 2020), we add new nodes to the semantic graph to represent the question and each step of the process. We connect the question node to any node in the graph representing the entity of interest e , and each step node to all the tokens in their corresponding sentence. An example of the QA-based graph can be found in Appendix D. All the node embeddings are initialized by the average embedding of their corresponding tokens in the procedure. We use C layers of graph attention network (TransformerConv), similar to Section 3.2.1, to encode the graph structure.

Location Prediction For predicting the locations of entities, that is, the answer to the question, we

predict the answer among the set of location candidates. This is different from the common practice of predicting start/end tokens. We represent each location candidate by combining representations from both the graph and the time-aware language model, denoted by $r_t^{lc} = [(h_e)_t^{lc}, (h_{gl}^e)_t^{lc}]$, where $(h_{gl}^e)_t^{lc}$ is the representation of the lc from the last layer of the GAT. The answer is then selected by calculating a *softmax* over the set of location candidates, $p_t^{lc} = \text{Softmax}(W^{location} r_t^{lc})$.

Action Prediction Similar to CGLI (Ma et al., 2022) model, we explicitly predict the actions of entities alongside the locations. First, the model extracts each timestamp’s ‘‘CLS’’ tokens and builds sequential pairs of (CLS_t^e, CLS_{t+1}^e) . Then, it produces a change representation vector for each of these pairs, denoted by $r_t^e = F(CLS_t^e; CLS_{t+1}^e)$. Lastly, the sequence of $[r_t^e]$ logits is passed through the same neural CRF layer used by the NCET model, introduced in Section 3.2.1, to generate the final probability of actions.

4 Evaluation

We use three evaluation metrics to analyze the performance of the symbolic, sub-symbolic, and neural baselines. The first metric is sentence-level and proposed in (Dalvi et al., 2018). The second metric is a document-level evaluation proposed by (Tandon et al., 2018). Both of these metrics evaluate higher-level procedural concepts that can be inferred from the predictions of the model rather than the raw decisions. These metrics give more importance to the actions compared to the location decisions. Although they can successfully evaluate some aspects of the models, they fail to measure the research progress in addressing the challenges of the procedural reasoning task. We extend these evaluations with a new decision-level evaluation metric that considers almost all model decisions with a similar weight and evaluates the models based on the difficulty of the reasoning process.

4.1 Propara Evaluation Metrics

With the **sentence-level** metrics, the predictions are evaluated in three different categories. **(Cat1)** evaluates whether an entity e has been created (destroyed/moved) during the process. **(Cat2)** evaluates when an entity e is created (destroyed/moved). **(Cat3)** evaluates where e is created (destroyed/moved).

With the **document-level** metrics, we evaluate the Inputs, Outputs, Conversion, and Moves separately and average over the F1 score of these four criteria to output one F1 score as the final metric. Here, **Inputs** are entities that did exist before the process started and are destroyed during. **Outputs** are entities that did not exist before the process but created during it. **Conversions** evaluates which entities converted to another entity. Lastly, **Moves** evaluates which entities have been moved from one place to another.

4.2 Extended Evaluation

Both sets of existing evaluation metrics of the Propara dataset do not directly evaluate the predictions of the model but rather evaluate higher-level procedural concepts which can be inferred from the sets of decisions (i.e. an entity being input/output). Given their evaluation criteria, one model may surpass another in the number of correct decisions but still obtain a lower performance.

Therefore, we propose a new evaluation metric (**decision-level**) that directly evaluates the models' decisions. This evaluation metric is designed to consider the difficulty of the reasoning process and help better identify the core challenges of the task. We divide the set of decisions into five categories based on the presence of the entity e and the location l at each step t . We denote any mention of e by m^e , any mention of l by m_l , the action for entity e at step t by tag_t^e , and the text of the current step by S_t . The following specifies the five categories and how a decision falls under them.

Local Decision: A decision where (1) $m^e \in S_t$, (2) $m^l \in S_t$, and (3) $tag_t^e \in \{Move, Create\}$

Global Location Decision: A decision where (1) $m^e \in S_t$, (2) $m^l \notin S_t$, and (3) $tag_t^e \in \{Move, Create\}$

Global Entity Decision: A decision where (1) $m^e \notin S_t$, (2) $m^l \in S_t$ or $l = \text{" - "}$, and (3) $tag_t^e \in \{Move, Create, Destroy\}$

Global Entity and Location Decision: A decision where (1) $m^e \notin S_t$, (2) $m^l \notin S_t$, and (3)

$tag_t^e \in \{Move, Create\}$

Ambiguous Local Action: A decision where (1) $m^e \in S_t$ and (2) S_t contains multiple action verbs.

Table 3 shows the detailed statistics of the number of decisions falling under each of these five categories for the Propara dataset. Evaluating the performance of models given the new decision-level metric will clarify the lower-level challenges in the reasoning over states and locations of entities simultaneously. Getting accurate predictions in any of these categories of decisions requires the models to have different reasoning capabilities.

The local decisions mostly require a sentence-level understanding of the action and its consequences. The global location decisions require reasoning over the current step and the ability to connect the local information to the global context. The predictions for the category of the global entity mostly require reasoning over complex co-references (we have already considered simple co-references such as pronouns as mentions of the entity) or the ability to recover missing pronouns in a sentence such as "Gradually mud piles over (them)". The global entity and location decisions are the most challenging cases, which require reasoning over local and global contexts, complex co-reference resolution, and handling of missing pronouns. The ambiguous decisions mainly require local disambiguation of (entity, role, predicate) connections when multiple predicates are present in the sentence. Moreover, common sense is required for a subset of all the decision categories.

5 Experiments

Here, we summarize the performance of strong baselines compared with the symbolic (PROPOLIS) and integrated models. The implementation details of the models are available in Appendix A. Table 4 shows the performance of models in the two conventional metrics of the Propara dataset, and Table 5 shows the performance of models based on the decision-level metric. We summarize our findings in a set of question-answer pairs.

Q1. Can semantic parsing alone solve the problem reasonably? Based on Table 4, the PROPOLIS model outperforms many of the neural baselines (document-level F1-score of row#4 compared to rows #1 to #3), showing that deep semantic parsing can provide a general solution for the procedural reasoning task to some extent without the need for training data. This model performs rel-

#Row	Models	Sentence-level evaluation					Document-level evaluation		
		Cat1	Cat2	Cat3	Macro-avg	Micro-avg	Precision	Recall	F1
1	ProLocal	62.7	30.5	10.4	34.5	34.0	77.4	22.9	35.3
2	ProGlobal	63	36.4	35.9	45.1	45.4	46.7	52.9	49.4
3	KG-MRC	62.9	40	38.2	47	46.6	64.5	50.7	56.8
4	PROPOLIS(ours)	69.9	37.71	5.6	37.74	36.67	70.9	50.0	58.7
5	NCET (re-implemented)	75.54	45.46	41.6	54.2	54.38	68.4	63.6	66
6	REAL(re-implemented)*	78.9	48.31	41.62	56.29	56.35	67.3	64.9	66.1
7	NCET + SRL(ours)	77.1	46.35	42	55.16	55.32	67.8	65.2	66.5
8	NCET + TRIPS(ours)	77.1	48.12	43.36	56.19	56.32	72.5	65.4	68.8
9	NCET + TRIPS(Edge)(ours)	75.68	47.6	45.71	56.33	56.37	69.9	65.5	67.6
10	NCET + PROPOLIS(ours) ⁺	78.54	48.69	44.26	57.16	57.31	74.6	65.8	69.9
11	DynaPro	72.4	49.3	44.5	55.4	55.5	75.2	58	65.5
12	KOALA	78.5	53.3	41.3	57.7	57.5	77.7	64.4	70.4
13	TSLM	78.81	56.8	40.9	58.83	58.37	68.4	68.9	68.6
14	CGLI	80.3	60.5	48.3	63.0	62.7	74.9	70	72.4
15	CGLI + TRIPS (ours)	80.62	58.94	49.08	62.88	62.68	74.5	68.5	71.4

Table 4: The table of results based on sentence-level and document-level evaluation of the Propara Dataset. * Since the code for the REAL model is not available, we have re-implemented the architecture based on the guidelines of the paper and the communications. ⁺ The graph is first abstracted using the PROPOLIS graph abstraction phase and then used instead of the Trips parse as input to the model.

Model	Local			Global Loc			Global Ent			Global Loc and Ent			Amb ⁺
	A	L	Both	A	L	Both	A	L	Both	A	L	Both	A
KOALA	74.3	65.7	59.0	86.9	24.6	22.9	1.0	7.0	0.0	5.6	11.1	0	73.63
PROPOLIS	55.2	19.0	19.0	63.9	1.6	1.6	0.0	9.9	0.0	0.0	0.0	0.0	52.7
NCET	69.5	62.8	60.0	70.5	36.1	29.5	3.1	5.6	0.0	0.0	0.0	0.0	57.2
NCET + SRL	68.6	65.7	61.9	77.0	36.1	31.1	10.2	5.6	0.0	5.5	5.5	0.0	62.7
NCET + TRIPS	71.4	67.6	63.8	75.4	42.6	36.1	10.2	9.9	2.8	5.5	11.1	0.0	63.6
NCET + PROPOLIS	71.4	64.8	61.9	83.6	36.1	34.4	3.1	7.0	0.0	5.5	5.5	0.0	70.9
CGLI	65.7	62.9	54.3	75.4	59.0	50.8	19.4	19.7	11.3	22.2	27.8	11.1	70.0
CGLI + TRIPS	75.2	70.5	61.9	80.3	60.6	52.2	17.3	22.5	12.7	27.8	27.8	16.7	74.5

Table 5: The results of the models on the new extended evaluation metric (decision-level) in terms of accuracy (%). ‘A’ means the action is correct, ‘L’ means the location is correct, and ‘Both’ means both the action and location are correct. ⁺ Local ambiguous cases.

atively well on action-based decisions (cat1) but fails to extract the proper location decisions (cat3). This is because many locations are inferred based on common sense rather than the verb semantic frames. Notably, the set of rules written on top of PROPOLIS is local and simple and can be further expanded to improve performance. Table 5 further indicates that the predictions of the PROPOLIS model on the actions are much closer to the SOTA models than its predictions for the entities’ location. The good performance of PROPOLIS on the action decisions for the “Global Location” category can further show that the local context can mostly indicate the action even if retrieving the result of the action (location) requires more reasoning steps. Lastly, since PROPOLIS is a model built over local semantic frames, it dramatically fails to make accurate decisions when the entity does not appear in the sentence (Global Ent).

Q2. Can the integration of semantic parsing improve the neural models? We evaluate this based on the two strong baselines, NCET and TSLM. When semantic parsers are integrated into NCET,

all three evaluation metrics improve (compare rows #7 to #10 with row #5). This improvement is even better if the source of the graph is the abstracted parse from the PROPOLIS method (row #10). Semantic parsers improve NCET’s performance in all categories of decisions, particularly in local ambiguous sentences and decisions requiring reasoning over global locations. Notably, the integration of PROPOLIS with the NCET model significantly boosts the ability to disambiguate local information in sentences with multiple action verbs.

The integration of the semantic graph slightly hurts the performance of the CGLI baseline when using conventional metrics (1%). However, it outperforms this baseline on “cat3” (0.78%), which is the only evaluation that directly considers location predictions. Notably, the original CGLI model (baseline) uses the pre-trained classifiers from SQUAD (Rajpurkar et al., 2016) to predict the start/end tokens from the paragraph as the locations (answer to the question). However, since the integrated method extracts candidates from the graph in the form of spans, it cannot reuse the

same pre-trained classifier parameters. This may contribute to the drop in performance since CGLI performs 2% lower on the document-level F1 score when SQUAD pre-training is removed (Ma et al., 2022). Despite the drop in performance based on the conventional metrics, the integrated QA-model (CGLI + TRIPS) outperforms the baseline in almost all the criteria in the new evaluation (Table 5), especially on decisions that only require local reasoning or local disambiguation. This is due to the global nature of the TSLM (or CGLI) backbone, which predicts the locations based on the whole story and ignores many of the local signals, whereas the graph can help directly extract the local relations.

Q3. How can the decision-level metrics help understand models' weaknesses and strengths?

Based on the results in Table 5, the NCET model is better at reasoning over the local context than the global context. It also clarifies that although the TSLM (or CGLI) model can properly reason over multiple steps, it is not as competitive as the NCET model in the local cases. However, the integration of semantic parsers could improve the models to close the gap on both local and global aspects and has a complimentary influence on the initial performance of the baselines. As a general conclusion based on our new evaluation metrics, we can argue that the most challenging decisions are the ones that require reasoning over missing mentions of entities in the local context. Addressing this challenge may require external reasoning over common-sense, performing the complex coreference resolution, or handling missing pronouns.

6 Discussion

Here, we discuss some of the potential concerns that may arise with the usage of symbolic systems such as TRIPS and the new evaluation criteria.

Coverage and rule crafting of PROPOLIS. Our implementation of the symbolic method and the integrated models rely on the knowledge extracted from very fine-grained semantics covered in TRIPS. Consequently, a small mapping effort was needed to create such a system. The mapping between actions in Propara and verbs is straightforward since verbs are automatically mapped onto ontological classes that provide the type of actions based on the parse. Hence, defining the mapping rules for the most general relevant ontology types of verbs is sufficient because all the descendent types will fol-

low the same mapping (See Table 1). More details are available in Appendix B). Additionally, the effort needed for the pre-processing and designing of the mapping rules is similar to the hyperparameter tuning of neural models. Since mapping is based on common sense rather than trial-and-errors in hyperparameter tuning, finding an optimal solution may even take less effort.

Out-Of-Vocabulary words in parses. TRIPS automatically maps words to ontology classes using WordNet (Miller, 1995). This gives us considerable vocabulary coverage and reduces OOV risk. TRIPS can identify the role of the unseen words (not available in WordNet) based on the sentence syntax and will not produce errors when encountering unseen words. In the same way, PROPOLIS and integrated models will not be affected.

Effectiveness of the new evaluation metric. The previously proposed high-level evaluations are strict and do not accurately reflect the quality and quantity of the lower-level model decisions. Thus they do not adequately reveal the models' abilities. For example, when compared at high-level metrics, two models may have the same performance value of 20%, while their decision accuracy may be 60% and 10%. This issue is reflected during training epochs too when the models' performance remains the same despite the decisions on the train set continuing to improve. Therefore, it seems more appropriate to evaluate the models based on the same objective criteria used for training them (decision-level). However, the previously used metrics can be secondary evaluations to measure how well the model captures higher-level procedural concepts.

7 Conclusion

We investigated whether semantic parsers could help with reasoning over procedural text. We proposed PROPOLIS, a symbolic model operating on deep semantic parsers to solve the procedural reasoning task. For this task, the symbolic model outperformed many recent neural architectures. We then evaluated the effects of integrating semantic parsers with two well-known SOTA neural backbones. All integrated models outperformed baseline architectures, particularly when the parser provided more detailed information and rich semantic frames. Furthermore, we proposed new evaluation metrics that show the pros and cons of the models and help identify the key challenges in reasoning over procedural text.

Acknowledgments

This project is supported by National Science Foundation (NSF) CAREER award 2028626 and partially supported by the Office of Naval Research (ONR) grant N00014-20-1-2005. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation nor the Office of Naval Research. We also want to thank Drew Hayward, who helped us with a subset of experiments on PROPOLIS during his research at Michigan State University.

Limitations

There are multiple limitations to methods that rely on semantic parsers for solving natural language tasks. First, semantic parsers and especially the ones that do not rely on noisy training data, are most susceptible to errors when the original sentence contains even small grammatical or spelling errors. Next, parsers such as TRIPS rely on general-purpose ontology and a pipeline for generating the output parses. The pipeline first understands the meaning of each word in the sentence. This is subject to errors when words/verbs can have multiple meanings and require the context to disambiguate their semantics. For instance, the TRIPS parser may map the verb ‘run’ to the ‘management’ class in ontology instead of the ‘physical activity’ class. Lastly, executing graph attention networks with many layers requires a powerful system with access to GPU and is more time-consuming than the baselines that do not require reasoning over a graph structure.

References

- James F Allen and Choh Man Teng. 2017. Broad coverage, domain-generic deep semantic parsing. In *2017 AAAI Spring Symposium Series*.
- Aida Amini, Antoine Bosselut, Bhavana Dalvi Mishra, Yejin Choi, and Hannaneh Hajishirzi. 2020. Procedural reading comprehension with attribute-aware context flow. In *Proceedings of the Conference on Automated Knowledge Base Construction (AKBC)*.
- Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 1499–1510.

- Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *Proceedings of the 6th International Conference for Learning Representations (ICLR)*.
- Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1595–1604, New Orleans, Louisiana. Association for Computational Linguistics.
- Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, and Peter Clark. 2019. Everything happens for a reason: Discovering the purpose of actions in procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4496–4505, Hong Kong, China. Association for Computational Linguistics.
- Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2018. Building dynamic knowledge graphs from text using machine reading comprehension. In *International Conference on Learning Representations*.
- Hossein Rajaby Faghihi and Parisa Kordjamshidi. 2021. Time-stamped language model: Teaching language models to understand the flow of events. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4560–4570.
- George Ferguson, James F Allen, et al. 1998. Trips: An integrated intelligent problem-solving assistant. In *Aaai/Iaai*, pages 567–572.
- Aditya Gupta and Greg Durrett. 2019. Tracking discrete and continuous entity state for process understanding. In *Proceedings of the Third Workshop on Structured Prediction for NLP*, pages 7–12, Minneapolis, Minnesota. Association for Computational Linguistics.
- Hao Huang, Xiubo Geng, Jian Pei, Guodong Long, and Daxin Jiang. 2021. Reasoning over entity-action-location graph for procedural text understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5100–5109, Online. Association for Computational Linguistics.
- Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han.

2019. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.
- Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1456–1465, Berlin, Germany. Association for Computational Linguistics.
- Kaixin Ma, Filip Ilievski, Jonathan Francis, Eric Nyberg, and Alessandro Oltramari. 2022. Coalescing global and local information for procedural text understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1534–1545.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Kuntal Kumar Pal, Kazuaki Kashihara, Pratyay Banerjee, Swaroop Mishra, Ruoyu Wang, and Chitta Baral. 2021. [Constructing flow graphs from procedural cybersecurity texts](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3945–3957, Online. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.
- Qi Shi, Qian Liu, Bei Chen, Yu Zhang, Ting Liu, and Jian-Guang Lou. 2022. Lemon: Language-based environment manipulation via execution-guided pre-training. *arXiv preprint arXiv:2201.08081*.
- Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. 2020. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*.
- Shane Storcks, Qiaozhi Gao, Yichi Zhang, and Joyce Chai. 2021. Tiered reasoning for intuitive physics: Toward verifiable commonsense language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4902–4918.
- Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 57–66, Brussels, Belgium. Association for Computational Linguistics.
- Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. Wiqa: A dataset for “what if...” reasoning over procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6076–6085.
- Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. [A dataset for tracking entities in open domain procedural text](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.
- Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. Cite arxiv:1502.05698.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Te-Lin Wu, Alex Spangher, Pegah Alipoormolabashi, Marjorie Freedman, Ralph Weischedel, and Nanyun Peng. 2022. Understanding multimodal procedural knowledge by sequencing multimodal instructional manuals. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4525–4542.
- Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes. In *EMNLP*.
- Zhihan Zhang, Xiubo Geng, Tao Qin, Yunfang Wu, and Daxin Jiang. 2021. Knowledge-aware procedural text understanding with multi-stage training. In *Proceedings of the Web Conference 2021*.
- Chen Zheng and Parisa Kordjamshidi. 2020. Srlgrn: Semantic role labeling graph reasoning network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8881–8891.

A Implementation Details

We use the PyTorch geometric ⁵ library to implement all the graph attention models and Hugging-

⁵<https://pytorch-geometric.readthedocs.io/>

face library (Wolf et al., 2020) for implementing the language models. For the NCET model and its extensions based on semantic parsers, the best model is selected by a search over the $\lambda \in \{0.3, 0.4\}$, the learning rate in $\{3e - 5, 3.5e - 5, 5e - 5\}$. The number of graph attention layers are set to 2 and the batch size is set to 8 process. All models are using Bert-base as the selected language model for encoding the context. We further use RAdam (Liu et al., 2019) to optimize the model parameters of both the language models, the LSTM, and the classifiers. For the CGLI method, we use the exact hyper-parameters as specified in (Ma et al., 2022). We further use 15 layers of graph attention network with the input from the fifth layer of the time-aware language models. The gradients from the graph attention network (GAT) would not back-propagate to the original language model and only affect the parameters in the GAT model. The implementation code of our models and the re-implemented models will be available in the camera-ready version. The implementation code for all our models will be available on GitHub after acceptance.

B PROPOLIS

Here, we share more details on the steps in producing symbolic decisions over the actions and the locations of objects in the Propara dataset, based on the TRIPS parser. You can also find the ontology of TRIPS parser online⁶.

B.1 Graph Abstraction

From the logical forms produced by the TRIPS parser we need to extract the events and event relationships of interest. Because much of the variation expected in sentence constructions is handled by the TRIPS system, we are able to use a relatively compact specification for defining the events and relationships of interest, while coping with fairly complex and nested formulations.

We capitalized on the TRIPS ontology and parser to develop a compact and easy-to-maintain specification of event extraction rules. Instead of having to write one rule to match each keyword/phrase that could signify an event, many of these words/phrases have already been systematically mapped to a few types in the TRIPS ontology. For instance, demolish, raze, eradicate, and annihilate are all mapped to the TRIPS ontology type

“ONT::DESTROY”. In addition, the semantic roles are consistent across different ontology types. The parser handles various surface structures, and the logical form contains normalized semantic roles. For example, in the following sentence:

- The bulldozer demolished the building
- The building was demolished
- The demolition of the building
- Building demolition

, all the parses result in the same basic logical form with the semantic roles “AFFECTED: the building” and, where applicable, “AGENT: the bulldozer”. Thus, we needed very few extraction rule specifications for each event type, covering a wide range of words and syntactic patterns.

B.2 Rule-based Local Decisions

(New: The sets of heuristics used to detect the effect of each semantic frame on the arguments were shown in Table 1). To handle the location arguments from the parses, we also consider the two cases on ‘from_loc’ and ‘to_loc’. In the specific case of a destroy event, any location attached to the semantic frame is considered the ‘from_loc’ for the item being destroyed.

B.3 Global Reasoning

To perform the global reasoning over the local predictions, we first do a forward pass through the actions and location predictions and make sure that they are globally consistent. To do so, we start from the first predicted action and check the following on every next step prediction:

- If the current action is None, then we skip this step!
- If the last observed action is “Create” or “Move”,
 - If the current action in “Create” and the location of this action is the same as the last observed location, then the new “Create” action is transformed to “None”.
 - IF the current action is “Create” and the location of this action is different from the last observed location, then the new action is changed to “Move”.

⁶<https://www.cs.rochester.edu/research/trips/lexicon/browse-ont-lex-ajax.html>

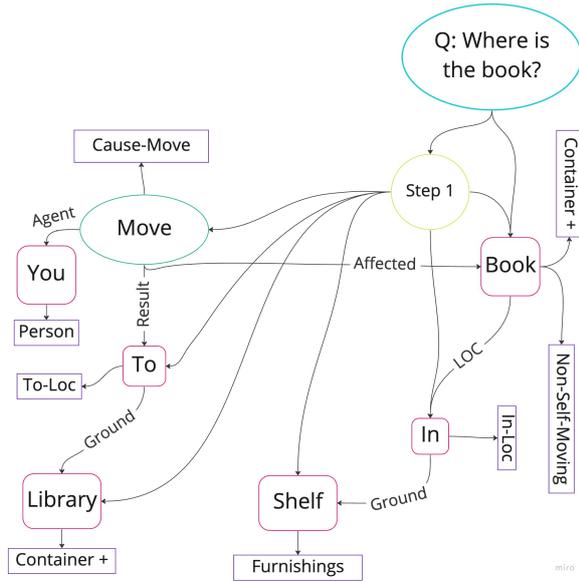


Figure 4: The QA graph for the query of “where is the book” and the sentence “Move the book on the shelf to the library”.

- Otherwise, the new action is kept the same, and the last observed action is updated.
- If the last observed action is “Destroy”,
 - If the current action is “Destroy” and it has a location different from the last observed location, then the action is changed to “Move”.
 - If the current action is “Destroy” and it has a location similar to the last observed location, then the action is changed to “None”.
 - Otherwise, the new action is kept the same, and the last observed action is updated.

After fixing the sequence of actions, we first check whether the entity gets created at any of the steps or is just moved or destroyed during the process. If the entity is not created, its initial location is equal to the first ‘from_loc’ in any subsequent actions. we then use the following criteria to fix the locations in a forward pass over the local decisions:

- If the action is “Move” but there is no final location, the final location is the first ‘from_loc’ from any of the subsequent actions before the next “Move” event.
- If the object is being “Moved”, then its final location should be changed. If the action does

not indicate a new location or the information is missing, we replace the final location with ‘?’ to indicate an unknown location.

- If the action is “None”, the last location is kept unchanged for the new step.

C Graph Attention Network

TransformerConv uses the following formula to update the representation of the nodes (v_i) in the graph.

$$\mathbf{v}_i^{l+1} = \mathbf{W}_1 \mathbf{v}_i^l + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \left(\mathbf{W}_2 \mathbf{v}_j^l + \mathbf{W}_6 \mathbf{e}_{ij} \right),$$

where $\mathcal{N}(i)$ represents the neighbors of node i in the graph, l is the layer, and the coefficient $\alpha_{i,j}$ is computed using the following formula:

$$\alpha_{i,j} = \text{softmax} \left(\frac{(\mathbf{W}_3 \mathbf{v}_i^l)^\top (\mathbf{W}_4 \mathbf{v}_j^l + \mathbf{W}_6 \mathbf{e}_{ij})}{\sqrt{d}} \right)$$

D Semantic Parsers

Figure 4 shows an example of the QA graph used in the integration model with CGLI baseline.

Named Entity Recognition in a Very Homogeneous Domain

Oshin Agarwal

University of Pennsylvania
oagarwal@seas.upenn.edu

Ani Nenkova

Adobe Research
nenkova@adobe.com

Abstract

Machine Learning models have lower accuracy when tested on out-of-domain data. Developing models that perform well on several domains or can be quickly adapted to a new domain is an important research area. Domain, however, is a vague term, that can refer to any aspect of data such as language, genre, source and structure. We consider a very homogeneous source of data, specifically sentences from news articles from the same newspaper in English, and collect a dataset of such “in-domain” sentences annotated with named entities. We find that even in such a homogeneous domain, the performance of named entity recognition models varies significantly across news topics. Selection of diverse data, as we demonstrate, is crucial even in a seemingly homogeneous domain.

1 Introduction

Supervised neural models for named entity recognition achieve high accuracy when used in-domain. When models are evaluated or adapted (Daumé III, 2007; Wang et al., 2020; Gururangan et al., 2020) for out-of-domain text, or even developed for specialized domains (Nguyen et al., 2020; Beltagy et al., 2019), the term domain generally refers to broad genres such as news, social media, or biomedical text. However, text can be (dis)similar in aspects beyond genre, such as the source of the data, its structure, or the time period. Dai et al. (2019) distinguish two aspects of domain—the genre and the tenor, which they describe as the participants in the discourse, their relationships and their purpose. They find that even though people consider genre to be more important for domain adaptation, tenor is important as well when selecting pre-training data.

The term domain encompasses more than just broadly defined genres. Online comments on different platforms can be considered different domains. So can news from different newspapers or different time periods. We show that even text from the

same genre and source needs to be examined finely for topical or structural differences. We collect a dataset of news articles from the New York Times and annotate it for named entities. We find that the performance of NER models varies significantly even in this dataset when it is stratified based on news topics. While entities unseen in the training data can be a factor that contributes to performance degradation, we find that structural differences in sentences and entity ambiguity are the main contributors. Selecting diverse data is therefore crucial even in such “in-domain” settings. We show that even a very small number of sentences from each topic can help narrow the performance gap, and selecting random sentences rather than full documents from the full corpus, will ensure that there is a good sample of diverse sentences.

2 Dataset

The dataset is available at <https://github.com/oagarwal/nyt-ner>. Here we describe the process of collecting it.

2.1 Data Collection

We sample sentences from the New York Times (NYT) Annotated Corpus (Sandhaus, 2008). The corpus consists of 1.8M articles from NYT between 1987 and 2007 along with article metadata provided by the New York Times Newsroom, the New York Times Indexing Service and the online production staff at nytimes.com. We select sentences from different years and news topics¹, both available as metadata. Variations in topic names are merged together resulting in a total of nine topics—Arts (+Weekend/Cultural), Business (+Financial), Classifieds (+Obituary), Editorial, Foreign, Metropolitan, Sports and Others. Others consists of all desks that did not have many articles such as Real Estate, New Jersey Weekly, Book Review, Job Market, Science and Health & Fitness.

¹desk in NYT newsroom that produced the article

2.2 Data Annotation

The selected sentences are labeled with person (*PER*), location (*LOC*) and organization (*ORG*) tags on Upwork², with CoNLL’03 (Tjong Kim Sang and De Meulder, 2003) guidelines and annotation scheme. For efficiency, we first annotate the sentences with entities from the article metadata. The metadata consists of relevant persons, locations and organizations selected from a fixed vocabulary, manually assigned as part of NYT indexing. This first pass of annotation is done using phrase matching, similar to a gazetteer lookup. The resulting annotations are expected to be better than looking up in a general gazetteer since the available entities are assigned manually per article.

We use one annotator per example, but the annotators are first trained for the task. Each annotator is given 10-20 sentences to correct the entity labels from the first pass. The corrected sentences are reviewed by one of the authors and the feedback is shared with the annotator. Another 10-20 sentences are then shared with the annotator. These sentences are a mix of previously annotated but problematic sentences and new sentences, focusing on the types of mistakes made by the particular annotator in the earlier batch. If the annotator makes several mistakes in this round overall, or even one mistake on a sentence re-selected from the previous round, they are not asked to do further annotations. The annotators are encouraged to ask clarifying questions during the training rounds as well as the actual annotations. If they are uncertain about the correct label for any example, they are asked to indicate this in their comments. Finally, one of the authors goes over a random selection of examples to ensure quality and also over the ones marked as uncertain to correct if necessary.

2.3 Data Splits

We split sentences in each news topic into training, development and test splits in the ratio 35:15:50. The proportions are different from the typical 80:10:10 splits but ensure that there are a sufficient number of test examples in each topic for stable and reliable results. The number of sentences and entities in each topic are shown in Table 1.

3 Results

We finetune BERT-large-cased (Devlin et al., 2019) on each topic, evaluating on all others. Hyperpa-

²<https://www.upwork.com/>

	# sentences			# entities		
	train	dev	test	train	dev	test
arts	3570	1531	5101	2451	1112	3542
business	2454	1052	3507	2055	870	2923
classified	1052	451	1503	1380	568	1895
editorial	2872	1232	4104	2198	939	3113
foreign	4654	1995	6649	3961	1672	5906
metropolitan	2873	1232	4106	2254	888	3141
national	3888	1667	5555	3062	1310	4303
sports	3664	1571	5235	3475	1572	4995
others	3221	1380	4602	2397	988	3413

Table 1: Dataset Statistics

parameter details are listed in the appendix. We report micro-F1 at the span-level averaged over three runs with different seeds. The full evaluation table is shown in the appendix for reference. Here we discuss the aggregated results. Since domain is used to refer to the genre of text (news in this case), we use the term sub-domain to refer to the news topics. However, we still use in-subdomain (InD) and out-of-subdomain (OOD) to refer to in-subdomain and out-of-subdomain training and evaluation in the following sections.

3.1 Evaluation Sub-domain Difficulty

First, we report the performance on each test sub-domain, when a model is trained on sentences from the same sub-domain and when trained on sentences from a different sub-domain. The goal is to determine if it is easier to recognize entities in some sub-domains. The results are shown in Table 2. InD refers to the models trained on the same sub-domain as the test, and OOD refers to models trained on each of the remaining sub-domains. The OOD mean and median are aggregated over the eight models trained on each of the remaining sub-domains. As expected, in-subdomain training results in incredibly high F1 on all sub-domains. The F1 with OOD training is lower than that for in-subdomain, especially when testing on classified and sports. For OOD, we also report the minimum and maximum F1 on each test sub-domains, along with the corresponding training sub-domain, showing that the range of F1 also varies considerably. The lowest test F1 on most sub-domains occurs with the model trained on classified, and the highest occurs with training on national or metropolitan. For a better understanding of the variation in the performance on a given test sub-domain with different OOD sub-domains, we also show box plots (Figure 1) for the test sub-domains of classifieds

InD	OOD						
	mean		median	min		max	
	F1	F1	F1	F1	trn-d	F1	trn-d
a	92.1	86.9	88.3	78.4	c	89.7	m
b	95.7	88.2	90.9	72.0	c	93.2	m
c	94.7	77.7	76.7	67.1	f	90.4	e
e	96.4	88.7	93.0	67.0	c	94.6	n
f	96.9	87.5	92.5	64.2	c	93.9	n
m	95.0	89.2	90.8	78.4	c	92.8	n
n	96.2	90.9	93.8	79.8	c	94.9	m
s	94.8	81.0	81.0	77.9	n	84.6	a
o	92.0	87.4	89.0	76.7	c	90.8	m

Table 2: F1 on each test sub-domain, one per row, with models trained on different domains. Each row represents a test sub-domain. InD is the F1 with in-subdomain training. OOD mean and median are over the remaining eight training domains. Min and max show the F1 and training sub-domain with minimum and maximum F1 on the given test sub-domain.

InD	OOD						
	mean		median	min		max	
	F1	F1	F1	F1	tst-d	F1	tst-d
a	92.1	87.7	90.4	73.2	c	91.6	b
b	95.7	88.7	90.0	78.1	c	94.0	e
c	94.7	74.3	77.5	64.2	f	79.8	n
e	96.4	89.4	90.3	80.4	s	94.2	n
f	96.9	85.8	88.8	67.1	c	93.6	n
m	95.0	90.6	92.0	84.0	s	94.9	n
n	96.2	89.2	91.1	77.9	s	94.6	e
s	94.8	82.4	85.1	68.8	c	86.6	n
o	92.0	89.2	92.3	75.3	c	94.1	e

Table 3: F1 of each training sub-domain, one per row, across different test sub-domains. Each row represents a training sub-domain. InD is the F1 for in-subdomain testing. OOD mean and median are over the remaining eight test domains. Min and max show the F1 and test sub-domain with minimum and maximum F1 for the given training sub-domain.

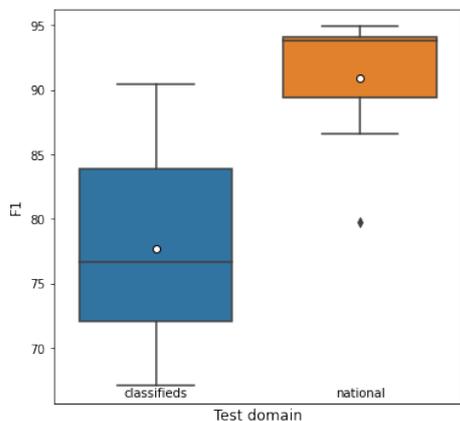


Figure 1: Box plot for two test sub-domains (classifieds and national) showing the range of F1 with training on OOD sub-domains

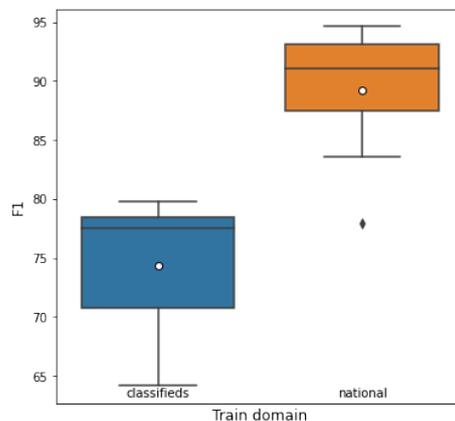


Figure 2: Box plot for two training sub-domains (classified and national), showing the range of F1 when tested on these as OOD sub-domains.

and national. Depending upon the sample of sub-domains in the test set, the model performance can vary significantly even in such a homogeneous domain, leading to an incorrect characterization of the domain/dataset difficulty.

3.2 Training Sub-domain Quality

Next, we report the performance of models trained on each sub-domain when tested on the same sub-domain and on other sub-domains. The goal is to determine if it is better (or worse) to train on certain sub-domains for good performance overall. The results are shown in Table 3. InD refers to testing a model on the same sub-domain as the training data, and OOD refers to testing it on the remaining eight sub-domains. The OOD mean and median

are aggregated over the eight OOD sub-domains. As expected, in-subdomain testing results in incredibly high F1 on all sub-domains. The F1 with OOD testing is lower than that for in-subdomain, especially for models trained on classified and sports. For OOD, we also report the minimum and maximum F1 obtained by each model along with the corresponding test sub-domain, showing that the range of F1 also varies significantly. The lowest F1 for most models occurs when tested on classified or sports, and the highest F1 occurs when tested on national or editorial. For a better understanding of the variation in the performance of a model trained on sub-domains when tested on other sub-domains, we also show box plots (Figure 1) for the training sub-domains of classified and national. Depending

Domain	Sentence
Classifieds	WEISER–Joel, passed away on March 31st, 2007.
Sports	Pollin clashed with Jordan at a bargaining session during the long labor standoff in November 1998.

Table 4: Example of sentences by sub-domain

upon the sample of sub-domains in the training set, the model performance can vary significantly even in such a homogeneous domain, leading to a much better or worse resulting model.

Classified and Sports stand out, exhibiting lower performance than other sub-domains for both training and testing. Examples sentences for both are shown in Table 4. Classified has several sentences that have atypical sentence structures, beginning with the last name in uppercase. For Sports, the entity type cannot be determined from the sentence-level context in several cases. In the example, it is hard to say whether the entities are names of person, location or team (organization). If this ambiguity of these entities isn’t captured in the training data, labeling them correctly is unlikely.

4 Data Selection

Datasets are typically collected by selecting some documents and then annotating all sentences in each document. The training set in CoNLL’03 (Tjong Kim Sang and De Meulder, 2003) has 15k sentences from 946 documents, Wikigold (Bala-suriya et al., 2009) has 1.7k sentences from 145 pages, and MUC-7 (Chinchor, 1998) has 3.5k sentence from 100 articles.³ This method of data selection is reasonable and intuitive. It also supports the development of models that utilize document-level context (Ratinov and Roth, 2009) which can help resolve the entity types in sentences such as the above example from sports. However, most commonly used models are built at the sentence level and the selection of full documents could result in performance similar to a model trained on the same sub-domain, with all sentences in a document representing the same sub-domain and fewer chances to cover rare sub-domains (types of documents). To illustrate this, we train models for NER using CoNLL ’03. We randomly select 3,000 training sentences as this is roughly the number of sentences in each of the sub-domains. We train three

³MUC-7 consists of sentences from the New York Times. However, we were unable to map the documents in MUC-7 to the NYT Annotated Corpus. Regardless, MUC-7 consists only of articles on aircraft accidents and launch events, and would likely not span enough sub-domains for our analysis.

models with different seeds and report the average F1 in the third column of Table 5. CoNLL consists of news on mainly business, national, foreign and sports. Therefore, F1 on these sub-domains is closer to that with in-subdomain training, and F1 on the remaining sub-domains is close to that with out-of-subdomain training.

It is therefore essential to ensure a diverse set of sentences in the training data. Even a small number of sentences of each sub-domain in the training data can make a vast difference. Columns ‘C’ and ‘N’ in Table 5 show the F1 on various test sub-domains with a model trained on just classified or just national news. In columns ‘C+10’ and ‘N+10’, we add just 10 sentences from each of the remaining eight sub-domains. For classified, this affects each of the test sub-domains with an improvement of up to 12 points F1. On national, this mainly improves F1 on classified by 10 points and that on sports by 2 points. These two sub-domains, as shown above, exhibit different properties than the rest of the data and therefore including even a few relevant examples helps the models substantially.

One way to select relatively diverse sentences is by data selection at the sentence level instead of the document level. First, segment each document in a corpus into sentences and then select sentences randomly. While new future domains or those that evolve significantly will still be missed, this method would result in the selection of some representative samples of each existing domain. Such explicit sentence selection has been performed for domains such as Twitter where explicit documents⁴ do not exist. Derczynski et al. (2016) selects tweets from different countries and different types of user accounts for linguistic variations and topics. They also account for temporal variation taking tweets from different years, months, weeks and days.

We build models with this random sentence selection scheme. We first downsample the data such that it follows the same distribution of sub-domains as the NYT corpus with 20 years of articles. This results in 10,500 training and 4,494 development sentences with 14% arts, 11% business, 3% classi-

⁴A thread could be considered a document.

	InD	OOD	CoNLL	C	C+10	N	N+10	Rndm
arts	92.1	86.9	85.8	78.4	82.4	88.7	88.4	90.6
business	95.7	88.2	91.4	72.0	83.8	91.9	92.2	93.9
classified	94.7	77.7	64.8	94.7	94.6	83.6	90.2	93.9
editorial	96.4	88.7	89.2	67.0	83.7	94.6	94.4	93.7
foreign	96.9	87.5	90.4	64.2	82.6	93.9	94.0	93.2
metropolitan	95.0	89.2	89.0	78.4	83.5	92.8	92.8	91.8
national	96.2	90.9	90.0	79.8	86.2	96.2	96.3	93.0
sports	94.8	81.0	89.7	78.3	80.1	77.9	79.7	91.7
others	92.0	87.4	86.3	76.7	82.2	90.3	90.1	90.2
Avg	94.9	86.4	86.3	76.6	84.4	90.0	90.9	92.4

Table 5: F1 on each test sub-domain with different models. InD is in-domain training and OOD is the average of out-of-domain training. CoNLL refers to training on CoNLL '03. C and N are trained on classified and national only. C+10 and N+10 additionally include 10 sentences from each sub-domain. Rndm is random selection of sentences from a corpus with sentences in the same proportion of sub-domains as the full NYT corpus. Highest F1 in each row (excluding InD) is boldfaced.

fied, 5% editorial, 7% foreign, 11% metropolitan, 8% national, 11% sports and 30% others. We then select 3,000 training and 1,284 development sentences randomly from this set. This is roughly the average number of sentences in each of the sub-domains and seeks to eliminate the impact of the training data size. Every sub-domain has at least 39 sentences in the selected training set. With models trained on this dataset, the average F1 is almost the same as in-subdomain training (col Rndm).

5 Conclusion

Perform fine-grained inspection of data even when it seems that the domain is homogeneous, and perform training data selection at the sentence level rather than the document level.

6 Limitations

We develop a new corpus for a standard NER task, drawn from a reputable news source, New York times. Our analysis is based on the sub-domains available in the metadata of the news article. To extend it to other datasets, automatic predictors of domain are necessary. Furthermore, for a random sentence selection that includes all representative samples, a corpus spanning the entire space of sentences is needed. This is straightforward for newspapers or Wikipedia, but infeasible for domains such as Reddit or Twitter. In such cases, domain knowledge is used to select diverse sentences (Derczynski et al., 2016), again pointing to the need for automatic domain prediction. We performed domain classification experiments on our dataset via unsupervised clustering as well as zero-shot classi-

fication⁵ (Yin et al., 2019), using both the known domains from the metadata and dummy domains as candidates. The accuracy of the best classifier on our data was only 30%, insufficient for better performance than a random sentence selection.

References

- Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. [Named entity recognition in Wikipedia](#). In *Proceedings of the 2009 Workshop on The People’s Web Meets NLP: Collaboratively Constructed Semantic Resources (People’s Web)*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Nancy A. Chinchor. 1998. [Overview of MUC-7](#). In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.
- Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2019. [Using similarity measures to select pre-training data for NER](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1460–1470, Minneapolis, Minnesota. Association for Computational Linguistics.
- Hal Daumé III. 2007. [Frustratingly easy domain adaptation](#). In *Proceedings of the 45th Annual Meeting of*

⁵<https://huggingface.co/facebook/bart-large-mnli>

the Association of Computational Linguistics, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. **Broad Twitter corpus: A diverse named entity recognition resource**. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. **Don’t stop pretraining: Adapt language models to domains and tasks**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. **BERTweet: A pre-trained language model for English tweets**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. **Design challenges and misconceptions in named entity recognition**. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. **Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition**. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Jing Wang, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2020. **Multi-domain named entity recognition with genre-aware and agnostic inference**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8476–8488, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric

Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. **Huggingface’s transformers: State-of-the-art natural language processing**. *ArXiv*, abs/1910.03771.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. **Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

A Hyperparameters and Infrastructure

Hyperparameters are optimized via grid search over the learning rate (3e-05, 5e-06, 5e-06), batch size (2, 4, 8, 16, 32) and number of epochs (1, 2, 3, 4, 5) on each sub-domain. Models were fine-tuned using the implementation in HuggingFace (Wolf et al., 2019) on 2 V100 GPUs. The training time for models varies by the sub-domain and hyperparameters, and is typically 10-20 min. The best checkpoint on the development set is selected.

	LR	BS	EP
arts	3e-05	16	4
business	5e-05	8	2
classified	5e-05	4	1
editorial	5e-05	8	2
foreign	5e-05	4	2
metropolitan	5e-05	16	2
national	5e-05	16	2
sports	3e-05	8	3
others	5e-05	8	2

Table 6: Hyperparameters, namely the learning rate, the total batch size and the number of epochs.

B Full Evaluation

Test	Training Domain								
	a	b	c	e	f	m	n	s	o
a	92.1	88.7	78.4	87.9	87.1	89.7	88.7	85.7	89.2
b	91.6	95.7	72.0	90.2	90.0	93.2	91.9	84.7	92.0
c	73.2	78.1	94.7	90.4	67.1	84.7	83.6	68.8	75.3
e	91.0	94.0	67.0	96.4	92.1	94.3	94.6	82.1	94.1
f	90.4	92.9	64.2	92.1	96.9	93.3	93.9	80.0	93.1
m	90.4	91.0	78.4	91.4	90.6	95.0	92.8	86.2	92.5
n	90.3	94.0	79.8	94.2	93.6	94.9	96.2	86.6	94.1
s	84.6	81.7	78.3	80.4	78.1	84.0	77.9	94.8	82.9
o	90.1	89.0	76.7	88.9	87.6	90.8	90.3	85.5	92.0

Table 7: F1 on model trained on each sub-domain on each of the sub-domains

Crawling The Internal Knowledge-Base of Language Models

Roi Cohen¹ Mor Geva^{2*} Jonathan Berant¹ Amir Globerson¹

¹Blavatnik School of Computer Science, Tel Aviv University ²Allen Institute for AI

roi1@mail.tau.ac.il, pipek@google.com, joberant@cs.tau.ac.il, gamir@tauex.tau.ac.il

Abstract

Language models are trained on large volumes of text, and as a result their parameters might contain a significant body of factual knowledge. Any downstream task performed by these models implicitly builds on these facts, and thus it is highly desirable to have means for representing this body of knowledge in an interpretable way. However, there is currently no mechanism for such a representation. Here, we propose to address this goal by extracting a knowledge-graph of facts from a given language model. We describe a procedure for “crawling” the internal knowledge-base of a language model. Specifically, given a seed entity, we expand a knowledge-graph around it. The crawling procedure is decomposed into sub-tasks, realized through specially designed prompts that control for both precision (i.e., that no wrong facts are generated) and recall (i.e., the number of facts generated). We evaluate our approach on graphs crawled starting from dozens of seed entities, and show it yields high precision graphs (82-92%), while emitting a reasonable number of facts per entity.

1 Introduction

Modern language models (LMs) (Raffel et al., 2020; Brown et al., 2020) are trained on vast amounts of text that captures much of human knowledge, including scientific articles, Wikipedia, books, and other sources of information (Gao et al., 2020). Consequently, such models encode world knowledge in their parameters, allowing them to generate rich and coherent outputs.

Past work has illustrated LMs can be viewed as knowledge-bases (Petroni et al., 2019) as well as analyzed the encoded knowledge (e.g., see AlKhamissi et al., 2022) and leveraged it for applications such as closed-book QA (Roberts et al., 2020; Brown et al., 2020) and search (Tay et al., 2022), illustrating LMs can be viewed as

knowledge-bases (Petroni et al., 2019). But what are the facts stored in the internal knowledge bases of modern LMs, and how can these be represented explicitly? This is the challenge we address in this work. Our motivation is to obtain an interpretable and transparent representation that will allow humans to inspect what the LM knows, what it does not know, why it makes certain mistakes, and what are the biases it encodes. Moreover, with such a representation, one can leverage general-purpose tools, such as query languages, for interacting with this knowledge.

The first question in this endeavour is what is a suitable explicit knowledge representation. A natural candidate structure is a knowledge graph (KG). Namely, a graph whose nodes are entities and whose edges represent relations between entities. KGs are appealing since information can be readily “read-off” from the graph, they can be reliably queried, and different KGs can be easily compared. KGs have been extensively used to represent knowledge (Bollacker et al., 2008; Vrandečić and Krötzsch, 2014), but a key limitation is their *low coverage*, as they usually require manual curation and depend on a closed schema. Conversely, LMs might have very high coverage as they are trained on a vast body of knowledge represented as raw text. We thus ask if it is possible to convert an LM to a KG, such that we enjoy its advantages while achieving high coverage.

As the full KG encoded in an LM can be large, we reduce the problem to the task of constructing a KG around a given seed entity. For example, Fig. 1 shows a KG extracted by our method for the seed entity *Alan Turing*. This can be viewed as a crawling procedure which starts from the seed entity and recursively expands it to expose additional facts. This crawling problem introduces several new challenges. First, unlike prior work (Petroni et al., 2019; Alivanistos et al., 2022; Hao et al., 2022), we are given only *an entity*, without know-

* Now at Google Research.

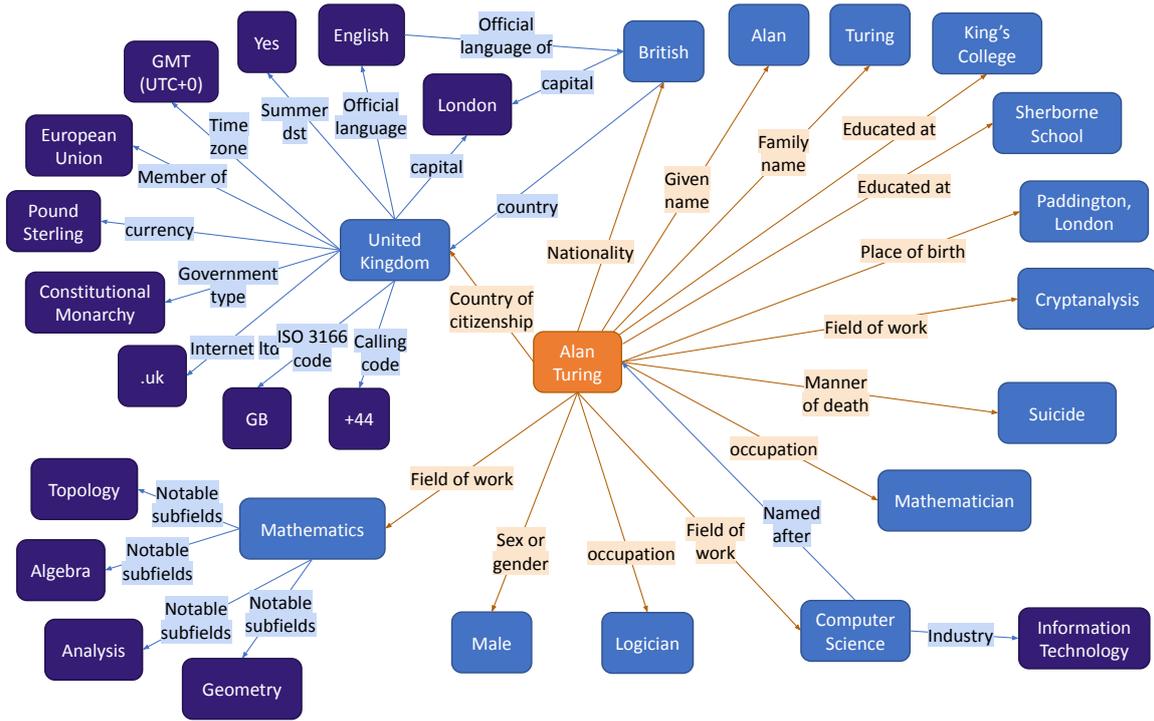


Figure 1: An example of a generated depth-2 knowledge graph around the seed entity ALAN TURING, applying LMCRAWL (see Sec. 3-Sec. 4). Additional graphs are in Sec. E.

ing what relations are associated with it. Thus, we have to extract those relations and then find the objects for each relation. Second, KGs are expected to exhibit very high precision, and thus it is necessary to generate as many relevant facts as possible while maintaining almost perfect factual correctness.¹

We address the above challenges by decomposing crawling into multiple sub-tasks, and handle each task using few-shot in-context learning (Brown et al., 2020). Explicitly, we do not fine-tune a model, but instead manually design a prompt and a few examples for each task, an approach recently-proven successful (Wei et al., 2022; Drozdov et al., 2022; Chowdhery et al., 2022; Khot et al., 2022). We use the following sub-tasks (see Tab. 1 for the full list and examples). First, given an entity e (e.g., ALAN TURING), we generate the relations relevant for e (e.g., EDUCATED AT, PLACE OF BIRTH). Second, for each entity e and relation r , we generate the corresponding set of objects O and add to the KG triplets (e, r, o) for each $o \in O$. For example, for ALAN TURING and EDUCATED AT, we gener-

¹We note that there is a deeper philosophical aspect to this issue, which is at the core of the field of epistemology. Namely, what does it mean for a model to “believe” a fact, as opposed to the model “knowing” a fact. Here we adopt a “dispositional” view of belief, whereby a belief corresponds to a statement by the model, and knowledge is a belief that is true in the world.

ate triplets with the *objects* KING’S COLLEGE and SHERBORNE SCHOOL. To maintain high precision, we prompt the model to emit “Don’t know” whenever it is not confident about the target objects. All the above outputs are generated through in-context learning, where we use the WIKIDATA KG (Vrandečić and Krötzsch, 2014) to construct in-context examples. *Don’t know* examples are constructed by finding true facts in WIKIDATA that are unknown to the LM. Finally, we increase recall by prompting the LM to generate paraphrases for entities and relations, and use those to obtain additional triplets.

We test our approach with GPT-3 (text-davinci-002) on 140 seed entities, and show that we can extract accurate KGs (~82-92% precision) that contain a plausible number of facts per entity. Importantly, large LMs are not constrained to a predefined schema, and indeed our procedure with GPT-3 generates facts outside the schema of WIKIDATA, e.g., (BOSTON CELTICS, CHAMPIONSHIPS, 17).

To conclude, our contributions are: 1) Formulating the problem of crawling a KG from an LM, 2) Presenting a prompt-based approach that decomposes the problem into multiple sub-tasks, and 3) Evaluating the approach with GPT-3, which leads to high-precision graphs.

2 Problem Setup

Our goal is to uncover the knowledge-base of a given LM. We represent a knowledge-base via a KG, which is a collection of triplets. Formally, a KG is a graph $G = (N, R, E)$, where N is a set of entities, R is a set of relations, and E is a set of subject-relation-object triplets (s, r, o) where $s, o \in N$ and $r \in R$.

To simplify the setup, we assume we are given a “seed entity” around which we will expand the graph (for example Fig. 1). Conceptually, we can also let the LM generate seed entities, but we argue seed expansion is a more realistic scenario, where a user is interested in a graph about a certain entity.

Entities and relations are represented via strings and are not constrained to a given vocabulary (similar to open information extraction. e.g., see [Vo and Bagheri, 2017](#)).

3 Crawling KGs via Prompting

The core component of our approach is a procedure that takes an entity e , and extracts all relations associated with it, and the corresponding objects. Namely, we expand the KG around this entity. We can then recursively apply this procedure to further expand the KG. We refer to this as ‘entity expansion’, and break it into two high-level steps:

- **Relation generation** (Sec. 3.1): For an entity e , generate a set of relations R , where e is the subject.
- **Object generation** (Sec. 3.3-Sec. 3.4): Given the entity e and the relation set R , find the corresponding objects. Namely, for each $r \in R$, find a list of entities O such that (e, r, o) is in the KG for $o \in O$. We consider lists since many relations (e.g., CHILDREN) potentially have multiple correct objects. Furthermore, we also consider the case where the object corresponding to (e, r) is unknown to the model (e.g., the model does not know who is the daughter of a given entity e). In this case we take O to be empty, and the edge is not added to the KG. This is crucial for maintaining a high-precision KG.

Both steps are achieved via few-shot in-context learning. Namely, we construct prompts with in-context examples (stay fixed throughout the process) that exhibit the desired behaviour (Tab. 1).

To improve recall, we employ an additional paraphrasing procedure (Sec. 3.2 and Sec. 3.5), which generates alternative strings for a given entity or re-

lation. For example, the entity WILLIAM CLINTON can be referred to as WILLIAM JEFFERSON CLINTON or BILL CLINTON, and the relation OCCUPATION may be expressed as PROFESSION. Thus, we run object and relation generation for all these variants, and pool the results to construct the final graph. Paraphrases are also obtained through the LM, without use of external knowledge. The entire flow is illustrated in Fig. 2, and we next elaborate on each of the components.

3.1 Relation Generation

Our task is to generate a set of relations R for a given subject entity e . To achieve this, we leverage WIKIDATA to construct in-context examples. Specifically, we pick a list of WIKIDATA entities e_1, \dots, e_{K_r} and for each entity e_i , extract its set of WIKIDATA relations. This results in K_r in-context examples for relation generation. We concatenate the target entity to the in-context examples, feed this prompt to the LM and use its output as the set R for e . Tab. 1 shows an example prompt. We note that this generation process can produce relations that are not included in the prompt, and are not part of WIKIDATA at all.² Full prompt with in-context examples is presented in Sec. B.1.

3.2 Relation Paraphrasing

A relation r may be described in multiple ways, and the LM might work better with some of these paraphrases ([Jiang et al., 2021](#)). Thus, we use a procedure to obtain a set of paraphrases of r , denoted by $P(r)$, and run all downstream crawling tasks for all strings in $P(r)$.

For relation paraphrasing we find that in-context examples are not necessary and an instruction prompt is sufficient. Tab. 1 shows a specific example under the sub-task “Relation Paraphrasing”. See Sec. A.1 for the three prompts and more technical details.

3.3 Object Generation

Our next goal is, for each $r \in R$, to generate a set of objects O such that (e, r, o) is in the KG for all $o \in O$. Importantly, we should also let the LM declare it does not know the object, and thus O would be empty. In this case, no edge will be added to the output KG.

²For example, when the subject is a sports team, the model repeatedly generated a relation regarding its MASCOT or LARGEST WIN, which are facts outside of WIKIDATA.

Sub-task	Query	Prompt	Expected Output
Relation Generation	Philippines	Q: René Magritte A: ethnic group, place of birth, place of death, sex or gender, spouse, country of citizenship, member of political party, native language, place of burial, cause of death, residence, family name, given name, manner of death, educated at, field of work, work location, represented by Q: Stryn A: significant event, head of government, country, capital, separated from Q: Philippines A:	leader name # cctld # capital # calling code
Pure Object Generation	Barack Obama # child	Q: Monte Cremasco # country A: Italy Q: Johnny Depp # children A: Jack Depp # Lily-Rose Depp Q: Wolfgang Sauseng # employer A: University of Music and Performing Arts Vienna Q: Barack Obama # child A:	Sasha Obama # Malia Obama
DK Object Generation	Queen Elizabeth II # date of death	Q: Heinrich Peters # occupation A: Don't know Q: Monte Cremasco # country A: Italy Q: Ferydoon Zandi # place of birth A: Don't know Q: Hans Ertl # sport A: mountaineering Q: Queen Elizabeth II # date of death A:	Don't know
Subject Paraphrasing	Alan Turing	Alan Turing is also known as:	The father of computing
Relation Paraphrasing	notable work	'notable work' may be described as	a work of 'great value' or a work of 'importance'

Table 1: The full list of sub-tasks in our approach, where for each sub-task we provide its name, a query, a corresponding prompt, and the expected output. In ‘DK Object Generation’ the prompt declares in one of the in-context examples that the model does not know the place of birth of Ferydoon Zandi, since querying for it leads to a wrong answer (the query with the wrong answer isn’t shown).

We first explain prompt construction without the use of “*Don’t Know*” output, and refer to this as “Pure Object Generation”. We take K_o entities e_1, \dots, e_{K_o} from WIKIDATA. For each entity e_i , we choose one of its relations r_i , and all the objects O_i for this entity-relation pair in WIKIDATA. This creates K_o examples for object generation. Similar to relation generation, the target entity-relation pair is concatenated to the K_o examples, and the list of objects is parsed from the generated LM output (see exact format in Tab. 1, under the sub-task “Pure Object Generation”, and the full prompt with in-context examples in Sec. B.2). Recall that for each relation, we have multiple paraphrases. To maintain high precision, we only accept objects that were generated by at least two realizations of the relation.

3.4 Learning to Output “*Don’t Know*”

A key desideratum for KGs is high precision, namely the facts in the graph should be correct with high probability. Towards this end, we want to prompt the LM to output “*Don’t Know*” (DK) for facts where it is likely to make an error.³

³A model might make an error because it is not confident about the answer, or because its training data contains false facts. In this work, we are agnostic to this distinction and our

But how do we know what the model does not know? To capture this, we find cases where the LM outputs erroneous facts, and use these to construct in-context examples with a DK target. For example, suppose we run ‘Pure Object Generation’ with $e = \text{BILL CLINTON}$ and $r = \text{CHILDREN}$ and the model outputs $O = \text{KLAY THOMPSON}$. We deduce that the model does not know who Clinton’s children are, and therefore, can add the example $e_i = \text{BILL CLINTON}, r_i = \text{CHILDREN}, o_i = \text{Don’t know}$ to the prompt. In other words, we find examples where o_i is *Don’t know* through cases where the model errs on its predicted objects. We then construct a prompt with a total of K_{dk} examples, half of which are failure cases where with $o_i = \text{Don’t know}$ and the other half are correct predictions. We refer to this as “DK Object Generation”. See the corresponding row in Tab. 1 and the full prompt with in-context examples in Sec. B.3.

3.5 Subject Paraphrasing

Similar to relations, an entity e may have several names, and it may be easier for the LM to complete the triplet $(e, r, ?)$ with one of these. Thus, we take a paraphrasing approach to extend an entity name e into a set $P(e)$. The procedure is identical prompt’s goal is to encourage generation of correct outputs.

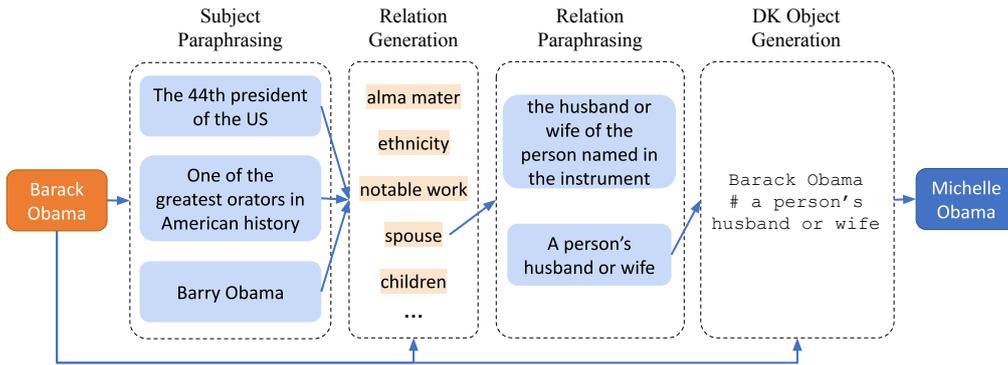


Figure 2: An illustration of the full method for crawling a subgraph (LMCRAWL), starting from BARACK OBAMA as the subject, until obtaining the triplet (BARACK OBAMA, SPOUSE, MICHELLE OBAMA).

to relation paraphrasing (Sec. 3.2), except we use a single prompt instructing the LM to complete the sentence “*s* is also known as”, where *s* is the subject. To increase the number of paraphrases, we sample from the model three times, resulting in up to three paraphrases.

Both here and in Relation Paraphrasing (Sec. 3.2), the LM occasionally generates nonsensical paraphrases. Nevertheless, the DK method handles those cases well, outputting “Don’t know” for most of them. Thus, we argue that paraphrasing combined with DK emission is an effective approach for controlling recall and precision.

3.6 LMCRAWL

Fig. 2 shows the application of the complete pipeline (which we refer to as LMCRAWL) for the entity BARACK OBAMA. First, we obtain all paraphrases for *e* (Sec. 3.5). Then, we extract all relations for these (Sec. 3.1). Next, we paraphrase relations (Sec. 3.2). Finally, we extract the known objects for these relations (Sec. 3.3-Sec. 3.4).

4 Experimental Setup

As mentioned in Sec. 3, we use WIKIDATA (publicly available) in constructing the in-context prompts. The number of in-context examples is $K_r = 7$, $K_o = 8$, $K_{dk} = 10$.

Additionally, we use WIKIDATA to select seed entities for evaluating our approach. For these seeds, we consider the task of constructing KGs around the corresponding entities.

We split the seed entities into a validation set (20 entities), which is used to make design choices (e.g., choosing prompt format), and a test set (120 entities), which is used only for the final evaluation.

For the development set, we manually chose 20 entities from WIKIDATA. These included women and men with various professions, cities, countries, and various cultural entities such as movies and books. We also aimed to represent both head and tail entities in this list.

To construct our test set, we defined 25 specific world-entities related categories, which we refer to as the *test categories*. Some of these were more specific, such as *AI Researchers*, and some are more general, such as *Scientists* (see Table.6 for the full list). We chose 4 seeds out of each category as follows. We first sorted the set of entities of each group based on the number of WIKIDATA facts associated with them (we view this count as an approximate measure of popularity). Then, we randomly sampled two entities out of the full list, and an additional two out of the first 1000. Intuitively, the first two represent tail entities, while the other two represents head ones. Thus we ended up with 100 seed entities (i.e., 4 different entities out of each of the 25 different subgroups). We refer to these as the *main test set* (see Tab. 6). We created an additional test set of 20 entities that is meant to contain very popular entities. Its entities were randomly sampled out of a set of size 1000, which was manually constructed by choosing 40 very well-known entities (i.e., that all people would know) from each of the 25 test categories.

All 140 entities were not used in the construction of any of the prompts in Sec. 3. Tab. 2 shows the full list of validation and head test entities.

Evaluation metrics Given an entity *s*, our entity expansion process returns a knowledge graph *G*, that contains the entity *s*, other entities and relations between them.

Dev Seeds	Head Test Seeds
ABBA	Aristotle
Alan Turing	Canada
Angela Merkel	Celine Dion
Augustin-Louis Cauchy	China
Barack Obama	Emanuel Macron
Bob Dylan	Franz Kafka
Boston Celtics	Grease
David Bowie	Hamlet
Diana, Princess of Wales	Jacinda Ardern
Eike von Repgow	Lionel Messi
Inglourious Basterds	Little Women
Marble Arch	Manchester United F.C.
Marie Curie	Margaret Hamilton
Mikhail Bulgakov	Michelangelo
Moby-Dick	Mike Tyson
Pablo Picasso	Oprah Winfrey
Paris	Rosalind Franklin
Philippines	Steven Spielberg
Rachel Carson	Serena Williams
Shahar Pe'er	The Rolling Stones

Table 2: List of all validation and head test seeds.

Ideally, we want to compare G to a ground truth graph that results from expanding the entity s . Given such a graph, we could measure precision and recall over the gold and predicted sets of triplets. However, using large LMs to generate graphs leads to several challenges. First, there is no ground-truth graph. While we could presumably use the WIKIDATA graph, we found that it is missing many correct facts predicted by the LM. In fact, improving coverage is a key motivation for our work! Second, facts may be reworded in several equivalent ways, rendering comparison between WIKIDATA graphs and predicted graphs difficult.

To circumvent these challenges, we use the following notions of precision and recall.

- **Precision:** To estimate precision we conducted both manual and automatic evaluations (the automatic approach was more scalable). For the manual evaluation we simply tried to validate each of the generated facts by manually browsing highly trustful web sources (Google, Wikipedia, etc.) to check if the fact is true. The automatic evaluation approach was implemented as follows. In order to check the correctness of a given predicted triplet (e, r, o) , we issue a query containing (e, r) to Google search, and search whether o appears in the result. We limit the result to first 40 words which are not HTML labels or URL links. If it does, we assume the triplet is correct.

⁴ See Sec. 5.3 for an accuracy estimation of the

⁴This paragraph typically contains either an “answer box” or some summary of the first result page, in case there is no answer box.

automatic method.

Manual evaluation was done for all the *head test set* graphs, as well as all the 1-hop graphs of the *main test set*. Additionally, we performed manual evaluation for 20% randomly sampled triplets from the 2-hop graphs (altogether, the total portion of manually labeled facts from each graph was $\sim 30\%$). The rest of the triplets were automatically evaluated.

- **Recall:** Estimating recall is not possible since we do not have access to the true ground truth graph. Moreover, using WIKIDATA graph size as an estimate for the number of true facts will be misleading since it has low coverage in general, and *high variance* in terms of coverage for different entities. Thus, we simply report the number of verified triplets in our KG. In other words, we report recall without the denominator. We refer to this as **# of facts**. This practice is similar to open information extraction (Vo and Bagheri, 2017), where it is impossible to know the set of all true facts and thus the convention is to report the number of generated facts only.

Implementation details As the LM in our experiments, we used the OpenAI text-davinci-002 model. We experiment with both greedy decoding and sampling 3 outputs per query (temperature 0.8). We generate graphs with either a single expansion step or two expansion steps, recursively expanding entities found in the first step. After a graph is generated, we remove duplicates by iterating through the facts and removing a fact if the token-wise F_1 between it and another fact is higher than 0.85.

Base Model and Ablations The simplest version of our model includes only ‘Relation Generation’ (Sec. 3.1) and ‘Pure Object Generation’ (Sec. 3.3), without the “Don’t Know” and paraphrasing components. We refer to this version as *Pure-Greedy* and *Pure-Sampling*, depending on the decoding used (see Sec. 4). In other model variants, we use *DK* to refer to using ‘DK Object Generation’ instead of ‘Pure Object Generation’. Additionally, *SP* and *RP* refer to adding ‘Subject Paraphrasing’ and ‘Relation Paraphrasing’ respectively.

5 Results

We next report results showing that our expansion method is able to generate meaningful knowledge subgraphs, when expanding seed entities.

	Main Test Set				Head Test Set			
	one-hop		two-hop		one-hop		two-hop	
	Precision	# of Facts	Precision	# of Facts	Precision	# of Facts	Precision	# of Facts
Pure-Greedy	54.6 ± 8.2	6.2 ± 2.8	43.4 ± 6.1	26.1 ± 5.5	80.3 ± 8.4	14.4 ± 3.9	62.1 ± 7.3	82.3 ± 15.4
LMCRAWL	83.3 ± 7.9	5.4 ± 1.1	82.0 ± 7.5	21.4 ± 4.7	91.5 ± 11.4	11.0 ± 4.6	90.9 ± 4.9	61.2 ± 25.1

Table 3: Averaged results across all 100 **main test** seeds (left), as well as all the 20 **head test** ones (right).

Example graph: We begin with an illustrative example for the graph of the seed entity ALAN TURING. Fig. 1 shows a subset of the two-hop extracted graph in this case. It can be seen that all facts are sensible, except for the fact that the field of Computer Science is named after Alan Turing (although he is certainly one of its fathers). See also Figs. 4 and 5 for additional example graphs.

Results on the Main Test set: Tab. 3 reports averaged results of the Pure-Greedy base model and LMCRAWL across the 100 main test seeds. We observe that precision of Pure-Greedy is too low to be useful for a KG – 54.6% for 1-hop graphs and 43.4% for 2-hop graphs. Conversely, precision with LMCRAWL is much higher: 83.3% for 1-hop graphs and 82.0% for 2-hop graphs. While we suffer a small hit in ‘# of facts’, the sizes of KGs output by our approach are quite reasonable.

Results on the Head Test set: Tab. 3 reports averaged results of the Pure-Greedy base model and LMCRAWL across the 20 head test seeds. Specifically, we achieve precision of **91.5%** while applying LMCRAWL for 1-hop graphs, and for 2-hop we have **90.9%**. It can be seen that both precision and number of facts in this case are higher than in the main test set. This suggests that either it is easier to extract facts from the LM about popular entities, or that the LM indeed encodes more facts for these (see Sec. 5.2 for further analysis).

5.1 Ablations

Next, we examine the contribution of each component in our final approach on the validation set.

The Effect of Don’t Know Generation: The goal of allowing the model to output “Don’t Know” is to improve precision. Tab. 4 and 5 show results for the model without using DK prompting (in *Pure* rows) as well as with (*DK* rows) for both sampling and greedy decoding. In both cases, the DK option leads to much higher precision, but reduces the number of generated facts. However, we later recover some of these lost facts using subject and relation paraphrasing.

Method	Precision	# of Facts
Pure-Sampling	64.9 ± 20.2	22.2 ± 9.7
Pure-Greedy	77.5 ± 17.4	12.5 ± 6.0
DK-Sampling	71.4 ± 19.9	17.7 ± 9.4
DK-Greedy	82.9 ± 16.0	10.2 ± 5.9
+RP	80.9 ± 17.0	12.7 ± 5.4
+SP	80.6 ± 17.0	12.2 ± 7.0
LMCRAWL	88.3 ± 8.2	13.0 ± 5.9

Table 4: Averaged results over the 20 **validation** seed (**one-hop**). DK: “Don’t know”. SP: Subject Paraphrasing. RP: Relation Paraphrasing.

Method	Precision	# of Facts
Pure-Sampling	40.0 ± 9.5	224.0 ± 81.1
Pure-Greedy	55.9 ± 9.7	87.8 ± 39.7
DK-Sampling	54.7 ± 8.6	144.0 ± 83.5
DK-Greedy	72.4 ± 7.5	45.8 ± 30.3
LMCRAWL	86.4 ± 6.1	69.8 ± 52.9

Table 5: Averaged results across all 20 **validation** seeds (**two-hop**). DK: “Don’t know”. SP: Subject Paraphrasing. RP: Relation Paraphrasing.

The Effect of Paraphrasing: Tab. 4 shows results without the paraphrasing component in the *DK-Greedy* row. Both paraphrasing techniques, RP and SP, separately increase coverage, while causing a minimal hit to precision. Interestingly, combining RP and SP leads to improvements in *both* precision and coverage for 1-hop *and* 2-hop graphs (Tab. 4, 5).

5.2 Coverage vs. Entity Frequency

The frequency of entities on the Web is highly skewed. That is, some entities appear many times, while others are rare. We expect this will be reflected in the number of facts extracted for these entities. Indeed, on WIKIDATA, head entities usually have many more facts compared to tail entities. Here, we ask whether a similar phenomenon exists in our predicted KGs.

Fig. 3 shows the number of facts generated for a depth-1 graph by LMCRAWL for all entities of type PERSON, as a function of the number of facts that appear in the corresponding depth-1 WIKI-

DATA graph of the same seed. Clearly, there is high correlation (correlation coefficient is 0.61) between the number of extracted facts and entity frequency on WIKIDATA. This is rather surprising and encouraging since our procedure does not make any use of entity frequency, and head and tail entities are expanded in exactly the same way.

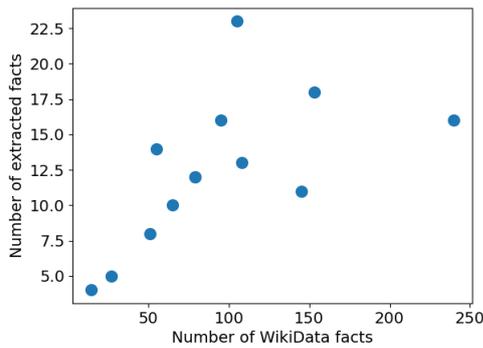


Figure 3: The # of triplets extracted by LMCRAWL as a function of the # of triplets in WIKIDATA, for the set of validation entities of type PERSON.

5.3 Precision is Possibly Underestimated

Our automatic approach for evaluating precision uses Google search (see Sec. 4). We view this as a conservative estimate of precision, since a fact judged as true via this mechanism is highly likely to be true. Conversely, a true fact might not be verified due to search or string matching issues. To quantify this, we sampled 500 generated facts from *Pure-Greedy* and LMCRAWL that were judged to be incorrect through Google search, as well as 500 that were judged to be correct. We manually inspected them and found that 4.1% of the triplets that the automatic approach has labeled as correct, are actually wrong, while 22% of the triplets that the automatic approach has labeled to be incorrect, are true (few demonstrations are presented in Sec. D). Exact estimation of precision would require *full* manual annotation, which we avoided to minimize costs.

6 Related Work

Pretrained LMs are at the heart of recent NLP research and applications. As mentioned earlier, Petroni et al. (2019) and other works have observed that LMs contain rich factual knowledge. We elaborate on other relevant works below.

Knowledge-base construction. KG construc-

tion typically involves both manual and automated aspects. For example, popular KBs such as WordNet (Fellbaum, 2020), ConceptNet (Speer et al., 2017) and WIKIDATA (Vrandečić and Krötzsch, 2014) were constructed by heavily relying on manual effort, gathering knowledge from humans. To reduce such manual labor, automated information extraction (IE) methods have been extensively developed (Yates et al., 2007; Fader et al., 2011; Angeli et al., 2015; Vo and Bagheri, 2017). Knowledge in LMs is a fairly recent topic of interest, and has mostly focused on probing for specific facts (Petroni et al., 2019; Razniewski et al., 2021).

Most similar to our work are Hao et al. (2022), who also extract KGs from LMs, However, they require defining the relations of interest through examples before crawling, while our specific goal is to start with a seed entity and allow the LM to determine the relevant relations. Another relevant recent work is Alivanistos et al. (2022) who also use in-context learning to extract a KG from GPT3. But they also assume relations are provided, whereas a key aspect of our approach is generating the relations.

To the best of our knowledge, ours is the first work to construct a knowledge graph via extracting knowledge directly from LMs, using only one seed entity (and no other given relations or entities).

Quantifying Uncertainty in LMs. Factual correctness in LMs has attracted recent interest, because it is a crucial requirement for LM applicability. In this context, some works have studied selective question answering, where LMs avoid answering particular questions (Varshney et al., 2022). Other works have considered calibration in LMs (Jiang et al., 2021; Desai and Durrett, 2020),

Finally, recent works have investigated whether models can express their certainty on output facts, either in words or by producing the probability of certainty (Lin et al., 2022; Kadavath et al., 2022). A key aspect of our approach is the use of a “Don’t know” mechanism, which is related to this line of work since it lets the LM declare its certainty as part of the output. Unlike Kadavath et al. (2022), we do so in the context of crawling a KG and via in-context learning (as opposed to fine-tuning).

7 Conclusion

Understanding large LMs is a key part of modern NLP, as they are used across the board in NLP applications. In particular, it is important to under-

stand the body of knowledge these models possess, so it can be used and revised as needed, thereby avoiding factual errors and biases. In this work we present an important step towards this goal by extracting a structured KB from an LM.

There are many possible exciting extensions for our work. The first is to expand it to a larger graph corresponding to more expansion hops. This would require many more calls to an API, which at present is also costly, and it would be important to develop more cost-effective approaches. Second, we have introduced several approaches to controlling the precision and recall of the proposed model, but certainly more can be envisioned. For example, we can introduce various consistency constraints to increase precision (e.g., check that FATHER OF and CHILD OF are consistent in the generated graph). Finally, once a larger KG has been extracted, one can query it to see how well it serves as a question answering mechanism.

Overall, we find the possibility of seamlessly converting LMs to KGs for better interaction and control to be an exciting and fruitful direction for future research.

Limitations

Producing the full internal KG out of an LM is still a significant challenge. One challenge is cost (as noted above). The other is error propagation issues. Once the model makes a generation mistake in a particular node of the generated graph, it may lead to an increasing number of mistakes during the next generation steps, expanding from that node. That is one of our main rationales for creating and evaluating only two-hop graphs, and not additional hops (although ideally, the real goal is to uncover the full internal KG).

Our automatic way of evaluating precision is only approximate, which means our reported accuracy numbers for 2-hop are an approximation of true precision (although we believe the true precision is in fact higher, as discussed in the text).

Another challenge we do not address is understanding the source of knowledge inaccuracies. Are they due to limitations of our model in extracting the knowledge, or due to the LM not containing these facts at all. This is certainly important to understand in order to improve knowledge representation in LMs. We are also aware to the fact that since the generated graphs are not perfectly accurate, they might contain disinformation and mis-

leading facts. That would hopefully be improved by future research.

Finally, the question whether we could have come up with a better-reflecting “recall” metric than the one we suggested is yet to be solved, as in general it is still unclear how to measure knowledge coverage.

References

- Dimitrios Alivanistos, Selene Báez Santamaría, Michael Cochez, Jan-Christoph Kalo, Emile van Krieken, and Thiviyan Thanapalasingam. 2022. Prompting as probing: Using language models for knowledge base construction. *arXiv preprint arXiv:2208.11057*.
- Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases. *arXiv preprint arXiv:2204.06031*.
- Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. pages 344–354.
- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Matusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. *Language models are few-shot learners*. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.
- Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. 2022. Compositional

- semantic parsing with large language models. *arXiv preprint arXiv:2209.15003*.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545.
- Christiane Fellbaum. 2020. *WordNet: An Electronic Lexical Database*. MIT Press.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.
- Shibo Hao, Bowen Tan, Kaiwen Tang, Hengzhe Zhang, Eric P Xing, and Zhiting Hu. 2022. Bertnet: Harvesting knowledge graphs from pretrained language models. *arXiv preprint arXiv:2206.14268*.
- Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. [How can we know when language models know? on the calibration of language models for question answering](#). *Transactions of the Association for Computational Linguistics*, 9:962–977.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.
- Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.
- Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Sebastian Miller, Alexander H. Riedel. 2019. Language models as knowledge bases? *EMNLP*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Simon Razniewski, Andrew Yates, Nora Kassner, and Gerhard Weikum. 2021. Language models as or for knowledge bases. *arXiv preprint arXiv:2110.04888*.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *EMNLP*.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. 31(1).
- Yi Tay, Vinh Q Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *arXiv preprint arXiv:2202.06991*.
- Neeraj Varshney, Swaroop Mishra, and Chitta Baral. 2022. Investigating selective prediction approaches across several tasks in iid, ood, and adversarial settings. *arXiv preprint arXiv:2203.00211*.
- Duc-Thuan Vo and Ebrahim Bagheri. 2017. Open information extraction. *Encyclopedia with semantic computing and Robotic intelligence*, 1(01):1630003.
- Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.
- Denny Vrandečić and Markus Krötzsch. 2014. [Wikidata: A free collaborative knowledge base](#). *Communications of the ACM*, 57:78–85.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26.

A Technical Details

A.1 Relation-Paraphrasing

We use 3 different instructions that have been manually constructed. If we denote a specific relation by r , then they are:

- " r may be described as"
- " r refers to"
- "please describe ' r ' in a few words:"

That is, for every original relation which has been generated by the model, we perform additional three different model calls, one with each of those instruction prompts, resulting in three paraphrases. If needed, we eliminate overlapping paraphrases.

B Full Prompts

B.1 Relation Generation

Q: Javier Culson
A: participant of # place of birth # sex or gender # country of citizenship # occupation # family name # given name # educated at # sport # sports discipline competed in

Q: René Magritte
A: ethnic group # place of birth # place of death # sex or gender # spouse # country of citizenship # member of political party # native language # place of burial # cause of death # residence # family name # given name # manner of death # educated at # field of work # work location # represented by

Q: Nadym
A: country # capital of # coordinate location # population # area # elevation above sea level

Q: Stryn
A: significant event # head of government # country # capital # separated from

Q: 1585
A: said to be the same as # follows

Q: Bornheim
A: head of government # country # member of # coordinate location # population # area # elevation above sea level

Q: Aló Presidente
A: genre # country of origin # cast member # original network

B.2 Pure Object Generation

Q: Kristin von der Goltz # mother
A: Kirsti Hjort

Q: Monte Cremasco # country
A: Italy

Q: Johnny Depp # children
A: Jack Depp # Lily-Rose Depp

Q: Theodor Inama von Sternegg # place of birth
A: Augsburg

Q: Wolfgang Sauseng # employer
A: University of Music and Performing Arts Vienna

Q: Hans Ertl # sport
A: mountaineering

Q: Nicolas Cage # sibling
A: Christopher Coppola # Marc Coppola

Q: Manfred Müller # occupation
A: Catholic priest

B.3 DK Object Generation

Q: Heinrich Peters # occupation
A: Don't know

Q: Monte Cremasco # country
A: Italy

Q: Nicolas Cage # sibling
A: Christopher Coppola # Marc Coppola

Q: Hans Ertl # sport
A: mountaineering

Q: Klaus Baumgartner # work location
A: Don't know

Q: Ruth Bader Ginsburg # educated at
A: Cornell University # Harvard Law School # Columbia Law School

Q: Ferydoon Zandi # place of birth
A: Don't know

Q: Wolfgang Sauseng # employer
A: University of Music and Performing Arts Vienna

Q: Apayao # head of government
A: Don't know

Q: Kristin von der Goltz # mother
A: Don't know

C Main Test Set

Table 6 provides our main test, which includes 100 different seeds - 4 from each of our predefined entity group categories.

D Automatic Precision Evaluation

As noted in the main text, the automatic precision evaluation method (i.e., the one based on Google search) may sometimes fail. Some of the failure cases are: (a) *Inexact string matching*. For example (BOSTON CELTICS, LEAGUE, NATIONAL BASKETBALL ASSOCIATION (NBA)) is not verified, but dropping (NBA) from the object would result in a successful verification. b) *Paraphrases*: For example (MARBLE ARCH, COUNTRY, UNITED KINGDOM) is not verified but changing the object to ENGLAND does succeed.

E Additional Generated Graphs

Figs. 4, 5 show additional example graphs (to the one shown in Fig. 1), generated around the seed entities ANGELA MERKEL and BOSTON CELTICS respectively.

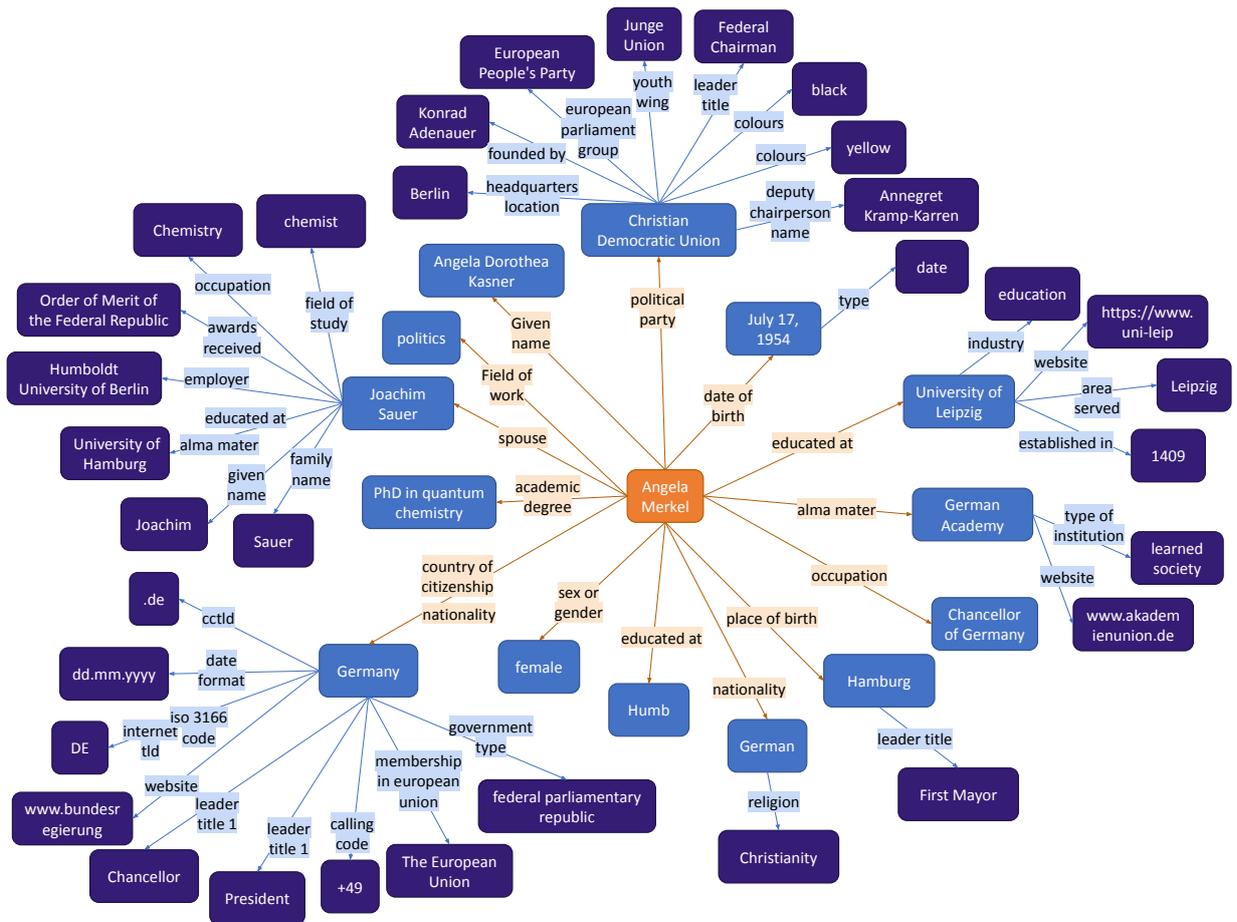


Figure 4: An example of a generated depth-2 knowledge graph around the seed entity ANGELA MERKEL, using LMCRAWL(see Sec. 3). For readability, back edges from 2-depth nodes to 1-depth nodes are omitted.

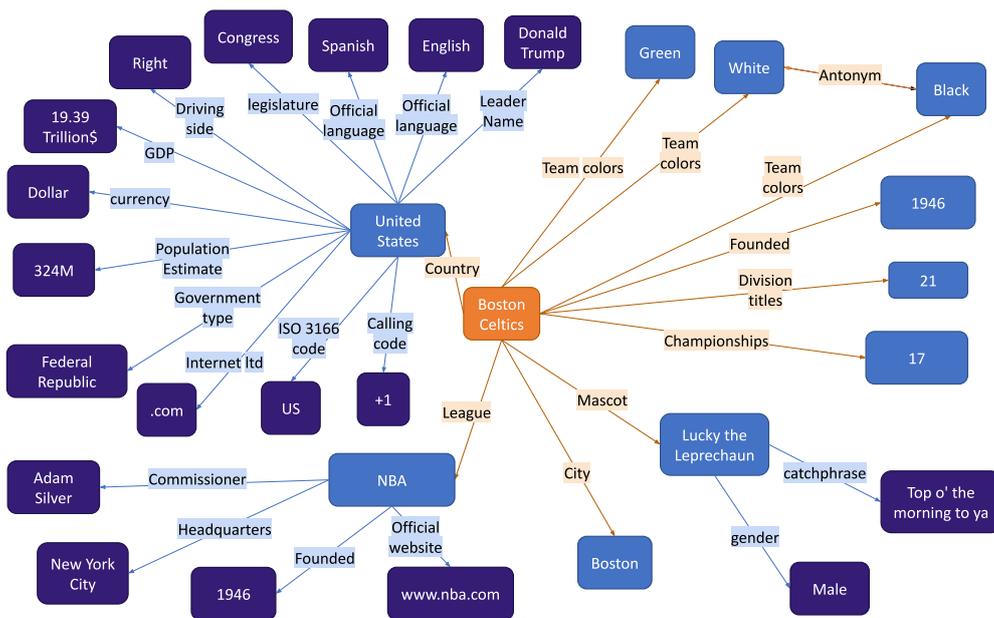


Figure 5: An example of a generated depth-2 knowledge graph around the seed entity BOSTON CELTICS, using LMCRAWL (see Sec. 3).

Categories	Sampled Seeds	Categories	Sampled Seeds
Politicians	Wang Zhi Cathy Rogers Kate Wilkinson Carles Campuzano	Producers	Alyssa Milano Lenny Kravitz Carter Harman Nancy Meyers
Scientists	Pavel Krotov Mirra Moiseevna Gukhman Axel Delorme Jesús Caballero Mellado	Actors	Jon Voight Boris Savchenko Tolga Tekin Virginia Keiley
Basketball	Tom McMillen Pat Kelly Steve Moundou-Missi Allen Phillips	Singers	Freddie Mercury Angélique Kidjo Camille Thurman Giorgio Ronconi
Sports	Peteca motorcycle racing Basque pelota mountain bike trials	Bands	Steve Miller Band Hypocrisy Afro Kolektyw Frailty
Artists	Bořek Šípek Loriot George William Wakefield George Trosley	TV Shows	Secrets and Lies Spirited Away Super Friends The Life and Legend of Wyatt Earp
Paintings	Portrait of a Man Landscape The foot washing The King's rival	Foods/Restaurants	Tahu petis Kandil simidi Jim Block kubang boyo
Writers	Aleksandr Volkov Osamu Tezuka Elizaveta Sergeevna Danilova Henry Saint Clair Wilkins	Animals	donkey jaguar mustang whale
Books	The Green Berets Alfred de Musset Demain le capitalisme The labyrinth	Plants	maple rose catmint conflower
Landmarks	Trafalgar Square Mount Everest Yosemite National Park Matterhorn	Architects	Louis Kahn Christopher Wren Michael Graves Domenico Fontana
Cities	Vatican City Cherdyn Toulon MiljøXpressen	Drummers	Alan Montagu-Stuart-Wortley-Mackenzie Mihály Deák Joey Kramer Stephanie Eulinberg
Countries	Niger Sweden England Singapore	Biologists	Wangari Muta Maathai James Rothman Joanna Siódmiak Barbara Bajd
Philosophy	Evgeny Torchinov Nikolay Umov Monica Giorgi Larysa Tsitarenka	AI Researchers	William T. Freeman Stephen Falken Joseph Weizenbaum Robby Garner
Movies	Spider-Man: Far from Home Sonic the Hedgehog Unearthed Another Man's Poison		

Table 6: List of all main test set seeds

Intent Identification and Entity Extraction for Healthcare Queries in Indic Languages

¹Ankan Mullick ²Ishani Mondal* ¹Sourjyadip Ray* ³R Raghav

¹G Sai Chaitanya and ¹Pawan Goyal

{¹ankanm, ¹sourjyadipray}@kgpian.iitkgp.ac.in, ²imondal@umd.edu

³rraghavr@cs.cmu.edu, {¹gajulasai@, ¹pawang@cse.}iitkgp.ac.in

¹Computer Science and Engineering Department, IIT Kharagpur, India

²University of Maryland, College Park, USA

³School of Computer Science, Carnegie Mellon University, USA

Abstract

Scarcity of data and technological limitations for resource-poor languages in developing countries like India poses a threat to the development of sophisticated NLU systems for healthcare. To assess the current status of various state-of-the-art language models in healthcare, this paper studies the problem by initially proposing two different Healthcare datasets, Indian Healthcare Query Intent-WebMD and 1mg (IHQID-WebMD and IHQID-1mg) and one real world Indian hospital query data in English and multiple Indic languages (Hindi, Bengali, Tamil, Telugu, Marathi and Gujarati) which are annotated with the query intents as well as entities. Our aim is to detect query intents and extract corresponding entities. We perform extensive experiments on a set of models in various realistic settings and explore two scenarios based on the access to English data only (less costly) and access to target language data (more expensive). We analyze context specific practical relevancy through empirical analysis. The results, expressed in terms of overall F1 score show that our approach is practically useful to identify intents and entities.

1 Introduction

Healthcare is a top priority for every country. People across the world ask millions of health-related queries, hoping to get a response from a domain expert (Gebbia et al., 2020). These queries mostly deal with medical history of patients, possible drug interactions, disease related concerns, treatment protocols and so on. Conversational agents for healthcare play a pivotal role by facilitating useful information dissemination (Li et al., 2020; Maniou and Veglis, 2020). In order to understand these

queries better, practical conversational systems for healthcare need to be developed. However, the primary obstacle in developing such technologies for low-resource languages is the lack of usable data (Mehta et al., 2020; Daniel et al., 2019; Liu, 2022).

India is a country with a diverse language speaking population suffering from abject poverty and low-economic status (Mohanty, 2010; Pande and Yazbeck, 2003). This linguistic diversity and complex socio-economic situation in India certainly poses significant challenges in developing automatic healthcare systems; and there is a lack of linguistic resources specific to the medical domain. For example, situations such as the patient and the doctor speaking in different languages, is not an uncommon situation in rural India. These individuals are unable to avail the existing systems and facilities which exist mainly in the English language. Recent efforts in developing automatic translation systems, even from extremely low resource languages such as ‘Mundari’ and ‘Gondi’ (Joshi et al., 2019), should ideally improve this situation, but there is no extensive study on that front.

In order to bridge this language barrier, massively Multilingual Transformer based Language Models (MMLM) (Devlin et al., 2019; Lample and Conneau, 2019) have made impressive advancements on a wide range of downstream applications. But the real-world implications of such advancements in the Indian healthcare system remain largely unexplored. In this paper, we aim to explore scarcity of the data and study the extent to which the existing language technologies can be leveraged to develop practically useful healthcare systems for the low-resource languages in developing countries.

With an aim to answer our research question,

* Authors contributed equally

Intent	Language	Query	Entities		
			disease	drug	treatment
treatment	English	can anesthesia during surgery cause memory loss or signs of senility?	memory loss, signs of senility	anesthesia	surgery
	Hindi	क्या सर्जरी के दौरान बेहोशी स्मृति हानि या बुढ़ापा के लक्षण पैदा कर सकता है?	स्मृति हानि, बुढ़ापा के लक्षण	बेहोशी	सर्जरी
	Bengali	অস্ত্রোপচারের সময় অ্যানেস্থেসিয়া কি স্মৃতিভ্রংশ বা বার্ধক্যের লক্ষণ প্রদর্শনের কারণ হতে পারে?	স্মৃতিভ্রংশ, বার্ধক্যের লক্ষণ	অ্যানেস্থেসিয়া	অস্ত্রোপচার

Figure 1: Example of a query of ‘treatment’ intent category for different languages along with associated entities.

we create two different multilingual healthcare datasets, namely, IHQID-WebMD and IHQID-Img. These datasets are created by crawling frequently asked questions from two healthcare websites, *WebMD* and *Img*. These datasets comprise frequently asked questions about drugs, diseases and treatment methods in seven different languages, namely English, Hindi, Bengali, Tamil, Telugu, Gujarati and Marathi. The queries are manually tagged with intent labels and entity tags by domain-experts and translated by native speakers of the corresponding languages. We also collect real world Indian hospital queries (annotated) in seven languages to check the empirical effectiveness of our approach. Fig. 1 shows an example of a health query belonging to ‘treatment’ intent class manually translated into three different languages. Then we evaluate the performance of state-of-the-art language models (LMs), for both English and multilingual setups on our datasets, to answer the questions regarding their deployability and practicality. Various experimental configurations (Section 4) have been tried on these datasets where we try to figure out the ways of using best technologies through extensive experimentation in two real-world scenarios. First, we assume to have access to only English training queries (less costly) and the test queries are multilingual in nature. We observe that translate-test setup on RoBERTa seems to be a reasonable choice of technology. Second, we assume to have access to manually written multilingual training and test queries in the target languages, which is indeed quite expensive in terms of data collection effort. However, back-translation of both train and test queries proves to be a reasonable choice if we have budget of collecting data in target languages.

In sum, our contributions are four folds:

- We propose two intent and entity labelled Indian healthcare datasets (annotated by domain-experts) comprising of frequently asked ques-

tions from users.

- Even though the large language models have proved their effectiveness in almost every NLU operation, we want to determine their effectiveness in determining the correct intent and slot filling operations for practical domain-specific healthcare scenarios in the Indian context. We intend to analyze how should we prioritize the research and resource building investments for the economically backward countries with a high percentage of multilingual population? This will make us aware about the best techniques of deploying the language models in various scenarios such as: availability of English training data vs multilingual training data. Keeping this in mind, all our experiments have been carried out using both monolingual and multilingual setups of these models. Through our experiments, we try to point out the best possible language models and techniques to develop practically useful NLU solutions (pipeline based approach for intent detection and corresponding entity extraction from the queries).
- Through extensive experiments on the datasets, we recommend the community to use back-translation of test queries to English in two real-life scenarios as a reasonable choice when we have access to English training data. However, the same strategy can be applied to both train and test queries if we have the budget of collecting data in target languages.
- Our findings imply that the back-translation of queries using an intermediate bridge language proves to be a useful strategy in the intent recognition experiments.

2 Related Work

We pivot our study of related works into the following buckets - generalised intent and entity detection, entity and intent detection in healthcare, health care in Indian languages and multi-lingual healthcare datasets.

A) Generalised Intent and Entity Detection Approaches: (Sun et al., 2016; Wang et al., 2020; Mu et al., 2017b,a) focus on detecting novel intents in the form of outlier detection. (Mullick et al., 2022a) explore intent classification on legal data. People also work on different detection approaches - few shot (Xia et al., 2021), zero shot (Xia et al., 2018), clustering frameworks (Mullick et al., 2022b). (Yani et al., 2022; Sufi and Alsulami, 2021; Zhao et al., 2021) all explore entity detection tasks. (Vanzo et al., 2019) develop a hierarchical multi-task architecture for semantic parsing sentences for cross-domain spoken dialogue systems. Most of these approaches are very domain and language specific and thus not very useful for the healthcare domain in Indian languages.

B) Entity and Intents in Health Care: Zhou et al. (2021) solve different tasks in smart healthcare. Bao et al. (2020) build a chat-bot framework using user intents. Bai et al. (2022) aim at incremental medical intent detection. Razzaq et al. (2017); Amato et al. (2017) develop an e-Health application using intent-context relationships. Zhang et al. (2017) explore medical query intents. Most of the works are done for English and Chinese languages and there is no proper architecture for Indian multi-lingual scenarios for intent and entity extraction..

C) Health Care in Indian Languages: Some researchers focus on Indian Languages - Hindi Medical Conversation system, MedBot (Bharti et al., 2020), detecting Hindi and English COVID-19 posts (Patwa et al., 2021), Tamil health information (Santosh, 2019), Bengali health-bot (Badlani et al., 2021), Telugu COVID-19 health information (Vishwas et al., 2017). But none of the work aims at Indian health query datasets and model analysis. (Mondal et al., 2022) highlights the gaps when using existing state-of-the-art commercial frameworks for NLU tasks in a few Asian and African low-resource languages, especially when the goal is to develop conversational agents for healthcare during COVID. In our work, we strengthen the claims made in their paper for generic healthcare specific datasets in Indian context, and highlight the potential drawbacks of the existing LMs.

D) Multilingual Health Care Dataset: Liu et al. (2020) develop MedDG (Medical Dialogue dataset of common Gastrointestinal diseases) in Chinese. Zeng et al. (2020) proposes MedDialog, a Chinese and English medical dataset, and explores medical dialogue generation tasks. Zhang et al. (2021) build a medical intent evaluation dataset in Chinese and Kim et al. (2022) has constructed a Korean health intent dataset. Our work differs from the existing research in two ways: 1) We focus on developing a gold standard healthcare NLU dataset in Indian languages, 2) cost parameter and availability oriented usage of models for intent detection and entity extraction, and 3) end-to-end evaluation of the state-of-the-art solutions for healthcare in both English and Indic languages which leads to interesting implications and generates important future recommendations for the language community.

3 Dataset and Pre-Processing

3.1 Necessity of a new dataset

India is a country with a diverse language speaking population. There is an increasing population of users consuming Indian language content. This linguistic diversity certainly poses significant challenges in healthcare setup, particularly in the situation when healthcare providers and patients speak different languages (also termed as *Language Discordance*) (Shamsi et al., 2020). Therefore, individuals with limited English proficiency are left behind and suffer from worse health outcomes than those who speak English with high proficiency. The growing need for the deployment of multilingual conversational agents in hospital and healthcare facilities in India, especially highlighted by the plight of the healthcare workers during the COVID-19 pandemic, warrants a multilingual healthcare query intent dataset in Indian languages (Daniel et al., 2019). Therefore, we resort to create two novel Indian Healthcare Query Intent Datasets - (IHQID-WebMD and IHQID-1mg) and one real-world healthcare dataset from hospitals.

3.2 Source of the dataset

Due to the unavailability of open-source multilingual NLU datasets in healthcare setup, we sample frequently asked medical queries (FAQs) in English from two popular data sources:

WebMD¹: It is an American website containing a large repository of healthcare data. The queries,

¹<https://www.webmd.com/>

Class	Intents		Entities		Real World Hospital Query Data (#Intent / #Entity)						
	#WebMD	#Img	#WebMD	#Img	#En	#Hi	#Bn	#Ta	#Te	#Ma	#Gu
Disease	283 (207+76)	111 (87+24)	629 (464+165)	240 (185+55)	28/37	31/35	29/37	27/35	31/39	28/35	29/35
Drug	234 (181+53)	198 (144+54)	400 (302+98)	224 (166+58)	34/44	33/43	31/37	30/35	32/38	34/40	32/37
Treatment	166 (127+39)	67 (46+21)	218 (165+53)	64 (44+20)	21/24	20/26	21/25	23/29	19/24	17/23	20/26
Other	278 (205+73)	41 (28+13)	-	-	17/-	16/-	19/-	20/-	18/-	21/-	19/-
Total	961 (720+241)	417 (305+112)	1247 (931+316)	528 (395+133)	100/105	100/104	100/99	100/99	100/101	100/98	100/98

Table 1: Distributions of different types of intent and entity labels in WebMD, 1mg datasets (IHQID) and Real World Hospital Query Data. (- + -) represents (train + test) division. # denotes the count.

taken from the WebMD health forum are asked by ordinary users regarding a wide range of problems. **1mg²**: 1mg is an Indian website, which is also a rich source for healthcare data, especially in the Indian context. The English queries are scraped from the FAQ section in drug and disease pages.

Although, both the above datasets are curated from online forums where users post healthcare concerns, in order to evaluate our approach in a practical Indian context, we develop a real world healthcare query dataset in Indian scenario. We collect real world healthcare queries (asked by patients) from the doctors in local hospitals. All queries are anonymous without identity or any details of the patients. For each language, we fetch 100 queries (some of which overlap) belonging to different categories.

3.3 Dataset Sampling

The FAQs sampled from these data sources are unlabeled. Hence, for the purpose of supervised classification, it is necessary to categorize each query into a specific intent and list of corresponding entities. We broadly categorize queries into four different intent types, namely, ‘Disease’, ‘Drug’, ‘Treatment Plan’ and ‘Other’. Each query is assigned one of the four intent labels. Two English-speaking medical graduate doctors annotate the intents from the English queries to prepare the datasets. Annotators also mark entities, belong to three different medical entity categories present in the datasets - ‘Disease’, ‘Drug’ and ‘Treatment’. The queries with their intent labels are retained where both annotators agree, otherwise discarded. On an average, this filtering lead to an average rejection of around 10% samples of the dataset for all our setups and languages. Overall Inter-annotator agreement, Cohen κ is 0.89.

²<https://www.1mg.com/>

3.4 Parallel Data Generation

In order to generate parallel corpora of these frequently asked questions in English, we choose six Indian languages apart from English.

Language Selection: The language set includes English: USA version (EN-US) termed as (‘En’), Hindi (‘Hi’), Bengali (‘Bn’), Tamil (‘Ta’), Telugu (‘Te’), Gujarati (‘Gu’) and Marathi (‘Mr’). The choice of languages was driven by (a) the number of native speakers of those languages in India, (b) number of annotators available for creating the dataset, (c) combined with typological diversity amongst the languages - we choose languages from various language families. For instance, Bengali, Hindi, Gujarati, Marathi belong to the Indo-Aryan family whereas Tamil and Telugu belong to the Dravidian group.

Annotation and Quality Control: Since the gold standard annotated queries are not available online in Indian languages, the English queries of 1mg and WebMD have to be manually translated. **After discussions with the doctors and different patients, we create the annotation guidelines.** Annotators are told to formulate the queries on their own regional languages with the help of Bing Translator API³. Annotators are also asked to annotate the entities and their types (in their respective native languages) for each query being corrected with the idea of what common people of corresponding native language generally ask healthcare queries to doctors.

Three annotators are selected per language after several discussions and conditions of fulfilling many criteria like annotators should have native proficiency in their language of annotation, domain knowledge expertise along with a good working proficiency in English. Initial labeling is done by two annotators and any annotation discrepancy is checked and resolved by the third annotator after discussing with others. While formulating the

³<https://www.microsoft.com/en-us/translator/business/translator-api/>

query on their own manually, the annotators are also asked to annotate the entities and their types (in their respective native languages) for each query being corrected. The above quality control measures ensure that the translated data is of high quality, resembling real world data in the target language. In the case of a word such as a proper noun like ‘*Paracetamol*’ (drug), which does not have a translation in the target vocabulary, the word is asked to be simply transliterated in the target language.

In order to prepare the real world hospital query dataset in Indian healthcare contexts, we collect healthcare queries from the doctors of local hospitals. It also consists of six different Indic languages along with English. There are a hundred queries for each of the language. These queries also have similar intent classes and entity categories, which are labelled by the doctors. During collection of queries, we fix the minimum number of samples for each intent classes across all languages.

In order to maintain the quality of the Indian language annotations, the annotators are directed to use the native language words and grammar, keeping the original interpretation of the query. All query logs, annotations and changes are recorded in order to conduct future verification and analysis. On completion of the translation process, the annotators are asked to exchange their work and check the quality of translation for fluency and semantic stability. Inaccuracies are noted, and the respective queries are rectified in the dataset.

At the end, we finally have three multilingual intent and entity recognition labelled datasets - **IHQID-WebMD**, **IHQID-1mg** and a real world hospital query test dataset in seven different languages, the dataset distributions of which are provided in Table 1. The first two datasets (IHQID-WebMD and IHQID-1mg) help to build the models and real world hospital dataset is used to evaluate our approaches in real world contexts. Table 1 also shows the statistical details across different intent classes (‘Disease’, ‘Drug’, ‘Treatment’ and ‘Other’) and corresponding entities (of ‘Disease’, ‘Drug’ and ‘Treatment’ categories) along with the total counts and train-test divisions. It also shows the distribution of hospital collected practical healthcare queries across different languages (Right part of the table).

4 Strategies of Evaluation

In this section, we illustrate the strategies of evaluating the state-of-the-art LMs on our dataset. Our evaluation of these models for Healthcare is scoped down to two fundamental NLU tasks:

- a) Intent Recognition (Section 5.1)
- b) Entity Extraction (Section 5.2)

Evaluation Setup Description: Our evaluation of the models has been conducted while keeping in mind about the availability of human-translated monolingual and multilingual training data in two possible real-life scenarios: 1) **Scenario A:** In this setup, we assume to have access to only English training data (less costly) and in 2) **Scenario B:** we assume to have access to manually written training queries in all the target languages (very expensive). During inference/testing, we expect all the queries are in the corresponding target languages.

Scenario A:

Setup 1) Backtranslated Test (S1): [Translate-Test] Here we develop our system by training the models on the English queries, and evaluate the intent detection and entity extraction systems in different languages by automatically backtranslating the test queries into English (e.g. similar to (Gupta et al., 2021)). **Setup 2) Zero-Shot Cross-Lingual Test (S2):** Cross lingual transfer learning is a useful methodology used for tasks involving scarce data (Zhou et al., 2016; Karamanolakis et al., 2020). In this setup, the models make use of zero-shot based cross-lingual capabilities from training on the English data (scraped from WebMD and 1mg) and use it for inference on test queries in Indic languages. **Setup 3) Bridge Language Back-translation (S3):** Here a relatively low-resource language is first translated to an intermediate language and then finally to English. The motivation behind this setup lies in the fact that even though these Indic languages belong to different scripts, there are linguistic and morphological similarities among them which may improve the translation to English if they are used as intermediate languages. In this paper, we have considered ‘Hindi’ as the bridge language. This notion of such “bridge” languages has been explored previously in the context of Machine Translation (Paul et al., 2013) and zero/few-shot transfer in MMLMs (Lauscher et al., 2020).

Scenario B:

Setup 4) Train and Test on Indic Data (S4): In this setup, we use the training dataset in indic lan-

languages to train our NLU models in different target languages. Here, we use the IHQID-WebMD and IHQID-1mg Indic data (non-English) to evaluate the NLU detection performances of the developed models. Jennifer Bot (Li et al., 2020) use a similar setup to extend their English bot to Spanish. **Setup 5) Full Backtranslation (S5):** In this setup, both train and test data are backtranslated to English. This is useful for the countries with poor technical setups for low-resource languages, since an automated approach can translate low-resource medical queries to resource-rich language and test.

In all back translation experiments, we use Bing Translation Api ⁴.

5 Experiments and Results

Experimental Setup: Our experiments are conducted on two Tesla P100 GPUs with 16 GB RAM, 6 Gbps clock cycle and GDDR5 memory. All methods of entity extraction and intent detection took less than 30 GPU minutes for training. We perform a hyperparameter search and report the results of the settings which achieve the best results, and then fixed the same for all the models. The batch size is kept at 16, number of epochs is 10, optimization algorithm used is AdamW and the learning rate is 1e-5 with cross-entropy as the loss function.

5.1 Intent Detection

Task Description: It can be defined as a multi-class classification task of correctly assigning a medical query with an intent label from a fixed set of intents (*drug, disease, treatment and other*).

Classification Models: Since in Setups 1, 3 and 5, we take both the training and test set in English, we use state-of-the-art LMs pre-trained on English corpora (as shown in (i)) for our classification experiments. Whereas in Setup 2 and 4, we make use of multilingual LMs (as shown in (ii)) which have been widely used for various benchmark tasks in Indian languages. Following are the baselines:

(i) **Pre-trained English Models:** For setups 1, 3 and 5, we fine-tune the last layer of RoBERTa (Liu et al., 2019) and Bio_ClinicalBERT (Alsentzer et al., 2019) models on the English queries for intent detection by adding a classification layer that takes [CLS] token as input. The latter is a state-of-the-art domain-specific transformer based

language model pre-trained on MIMIC III notes⁵, which is a collection of electronic health records and discharge notes.

(ii) **Pre-trained Multilingual Models:** Two pre-trained multilingual LMs are used, mBERT (*bert-base-multilingual-uncased*) (Pires et al., 2019) and XLM-Roberta (*xlm-roberta-base*) (Conneau et al., 2020), both support all Indic languages in the datasets along with English. In Setup 2, we perform zero-shot classification using these models. The zero shot setting involves fine-tuning the model using English data, and testing on Indic languages. Whereas in Setup 4, we first train these models using the entire train sets in the target languages, separately for WebMD and 1mg, and check the performance on the test sets.

5.2 Entity Recognition

Task Description: This task is analogous to performing a Named Entity Recognition (NER) for three categories, namely, *drugs, diseases and treatments* on the query texts. We follow the standard BIO-tagging system while annotating the entities word-by-word. The train and test files for each configuration and language respectively are constructed from our WebMD and 1mg datasets.

Extraction Frameworks: For entity recognition, we follow the same strategies of evaluating the predictive performance of the LMs as described in Section 4. The same models (as described in section 5.1) are also used for entity recognition experiments.

5.3 Evaluation

For all our experiments on intent detection and entity recognition, we calculate the Precision, Recall and report the F1-score.

5.4 Results and Analysis

Intent Detection: Table 2 shows the results of intent detection of five experimental strategies on the IHQID-WebMD and IHQID-1mg datasets in terms of Macro F1-score (in percentage).

Finding 1: We observe that in general, Backtranslated Test (Setup 1) performs better than Zero-Shot Cross-Lingual Test (Setup 2). Moreover, it is interesting to notice that even though the performance of these models for most of the target languages in Setup 1 are comparable with that of English in

⁴<https://www.microsoft.com/en-us/translator/business/translator-api/>

⁵https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT

Lang -uage	Backtranslated Test (S1)				Zero-Shot Cross-Lingual Test (S2)				Bridge Language Backtranslation (S3)				Train and Test on Indic Data (S4)				Full Backtranslation (S5)			
	RoBERTa		bcBERT		mBERT		XLM-RoBERTa		RoBERTa		bcBERT		mBERT		XLM-RoBERTa		RoBERTa		bcBERT	
	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg
En	<u>76.34</u>	<u>73.33</u>	75.38	68.72	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Hi	73.90	67.48	66.50	66.50	42.46	46.45	58.68	43.30	-	-	-	-	56.18	51.41	41.14	40.09	<u>75.42</u>	<u>72.32</u>	75.21	63.81
Bn	75.18	66.42	75.02	63.66	35.85	35.62	55.85	43.69	70.76	<u>71.94</u>	71.91	64.87	50.26	46.65	41.07	39.73	<u>78.83</u>	<u>70.13</u>	75.41	57.52
Ta	73.63	64.29	73.99	62.88	38.50	39.34	57.47	42.14	69.51	66.42	73.06	64.43	51.17	50.49	40.04	32.63	<u>74.36</u>	<u>69.44</u>	73.79	62.82
Te	73.25	63.48	73.79	66.17	36.40	30.75	55.38	38.53	69.89	66.63	72.19	63.67	51.28	51.69	45.26	41.07	71.80	<u>66.90</u>	<u>75.30</u>	65.63
Gu	71.76	66.85	73.05	68.80	35.61	32.67	51.50	34.58	71.61	72.07	<u>73.75</u>	68.02	50.07	51.17	42.23	46.23	72.93	<u>72.54</u>	<u>71.76</u>	70.52
Mr	72.70	70.64	73.47	73.26	43.57	38.22	60.58	44.16	71.50	72.47	73.13	<u>74.28</u>	54.44	55.24	43.18	46.11	<u>76.32</u>	70.65	75.42	63.83
Avg	73.82	67.50	73.03	67.14	38.73	37.18	56.58	41.07	70.65	69.91	67.25	67.05	52.23	51.10	42.15	40.98	<u>74.94</u>	<u>70.33</u>	74.48	64.03

Table 2: Macro-F1 scores for intent classification on the WebMD (WMD) and 1mg datasets for five Setups (three different setups for Train on English (Scenario A) and two setups of Train on Indic Data (Scenario B)). bcBert indicates BioClinicalBERT, mBERT indicates Multilingual BERT. Underline denotes the best across five settings.

WebMD (an average of 3% drop for all the languages compared to English), there is a significant drop (average of 6%) in the F1 scores for the Setup 1 results in 1mg Dataset. This holds true for both RoBERTa and BcBERT experiments. This denotes that the state-of-the-art English models, which are performing decently after backtranslation of the medical queries in English, pre-trained on both generic and medical domain, are lagging behind when the vocabularies of the medical entities are in the Indian context. This definitely calls for an immediate attention to developing LMs pre-trained on India-specific medical datasets.

Finding 2: Another interesting observation was that the use of Bridge Language Backtranslation (Setup 3) in Table 2, helps to boost performance of most of the languages in the case of 1mg dataset in comparison to Setup 1. The observation does not hold true for intent recognition in WebMD dataset. This might be attributed to the fact that using a bridge Indian language as an intermediate helps preserve the domain-specific sense of the queries instead of directly converting the queries from the target language to English. This seems like a reasonable alternative to develop useful intent recognition models for healthcare in Indian languages.

Finding 3: In comparison with zero-shot cross-lingual transfer (Setup 2), both mBERT and XLM-

R models are outperformed by few-shot experiments (Setup 4) for intent detection. This observation holds true for both WebMD and 1mg datasets. However, Setup 4 is much more cost-intensive than the Setup 2.

Finding 4: We report the average (Avg) F1-score across all languages. The best performing model is RoBERTa (Setup 1 for English and Setup 5 for non-English) for both WebMD (74.94%) and 1mg (70.33%). RoBERTa is used for further evaluations.

Entity Extraction: Table 3 displays the results of entity recognition task under five different strategies on IHQID-WebMD and IHQID-1mg datasets.

Finding 1: In the Backtranslation test performed in Setup 1, we observe that for WebMD dataset, the difference in the performance of the models (Performance on English is 0.33% more average F1 Score for RoBERTa and 3.58% more than average F1 for bcBERT) is far less significant than the drop observed for 1 mg (Performance on English is 9.66% more average F1 Score for RoBERTa and 10.49% more than average for bcBERT). This implies that loss of information is quite high for the entities in Indian context during backtranslation.

Finding 2: Unlike our findings on Setup 3 in intent recognition, we observe that backtranslation using a bridge language seems to induce more loss of

Lang -uage	Backtranslated Test (S1)				Zero-Shot Cross-Lingual Test (S2)				Bridge Language Backtranslation (S3)				Train and Test on Indic Data (S4)				Full Backtranslation (S5)			
	RoBERTa		bcBERT		mBERT		XLM-RoBERTa		RoBERTa		bcBERT		mBERT		XLM-RoBERTa		RoBERTa		bcBERT	
	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg	WMD	1mg
En	61.95	69.93	<u>65.50</u>	<u>73.68</u>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Hi	61.75	69.58	65.20	73.82	34.75	34.01	36.53	46.95	-	-	-	-	17.55	36.85	52.43	71.68	60.60	69.32	<u>65.90</u>	<u>76.55</u>
Bn	64.21	56.79	64.25	62.56	35.12	35.02	34.31	41.42	61.05	47.45	59.70	50.08	27.28	42.37	63.73	62.69	64.47	54.81	<u>64.75</u>	<u>65.80</u>
Ta	60.44	60.58	60.22	60.72	30.62	30.10	34.31	36.29	55.35	56.75	56.28	63.48	22.07	29.87	59.91	67.59	61.07	69.63	<u>64.91</u>	<u>71.40</u>
Te	62.76	62.56	62.37	63.44	30.00	31.50	32.95	41.71	62.26	55.75	63.89	62.27	27.95	27.82	59.81	68.93	65.27	67.06	<u>66.19</u>	<u>69.17</u>
Gu	60.02	51.13	58.20	52.62	23.56	27.24	23.90	42.19	56.36	47.12	57.60	51.47	21.93	25.82	49.19	<u>73.77</u>	<u>60.78</u>	59.62	58.26	70.78
Mr	<u>60.18</u>	51.31	57.68	55.45	26.32	22.54	29.51	50.84	54.63	59.61	55.02	<u>60.96</u>	20.48	23.52	52.61	57.56	59.10	57.38	58.83	58.45
Avg	61.62	60.27	61.92	63.19	30.56	30.07	31.92	43.23	57.92	53.34	58.49	57.65	22.88	26.61	58.28	67.04	61.88	62.98	<u>63.14</u>	<u>68.69</u>

Table 3: Macro-F1 scores for entity extraction on the WebMD (WMD) and 1mg datasets for five Setups (three different setups for Train on English (Scenario A) and two setups of Train on Indic Data (Scenario B)). bcBert indicates BioClinicalBERT, mBERT indicates Multilingual BERT. Underline denotes the best across five settings.

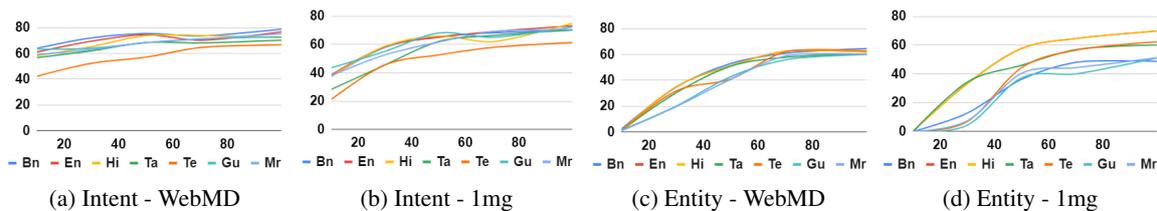


Figure 2: Intent Detection and Entity Extraction F1-score (Y-axis) for Different Percentage of Training Data (X-axis) for WebMD and 1mg

information on the entities compared to Setup 1. This observations holds true for both the models across two datasets.

Finding 3: Similar to intent recognition, we observe that completely backtranslating both training and test data to English performs the best among S1, S3 and S5. This holds true for both the datasets and both the models. However, this operation is indeed expensive in terms of data curation cost, since it requires original data in the target languages for both training and testing.

Finding 4: The abysmal performances of the multilingual models as shown in Table 3, for both S2 and S4 indicate that these approaches are not so useful in our case.

Finding 5: We report the average (Avg) F1-score across all languages. BioClinicalBERT performs the best (Setup 1 for English and Setup 5 for non-English (Avg)) for both WebMD (63.14%) and 1mg (68.69%). It is used for further evaluations.

5.5 Ablation Study

Experiments with Varying Training Size: We experiment with varying training sizes on both intent detection and entity extraction tasks using the best performing models, by taking 10%, 30%, 50%, 70% and 100% of the training set. We then show the F1-scores (Y-axis) for all the languages with different training sizes (X-axis) in Fig. 2. Fig. 2a and 2b show that the performance of the intent detection models do not vary too much with increasing training sample data. However, Fig. 2c and 2d clearly show that entity extraction F1-scores increase significantly with the increase of training data. Thus, we can conclude that the intent detection model does not require a large amount of data to generalise, as opposed to the requirements of the entity extraction model.

Category wise intent detection and entity extraction for the best model: We evaluate the F1-scores for different intent classes for the RoBERTa Model

(Setup 1 for English and Setup 5 for non-English) trained on WebMD and 1mg (See Section 4 for setup descriptions). Similarly, with the help of BioClinicalBERT (Setup 1 for English and Setup 5 for Non-English), we find the individual entity class wise F1-scores. The results in Table 4 show that the model is able to detect ‘disease’, ‘drug’ and ‘treatment’ intent classes with high F1-score but the performance on the ‘Other’ class is poor, thus bringing the macro averaged F1 score down considerably. This may be due to the fact that the system fails to detect open ended query types, present in the ‘Other’ class. This is supported by the intent class wise entity distribution, which shows an overwhelming dominance of ‘drug’, ‘disease’ and ‘treatment’ entities in their corresponding intent categories (‘drug’, ‘disease’ and ‘treatment plan’ intents, respectively), whereas the ‘other’ intent class, of which there are very few instances comparatively anyway, has no such dominant entity class associated with it. In the entity extraction task, the best performing model is able to extract all three entity categories with a similar F1-score performance.

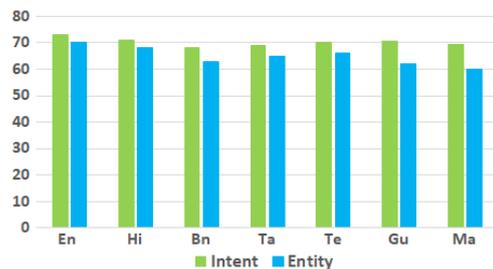


Figure 3: Macro Average F1 Score for Intent detection and Entity Extraction across all different languages in Real World Hospital Data

Real World Hospital Data Evaluation: We use the real world healthcare query dataset (100 queries per language) to test the usability of our models in practical Indian hospital scenarios. We run the best performing models trained on WebMD and 1mg

Lang	Intent								Entity					
	WebMD				1mg				WebMD			1mg		
	Disease	Drug	Treatment	Other	Disease	Drug	Treatment	Other	Disease	Drug	Treatment	Disease	Drug	Treatment
En	75.86	81.42	74.16	74.07	80.00	94.64	85.00	35.29	63.16	72.13	61.39	66.67	88.00	47.06
Hi	73.10	80.00	66.67	69.50	72.97	78.57	67.47	69.06	64.37	70.41	58.72	67.27	85.04	55.56
Bn	80.79	80.39	71.91	75.71	80.77	94.74	87.18	52.63	65.19	69.35	53.23	73.50	68.48	47.72
Ta	77.63	73.27	60.00	73.38	80.77	94.74	80.95	25.00	60.05	69.07	49.23	76.11	72.87	50.00
Te	72.85	78.10	70.45	72.46	80.77	94.55	77.27	33.33	62.09	65.00	60.71	75.63	65.37	66.67
Gu	75.64	78.50	69.77	72.18	83.02	93.81	80.00	33.33	53.41	66.32	58.82	72.41	69.44	52.86
Mr	76.82	78.85	75.29	71.83	79.25	94.64	83.72	25.00	59.48	59.26	56.45	60.78	59.74	49.58

Table 4: Macro-F1 scores for intent identification and entity extraction on the WebMD (WMD) and 1mg datasets. For each language, we portray the results of the best model obtained for the corresponding dataset.

data for intent detection (RoBERTa in Setup 1 for English and Setup 5 for Non-English) and entity extraction (BioCLinicalBERT in Setup 1 for English and Setup 5 for non-English) and report the average of two models (trained on IHQID-WebMD and IHQID-1mg) for each language. Fig. 3 shows the average F1-score for each language, which is consistent with the earlier results shown in Table 2 and 3. This shows that the best performing proposed setup performs satisfactorily on real world data in Indic languages.

5.6 Demonstration

To be able to make the proposed methods accessible and usable by the community, we create an on-line interface, which could be found in our GitHub repository⁶. With the help of this website, one can post health query in the allowed language and obtain the predictions using our best models.

6 Discussion and Error Analysis

We categorize the issues in mis-classification and identify two broad themes of the reasons. The primary reason is model prediction error. Figure 4 shows the model prediction errors for various intents in different languages. For an example, ‘How common is syphilis’ is of ‘disease’ intent category but model wrongly predicts it as ‘other’ category. Another reason is the misclassification due to incorrect translation of the medical entities such as the disease ‘*uticartia*’ has been transformed into ‘*ambat*’ during backtranslation as shown in Figure 5 which is not detected as an entity. So, the backtranslation error leads to intent mis-classification and entity extraction error. We speculate such random absurd behaviour due to the context of the query and languages are semantically different. Secondly, there are also certain issues in fluency and grammatical meaning after backtranslation. For instance,

⁶<https://github.com/indichealth/indic-health-demo>

ENGLISH QUERY	INDIC QUERY	GOLD INTENT	PREDICTED INTENT
How common is syphilis?	সিফিলিস কতটা সাধারণ? (bengali)	disease	other
Do all women experience discomfort after menopause?	সব মহিলাই কি মেনোপজের পরে অস্বস্তি অনুভব করেন? (bengali)	other	disease
How long can I take Ganaton Tablet?	నేను గానాటన్ మోతాదు ఎంతకాలం తీసుకోవచ్చు? (telugu)	drug	other
What home remedies can help with cough?	దగ్గుకు ఏ ఇంటి నివారణలు సహాయపడతాయి? (telugu)	treatment	other

Figure 4: Error in Prediction

English Query	Indic Query	Back Translated Query	Error	Intent Gold Label	Intent Predicted	Entity Detected
Is <i>urticaria</i> an autoimmune condition?	আমবাত একটি অটোইমিউন অবস্থা?	Is <i>ambat</i> an autoimmune condition?	Disease Translation	Disease	Other	
Is <i>Domstal 10</i> Tablet an over the counter drug?	டொস্টাল 10 టాబ్లెట్ డి.ఎం.ఎల్.ఓ.టి.సి. యాన్ ఓవర్ ది కౌంటర్ డ్రగ్?	Is <i>Domstal 10</i> Tablet an opposite drug?	Grammatical Error	Drug	Other	Domstal 10

Figure 5: Error in Back-Translation

‘over the counter drug’ gets changed to ‘over the opposite drug’. Entity recognition errors are also occurring along with the intent mis-classification.

7 Conclusion

We focus on developing novel Indian HealthCare Query Datasets and propose frameworks to detect intents and extract entities from queries in different Indian languages. Through extensive experiments on our proposed datasets, we recommend the community to use backtranslation of test queries to English in two real-life scenarios as a reasonable choice when we have access to English training data. However, the same strategy can be applied to both train and test queries if we have the budget of collecting data in target languages. Backtranslation of queries using an intermediate bridge language also proves to be a useful strategy in some cases.

Acknowledgements

The project was supported in part by the grant given by I-Hub Foundation for Cobotics, IIT Delhi for the project, "Voice based Natural Interaction for Goal Oriented Tasks in Healthcare".

Limitations

Our dataset needs to be scaled up in terms of size and intent labels which we aim to do as a part of future work. Another constraint is that we do not consider cases where queries are multi-labelled (e.g. - drug and disease both). We shall explore in future.

Ethical Concerns

We propose to release the dataset which neither reveals any personal sensitive information of the patients nor any toxic statement. Besides, we have paid enough token money (exact remuneration will be revealed once accepted to the conference) to the domain-expert annotators who have helped us in manually tagging the medical queries.

References

- Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.
- Flora Amato, Stefano Marrone, Vincenzo Moscato, Gabriele Piantadosi, Antonio Picariello, and Carlo Sansone. 2017. Chatbots meet ehealth: Automating healthcare. In *WAIAH@ AI* IA*, pages 40–49.
- Sagar Badlani, Tanvi Aditya, Meet Dave, and Sheetal Chaudhari. 2021. Multilingual healthcare chatbot using machine learning. In *2021 2nd International Conference for Emerging Technology (INCET)*, pages 1–6. IEEE.
- Guirong Bai, Shizhu He, Kang Liu, and Jun Zhao. 2022. Incremental intent detection for medical domain with contrast replay networks. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3549–3556.
- Qiming Bao, Lin Ni, and Jiamou Liu. 2020. Hhh: an online medical chatbot system based on knowledge graph and hierarchical bi-directional attention. In *Proceedings of the australasian computer science week multiconference*, pages 1–10.
- Urmil Bharti, Deepali Bajaj, Hunar Batra, Shreya Lalit, Shweta Lalit, and Aayushi Gangwani. 2020. Medbot: Conversational artificial intelligence powered chatbot for delivering tele-health after covid-19. In *2020 5th international conference on communication and electronics systems (ICCES)*, pages 870–875. IEEE.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Jeanne E. Daniel, Willie Brink, Ryan Eloff, and Charles Copley. 2019. [Towards automating healthcare question answering in a noisy multilingual low-resource setting](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 948–953, Florence, Italy. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#).
- Vittorio Gebbia, Dario Piazza, Maria Rosaria Valerio, Nicolò Borsellino, and Alberto Firenze. 2020. [Patients with cancer and covid-19: A whatsapp messenger-based survey of patients’ queries, needs, fears, and actions taken](#). *JCO Global Oncology*, (6):722–729. PMID: 32412811.
- Ankur Gupta, Yash Varun, Prarthana Das, Nithya Muttineni, Parth Srivastava, Hamim Zafar, Tanmoy Chakraborty, and Swaprava Nath. 2021. [Truthbot: An automated conversational tool for intent learning, curated information presenting, and fake news alerting](#).
- Pratik Joshi, Christain Barnes, Sebastin Santy, Simran Khanuja, Sanket Shah, Anirudh Srinivasan, Satwik Bhattamishra, Sunayana Sitaram, Monojit Choudhury, and Kalika Bali. 2019. [Unsung challenges of building and deploying language technologies for low resource language communities](#). *arXiv preprint arXiv:1912.03457*.
- Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. 2020. [Cross-lingual text classification with minimal resources by transferring a sparse teacher](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3604–3622, Online. Association for Computational Linguistics.
- Young-Min Kim, Tae-Hoon Lee, and Seon-Ok Na. 2022. [Constructing novel datasets for intent detection and ner in a korean healthcare advice system: guidelines and empirical results](#). *Applied Intelligence*, pages 1–21.
- Guillaume Lample and Alexis Conneau. 2019. [Cross-lingual language model pretraining](#). *arXiv preprint arXiv:1901.07291*.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. [From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

- Yunyao Li, Tyrone Grandison, Patricia Silveyra, Ali Douraghy, Xinyu Guan, Thomas Kieselbach, Chengkai Li, and Haiqi Zhang. 2020. [Jennifer for COVID-19: An NLP-powered chatbot built for the people and by the people to combat misinformation](#). In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.
- Wenge Liu, Jianheng Tang, Jinghui Qin, Lin Xu, Zhen Li, and Xiaodan Liang. 2020. [Meddg: A large-scale medical consultation dataset for building medical dialogue system](#). *arXiv preprint arXiv:2010.07497*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Zihan Liu. 2022. [Effective transfer learning for low-resource natural language understanding](#).
- Theodora A Maniou and Andreas Veglis. 2020. [Employing a chatbot for news dissemination during crisis: Design, implementation and evaluation](#). *Future Internet*, 12(7):109.
- Devansh Mehta, Sebastin Santy, Ramaravind Kommiya Mothilal, Brij Mohan Lal Srivastava, Alok Sharma, Anurag Shukla, Vishnu Prasad, Venkanna U, Amit Sharma, and Kalika Bali. 2020. [Learnings from technological interventions in a low resource language: A case-study on Gondi](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2832–2838, Marseille, France. European Language Resources Association.
- Ajit K. Mohanty. 2010. [Languages, inequality and marginalization: implications of the double divide in indian multilingualism](#). 2010(205):131–154.
- Ishani Mondal, Kabir Ahuja, Mohit Jain, Jacki O Neil, Kalika Bali, and Monojit Choudhury. 2022. [Global readiness of language technology for healthcare: What would it take to combat the next pandemic?](#) *arXiv preprint arXiv:2204.02790*.
- Xin Mu, Kai Ming Ting, and Zhi-Hua Zhou. 2017a. [Classification under streaming emerging new classes: A solution using completely-random trees](#). *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1605–1618.
- Xin Mu, Feida Zhu, Juan Du, Ee-Peng Lim, and Zhi-Hua Zhou. 2017b. [Streaming classification with emerging new class by class matrix sketching](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Ankan Mullick, Abhilash Nandy, Manav Nitin Kapadnis, Sohan Patnaik, and R Raghav. 2022a. [Fine-grained intent classification in the legal domain](#). *arXiv preprint arXiv:2205.03509*.
- Ankan Mullick, Sukannya Purkayastha, Pawan Goyal, and Niloy Ganguly. 2022b. [A framework to generate high-quality datapoints for multiple novel intent detection](#). *arXiv preprint arXiv:2205.02005*.
- Rohini P Pande and Abdo S Yazbeck. 2003. [What’s in a country average? wealth, gender, and regional inequalities in immunization in india](#). *Social Science Medicine*, 57(11):2075–2088.
- Parth Patwa, Mohit Bhardwaj, Vineeth Guptha, Gitanjali Kumari, Shivam Sharma, Srinivas Pykl, Amitava Das, Asif Ekbal, Md Shad Akhtar, and Tanmoy Chakraborty. 2021. [Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts](#). In *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation*, pages 42–53. Springer.
- Michael Paul, Andrew Finch, and Eiichiro Sumita. 2013. [How to choose the best pivot language for automatic translation of low-resource languages](#). *ACM Transactions on Asian Language Information Processing*, 12(4).
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. [How multilingual is multilingual BERT?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.
- Muhammad Asif Razzaq, Wajahat Ali Khan, and Sungyoun Lee. 2017. [Intent-context fusioning in healthcare dialogue-based systems using jdl model](#). In *International Conference on Smart Homes and Health Telematics*, pages 61–72. Springer.
- KC Santosh. 2019. [Speech processing in healthcare: can we integrate?](#) In *Intelligent Speech Signal Processing*, pages 1–4. Elsevier.
- Hilal Salim Al Shamsi, Abdullah Ghthaiht Almutairi, Sulaiman Salim Al Mashrafi, and Talib Salim Al Kalbani. 2020. [Implications of language barriers for healthcare: A systematic review](#). *Oman Medical Journal*, 35:e122 – e122.
- Fahim K Sufi and Musleh Alsulami. 2021. [Automated multidimensional analysis of global events with entity detection, sentiment analysis and anomaly detection](#). *IEEE Access*, 9:152449–152460.
- Yu Sun, Ke Tang, Leandro L Minku, Shuo Wang, and Xin Yao. 2016. [Online ensemble learning of data streams with gradually evolved classes](#). *IEEE Transactions on Knowledge and Data Engineering*, 28(6):1532–1545.
- Andrea Vanzo, Emanuele Bastianelli, and Oliver Lemon. 2019. [Hierarchical multi-task natural language understanding for cross-domain conversational ai: Hermit nlu](#). *arXiv preprint arXiv:1910.00912*.

- Hunsur Nagendra Vishwas, Jillella Sandeep Reddy, Pradeep Kumar Katravath, Naveen Kumar Posanpally, Lakshmi Sowjanya Vakkapatla, Prasanna Kumar Bojja, and Harish Gattikoppula. 2017. Translation and validation of telugu version of marital satisfaction scale (t-mss). *Indian Journal of Pharmacy Practice*, 10(1).
- Min Wang, Ke Fu, Fan Min, and Xiuyi Jia. 2020. Active learning through label error statistical methods. *Knowledge-Based Systems*, 189:105140.
- Congying Xia, Wenpeng Yin, Yihao Feng, and Philip Yu. 2021. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. *arXiv preprint arXiv:2104.11882*.
- Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S Yu. 2018. Zero-shot user intent detection via capsule neural networks. *arXiv preprint arXiv:1809.00385*.
- Mohammad Yani, Adila Alfa Krisnadhi, and Indra Budi. 2022. A better entity detection of question for knowledge graph question answering through extracting position-based patterns. *Journal of Big Data*, 9(1):1–26.
- Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, et al. 2020. Medialog: Large-scale medical dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Chenwei Zhang, Nan Du, Wei Fan, Yaliang Li, Chun-Ta Lu, and S Yu Philip. 2017. Bringing semantic structures to user intent detection in online medical queries. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 1019–1026. IEEE.
- Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, et al. 2021. Cblue: A chinese biomedical language understanding evaluation benchmark. *arXiv preprint arXiv:2106.08087*.
- Lingyun Zhao, Lin Li, Xinhao Zheng, and Jianwei Zhang. 2021. A bert based sentiment analysis and key entity detection approach for online financial texts. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 1233–1238. IEEE.
- Bingui Zhou, Guanghua Yang, Zheng Shi, and Shaodan Ma. 2021. Natural language processing for smart healthcare. *arXiv preprint arXiv:2110.15803*.
- Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. [Cross-lingual sentiment classification with bilingual document representation learning](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1403–1412, Berlin, Germany. Association for Computational Linguistics.

Text-Derived Knowledge Helps Vision: A Simple Cross-modal Distillation for Video-based Action Anticipation

Sayontan Ghosh¹

Tanvi Aggarwal¹

Minh Hoai¹

Niranjan Balasubramanian¹

¹Stony Brook University

{sagghosh, taggarwal, minhhoai, niranjan}@cs.stonybrook.edu

Abstract

Anticipating future actions in a video is useful for many autonomous and assistive technologies. Most prior action anticipation work treat this as a vision modality problem, where the models learn the task information primarily from the video features in the action anticipation datasets. However, knowledge about action sequences can also be obtained from external textual data. In this work, we show how knowledge in pretrained language models can be adapted and distilled into vision-based action anticipation models. We show that a simple distillation technique can achieve effective knowledge transfer and provide consistent gains on a strong vision model (Anticipative Vision Transformer) for two action anticipation datasets (3.5% relative gain on EGTEA-GAZE+ and 7.2% relative gain on EPIC-KITCHEN 55), giving a new state-of-the-art result¹.

1 Introduction

Anticipating future actions in the video of an unfolding scenario is an important capability for many applications in augmented reality (Salamin et al., 2006; Azuma, 2004), robotics (Duarte et al., 2018; Schydlo et al., 2018), and autonomous driving (Chaabane et al., 2020; Suzuki et al., 2018). Anticipating what actions will likely happen in a scenario, requires one to both recognize what has happened so far, and use anticipative general knowledge about how action sequences tend to play out. Most models for this task use a pre-trained video encoder to extract information about what has happened so far in the scenario, and use a text-based decoder to predict what action is likely to happen in the future (Carion et al., 2020; Dessalene et al., 2021; Liu et al., 2020; Sener et al., 2020).

However, when trained on the target video datasets, the generalization of the models depends

¹The models and code used are available at: <https://github.com/StonyBrookNLP/action-anticipation-lmtovideo>

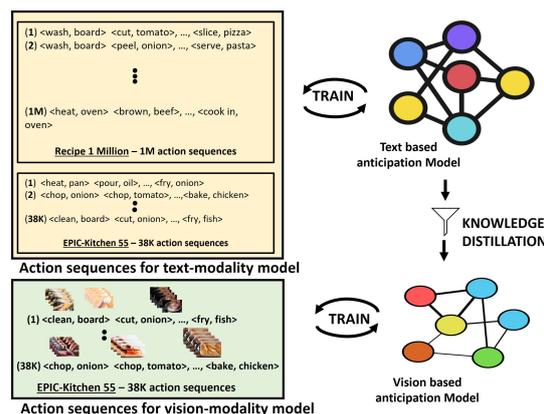


Figure 1: A model learning the action anticipation from only the vision modality (video frames) is essentially exposed to a very limited set of action sequences. Language models, which are pre-trained on large-scale text, can learn this distribution from the task, and a much larger domain-relevant text. We propose distilling this knowledge from text modality models to vision modality model for video action anticipation task.

on how well these video datasets cover the space of action sequence distributions. In other words, the knowledge that is learnt for predicting future actions is, in effect, limited to the information in the target video datasets, where obtaining large scale coverage of action sequences is difficult.

Knowledge about action sequences can also be obtained from text resources at scale. Language models, (e.g. BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b)), are typically pre-trained on large collections of unlabeled texts with billions of tokens, where they acquire a wide-variety of knowledge including large scale knowledge about common action sequences. For example, Table 1 illustrates how the pre-trained BERT is able to predict the next action in a sequence of actions extracted from a recipe video in terms of its verb and the object. Also, it is easier to collect a much larger collection of action sequences from text sources compared to video annotated with

Masked action sequence	BERT@top5
Clean the board → takeout pan → wash the onion → clean the fish → cut the onion → heat the pan → pour oil in pan → [MASK] the fish.	fry, cook, boil, wash, clean
Clean the board → takeout pan → wash the onion → clean the fish → cut the onion → heat the pan → pour oil in pan → fry [MASK].	pan, fish, chicken, it, onion

Table 1: Given a sequence of actions extracted from a video, **BERT@top5** shows the top5 prediction made by a standard pre-trained BERT for the masked verb and object of the next action.

segments. As illustrated in Figure 1, EPIC55, a video dataset of about 800GB only has about 38K action sequences, whereas there are around 1M sequences in the text recipes dataset Recipe1M. Text modality models can thus be exposed to a much larger variety of action sequences compared to video-modality anticipation models. However, because the task is defined only over the video inputs there is a question of how one can transfer this knowledge.

In this work, we show that we can augment video-based anticipation models with this external text-derived knowledge. To this end, we propose a simple cross-modal distillation approach, where we distill the knowledge gained by a language model from the text modality of the data into a vision-modality model. We build a teacher using a pre-trained language model which already carries general knowledge about action sequences. We adapt this teacher to the action sequences in the video domain by fine-tuning them for the action anticipation task. Then, we train a vision-modality ² student, which is now tasked with both predicting the target action label as well as matching the output probability distribution of the teacher.

There are two aspects of language models that can be adjusted further for improved distillation. First, while they may contain knowledge about a broad range of action sequences, we can focus them towards specific action sequences in the target dataset. Second, the text modality teacher can be further improved by pretraining on domain-relevant texts (e.g. cooking recipes), to further adapt it to the action sequences in the task domain.

²The task requires the anticipation model to make inference based on the vision modality (video frames) of the video

Our empirical evaluation shows that this cross-modal training yields consistent improvements over a state-of-the-art Anticipative Vision Transformer model (Girdhar and Grauman, 2021) on two ego-centric action anticipation datasets in the cooking domain. Adapting the teacher to the task domain by pretraining on domain relevant texts yields further gains and the gains are stable for different language models. Interestingly, our analysis shows that the language model based teacher can provide gains even when it is not necessarily better than the vision student, suggesting that distillation benefits can also come from the complementarity of knowledge, as in the case of the text modality.

In summary we make the following contributions: (i) We show that a simple distillation scheme can effectively transfer text-derived knowledge about action sequences (i.e. knowledge external to the video datasets) to a vision-based action anticipation model. (ii) We show that text-derived knowledge about actions sequences contain complementary information that is useful for the anticipation task, especially for the case where the action label space is large. (iii) Using a strong action anticipation model as a student, we achieve new state-of-the-art results on two benchmark datasets.

2 Related Work

There has been a wide range of solutions for action anticipation ranging from hierarchical representations (Lan et al., 2014), unsupervised representation learning (Vondrick et al., 2016), to encoder-decoder frameworks that decode future actions at different time scales (Furnari and Farinella, 2019), and transformers trained on multiple auxiliary tasks (Girdhar and Grauman, 2021). However, these only use the vision modality features of the observed video to train the model for the anticipation task. Our work aims to distill text-derived knowledge to improve action anticipation. Here we relate our work to others that have made use of (i) textual knowledge for related tasks, (ii) general knowledge distillation, and (iii) multimodal models which also allow for integration of information from different modalities.

Textual Knowledge for Action Anticipation: Other works have also shown the utility of modeling text-modality. Sener and Yao (2019) transfer knowledge in a text-to-text encoder-decoder to a video-to-text encoder-decoder, by substituting the text encoder with the video encoder. However, this

relies on projecting the image and text features in a shared space, which requires lots of properly aligned text and its corresponding image. [Camporese et al. \(2021\)](#) model label semantics with a hand engineered deterministic label prior based on the global co-occurrence statistics of the action labels from the overall training data, which can be ineffective in case the underlying joint action distribution is complex. In contrast, our work proposes a different approach to leverage the text in the training data by using language models to learn the complex underlying distribution of action sequences in the video and then distill this knowledge into a vision model to improve their performance.

Cross-modal Knowledge Distillation: [Thoker and Gall \(2019\)](#) propose learning from RGB videos to recognize actions for another modality. Others have used cross-modal distillation for video retrieval tasks ([Hu et al., 2020](#); [Chen et al., 2020](#)) and for text-to-speech ([Wang et al., 2020](#)). Most relevant to ours is a recent system that improves language understanding of text models by transferring the knowledge of a multi-modal teacher trained on a video-text dataset, into a student language model with a text dataset ([Tang et al., 2021](#)). In contrast, our proposed method for action anticipation transfers knowledge gained by a text-based teacher model into a vision-based student model.

Multimodal Models: Due to the recent prevalence of multimodal data and applications ([Lin et al., 2014](#); [Sharma et al., 2018](#); [Antol et al., 2015](#); [Krishna et al., 2017](#); [Ordonez et al., 2011](#); [Abu Farha et al., 2018](#); [Talmor et al., 2021](#); [Afouras et al., 2018](#)), there has been plethora of recent work on multimodal transformers. One commonly used approach used to train these models is to learn a cross-modal representation in a shared space. Examples include learning to align text-image pairs for cross-modal retrieval ([Radford et al., 2021](#); [Wehrmann et al., 2020](#)), grounded image representations ([Liu et al., 2019a](#)), and grounded text representations ([Tan and Bansal, 2020](#); [Li et al., 2019](#)). [Hu and Singh \(2021\)](#) extend the idea for multi-task settings with multiple language-vision based tasks. [Tsimpoukelli et al. \(2021\)](#) adapt a vision model to a frozen large LM to transfer its few-shot capability to a multimodal setting (vision and language). However these methods rely on large-scale image-text aligned datasets for the training the model, which may not always be available, for

e.g. EGTEA-GAZE+ video dataset has only 10.3K labelled action sequences. In contrast our distillation approach does not require any image-text alignment for the anticipation task.

3 Language-to-vision knowledge distillation for action anticipation

The action anticipation task asks to predict the class label of a future action based on information from an observed video sequence. In this task setting, the model has access to both, video and annotated action segments (action text) during the train time, but needs to make the inference only using the video sequence. The input to the prediction model is a sequence of video frames up until time step t : $\mathbf{X} = (X_1, X_2, \dots, X_t)$, and the desired output of the model is the class label Y of the action at time $t + \tau$, where τ is the anticipation time.

To learn an anticipation model, we assume there is training data of the following form: $\mathcal{D} = \{(\mathbf{X}^i, \mathbf{L}^i, Y^i)\}_{i=1}^n$, where $\mathbf{X}^i = (X_1^i, \dots, X_{t_i}^i)$ is the i^{th} training video sequence, Y^i is the class label of the future action at time $t^i + \tau$, and $\mathbf{L}^i = (L_1^i, \dots, L_{k^i}^i)$ is the sequence of action label of the action segments in the video sequence \mathbf{X}^i . Each human action can span multiple time steps, so the number of actions k^i might be different from the number of video frames t^i .

Our task is to learn a model g that can predict the future action label based on the vision modality of the video sequence \mathbf{X}^i only. A common approach is to optimize cross entropy loss \mathcal{L} between the model’s predicted label $g(\mathbf{X}^i)$ and the ground truth label Y^i of each training instances, i.e., to minimize: $\sum_i \mathcal{L}(g(\mathbf{X}^i), Y^i)$. Although the sequence of action labels \mathbf{L}^i is available in the training data, the semantics associated with these labels is not properly used by the existing methods for training the anticipation model.

Here we propose to learn a text-based anticipation model g_{text} and use it to supervise the training of the vision-based anticipation model g . This training approach utilizes the knowledge from the text domain, which is easier to learn than the vision-based knowledge, given the abundance of event sequences described in text corpora. Hereafter, we will refer to the language-based model as the teacher, and the vision-based model as the student.

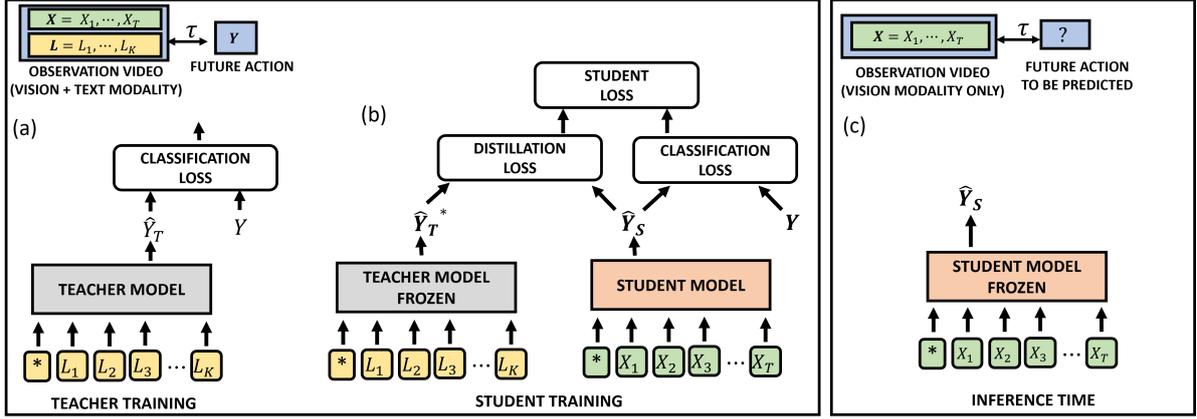


Figure 2: **METHOD OVERVIEW: Training**- The observation video has two set of features, a sequence of T image frames \mathbf{X} , and a sequence action labels (e.g. cut-onion, peel-onion etc.) \mathbf{L} corresponding to the K action segments in \mathbf{X} . (a) We train a teacher model to predict Y using the text features \mathbf{L} . Then we distill the knowledge gained by the teacher on text features into the student model that operates on vision modality \mathbf{X} . For this, (b), we train a student model on the vision modality feature \mathbf{X} while using the corresponding prediction from the teacher model as a label prior. **Inference**- During the inference or test time, the trained student model is used to predict the future action using only the vision modality of the observed video.

3.1 Overview

The overview of our proposed method is shown in Figure 2. We augment vision-based anticipation models (students) with knowledge distilled from text-based models (teachers) that have access to knowledge from large scale action sequences. To this end we fine-tune a pre-trained language model on the action sequences in the training data. However, unlike the student, the teacher gets to see the action labels of the input video segment to make its predictions (Figure 2a). Then, we train a vision-based student that learns from the text-based teacher (Figure 2b).

The teacher in our setting is built using a pre-trained language model g_{txt} that has access to broad knowledge about action sequences. We fine-tune it on the target dataset as follows. For each instance, the teacher is given the textual action sequence \mathbf{L}^i in the input video as the input (or conditioning context), which then predicts the anticipated future action \hat{Y}_{txt}^i . The teacher is trained to minimize the loss defined over the predicted and true labels, i.e., to minimize: $\sum_i \mathcal{L}_{txt}(g_{txt}(\mathbf{L}^i), Y^i)$, where \mathcal{L} denotes the cross-entropy loss and $\hat{Y}_{txt}^i = g_{txt}(\mathbf{L}^i)$ is the output of the text-based teacher model.

We then freeze the *teacher*, and train a vision-based *student* model g that predicts the future action using the vision features \mathbf{X}^i . The student is trained to minimize the loss $\mathcal{L}_S(Y^i, \hat{Y}^i, \hat{Y}_{txt}^i)$ such that it's output probability distribution $\hat{Y}^i = g(\mathbf{X}^i)$

matches that of the teacher's output \hat{Y}_{txt}^i , in addition to matching the true label Y^i .

3.2 Teacher

The input to the teacher is a sequence of action phrases $\mathbf{L} = (L_1, \dots, L_k)$ that denotes the sequence of actions observed in the input video segment. The teacher first uses a standard language model ϕ_{LM} to produce a vector f_{txt} , of the input sequence \mathbf{L} . In transformer-based language models, a special token (e.g. [CLS] in BERT) is prepended to the input sequence. The output contextual representation of this special token is used as the final representation of the entire input sequence.

The teacher uses this f_{txt} vector to predict the output labels using the standard linear transformation (\mathbf{W} , b) followed by a softmax layer. In addition we also train the teacher to predict the main verb Y_{vb} and the object Y_{ob} of the action Y . These are predicted using separate linear transformations ($\mathbf{W}_v, \mathbf{b}_v$) and ($\mathbf{W}_o, \mathbf{b}_o$), followed by softmax.

The full set of predictions for input $\mathbf{L} = (L_1, \dots, L_k)$ is obtained as:

$$\begin{aligned}
 f_{txt} &= \phi_{LM}(L_1, \dots, L_k) \\
 \hat{Y}_{txt} &= \text{softmax}(\mathbf{W} f_{txt} + \mathbf{b}) \\
 \hat{Y}_{ob} &= \text{softmax}(\mathbf{W}_o f_{txt} + \mathbf{b}_o) \\
 \hat{Y}_{vb} &= \text{softmax}(\mathbf{W}_v f_{txt} + \mathbf{b}_v)
 \end{aligned}$$

To fine-tune the teacher model, we minimize the weighted sum of the cross-entropy loss between

the predicted triplet of action, verb and noun and their corresponding ground truth values.

$$\begin{aligned} \mathcal{L}_{txt}(Y, (\hat{Y}_{txt}, \hat{Y}_{ob}, \hat{Y}_{vb})) &= \lambda \mathcal{L}(Y, \hat{Y}_{txt}) \\ &+ \lambda_o \mathcal{L}(Y_{ob}, \hat{Y}_{ob}) + \lambda_v \mathcal{L}(Y_{vb}, \hat{Y}_{vb}) \end{aligned} \quad (1)$$

3.2.1 Adapting Teacher using Domain Relevant Texts

Pre-trained LMs have been shown to contain a wide variety of knowledge, which we hope to distill into the vision student model. However, there are two aspects about LMs which limit their applicability. First, LMs are general purpose models that cover many domains, but the target video datasets cover specific domains. For example, many action anticipation datasets are built for the cooking domain. Second, unlike fluent texts that LM’s are trained on, the action sequences in the videos are annotated using simpler verb/object constructions. Adapting LMs to these differences can benefit the knowledge distillation. To this end, we make use of domain-relevant texts (for e.g. the recipes in Recipe1M (Marin et al., 2019) dataset). The recipes are natural language instructions. To mimic the target sequences in the video datasets, we convert these into simpler verb-object constructs, and then use the standard Masked Language Modeling training task. Thus, this allows us to not only make use of generic knowledge about action sequences but also adapt the text-derived knowledge to the target domain.

3.3 Student

The student is trained to take the video frames in the video segment $\mathbf{X} = (X_1, \dots, X_t)$ as input and predict the future action Y as output. Though the applicability of the proposed distillation method is not restricted to any particular class of student model, we use the recent state-of-the-art Anticipative Vision Transformer (AVT) (Girdhar and Grauman, 2021) as our student model. In AVT, the video to action prediction is done in two stages, first a *backbone* network \mathcal{B} generates the feature representation of the individual frames in \mathbf{X} in a non-contextual manner.

$$z_1, \dots, z_t = \mathcal{B}(X_1), \dots, \mathcal{B}(X_t)$$

This is then followed by a transformer based decoder *head* network \mathcal{D} , that generates the contextual representation of the frames by transforming

the frame features z_i ’s in an autoregressive manner.

$$\begin{aligned} f_{v_1}, \dots, f_{v_t} &= \mathcal{D}(z_1, \dots, z_t) \\ \hat{Y}_{v_j} &= \text{softmax}(\mathbf{W}_s f_{v_j} + \mathbf{b}_s) \quad \forall j \in \{1, \dots, t\} \\ \hat{Y} &= \hat{Y}_{v_t} \end{aligned}$$

The feature representations from the *head* network f_{v_j} ’s are then used to make predictions for the anticipated action \hat{Y}_{v_j} at time unit j . The anticipated action \hat{Y} for the input video \mathbf{X} is simply the predicted label at time unit t i.e. \hat{Y}_{v_t} . During training the model is also supervised for two other auxiliary tasks namely future feature prediction and intermediate action prediction (see (Girdhar and Grauman, 2021) for details). We denote this combined training loss function as \mathcal{L}_{AVT} .

For the teacher to student distillation, we want the AVT’s output distribution over action classes \hat{Y} to match the teacher’s distribution \hat{Y}_{txt} . To this end, we minimize the KL divergence between the teacher prediction \hat{Y}_{txt}^γ and student predictions \hat{Y}^γ , after smoothing the distributions using a temperature parameter γ , following the standard distillation technique (Hinton et al., 2015).

$$\mathcal{L}_S = \mathcal{L}_{AVT} + \lambda_S \cdot \mathcal{D}_{KL}(\hat{Y}_{txt}^\gamma \parallel \hat{Y}^\gamma) \quad (2)$$

Dataset	Segments	Classes	τ
Epic 55	28.6K + 9 K	2,513	1.0 sec
EGTEA-Gaze+	7.3K + 3K	106	0.5 sec

Table 2: **Datasets** on which the proposed method is benchmarked. **Segments** are the number of action segments in the train + test set, **Classes** are the number of action classes in the dataset, τ is the anticipation time.

4 Experimental Setup

4.1 Datasets

1. Anticipation Datasets We evaluate the proposed method on two different datasets that are summarised in Table. 2. Both the datasets, Epic-Kitchen 55 (Damen et al., 2018) and EGTEA-GAZE+ (Li et al., 2018), are egocentric (first-person) videos of people cooking some recipe. Note the proposed method is broadly applicable to other types of dataset as long as the input video segments in the training set contain action sequence annotations. For the Epic-Kitchen 55 dataset, we use the standard train-test split followed in Furnari and Farinella (2019). For the EGTEA-GAZE+ dataset,

we report performance on the first of the three train-test splits following previous work by [Girdhar and Grauman \(2021\)](#).

2. Domain-Relevant Dataset The teacher can be improved further by adapting its language model (LM) to domain relevant texts. To test the effectiveness of this, we use the Recipe1M dataset ([Marin et al., 2019](#)) to pre-train the LM. The Recipe1M dataset contains one million recipes along with associated images (which are not used in this work). The instructions in a recipe can be seen as a sequence of cooking actions to be performed.

4.2 Performance Metrics

For the EGTEA-Gaze+, we report the performance on top1 accuracy (**Acc@1**) and class mean recall (**Rec@1**)-mean recall of the individual classes, as reported by [Girdhar and Grauman \(2021\)](#). For the Epic-Kitchen 55 dataset, there are a set of action classes that occur only in the train set but not in the test set and vice versa. Existing anticipation methods, including our proposed work does not support zero-shot learning. Therefore top5 many-shot class-mean recall (**MS-Rec@5**)-mean top5 recall of the classes in the many-shot-classes, as mentioned in [Furnari et al. \(2018\)](#), is our primary metric for model evaluation.

4.3 Implementation Details

1. Teacher Training: The teacher model is a classification layer on top a pre-trained language model. For the main set of experiments we used **ALBERT** ([Lan et al., 2019](#)) as the base language model. Our choice here is motivated by two main factors: (i) the pre-training task for **ALBERT** focuses on modeling the inter-sentence coherence which is important when modeling the sequence of disparate action phrases (ii) it enables faster training of deeper models. For the EGTEA-GAZE+ dataset, we trained the model for 4 epochs by minimizing the weighted cross-entropy loss (inversely weighted by the relative class frequency) due to the high degree of class imbalance in the dataset ($\sim 1 : 24$). For the EPIC-Kitchen-55 dataset, the model was trained for 8 epochs using regular cross-entropy loss instead of weighted cross-entropy as a lot of classes in the test label set are not present in the train-set, and vice versa.

The classification head is a single linear layer ($\mathbf{W} \cdot \mathbf{x} + \mathbf{b}$) that projects the feature representation of the input action sequence into the label

space of the target dataset. For optimizing on both the datasets, we used the AdamW ([Loshchilov and Hutter, 2017](#)) optimizer, with a learning rate of 10^{-5} and weight decay of 10^{-7} . The context window for the Epic-Kitchen was set to 5 action segments whereas for the EGTEA-GAZE+ it was set to 15 action segments. The teacher training was performed on two Nvidia RTX Titan-X GPUs. The teacher training for the EGTEA-GAZE+ takes about 2-4 hours depending on the LM base whereas the EPIC-Kitchen-55 takes about 3-5 hours to train.

2. Teacher Pre-training: We first parse each instruction in the Recipe1M dataset into a sequence event tuples of the form (subject, verb, object) using an open information extraction system ([Stanovsky et al., 2018](#)) made available by AllenNLP ([Gardner et al., 2018](#)). To match the action label structure we see in the video datasets, we represent each instruction using the sequence of action, i.e. <verb, object> part of the event. The actions in the action sequence are sorted by the discourse order of their corresponding verb in the instruction. The language model is pretrained on these (verb, object) sequences using the standard masked language modeling objective ([Devlin et al., 2019](#)), where some token in the sequence is masked at random and the model is tasked with predicting the masked token.

For pre-training, the language models were trained on the Recipe1M dataset for 200K steps with a batch size of 16. The optimizer used was AdamW ([Loshchilov and Hutter, 2017](#)), with a learning rate of 10^{-5} and weight decay of 10^{-7} . LM pre-training was performed on a single Nvidia A100 GPU with the training time varying from 12 hrs for the smallest model (**DistilBERT**) to 24 hrs for **BERT**, **RoBERTa**, and **ALBERT**.

3. Student Training: For the student training, all the hyperparameters and initial conditions (parameter initialization) are exactly identical to the ones used to train the **AVT** ([Girdhar and Grauman, 2021](#)) baseline model. So any change in the performance from the baseline is the result of adding the knowledge distillation. The distillation loss coefficient λ_S , for the EGTEA-GAZE+ dataset was set to 150, and 20 for EPIC-Kitchen 55.

4. Top-K logit distillation: The label space of EPIC-Kitchen 55 has 2,513 classes, out of which only 31% of the classes in the training data are present in the test data. This leads to the teacher

model assign relatively low probability values to many classes, which may not be reliable signals for distillation. Therefore, instead of matching the probability distribution over all the action classes, we only match the relative probability distribution of the top-50 classes with the highest teacher probabilities. For this, we consider the classes corresponding to the top-50 logits from the teacher prediction, normalize them, and only minimized the KL-Divergence between them and their corresponding logits of the student prediction. The student training was performed on either Nvidia Tesla V100 GPU and the training time was ~ 24 hrs for EGTEA-GAZE+ and ~ 6 hrs for the EPIC-55 dataset.

5 Results and Analysis

We present the results of text to video knowledge distillation on the AVT (Girdhar and Grauman, 2021) model as the student. AVT is the state-of-the-art model for action anticipation on the EGTEA-GAZE+ and EPIC-Kitchen 55 datasets on all performance metrics.

For each of these datasets, we consider the AVT variants with the best performance as our baseline and student model. For the EGTEA-GAZE+ dataset, we consider AVT-h + AVT-b in (Girdhar and Grauman, 2021) as our baseline model. Similarly for the EPIC-Kitchen 55 dataset, we consider, AVT-h + irCSN152 in (Girdhar and Grauman, 2021) as our baseline model.³ Throughout this section, we refer to AVT-h + AVT-b and AVT-h + irCSN152 as AVT-1 and AVT-2 respectively. The baseline models distilled with LM based teacher is denoted as AVT-1(or 2) + LM Distillation and in case teacher LM is pre-trained on the recipe domain text, the resulting model is referred to as AVT-1(or 2) + RcpLM Distillation. We tried to reproduce the AVT model to use as our student and obtain stronger results than the published version (see Table 3), on all but one metric. We use this stronger implementation as our baseline and our student model.

5.1 Does Knowledge distillation from Language Models help ?

Table 3 shows the result of training the state-of-the-art baseline model AVT, with and without the text to vision knowledge distillation, for the EGTEA-GAZE+ and EPIC-Kitchen 55 dataset. We

³✦AVT variants used for the EGTEA-GAZE+ and EPIC-Kitchen 55 baselines are AVT-1 and AVT-2.

can observe that applying text to vision knowledge distillation to the AVT leads to performance gains on both the datasets. For EGTEA-GAZE+, adding knowledge distillation leads to 2.1% and 2% relative percentage improvement over AVT-1 on the Acc@1 and Rec@1 metrics respectively. For the EPIC-Kitchen 55 dataset, knowledge distillation leads to a relative performance gain of 3.5% over AVT-2 on MS-Rec@5 metric.

5.2 Does domain-adaptive pre-training of teacher improves the task performance ?

To analyze the effect domain adaptive pre-training on the task, we pre-train the teacher LM on the Recipe1M dataset through the MLM task. The pre-trained model was then finetuned on the task-specific video dataset for the anticipation task. As seen in Table 3, the performance gain of the teacher directly translates to the performance gain of the student. For EGTEA-GAZE+ dataset, pre-training teacher leads to 3.9% and 3.4% relative improvement over the AVT-1 on Acc@1 and Rec@1 metric compared to 2.1% and 2% relative improvement when not pretraining the teacher. For the EPIC-Kitchen 55 dataset, teacher pre-training leads to a relative improvement of 7.2% on MS-Rec@5 metric compared to only 3.5% when not pre-training the teacher.

Model	EGTEA-GAZE+		EPIC-55
	Acc@1	Rec@1	MS-Rec@5
AVT (Girdhar and Grauman, 2021) -published	43.0	35.5	13.6
AVT (Girdhar and Grauman, 2021) -reproduced	43.52	34.87	15.25
AVT + LM Distillation	44.41	35.54	15.79
AVT + RcpLM Distillation	45.2	36.1	16.36

Table 3: **Effect of knowledge distillation:** Distilling knowledge from teacher (ALBERT LM) trained on text-modality of the video data, into vision based student model leads to student performance gain. Pre-training the teacher on domain relevant text before task specific finetuning leads to further performance improvement. ✦

5.3 How sensitive is the distillation to the choice of Language Model ?

In order to analyze the sensitivity of the distillation scheme towards the choice of the language model,

Model	EGTEA-GAZE+		EPIC-55
	Acc@1	Rec@1	MS-Rec@5
AVT (Girdhar and Grauman, 2021)	43.52	34.87	15.25
+ Rcp-ALBERT Distillation	45.2	36.1	16.36
+ Rcp-BERT Distillation	44.81	35.57	15.98
+ Rcp-RoBERTa Distillation	45.5	36.53	15.97
+ Rcp-ELECTRA Distillation	45.2	35.58	15.34
+ Rcp-DistillBERT Distillation	44.86	35.64	16.23

Table 4: **Effect of the choice of teacher LM** on the distillation performance. Each of the pre-trained LMs that we tested as a teacher, showed performance gain over the baseline AVT model for both the datasets. ♣

we also trained multiple teachers with different pre-trained LMs. The result of using different teachers for the anticipation task is specified in Table 4. From the table, we can observe that all the teacher distilled models perform better than the baseline AVT on all the metrics for both the datasets. This indicates that the text modality has some information that complementary to the vision modality that if properly exploited can lead to improved performance for the anticipation task.

5.4 Should the teacher be always better than the student ?

To understand the impact of the quality of the teachers, we measured the performance of the teacher models by themselves on the anticipation task as show in Table 5. For the EPIC-Kitchen 55 dataset the teacher performance is much better than the video-only baseline, whereas, for the EGTEA-GAZE+ dataset, the baseline vision model’s performance is much better than any of the teachers. Despite this, the performance gain due to distillation is greater for the EGTEA-GAZE+ dataset compared to the EPIC-Kitchen 55 dataset, as seen in Table 3. This suggests that what matters more for distillation in this case is the complementary information gained from the text modality that is not already present in the vision modality.

6 Conclusions

Action anticipation is a challenging problem that requires training large capacity video models. In

Model	EGTEA-GAZE+		EPIC-55
	Acc@1	Rec@1	MS-Rec@5
AVT (Girdhar and Grauman, 2021)	43.52	34.87	15.25
Rcp-ALBERT Teacher	21.66	22.63	21.78
Rcp-BERT Teacher	22.05	23.39	21.43
Rcp-RoBERTa Teacher	19.98	21.58	22.41
Rcp-ELECTRA Teacher	21.46	23.71	15.19
Rcp-DistillBERT Teacher	21.86	22.58	21.56

Table 5: **Teacher performance** on the anticipation task. For the EGTEA-GAZE+ dataset, the teacher performance is much lower than the video only AVT model, where as for the EPIC-Kitchen 55 dataset, the teacher performance is much better than the video-only AVT model. ♣

this work, we showed how the textual modality of the input videos, which is often ignored in training, can be leveraged to improve the performance of the video models. In particular, we can exploit the large scale knowledge acquired by pre-trained language models to build a text-modality teacher that can provide useful complementary information about the action sequences to a vision modality student. This cross-modal distillation strategy yields consistent gains achieving new state-of-the-art results on multiple datasets. Last, the gap between the performance of the teacher and the student models for domains with large label space suggests that there is still room for improvement with better distillation techniques.

7 Limitations

Real life scenarios have a large space of human actions which cannot be exhaustively covered by manually annotated training data. As such it is important to have models with zero-shot anticipation capabilities to predict unseen actions. This work did not explore zero-shot settings but we believe text-to-video distillation holds promise given the recent successes of language models in zero-shot tasks.

In this work we have shown the capability of text based language models for action anticipation, especially when the action space is very large and sparse. Though this work is intended to be a proof of concept for leveraging text based model for im-

proving video based action anticipation, there is still a large performance gap between the a text based language model and vision modality model. This performance gap indicates fruitful research avenues in text to vision knowledge distillation for action anticipation task.

8 Ethical Considerations

Anticipation future action based on videos is an important for many applications such as assistive technologies, augmented reality etc. Our work demonstrates that knowledge derived from text sources can be used to further improve the performance of video based action anticipation model. Even though our proposed work is able to improve the current state-of-art numbers on the standard benchmark datasets, the absolute performance is still low, especially in the case where the action space is very large. As such we would recommend to carefully analyze the cost of erroneous prediction before deploying the system for real world application.

Since the proposed method involves distilling the knowledge gained by pre-trained language model from text sources into a vision based model for action anticipation, this can also transfer the biases that these languages models can learn from the training text. As such data on which these text-based teacher models are trained should be analyzed for potential biases before deploying the proposed system for actual application. Analysis of bias propagation during knowledge distillation and devising bias reduction techniques are some potential extension of this work that we are highly interested in.

9 Acknowledgement

This material is based on research that is supported in part by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003 and in part by the National Science Foundation under the award IIS #2007290.

References

Yazan Abu Farha, Alexander Richard, and Juergen Gall. 2018. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5343–5352.

Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2018. Deep

audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Ronald Azuma. 2004. Overview of augmented reality. In *ACM SIGGRAPH 2004 Course Notes*, pages 26–es.

Guglielmo Camporese, Pasquale Coscia, Antonino Furnari, Giovanni Maria Farinella, and Lamberto Ballan. 2021. Knowledge distillation for action anticipation via label smoothing. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3312–3319. IEEE.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Mohamed Chaabane, Ameni Trabelsi, Nathaniel Blanchard, and Ross Beveridge. 2020. Looking ahead: Anticipating pedestrians crossing with future frames prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2297–2306.

Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han. 2020. Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12655–12663.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736.

Eadom Dessalene, Chinmaya Devaraj, Michael Maynard, Cornelia Fermuller, and Yiannis Aloimonos. 2021. Forecasting action through contact representations from first person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

- Nuno Ferreira Duarte, Mirko Raković, Jovica Tasevski, Moreno Ignazio Coco, Aude Billard, and José Santos-Victor. 2018. Action anticipation: Reading the intentions of humans and robots. *IEEE Robotics and Automation Letters*, 3(4):4132–4139.
- Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. 2018. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0.
- Antonino Furnari and Giovanni Maria Farinella. 2019. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6252–6261.
- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. **AllenNLP: A deep semantic natural language processing platform**. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.
- Rohit Girdhar and Kristen Grauman. 2021. Anticipative video transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13505–13515.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).
- Hengtong Hu, Lingxi Xie, Richang Hong, and Qi Tian. 2020. Creating something from nothing: Unsupervised knowledge distillation for cross-modal hashing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3123–3132.
- Ronghang Hu and Amanpreet Singh. 2021. Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1439–1449.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.
- Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. 2014. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704. Springer.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.
- Yin Li, Miao Liu, and James M Rehg. 2018. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pages 619–635.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Fenglin Liu, Yuanxin Liu, Xuancheng Ren, Xiaodong He, and Xu Sun. 2019a. Aligning visual regions and textual concepts for semantic-grounded image representations. *Advances in Neural Information Processing Systems*, 32.
- Miao Liu, Siyu Tang, Yin Li, and James M Rehg. 2020. Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In *European Conference on Computer Vision*, pages 704–721. Springer.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):187–203.
- Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.
- Patrick Salamin, Daniel Thalmann, and Frédéric Vexo. 2006. **The benefits of third-person perspective in virtual and augmented reality?** In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '06*, page 27–30, New York, NY, USA. Association for Computing Machinery.

- Paul Schydlo, Mirko Rakovic, Lorenzo Jamone, and José Santos-Victor. 2018. Anticipation in human-robot cooperation: A recurrent neural network approach for multiple action sequences prediction. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5909–5914. IEEE.
- Fadime Sener, Dipika Singhania, and Angela Yao. 2020. Temporal aggregate representations for long-range video understanding. In *European Conference on Computer Vision*, pages 154–171. Springer.
- Fadime Sener and Angela Yao. 2019. Zero-shot anticipation for instructional activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 862–871.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. [Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.
- Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. [Supervised open information extraction](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.
- Tomoyuki Suzuki, Hirokatsu Kataoka, Yoshimitsu Aoki, and Yutaka Satoh. 2018. Anticipating traffic accidents with adaptive loss and large-scale incident db. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3521–3529.
- Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hananeh Hajishirzi, and Jonathan Berant. 2021. Multimodalqa: Complex question answering over text, tables and images. *arXiv preprint arXiv:2104.06039*.
- Hao Tan and Mohit Bansal. 2020. Vokenization: Improving language understanding via contextualized, visually-grounded supervision. In *EMNLP*.
- Zineng Tang, Jaemin Cho, Hao Tan, and Mohit Bansal. 2021. Vidlankd: Improving language understanding via video-distilled knowledge transfer. *Advances in Neural Information Processing Systems*, 34.
- Fida Mohammad Thoker and Juergen Gall. 2019. Cross-modal knowledge distillation for action recognition. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 6–10. IEEE.
- Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212.
- Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106.
- Disong Wang, Jianwei Yu, Xixin Wu, Songxiang Liu, Lifa Sun, Xunying Liu, and Helen Meng. 2020. End-to-end voice conversion via cross-modal knowledge distillation for dysarthric speech reconstruction. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7744–7748. IEEE.
- Jonatas Wehrmann, Camila Kolling, and Rodrigo C Barros. 2020. Adaptive cross-modal embeddings for image-text alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12313–12320.

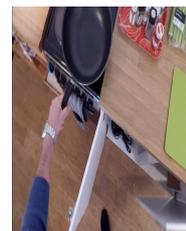
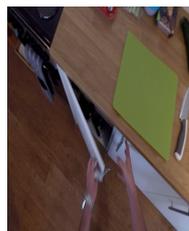
A Appendix

In this section, we present examples of model prediction for the video action anticipation task for the EPIC-55 dataset. For each instance we show the top-5 predictions for (i) video-only model - **AVT** (ii) text-based teacher model - LM-teacher (**Rcp-ALBERT**) (iii) a LM-teacher distilled student video model - AVT + LM teacher Distl (**AVT + Rcp-ALBERT Distillation**). Note that the end-task setting is such that, the inference has to be done only from the video frames, as the text label for the action segment won't be available during the inference time.

Figure 3 and 4 shows example of cases where the base video only model makes incorrect prediction, where as the text-based teacher and the teacher-distilled video model makes correct predictions. Figure 5 and 6 shows example of cases where the base video only model makes incorrect prediction, the text-based teacher makes correct prediction, however the teacher-distilled video model makes incorrect predictions.

EXAMPLE 1

INPUT



put-down_vegetable open_door take_greater take_pan put-down_pan

TARGET: close_door

PREDICTIONS

AVT : [put-down_pan, take_pan, turn-on_hob, open_door, open_drawer]

LM-teacher : [open_door, close_door, put-down_pan, take_pan, turn-on_hob]

AVT + LM teacher Dist1 : [put-down_pan, take_pan, open_door, close_door, turn-on_hob]

EXAMPLE 2

INPUT



put-down_pan put_lid put-down_pan take_pan take_lid

TARGET: put_lid

PREDICTIONS

AVT : [put-down_pan, turn-on_hob, open_door, take_pan, close_door]

LM-teacher: [put-down_pan, open_door, take_pan, put_lid, wash_pan]

AVT + LM teacher Dist1: [put-down_pan, turn-on_hob, open_door, take_pan, put_lid]

Figure 3: Example of instances where the base video-only model makes wrong prediction, whereas the text-based teacher and the teacher distilled video model makes correct prediction.

EXAMPLE 3

INPUT



put-down_board:cutting

put_onion

put_knife

pick-up_kettle

open_kettle

TARGET: fill_kettle

PREDICTIONS

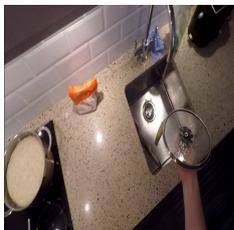
AVT: [open_door, turn-on_tap, pour_water, close_bin, open_tap]

LM-teacher: [pour_water, fill_kettle, put-down_kettle, open_kettle, close_kettle]

AVT + LM teacher Distl: [open_door, pour_water, close_bin, fill_kettle, turn-on_tap]

EXAMPLE 4

INPUT



put_lid

move_spoon

take_flour

open_flour

pour_flour

TARGET: put-down_flour

PREDICTIONS

AVT : [pour_flour, put-down_bag, mix_mixture, roll_dough, knead_dough]

LM-teacher : [put-down_flour, pour_flour, ' stir_flour, mix_mixture, check_flour]

AVT + LM teacher Distl: [pour_flour, roll_dough, mix_mixture, put-down_bag
put-down_flour]

Figure 4: Example of instances where the base video-only model makes wrong prediction, whereas the text-based teacher and the teacher distilled video model makes correct prediction.

EXAMPLE 1

INPUT



open_fridge



take_carrot



open_drawer



close_fridge



putdown_vegetable

TARGET: open_door

PREDICTIONS

AVT: [close_door, close_fridge, put_container, open_drawer, take_knife]

LM-teacher: [close_fridge, open_drawer, open_door, close_door, take_sausage]

AVT + LM teacher Distl: [close_door, put_container, take_knife, open_drawer, take_container]

EXAMPLE 2

INPUT



put_filter:water



drink-from_cup



put_cup



take_lid



take_pan

TARGET: put-down_pan

PREDICTIONS

AVT : [put_lid, stir_pasta, put-down_spoon, change_temperature, stir_pan]

LM-teacher : [put-down_pan, wash_pan, open_door, take_pan, dry_saucepan]

AVT + LM teacher Distl: [put_lid, stir_pasta, put-down_spoon, change_temperature, open_door]

Figure 5: Example of instances where the base video-only model makes wrong prediction, the text-based teacher makes the correct prediction, however the teacher distilled video model makes incorrect prediction.

EXAMPLE 3

INPUT



take_onion put-down_onion close_container take_spatula take_knife

TARGET: cut_onion

PREDICTIONS

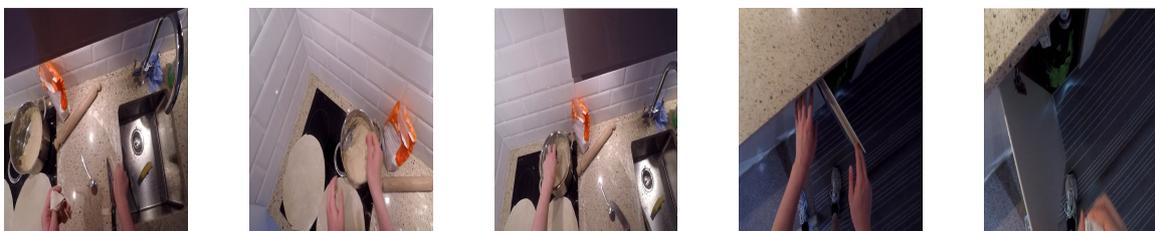
AVT: [put_container, take_knife, turn-on_tap, open_fridge, put-down_onion]

LM-teacher: [put-down_knife, cut_onion, mix_food, open_drawer, take_spoon]

AVT + LM teacher Distl: [put_container, take_knife, put-down_knife, take_container, open_fridge]

EXAMPLE 4

INPUT



take_dough put_dough put_lid open_door take_tomato

TARGET: put-down_tomato

PREDICTIONS

AVT : [open_fridge, open_door, turn-on_tap, open_drawer, rinse_hand]

LM-teacher : [put-down_tomato, close_door, take_tomato, take_plate, take_pan]

AVT + LM teacher Distl: [open_fridge, open_door, open_drawer, close_door, take_bowl]

Figure 6: Example of instances where the base video-only model makes wrong prediction, the text-based teacher makes the correct prediction, however the teacher distilled video model makes incorrect prediction.

Simple Yet Effective Synthetic Dataset Construction for Unsupervised Opinion Summarization

Ming Shen^{♥*} Jie Ma[♣] Shuai Wang[♣] Yogarshi Vyas[♣]
Kalpit Dixit[♣] Miguel Ballesteros[♣] Yassine Benajiba[♣]

[♥]Arizona State University [♣]AWS AI Labs

mshen16@asu.edu; {jlieman, wshui, yogarshi, kddixit, ballemig, benajiy}@amazon.com

Abstract

Opinion summarization provides an important solution for summarizing opinions expressed among a large number of reviews. However, generating aspect-specific and general summaries is challenging due to the lack of annotated data. In this work, we propose two simple yet effective unsupervised approaches to generate both aspect-specific and general opinion summaries by training on synthetic datasets constructed with aspect-related review contents. Our first approach, *Seed Words Based Leave-One-Out* (SW-LOO), identifies aspect-related portions of reviews simply by exact-matching aspect seed words and outperforms existing methods by 3.4 ROUGE-L points on SPACE and 0.5 ROUGE-1 point on OPOSUM+ for aspect-specific opinion summarization. Our second approach, *Natural Language Inference Based Leave-One-Out* (NLI-LOO) identifies aspect-related sentences utilizing an NLI model in a more general setting without using seed words and outperforms existing approaches by 1.2 ROUGE-L points on SPACE for aspect-specific opinion summarization and remains competitive on other metrics.

1 Introduction

Customer reviews play a vital role in decision-making for customers and product (or business) providers, as customers usually resort to reviews to guide their purchasing decisions and product providers improve their products based on reviews as feedback. However, it becomes hard for customers or product providers to read through all reviews before making decisions with the explosion of online reviews in recent years. Opinion summarization (Hu and Liu, 2006; Wang and Ling, 2016; Angelidis and Lapata, 2018; Bražinskas et al., 2020; Brazinskas et al., 2022; Angelidis et al., 2021; Amplayo et al., 2021a; Basu Roy Chowdhury

et al., 2022), the task of generating a general summary of *salient* opinions expressed among reviews, provides a feasible solution to this problem.

Different from summarization in Wikipedia and news domains (Nallapati et al., 2016; Narayan et al., 2018a; See et al., 2017; Narayan et al., 2018b; Liu and Lapata, 2019; Cachola et al., 2020), opinion summarization cannot rely on reference summaries for model training since it is difficult and expensive to annotate large scale reviews-summary pairs. Also, customers usually care about specific aspects of a product instead of a *general* high-level summary. Thus, fine-grained *aspect-specific* opinion summaries are required, and this makes the annotation process even more difficult and expensive.

Amplayo et al. (2021a) propose an abstractive approach to generate aspect-specific opinion summaries by training on synthetic datasets. They construct synthetic datasets with review elements (words, phrases, or sentences) identified by a multiple instance learning (MIL) module (Keeler and Rumelhart, 1991) learned with silver-standard labels obtained using aspect seed words. We first follow this direction to propose a more straightforward and effective method that excludes the complex learning module to identify aspect-related elements to construct synthetic datasets. Moreover, aspect seed words, which again require human efforts, may not always be available when moving to new domains. Thus we propose another more general solution without the curation and supervision of aspect seed words.

Specifically, we propose two simple yet effective methods to identify aspect-related review sentences and construct aspect-specific synthetic datasets in a *Leave-One-Out* (LOO) (Bražinskas et al., 2020; El-sahar et al., 2021; Brazinskas et al., 2022) style and then finetune pretrained language models (PLMs) on the synthetic datasets: (a) SW-LOO identifies aspect-related sentences by simply exact-matching aspect seed words and outperforms existing ap-

*Work done during an internship at AWS AI Labs.

proaches by 3.4 ROUGE-L points and 0.5 ROUGE-1 point on aspect opinion summaries of SPACE and OPOSUM+ respectively; (b) NLI-LOO identifies aspect-related sentences with a finetuned NLI (Bowman et al., 2015; Williams et al., 2018) model. Being the first approach that does not use aspect seed words, it outperforms existing approaches on aspect opinion summarization by 1.2 ROUGE-L points for SPACE and falls behind at most 1 ROUGE point on other metrics.

2 Problem Formulation

Let C denote a corpus of reviews on entities $\{e_1, e_2, \dots\}$ (products or business). Let $A_e = \{a_1, a_2, \dots, a_M\}$ denotes a set of aspects (e.g., *food* or *location* for a hotel) that are relevant for the domain of entities. For each entity e , we define its review set as $R_e = \{r_1, r_2, \dots, r_N\}$. Each review r is a collection of sentences $\{x_1, x_2, \dots\}$ and each sentence x is a sequence of tokens $\{w_1, w_2, \dots\}$. Each aspect a is represented by a small set of *seed words* (e.g., *meal* or *buffet* for *food* aspect) $S_a = \{v_1, v_2, \dots\}$. Our approaches generate two types of opinion summaries: (a) *general* summary that contains salient opinions over *all* aspects of the entities; and (b) *aspect* summary that focuses on only one specific aspect $a \in A_e$.

3 Synthetic Dataset Construction

Leave-One-Out (LOO) We construct synthetic datasets in a LOO style: from a pool of review elements (reviews or review sentences), an element is randomly sampled as a *pseudo-summary*, then we select input reviews from the remaining review elements.

3.1 Seed Words Based LOO

To build a synthetic reviews-summary pair for aspect a , as shown in the upper diagram of Figure 1, we first filter each review r into its aspect-related portion r^i where $r^i \subseteq \{x_1, x_2, \dots\}$ with each sentence in r^i containing at least one seed word in A_e . For example, for *food* aspect with its seed words $\{\text{breakfast, buffet, ...}\}$, a hotel review r_i : "*They have the most wonderful buffet in Bay Area. And the hotel is close to the airport. Forgot to mention, especially the breakfast is terrific.*" will be filtered into its aspect-related review portion r_i^i : "*They have the most wonderful buffet in Bay Area. Forgot to mention, especially the breakfast is terrific.*". Noticed that r_2^i is empty suppose there is no sentence in r_2

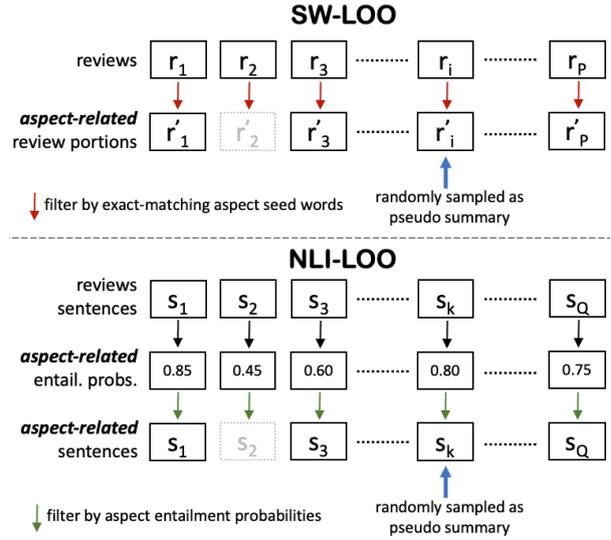


Figure 1. One synthetic data pair construction for aspect a in SW-LOO and NLI-LOO.

containing any seed word. Then we apply LOO construction on the filtered aspect-related review portions $\{r_1^i, r_3^i, \dots, r_p^i\}$ as shown in the diagram: r_i^i is randomly sampled as the pseudo summary and inputs are chosen from $\{r_1^i, r_3^i, \dots, r_p^i\} \setminus \{r_i^i\}$ by first ranking them with the pseudo-summary r_i^i based on ROUGE-1 score (Lin, 2004) and then truncating with a token budget j (truncate up to j tokens) since a concatenation of all filtered reviews cannot fit into the encoder of a PLM. Please refer to Appendix B for more details and analysis on SW-LOO.

3.2 NLI Based LOO

NLI Component In order to relax the requirement of aspect seeds (provided by humans) and to make a more scalable and general solution, we propose to use an NLI model to infer whether a review sentence is related to an aspect. Specifically, we set a review sentence as the premise and verbalize an aspect with the template: *the text is about {aspect}*, which we use as the hypothesis. If the entailment probability is higher than a threshold (0.9 for SPACE and 0.8 for OPOSUM+), we identify the sentence as related to the aspect with this entailment probability, else we set the aspect-related probability to 0.

To build a synthetic pair for aspect a , we first break all reviews into review sentences and filter out those that are not related to aspect a with the NLI model. As shown in the lower diagram of Figure 1, each sentence is first passed through the NLI model to infer its probability of relatedness

to aspect a , so s_2 with entailment probability of 0.45 will be filtered out if the threshold is set to 0.5. Then we apply LOO construction on all aspect-related sentences $\{s_1, s_3, \dots, s_Q\}$ and we also use a token budget to truncate ranked synthetic input similar to SW-LOO, however, different from SW-LOO where we use ROUGE-1 scores to rank, we calculate similarities based on entailment probabilities. Please refer to Appendix C for more details and analysis on NLI-LOO. Note that we filter the input reviews at sentence level for NLI-LOO and at review level in SW-LOO.

4 Summarization Model

We use T5 (Raffel et al., 2020), a sequence-to-sequence Transformer-based (Vaswani et al., 2017) PLM, to finetune our synthetic datasets similar to previous works (Ke et al., 2022; Amplayo et al., 2021a). For SW-LOO, we use the following template: “summarize based on aspect: [ASPECT] $\{aspect\}$ [ASPECT] with seed words: [SEED] $\{seed\ words\}$ [SEED]: $\{filtered\ review\}$ [SEP] $\{filtered\ review\}$... ” to convert synthetic input and for NLI-LOO, we use: “[ASPECT] $\{aspect\}$ [SEP] $\{aspect-related\ sent\}$ [SEP] $\{aspect-related\ sent\}$... ”. [ASPECT], [SEED], and [SEP] are special tokens, $\{aspect\}$ is an aspect name, $\{seed\ words\}$ are concatenation of seed words for an aspect, each $\{filtered\ review\}$ is a r_i^j in SW-LOO synthetic input, and each $\{aspect-related\ sent\}$ is a s_k in NLI-LOO synthetic input. For both methods, outputs are pseudo summaries.

5 Experiment

5.1 Datasets

We evaluate our methods on two opinion summarization datasets: SPACE (Angelidis et al., 2021), containing reviews from *hotel* domain, and OPOSUM+ (Amplayo et al., 2021a), containing Amazon product reviews from six different domains. Both datasets are comprised of a large corpus of raw reviews and a small development and test set with human-annotated aspects and general opinion summaries for evaluation. Aspect seed words are usually obtained with a small amount of human effort. For SW-LOO, we use the same seed words as in Amplayo et al. (2021a) (Appendix E). Refer to Appendix D for detailed descriptions and statistics of the two datasets.

Model	SPACE			OPOSUM+		
	R1	R2	RL	R1	R2	RL
LEXRANK	24.61	3.41	18.03	22.51	3.35	17.27
QT	28.95	8.34	21.77	23.99	4.36	16.61
ACESUM _{EXT}	30.91	8.77	23.61	26.16	5.75	18.55
SEMAE	31.24	10.43	24.14	-	-	-
SW-LOO _{EXT}	<u>33.14</u>	10.32	25.81	28.14	6.10	19.51
NLI-LOO _{EXT}	27.18	6.63	20.60	26.78	6.48	18.07
MEANSUM	25.68	4.61	18.44	24.63	3.47	17.53
COPYCAT	27.19	5.63	19.18	26.17	4.30	18.20
ACESUM	32.41	9.47	25.46	<u>29.53</u>	<u>6.79</u>	21.06
SW-LOO	34.68	11.50	28.83	30.00	6.92	<u>20.76</u>
NLI-LOO	31.57	<u>10.44</u>	<u>26.66</u>	28.90	6.60	20.11
HUMAN	44.86	18.45	34.58	43.03	16.16	31.53

Table 1. Evaluation for *aspect summaries* on SPACE and OPOSUM+ test sets. Best performances are in **bold** and second best performances are underlined.

5.2 Baselines

We compare our methods with several unsupervised extractive and abstractive approaches. Extractive approaches include CENTROID (Radev et al., 2004), LEXRANK (Erkan and Radev, 2004), QT (Angelidis et al., 2021), SEMAE (Basu Roy Chowdhury et al., 2022), and two extractive variants of our methods, SW-LOO_{EXT} and NLI-LOO_{EXT}, by feeding identified aspect-related sentences to LEXRANK instead of T5, similar to the idea in Amplayo et al. (2021a). Abstractive approaches include MEANSUM (Chu and Liu, 2019), COPYCAT (Bražinskas et al., 2020), and ACESUM (Amplayo et al., 2021a). Appendix F contains more details on baselines.

We also compare with two upper bounds reported in Amplayo et al. (2021a): an ORACLE that selects the review with the highest ROUGE score to the gold summary as the summary and a HUMAN upper bound that is calculated as the inter-annotator ROUGE scores.

5.3 Implementation

We first pre-process the raw corpus such as removing products with very few reviews and too long or short reviews as in Appendix G. We use T5-SMALL as our summarization models and larger T5 size does not show improvements as shown in Appendix L. We use a MNLI (Williams et al., 2018) finetuned BART-LARGE (Lewis et al., 2020) model in NLI-LOO. We choose this model given its better performance¹ in zero-shot topic classification. We perform simple hyper-parameter tuning

¹<https://joeddav.github.io/blog/2020/05/29/ZSL.html>

	Model	SPACE			OPOSUM+		
		R1	R2	RL	R1	R2	RL
Extractive	CENTROID	31.29	4.91	16.43	33.44	11.00	20.54
	LEXRANK	31.41	5.05	18.12	35.42	10.22	20.92
	QT	38.66	10.22	21.90	37.72	14.65	21.69
	ACESUM _{EXT}	35.50	7.82	20.09	38.48	15.17	22.82
	SEMAE	43.46	13.48	26.40	-	-	-
	SW-LOO _{EXT}	38.44	11.01	<u>25.62</u>	40.45	19.13	<u>23.20</u>
	NLI-LOO _{EXT}	25.07	4.52	<u>16.16</u>	<u>39.79</u>	<u>18.33</u>	23.49
Abstractive	MEANSUM	34.95	7.49	19.92	26.25	4.62	16.49
	COPYCAT	36.66	8.87	20.90	27.98	5.79	17.07
	ACESUM	40.37	11.51	23.23	32.98	10.72	20.27
	SW-LOO	<u>42.27</u>	<u>12.99</u>	23.47	36.19	12.17	21.11
	NLI-LOO	41.25	12.79	24.31	31.22	9.93	19.08
	ORACLE	40.23	13.96	23.46	41.88	21.52	29.30
	HUMAN	49.80	18.80	29.19	55.42	37.26	44.85

Table 2. Evaluation for *general summaries* on SPACE and OPOSUM+ test sets. Best performances are highlighted in bold and second-best performances are underlined.

on dev sets and select checkpoints with the best ROUGE-L scores to report performances on test sets. Please refer to Appendix H for more details such as training configurations and other analyses.

5.4 Results

We evaluate the quality of generated opinion summaries using ROUGE1/2/L F1 scores (Lin, 2004). Example summaries generated by our methods are shown in Table 12 and Table 13 in Appendix.

Aspect Opinion Summarization Table 1 contains the results of all baselines and our methods on the two benchmark datasets. Despite its simplicity, SW-LOO achieves the highest scores on both datasets across all metrics except RL for OPOSUM+ with only 0.3 points behind the best-performing baseline. On the other hand, NLI-LOO achieves higher R2 and RL scores on SPACE than existing methods despite using no seed words. While it falls behind other methods on OPOSUM+, it is at most 1 point behind across all metrics. This highlights that even without aspect seed words, NLI-LOO is possible to compete with SOTA aspect-based opinion summarization methods.

Next, we turn to the evaluation of extractive versions of our methods. We observe SW-LOO_{EXT} achieves higher R1 and RL scores on SPACE but falls behind on OPOSUM+ by at most 1.5 point compared with all baselines. This is consistent with the finding in Amplayo et al. (2021a) that a simple centrality-based extractive approach such as LEXRANK are strong baselines as long as input sentences are already aspect-related. And

Model	SPACE		OPOSUM+	
	Aspect	General	Aspect	General
SW-LOO	27.59	23.42	20.41	20.58
<i>w/ Training Random</i>	24.24	24.70	19.75	18.71
<i>w/ Inference Random</i>	23.46	22.04	18.76	19.41
<i>w/ Both Random</i>	14.71	21.82	18.06	18.15
NLI-LOO	25.92	25.13	19.21	19.32
<i>w/ Training Random</i>	22.05	22.06	18.42	19.37
<i>w/ Inference Random</i>	24.33	24.56	18.10	16.97
<i>w/ Both Random</i>	16.14	22.50	17.83	19.69

Table 3. *Training Random* means randomly selecting sentences as pseudo summary and input during synthetic dataset construction. *Inference Random* means randomly selecting sentences as input during inference. We report RL scores of our approaches on dev sets.

SW-LOO_{EXT} outperforming ACESUM_{EXT} further shows that our simple filtering method using exact-matching seed words already produces good enough aspect-related sentences compared with the extra learning module used in Amplayo et al. (2021a). However, NLI-LOO_{EXT}, is not able to outperform the best baseline, and we hypothesize the reason is that NLI model filtered aspect-related sentences are still noisy so that a summarization model is required to serve as regularization.

Finally, comparing our four methods, SW-LOO achieves the best performances with the supervision of seed words, NLI-LOO comes second despite the lack of seed words supervision, and our two extractive versions come last since the ground truth summaries are in nature abstractive.

General Opinion Summarization As shown in Table 2, on SPACE, SW-LOO and NLI-LOO outperform the SOTA abstractive system, ACESUM, but under-perform SOTA extractive system, SEMAE. We observe the same trend between SW-LOO_{EXT} and ACESUM_{EXT} as in aspect opinion summarization and this again shows the simple yet effective nature of our filtering method. For OPOSUM+, SW-LOO_{EXT} and NLI-LOO_{EXT} outperform existing methods given that the annotated general summaries for OPOSUM+ are extractive, SW-LOO outperforms existing abstractive approaches, and NLI-LOO falls behind with only 1 point.

5.5 Ablation Study

We conduct ablation experiments with random filtering to study the importance of the filtering strategies in our two methods. We introduce randomness in two different phases. First, when constructing synthetic pairs, instead of using our filtering strategies before applying LOO construction, we ran-

domly select sentences as pseudo-summary and input. This is essentially a random LOO baseline. Second, during inference, we sample random sentences to feed into T5 encoder instead of using our filtering strategies to select aspect-related elements. Finally, we combine these two random strategies. Results in Table 3 show that our sentence filtering strategies are crucial since ROUGE scores drop drastically as more randomness is introduced. This is more severe for aspect summarization since aspect-specific synthetic dataset construction needs to focus on particular aspects. However, randomly selecting sentences is possible to cover most aspects by chance for general summarization.

6 Conclusion

In this work, we propose two simple yet effective unsupervised approaches that generate aspect and general opinion summaries by training on synthetic datasets. SW-LOO constructs synthetic datasets simply by exact-matching aspect seed words and outperforms existing methods consistently on all metrics and datasets. Being the first work that generates aspect summaries without using aspect seed words, NLI-LOO constructs synthetic datasets with an out-of-the-box NLI model and achieves on-par and sometimes even better performances compared with existing methods.

Limitations

One of the biggest challenge in opinion summarization is the multi-document setting where each document represents one product review. Since the number of reviews for a product tends to be large, it would be unrealistic to concatenate all input reviews and train to generate a summary in an end-to-end fashion limited by modern hardware capacity, for example, the GPU memory needed is quadratic w.r.t the input length for all transformer-based PLM. In this work, we tackle this problem by pre-filtering reviews using some heuristics (aspect seed words matching and NLI model selecting) into sub-elements of reviews with much smaller sizes. However, information is very likely to get lost and become incomplete in the pre-filtering phase, leading to inaccurate summarization. Our approach is exactly facing this problem. One way to address this drawback is to first condense each review into an encoding that contains key information of the review such as opinion aspect and opinion sentiment, and then aggregate all review vectors to gen-

erate a summary. Amplayo and Lapata (2021) call this pipeline as CONDENSE-ABSTRACT and it has been used in both supervised and unsupervised general opinion summarization (Chu and Liu, 2019; Coavoux et al., 2019; Iso et al., 2021; Amplayo and Lapata, 2021; Isonuma et al., 2021).

References

- Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021a. [Aspect-controllable opinion summarization](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6578–6593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Reinald Kim Amplayo, Stefanos Angelidis, and Mirella Lapata. 2021b. [Unsupervised opinion summarization with content planning](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12489–12497.
- Reinald Kim Amplayo and Mirella Lapata. 2020. [Unsupervised opinion summarization with noising and denoising](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1934–1945, Online. Association for Computational Linguistics.
- Reinald Kim Amplayo and Mirella Lapata. 2021. [Informative and controllable opinion summarization](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2662–2672, Online. Association for Computational Linguistics.
- Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. [Extractive opinion summarization in quantized transformer spaces](#). *Transactions of the Association for Computational Linguistics*, 9:277–293.
- Stefanos Angelidis and Mirella Lapata. 2018. [Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.
- Somnath Basu Roy Chowdhury, Chao Zhao, and Snigdha Chaturvedi. 2022. [Unsupervised extractive opinion summarization using sparse coding](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1209–1225, Dublin, Ireland. Association for Computational Linguistics.
- Laura-Ana-Maria Bostan and Roman Klinger. 2018. [An analysis of annotated corpora for emotion classification in text](#). In *Proceedings of the 27th International Conference on Computational Linguistics*,

- pages 2104–2119, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. [Unsupervised opinion summarization as copycat-review generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online. Association for Computational Linguistics.
- Arthur Bražinskas, Ramesh Nallapati, Mohit Bansal, and Markus Dreyer. 2022. [Efficient few-shot fine-tuning for opinion summarization](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1509–1523, Seattle, United States. Association for Computational Linguistics.
- Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. [TLDR: Extreme summarization of scientific documents](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.
- Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. [Importance of semantic representation: Dataless classification](#). In *AAAI*.
- Eric Chu and Peter Liu. 2019. [Meansum: a neural model for unsupervised multi-document abstractive summarization](#). In *International Conference on Machine Learning*, pages 1223–1232. PMLR.
- Maximin Coavoux, Hady Elsahar, and Matthias Gallé. 2019. [Unsupervised aspect-based multi-document abstractive summarization](#). In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.
- Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. [Predicting query performance](#). In *SIGIR '02*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Bogdan Dumitrescu and Paul Irofti. 2018. [Dictionary learning algorithms and applications](#). Springer.
- Hady Elsahar, Maximin Coavoux, Jos Rozen, and Matthias Gallé. 2021. [Self-supervised and controlled multi-document opinion summarization](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1646–1662, Online. Association for Computational Linguistics.
- Günes Erkan and Dragomir R. Radev. 2004. [Lexrank: Graph-based lexical centrality as salience in text summarization](#). *J. Artif. Intell. Res.*, 22:457–479.
- Minqing Hu and Bing Liu. 2006. [Opinion extraction and summarization on the web](#). In *Aaai*, volume 7, pages 1621–1624.
- Hayate Iso, Xiaolan Wang, Yoshihiko Suhara, Stefanos Angelidis, and Wang-Chiew Tan. 2021. [Convex Aggregation for Opinion Summarization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3885–3903, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Masaru Isonuma, Junichiro Mori, Danushka Bollegala, and Ichiro Sakata. 2021. [Unsupervised abstractive opinion summarization by generating sentences with tree-structured topic guidance](#). *Transactions of the Association for Computational Linguistics*, 9:945–961.
- Wenjun Ke, Jinhua Gao, Huawei Shen, and Xueqi Cheng. 2022. [Consistsum: Unsupervised opinion summarization with the consistency of aspect, sentiment and semantic](#). In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 467–475.
- Jim Keeler and David Rumelhart. 1991. [A self-organizing integrated segmentation and recognition neural net](#). *Advances in neural information processing systems*, 4.
- Diederik P. Kingma and Max Welling. 2014. [Auto-encoding variational bayes](#). *CoRR*, abs/1312.6114.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yang Liu and Mirella Lapata. 2019. [Text summarization with pretrained encoders](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*

- (EMNLP-IJCNLP), pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *ICLR*.
- Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. [Image-based recommendations on styles and substitutes](#). In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15*, page 43–52, New York, NY, USA. Association for Computing Machinery.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence RNNs and beyond](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. [Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. [Ranking sentences for extractive summarization with reinforcement learning](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. [The pagerank citation ranking : Bringing order to the web](#). In *WWW 1999*.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2018. [A deep reinforced model for abstractive summarization](#). In *International Conference on Learning Representations*.
- Dragomir R. Radev, Hongyan Jing, Magorzata Sty, and Daniel Tam. 2004. [Centroid-based summarization of multiple documents](#). *Inf. Process. Manag.*, 40:919–938.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21(140):1–67.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Sascha Rothe, Joshua Maynez, and Shashi Narayan. 2021. [A thorough evaluation of task-specific pre-training for summarization](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 140–145, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.
- Aaron Van Den Oord, Oriol Vinyals, et al. 2017. [Neural discrete representation learning](#). *Advances in neural information processing systems*, 30.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Lu Wang and Wang Ling. 2016. [Neural network-based abstract generation for opinions and arguments](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 47–57, San Diego, California. Association for Computational Linguistics.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. [Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. [Pegasus: Pre-training with extracted gap-sentences for abstractive summarization](#). In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. [Character-level convolutional networks for text classification](#). In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Related Works

Unsupervised opinion summarization is the task of summarizing opinionated text such as customer reviews without training on gold reviews-summary pairs. Recent works have been using autoencoders (Kingma and Welling, 2014) and synthetic datasets construction, or a mix of both, to tackle the zero-shot setting.

An autoencoder model consists of an encoder that maps the input into latent embedding space and a decoder that reconstructs the original input from the latent space. The latent representation learned can be later aggregated or can be used to cluster and select text to perform both extractive and abstractive summarization. Chu and Liu (2019); Bražinskas et al. (2020) aggregate the input reviews latent representations by averaging then generate the summaries conditioned on it. Angelidis and Lapata (2018) utilizes the latent representation with aspect specificity and sentiment polarity to guide the selection of review texts as extractive summaries. Recently, Angelidis et al. (2021) proposes the first approach that generates both general and *aspect-specific* opinion summaries in an extractive manner. They first leverage Vector-Quantized Variational Autoencoder (Van Den Oord et al., 2017) to cluster review sentences and then use a popularity-driven extraction algorithm to summarize. Similar to Angelidis et al. (2021), Basu Roy Chowdhury et al. (2022) learns representations of texts over latent semantic units using dictionary learning (Dumitrescu and Irofti, 2018). Other autoencoder-related methods include denoising autoencoder (Amplayo and Lapata, 2020) and Coavoux et al. (2019), an encoder-decoder architecture that utilizes clustering of encoding space to extract summaries.

Another direction of work creates synthetic datasets utilizing the largely available amount of online customer reviews. Synthetic datasets are usually constructed in a *leave-one-out* (LOO) style that one review is first randomly sampled as a pseudo-summary, and then a subset of reviews are selected or generated as input reviews to be paired with the pseudo-summary to enable supervised training. Methods of selecting and generating input reviews include random sampling (Bražinskas et al., 2020), generating noisy versions of the pseudo-summary (Amplayo and Lapata, 2020), selecting reviews that have closer distribution with the pseudo-summary in the embedding space (Am-

playo et al., 2021b; Ke et al., 2022), and selecting more textual similar reviews (Elsahar et al., 2021; Bražinskas et al., 2022). Recently, Amplayo et al. (2021a) proposes the first abstractive approach that can generate both general and aspect summaries. Their method build synthetic datasets by identifying aspect-specific elements with a multiple instance learning (MIL) model (Keeler and Rumelhart, 1991) using aspect seed words. Our work is closest to Amplayo et al. (2021a) in that we also build synthetic datasets by identifying aspect-specific elements, however, our methods do not require extra learning components such as MIL but achieve better performances.

Besides unsupervised opinion summarization, our second method, NLI-LOO is related to the recent approach (Yin et al., 2019) that utilizes NLI (Bowman et al., 2015; Williams et al., 2018) models to tackle zero-shot text classification (Chang et al., 2008) (multi-class and multi-label) problem such as topic detection (Zhang et al., 2015) and emotion detection (Bostan and Klinger, 2018). The main idea is to solve the classification problem by casting the problem into NLI format. Specifically, the text to be classified becomes the premise, and class labels are converted into natural language format (verbalization) to be used as the hypothesis. If the text entails the verbalized class label, then the text belongs to this class. In our work, we identify the relatedness of a review sentence to an aspect in such a way to construct synthetic datasets.

B SW-LOO Details

For general synthetic pairs construction, after filtering each review with seed words for each aspect, we make sure to sample one review such that its aspect-related portions for all aspects are non-empty and concatenate them as pseudo-summary. We retrieve top similar filtered reviews to each aspect-related portion in the pseudo-summary and concatenate them as general synthetic input, and the retrieval process is the same as in aspect synthetic pairs construction. General synthetic input and output are both approximately M times the length of those in aspect synthetic pairs where M is the number of aspects. For the summarization model, $\{aspect\}$ and $\{seed\ words\}$ are the concatenation of all aspects and all seed words for general synthetic pairs. Finally, we train all synthetic pairs together.

At inference time, we also first filter each review

into aspect-related portions. However, since there is no reference pseudo summary, we cannot truncate based on similarities to fit into T5 encoder. We adopt the *principle strategy* used in PEGASUS (Zhang et al., 2020) Gap Sentences Generation pre-training objective to select important reviews as input for inference. We show the effectiveness of adopting the principle strategy in Appendix I.

C NLI-LOO Details

Different from SW-LOO where we use ROUGE-1 scores, we calculate similarities based on aspect entailment probabilities to rank and truncate aspect-related sentences as synthetic input. For aspect synthetic pairs, we simply calculate the absolute probability difference between pseudo summary and aspect-related sentences. For general synthetic pairs, each review sentence (no matter whether aspect-related) corresponds to a probability vector of dimension M where M is the number of aspects and each element is the probability of the sentence being related to each aspect, and we calculate cosine similarities between the probability vectors of pseudo summary and review sentences that are related to at least one aspect (sum of the probability vector is non-zero). We use the same token budget to truncate review sentences to fit into T5 encoder for both aspect and general synthetic pairs. We also train all synthetic pairs together. Another way to calculate similarities is directly using cosine similarity between sentence embeddings, however, results reported in Appendix J do not show better performance.

During inference, we use 1 and all-one vectors with dimension M as reference vectors to rank and truncate review sentences for aspect and general input construction.

D Datasets Details

Hotel reviews in SPACE are collected from TripAdvisor and each hotel in the evaluation sets is annotated with seven types of summaries: six aspect-specific and one general, with three gold summaries for each type. The number of reviews for a hotel in the raw corpus varies but each hotel in the evaluation sets comes with 100 reviews. Product reviews from six domains: *laptop bag*, *Bluetooth headset*, *boots*, *keyboard*, *television*, and *vacuum* in OPOSUM+ are initially down-sampled from *Amazon Product Dataset*² (McAuley et al., 2015) by Ange-

²<http://jmcauley.ucsd.edu/data/amazon/>

Statistics	SPACE	OPOSUM+
domain	1	6
aspects per entity	6	3
<i>raw review corpus</i>		
entities	11.40K	95.55K
total reviews	1.14M	4.13M
<i>dev / test set</i>		
entities	25	30
reviews per entity	100	10
summaries per entity	3	3
total aspect summaries	450	270
total general summaries	75	<u>90</u>

Table 4. Detailed statistic for SPACE and OPOSUM+ datasets. Note that only gold general summaries for OPOSUM+, which is underlined in the table, are extractive.

lidis and Lapata (2018) and then further expanded by Amplayo et al. (2021a). Each product in the evaluation sets is annotated with four types of summaries: three aspect-specific and one general, with also three gold summaries for each type. The number of reviews for a product in the raw corpus also varies but each product in the evaluation sets comes with 10 reviews. All human-annotated summaries are abstractive except that general summaries in OPOSUM+ are extractive. Detailed statistics of the datasets are shown in Table 4.

E List of Seed Words

Aspect seed words (listed in Table 5 and 6) are usually automatically extracted using a variant of clarity scoring function (Cronen-Townsend et al., 2002) applied on a small amount of aspect annotation as described in Angelidis and Lapata (2018), and they can be further manually improved by domain experts as in Amplayo et al. (2021a).

F Baselines Details

Extractive Approaches We first compare against two traditional approaches: CENTROID selects the review closest to the centroid of all reviews as the summary; LEXRANK selects the most salient review sentences as summary similar to PAGERANK (Page et al., 1999). BERT (Devlin et al., 2019) embedding is used to represent sentences in both traditional methods. More recent systems include QT (described in Section 1) and SEMAE. Inspired by QT, SEMAE represents text over latent semantic units using dictionary learning.

Abstractive Approaches MEANSUM generates summaries by reconstructing the mean of reviews'

Aspect	Hotel
building	lobby pool decor gym area
cleanliness	clean spotless garbage dirty stain
food	breakfast food buffet restaurant meal
location	location walk station distance bus
rooms	room bed bathroom shower spacious
service	staff service friendly helpful desk

Table 5. Seed words for *hotel* domain in SPACE dataset.

Aspect	Laptop Bag
looks	looks color stylish looked pretty
quality	quality material poor broke durable
size	fit fits size big space

Aspect	Boots
comfort	comfortable foot hurt ankle comfy
looks	cute look looked fringe style
size	size half big little bigger

Aspect	Bluetooth Headset
comfort	ear fit comfortable fits buds
ease of use	easy button simple setup control
sound quality	sound quality hear noise volume

Aspect	Keyboard
quality	working months build stopped quality
comfort	feel comfortable feels mushy shallow
layout	key keys delete backspace size

Aspect	TV
connectivity	hdmi computer port usb internet
image quality	picture color colors bright clear
sound quality	sound speakers loud tinny bass

Aspect	Vacuum
accessory	filter brush attachments attachment turbo
ease of use	easy push concerns awkward impossible
suction	suction powerful power hair quiet

Table 6. Seed words for various domains in OPOSUM+ dataset.

representations using autoencoder. COPYCAT uses a hierarchical variational autoencoder to learn latent codes for the summaries. The most recent approach is ACESUM. (described in Section 1).

Note that LEXRANK, MEANSUM, and COPYCAT do not support aspect-specific summary generation, Amplayo et al. (2021a) adopt a simple

sentence-filtering strategy to enable it. Specifically, after training a general opinion summarization model, during inference for aspect summaries, they filter out input review sentences that are not aspect-related using cosine similarities scores between BERT embeddings of review sentences and aspect seed words before feeding into general summarization model.

G Datasets Pre-Processing

We pre-process differently for our two methods on the same dataset since we want to control the constructed synthetic datasets to have reasonable sizes and resemble properties of test time data such as the number of reviews per product and average review length. We use dev sets to observe such properties. In SW-LOO, we first remove reviews with less than 20 words and then remove hotels with less than 10 reviews for SPACE; we first remove reviews less than 20 or more than 100 words then remove products with less than 12 reviews for OPOSUM+. In NLI-LOO, we remove reviews with less than 10 or more than 120 words for SPACE and remove reviews with less than 20 or more than 100 words for OPOSUM+.

H Implementation Details

We use T5 implementation from HuggingFace³ (Wolf et al., 2020). We use AdamW (Loshchilov and Hutter, 2019) optimizer without weight decay and set 0.9, 0.999, 1×10^{-8} for β_1 , β_2 , ϵ . We train all summarization models for a total of 25K steps on the combination of aspect and general synthetic pairs. We set ngram refraining size (Paulus et al., 2018) to 3 during inference. We tune initial learning rate in $[1e-6, 4e-5, 3e-4]$ and batch size in $[8, 16]$. We tune beam search size during inference in $[2, 4]$. For SW-LOO_{EXT} and NLI-LOO_{EXT}, we use [CLS] token embedding in the last layer of BERT as the sentence representation. We concatenate top 6 sentences returned by LEXRANK as general summary, and tune in $[2, 4]$ for aspect summary. We also use two sizes of BERT: BERT-BASE and BERT-LARGE. All computations are performed on 8-GPU p3.16xlarge Amazon instance. The best hyper-parameter settings for all experiments can be found in Table 7.

During preliminary studies for aspect synthetic pairs construction, we find that for SPACE, us-

³https://huggingface.co/docs/transformers/model_doc/t5

SW-LOO			
SPACE	asp.	lr=3e-4, bch=16, bm=2	
	gen.	lr=3e-4, bch=16, bm=2	
OPOSUM+	asp.	lr=3e-4, bch=16, bm=2	
	gen.	lr=1e-6, bch=16, bm=2	
NLI-LOO			
SPACE	asp.	lr=4e-5, bch=16, bm=2	
	gen.	lr=4e-5, bch=16, bm=2	
OPOSUM+	asp.	lr=3e-4, bch=8, bm=4	
	gen.	lr=1e-6, bch=16, bm=2	
SW-LOO _{EXT}			
SPACE	asp.	BERT-Base, n=2	
	gen.	BERT-Base, n=2	
OPOSUM+	asp.	BERT-Base, n=2	
	gen.	BERT-Large, n=2	
NLI-LOO _{EXT}			
SPACE	asp.	BERT-Large, n=2	
	gen.	BERT-Base, n=4	
OPOSUM+	asp.	BERT-Large, n=4	
	gen.	BERT-Large, n=4	

Table 7. Best hyper-parameter settings on SPACE and OPOSUM+ dev sets: lr stands for AdamW initial learning rate, bch stands for training batch size, and bm stands for beam search size at inference time.

ing sampled filtered aspect-related review portion as pseudo-summary rather than the original review that contains the pseudo-summary gives better downstream ROUGE scores, but it is the opposite way with OPOSUM+. Please refer to Appendix K for analyses on pseudo-summary granularity.

SW-LOO For SPACE, we add a linear learning rate warm-up in the first 500 steps and save checkpoints every 500 steps. Since there are totally 6 aspects for SPACE and very few reviews containing seed words for all 6 aspects can be sampled as pseudo summaries for general synthetic pairs construction, we relax the constraint of pseudo summaries containing seed words for all 6 aspects to 4 aspects. We set 200 as the token budget to truncate ranked aspect-related review portions for aspect synthetic pairs construction, and 150 as the token budget in principle strategy when selecting important sentences as input during inference for SPACE. We set 1536 and 200 as the maximum input and output token length of T5 for all SW-LOO experiments. Notice that this exceeds 512, which is the maximum token length that T5 is pretrained on, but recent works (Zhang et al., 2020; Rothe et al., 2021) have shown that seq2seq PLMs generalize

Model	Aspect			General		
	R1	R2	RL	R1	R2	RL
SW-LOO	33.11	10.98	27.59	40.26	12.04	23.42
<i>w/o Prin. Sel.</i>	30.88	9.74	25.78	35.84	10.28	21.80

Table 8. Randomly or using principle strategy to select aspect-related review portions in order to fit into the encoder of T5. Performances are reported on SPACE dev set.

well even when finetuned on longer sequences not observed at pretraining phase. For OPOSUM+, we add linear learning rate warm-up in the first 250 steps. We set 300 as the token budget to truncate for aspect synthetic pairs construction. There are ~ 50K aspect and ~ 5K general synthetic pairs for SPACE, ~ 70K aspect and ~ 6K general synthetic pairs for OPOSUM+.

NLI-LOO For SPACE, we add linear learning rate warm-up to first 1000 steps, and 500 steps for OPOSUM+. We set 0.9 and 0.8 as the entailment probability threshold for SPACE and OPOSUM+ based on our preliminary experiments (lower thresholds make identified aspect-related sentences too noisy and further hurt downstream ROUGE scores). For summarization models, we set 500 as the token budget for both aspect and general synthetic pairs construction for both datasets and set 512 and 150 for maximum input and output token length of T5. There are ~ 36K aspect and ~ 6K general synthetic pairs for SPACE, ~ 70K aspect-specific and ~ 28K general synthetic pairs for OPOSUM+.

I Principle Strategy Effectiveness

Unlike OPOSUM+, there are 100 reviews for each hotel in SPACE evaluation sets. During inference, we cannot simply concatenate all filtered reviews as input since they cannot fit into T5 encoder. We adopt the principle strategy introduced in PEGASUS to select the most important filtered reviews and concatenate them as input for inference. In Table 8, we show the effectiveness of the principle strategy by comparing it with randomly selecting filtered reviews as input for inference.

J Similarity Metric

Different from using aspect entailment probability, we can also use sentence embeddings (Reimers and Gurevych, 2019) to calculate the cosine similarity between pseudo summary and aspect-related review sentences to construct synthetic input. Specifi-

	Model	SPACE			OPOSUM+		
		R1	R2	RL	R1	R2	RL
Asp.	NLI-LOO	30.20	9.84	25.92	27.48	5.64	19.21
	w/ <i>Sent. Sim.</i>	29.87	9.30	25.26	27.00	6.20	19.06
Gen.	NLI-LOO	41.17	12.34	25.13	31.10	10.09	19.32
	w/ <i>Sent. Sim.</i>	25.01	9.68	17.34	31.11	10.43	19.86

Table 9. Calculate cosine similarity using aspect entailment probability or sentence embeddings when constructing synthetic datasets in NLI-LOO. Performances are reported on dev sets for both datasets.

Granularity	R1	Aspect			General		
		R2	RL	R1	R2	RL	
SPACE							
Sentence	33.11	10.98	27.59	40.26	12.04	23.42	
Review	25.01	6.42	18.04	39.86	11.21	23.07	
OPOSUM+							
Review	29.18	6.38	20.41	36.16	11.89	20.58	
Sentence	22.34	5.06	17.33	20.06	6.76	13.86	

Table 10. Pseudo summary granularity study for SW-LOO and NLI-LOO. Performances are reported on dev sets. Note that in our main experiments, we use sentence level pseudo summary for SPACE and review level for OPOSUM+

cally, we use `all-mpnet-base-v2`⁴, which is a sentence embedding model finetuned on a 1B sentence pairs dataset with a self-supervised contrastive learning objective. Results in Table 9 show that there is no significant difference except general summarization for SPACE where using sentence embeddings is much worse than using aspect entailment probability.

K Pseudo Summary Granularity

We use different pseudo-summary granularity for two datasets: sentence level for SPACE and review level for OPOSUM+. Sentence level directly uses sampled filtered aspect-related review portion (SW-LOO) or sampled aspect-related review sentence (NLI-LOO) as pseudo-summary, and review level uses the original review that contains the sampled pseudo-summary as pseudo-summary. Results in Table 10 show the importance of design choices for synthetic datasets construction.

L T5 Model Sizes

We use different T5 sizes including T5-SMALL, T5-BASE, and T5-LARGE as summarization models. Results in Table 11 show that larger summarization models do not necessarily guarantee better

⁴https://www.sbert.net/docs/pretrained_models.html

	Model	SPACE			OPOSUM+		
		R1	R2	RL	R1	R2	RL
Aspect Summary	SW-LOO						
	T5-SMALL	33.11	10.98	27.59	29.18	6.38	20.41
	T5-BASE	33.43	11.08	27.73	30.03	6.60	20.53
	T5-LARGE	33.70	10.77	27.60	28.98	6.15	20.32
	NLI-LOO						
	T5-SMALL	30.20	9.84	25.92	27.48	5.64	19.21
	T5-BASE	30.24	10.04	25.95	27.58	5.28	19.29
	T5-LARGE	30.61	9.50	25.68	27.14	5.47	19.55
	General Summary	SW-LOO					
T5-SMALL		40.26	12.04	23.42	36.16	11.89	20.58
T5-BASE		41.31	12.47	23.12	35.53	11.65	20.33
T5-LARGE		39.90	10.94	22.64	32.96	10.24	19.41
NLI-LOO							
T5-SMALL		41.17	12.34	25.13	31.10	10.09	19.32
T5-BASE		37.49	11.44	22.91	26.51	6.74	17.08
T5-LARGE		37.57	10.14	21.77	30.41	6.77	17.37

Table 11. Using different T5 sizes as summarization model. Performances are reported on dev sets for both datasets.

downstream ROUGE scores and sometimes even hurt downstream performances. Our hypothesis is that larger models overfit synthetic datasets and thus perform slightly worse on downstream evaluation sets.

SW-LOO Summaries	
Building	The pool area was very nice and the room was clean and comfortable.
Cleanliness	Our room was very clean and comfortable.
Food	The breakfast was great and the staff was very helpful and helpful.
Location	The hotel is located right next to the main road and is a short walk from the beach.
Rooms	The room was very clean and comfortable.
Service	The staff was very friendly and helpful.
General	The pool was very nice and clean. We were able to walk to the beach and Duval st. from the hotel, so we had a nice view of the harbor and the sea! The breakfast was great and we stayed in October and were very pleased with the location - right next to all the restaurants ... The room was small but very small and very comfortable with clean and comfortable beds.

NLI-LOO Summaries	
Building	The hotel is a beautiful old hotel.
Cleanliness	The room was clean and the staff was very helpful.
Food	The breakfast was great and the view from the rooftop was amazing.
Location	The location is great - just a short walk to the Spanish Steps and the metro station.
Rooms	The rooms are small by European standards, but very clean and comfortable.
Service	the service was excellent and the staff was very friendly and helpful.
General	The hotel is very clean and the staff is friendly and helpful. The room was very small and clean, but the bathroom was a bit small compared to the other rooms in the UK. It is OK to stay here again. I would stay there again if you want to go back to Europe! The location is great - the city is just ten minutes walk from the metro station andn't be disappointed with the price of the rooms.

Table 12. *General* and *aspect-level* summaries for a hotel in SPACE dataset generated by SW-LOO and NLI-LOO

SW-LOO Summaries	
Sound Quality	I love this headset. It's a great product, but it doesn't have any issues with the sound! It is OK if you are looking for something that can be used for your Samsung TV?
Comfort	I love this headset. It's a great headset for the price, but it doesn't fit my ear perfectly!
Ease of Use	I bought this for my Motorola. It is very easy to set up, and the buttons are very comfortable!
General	I haven't found any way of getting that to be consistently good. The earpieces are not as sturdy or high quality in material as a Motorola, but the buttons are quite accessible and the sound varies based on how it's fitting into my ears! The set is very comfortable and has great range (roughly 100 feet) and connects easily to my iPhone with me - but it is not too big for me to wear if it doesn't fit my TV.

NLI-LOO Summaries	
Sound Quality	I love these headphones. They are very comfortable, sound quality is good and they're very good quality for the price!
Comfort	I love these headphones. They are very comfortable, and the sound quality is great! They're a little tight on my ears but if you aren't sure how long they will last you...
Ease of Use	I bought these for my Motoactv. They are very comfortable to wear, and they don't touch my neck at all!
General	I bought these headphones in a package for the Motoactv. They are very comfortable, the neck band doesn't touch my neck at all allowing for free movement! The sound is very good and fits comfortably in my ears... but it takes some time to find the right angel and fit it right in.

Table 13. *General* and *aspect-level* summaries for a product in "Bluetooth Headset" domain of OPOSUM+ dataset generated by SW-LOO and NLI-LOO.

Towards Fine-tuning Pre-trained Language Models with Integer Forward and Backward Propagation

Mohammadreza Tayaranian^{*1} Alireza Ghaffari^{*1} Marzieh S. Tahaei¹

Mehdi Rezagholizadeh¹ Masoud Asgharian² Vahid Partovi Nia¹

¹ Huawei Noah’s Ark Lab, Montreal Research Center

² Department of Mathematics and Statistics, McGill University

{mohammadreza.tayaranian, alireza.ghaffari, marzieh.tahaei}@huawei.com

{mehdi.rezagholizadeh, vahid.partovinia}@huawei.com

masoud.asgharian2@mcgill.ca

Abstract

The large number of parameters of some prominent language models, such as BERT, makes their fine-tuning on downstream tasks computationally intensive and energy hungry. Previously researchers were focused on lower bit-width integer data types for the forward propagation of language models to save memory and computation. As for the backward propagation, however, only 16-bit floating-point data type has been used for the fine-tuning of BERT. In this work, we use integer arithmetic for both forward and back propagation in the fine-tuning of BERT. We study the effects of varying the integer bit-width on the model’s metric performance. Our integer fine-tuning uses integer arithmetic to perform forward propagation and gradient computation of linear, layer-norm, and embedding layers of BERT. We fine-tune BERT using our integer training method on SQuAD v1.1 and SQuAD v2., and GLUE benchmark. We demonstrate that metric performance of fine-tuning 16-bit integer BERT matches both 16-bit and 32-bit floating-point baselines. Furthermore, using the faster and more memory efficient 8-bit integer data type, integer fine-tuning of BERT loses an average of 3.1 points compared to the FP32 baseline.

1 Introduction

Over the past few years, integration of attention mechanisms into deep learning models led to the creation of transformer based models. BERT (Devlin et al., 2018) is a prominent transformer based language model which has shown state-of-the-art performance in natural language processing (NLP) tasks.

BERT requires high memory and computational resources due to its large number of parameters. Having large number of parameters incurs challenges for inference, training, and also fine-tuning

^{*}Equal contribution.

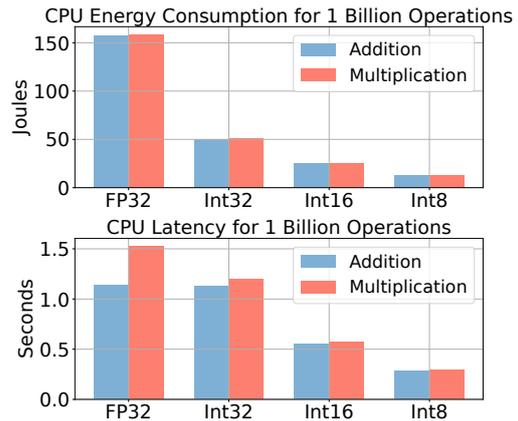


Figure 1: Energy consumption and latency of 1 billion operations using various data types, measured on an Intel[®] Xeon[®] CPU E5-2698 v4.

of this model. Moreover, the training phase i.e. pre-training and fine-tuning, involves more operations compared to the inference. More specifically, the training phase includes gradient computation and weight update that make the training more computationally intensive.

One method of reducing the computational complexity of deep learning models is to represent their parameters and activations in low bit-width data types. This reduces the memory footprint of the model and enables more efficient computations. For instance, Figure 1 shows that low-bit integer data types have higher throughput and better energy consumption compared to floating-point.

Previous research attempts at integer quantization of transformer based language models were only focused on forward propagation and the gradient computation were kept in 32-bit floating-point data type (FP32) (Bhandare et al., 2019; Kim et al., 2021; Zafir et al., 2019).

Furthermore, earlier efforts for using low bit-width data types for gradient computation of transformer based language models has only been limited to 16-bit floating-point (FP16). This method,

known as mixed precision training (Micikevicius et al., 2017), uses FP16 data type to represent weights, activations and gradients while using FP32 for the weight update.

Here we present an integer fine-tuning method for transformer based language models such as BERT. Unlike previous works, we use integer data types for both forward propagation and gradient computation during the fine-tuning of BERT. Moreover, we use the dynamic fixed-point format to represent floating-point numbers as integers.

Our integer mapping strategy can be used alongside floating-point numbers in fine-tuning and inference similar to mixed precision training. In our proposed strategy, the arithmetic of all the compute intensive layers for both forward and back propagation are performed using integer arithmetic while other components of the model, such as nonlinear functions and the weight updates are kept in FP32. We use integer versions of compute intensive layers such as linear, normalization (layer-norm), and embedding layers.

We study the effect of various bit-widths of the integer input activation and show that increasing the bit-width of the fixed-point mapping function improves the convergence behaviour of the model. This enables us to find the minimum bit-width required for integer fine-tuning of BERT.

Our fine-tuning experiments show that 16-bit integer BERT is able to match the metric performance of mixed precision FP16 and FP32 methods.

We also further reduce the bit-widths and show that integer fine-tuning of BERT with 8-bit integer weights and 12-bit integer activations has a score drop of 3.1 compared to the original performance.

To summarize, this paper makes the following contributions:

- Integer fine-tuning of transformer based language models that uses integer arithmetic for both the forward and back propagation of compute intensive layers such as linear, layer-norm, and embedding. To the best of our knowledge, this is the first time that integer data type is used for back propagation of pre-trained language models.
- Analyzing the effect of changing the bit-width of dynamic fixed-point format on the convergence of fine-tuning. **Remark 3** discusses that the convergence behaviour of our integer fine-tuning is directly related to the variance of dy-

namic fixed-point mapping and is controlled by the bit-width.

- We show that fine-tuning BERT using 16-bit integer numbers is able to outperform the FP16 mixed precision fine-tuning method.

The rest of this paper is structured as follows. Section 2 briefly discusses previous works in which low bit-width data types are used for inference and training of deep learning models. Section 3 provides details of our integer fine-tuning method, including the representation mapping functions and integer-only layers. The convergence behaviour of the dynamic fixed-point mapping is studied in Section 4 by providing empirical observations and theoretical analysis. The fine-tuning experiments on various integer and floating-point setups are presented in Section 5. Finally, Section 6 concludes the ideas proposed in this work.

2 Related Works

In this section we discuss the previous works that use low bit-width data types in transformer based language models. These works could be categorized into two major groups. In the first group, called low-bit inference, the low bit-width data types are used only in the forward propagation phase to improve computational complexity and reduce memory usage during the inference. In the second group, also known as low-bit training, lower bit-width data types are used for both the forward and back propagation phases.

2.1 Low-bit Inference

Previous research on low-bit inference quantize the model parameters and activations to speed up the forward propagation. This category is itself divided into quantization-aware training (QAT) and post-training quantization (PTQ) methods.

In QAT, quantization is performed during training, allowing the model parameters to adapt to the quantization noise. QAT relies on high-precision FP32 gradients to train the model and adapt it to the quantization noise.

For instance, (Zafrir et al., 2019) proposed Q8BERT which quantizes the inference computations of all linear and embedding layers of BERT to 8-bit integers and updates the quantization scale with a moving average. Similarly, (Shen et al., 2020) suggested Q-BERT which requires the computation of hessian matrix for each group of param-

eters to be used in a mixed precision fine-tuning with different bit-widths. (Kim et al., 2021) proposed I-BERT that uses a uniform quantization scheme to quantize input activations and weights of various components of BERT. In I-BERT, the quantization scaling factors are computed based on the distribution of the training data.

Unlike QAT that performs quantization of inference operations during training, Post-Training Quantization (PTQ) methods apply quantization to the parameters when the training is completed. Thus, they require extra calibration or parameter tuning to adapt the model to the quantized parameters.

For instance, (Bhandare et al., 2019) quantized the matrix multiplications of the original transformer architecture from (Vaswani et al., 2017) to 8-bit integer data type. Moreover, the quantization is done only for the forward propagation and requires extra calibration using validation data to tune the boundaries of the quantization function. (Zadeh et al., 2020) introduced GOBO which compresses the fine-tuned weights of BERT by grouping them into two categories of Gaussian and outlier. The outlier weights are kept in FP32, while the Gaussian weights are quantized to lower bits. For lower bit-width regimes, TernaryBERT and BinaryBERT are able to push the quantization to 2 and 1 bits respectively (Zhang et al., 2020a; Bai et al., 2020). They both rely on methods such as data augmentation and knowledge distillation to adapt the model to the low-bit weights.

2.2 Low-bit Training

Research on low-bit training try to perform both the forward propagation and gradient computation in low-bit arithmetic. Using low precision number formats for gradients reduces the model’s ability to adapt the parameters to the quantization noise, but increases the throughput and reduces the memory footprint.

FP16 mixed precision training (Micikevicius et al., 2017) is a common method currently for low-bit fine-tuning of transformer based language models. This method uses FP16 data type in both forward propagation and gradient computation, while using FP32 for the weight update. Unlike FP16 mixed precision training, our work uses dynamic fixed-point format which allows for multiple choices of bit-width for the data type. We show that our 16-bit integer fine-tuning method outperforms

FP16 mixed precision training in terms of metric score.

Using integer data types in the training of deep learning models has been previously studied for the computer vision tasks. For instance, (Zhang et al., 2020b) quantized the input activations, gradients and parameters of the linear layers for various convolutional neural networks (CNN). Similarly, (Zhao et al., 2021) adapted the quantization parameters by detecting the distribution of the gradients in the channel dimension. In both these works the quantization error is measured during training and is used to adjust the quantization scale, whereas our method does not require any information about distribution of data or gradients. (Zhu et al., 2020) applied a quantization scheme to train CNN architectures with “direction sensitive gradient clipping” and learning rate scaling to control the quantization error of gradients. Our integer fine-tuning method does not require gradient clipping and can follow the same loss trajectory as the floating-point baseline with the same hyperparameters. Our proposed method improves upon (Ghaffari et al., 2022) which uses dynamic fixed-point format for integer training of deep learning models. Unlike (Ghaffari et al., 2022), our work studies various bit-widths for both weights and activations to find the minimum bit-width required for fine-tuning BERT. Furthermore, we study integer training method on large language models where low-bit quantization is known to be a challenging task (Bondarenko et al., 2021). To the best of our knowledge, this is the first time where integer numbers are used for the back propagation of transformer based language models.

3 Methodology

3.1 Representation Mapping

We use the dynamic fixed-point format (Williamson, 1991) to map the floating-point numbers to integer data type. This format, also known as block floating-point, maps floating-point numbers to blocks of integer numbers, with each block having its unique scale. For more information on various number formats refer to Appendix A.

We use a linear fixed-point mapping function to map floating-point numbers to integer numbers. The linear fixed-point mapping converts a floating-point tensor \mathbf{F} to a tensor of integers and a single scale factor.

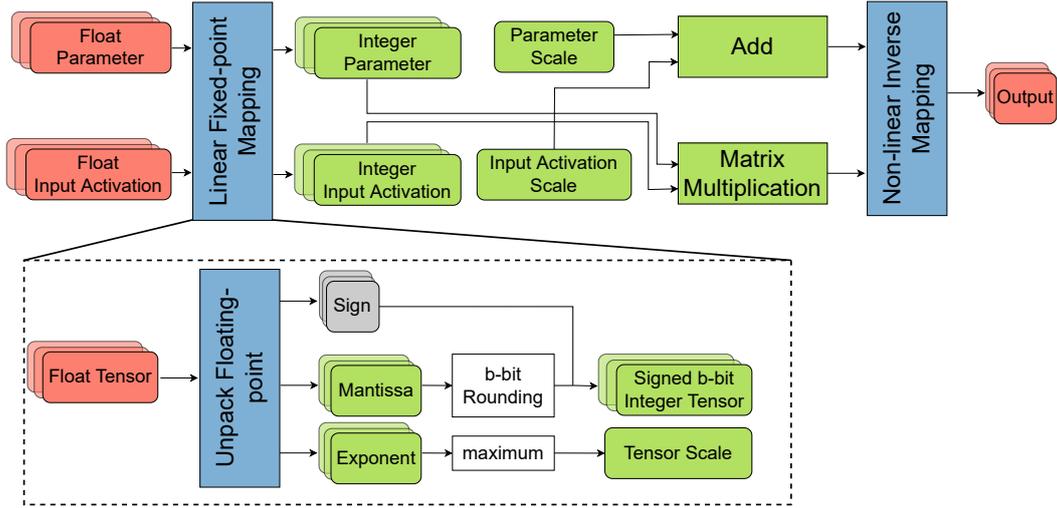


Figure 2: Forward propagation operations in an integer-only linear layer. Green boxes use integer arithmetic and red boxes use floating-point data type. Here, the integer output is generated using an integer matrix multiplication and the output scale is generated by a single add operation. The bottom panel shows the linear fixed-point mapping for the input tensors, that are the input activation and the parameter tensor in this figure.

The integers are obtained by rounding the floating-point mantissas. The scale is the maximum of the floating-point exponents of \mathbf{F} . The bottom section of Figure 2 shows the internal operations of the linear fixed-point mapping.

To map the fixed-point numbers to floating-point, a non-linear inverse mapping function is used. The inverse mapping converts integer numbers into normalized floating-point mantissas and packs each integer with its corresponding scale into a floating-point number.

Details of the representation mapping functions are provided in (Ghaffari et al., 2022). Our methodology differs in that it includes various bit-widths for both weights and activations for the fine-tuning of transformer based language models. We exploit this mapping strategy to explore various bit-widths for weights and activations in order to find the minimum bit-width for fine-tuning the model.

3.2 Integer Fine-tuning

Our method uses integer arithmetic for weights, activations and gradients, while the weight update is kept in FP32. Moreover, our proposed BERT setups use integer-only versions for all the linear, layer-norm and embedding layers in which internal operations are performed with integer arithmetic.

3.2.1 Linear Layer

Figure 2 depicts a high-level view of forward propagation operations of the integer-only linear layer. All the parameters and activations of the layer are

first mapped to dynamic fixed-point using the linear fixed-point mapping function. In the case of linear layer, the integer parameters and input activations are then sent to an integer matrix multiplication function to generate the integer output. If needed, the integer output could be mapped back to floating-point to be used by other layers of the model using the non-linear inverse mapping.

For back propagation, the gradients of the parameters and input activations are also computed using integer arithmetic. Using integer matrix multiplication, the output gradients are multiplied by input activations and parameters to compute the gradients. Since the weight update is performed in FP32, the integer gradients and their scales are passed to the non-linear inverse mapping to be mapped to FP32.

3.2.2 Layer-norm

The layer normalization or layer-norm performs the following operation on its input X (Ba et al., 2016):

$$\gamma \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta. \quad (1)$$

Here γ and β are the weight and bias parameters, and σ and μ are input standard deviation and mean respectively. For the forward propagation of integer layer-norm we map X to dynamic fixed-point format and compute σ and μ using integer arithmetic. Note that multiplication to γ and addition with β are also performed using integer arithmetic.

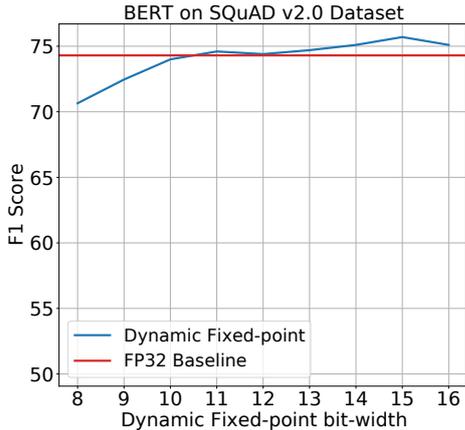


Figure 3: F1 score of fine-tuning BERT using b -bit gradients, and activations on SQuAD v2.0 dataset. For the 8-bit and 9-bit fixed-point bit-widths, we use 12-bit input activations.

Moreover, the back propagation also uses integer arithmetic to compute the gradients for the input, γ , and β .

3.2.3 Embedding Layer

The embedding layer is a lookup table that stores embeddings. The layer takes a list of indices as input and returns the list of corresponding embeddings for each index. The integer embedding layer, handles integer embeddings and needs less memory footprint to store these values. For the back propagation, the embedding layer applies the output integer gradients directly to each corresponding row of the lookup table.

4 Convergence Behaviour of Dynamic Fixed Point Mapping

4.1 Empirical Observations

Figure 2 shows that the bit-width, b , is controlled by adjusting the number of rounded bits in the rounding function. Here we study the effect of changing the integer bit-width on the metric performance of the model.

The motivation of varying the bit-width of the dynamic fixed-point is to control the variance induced by the linear fixed-point mapping. Our experiments show that using dynamic fixed-point with a bit-width of 10 achieves the same performance as the FP32 fine-tuning method. Figure 3 demonstrates the F1 score of fine-tuning BERT on SQuAD v2.0 dataset against the fixed-point bit-width. Note that the fixed-point arithmetic with a bit-width higher

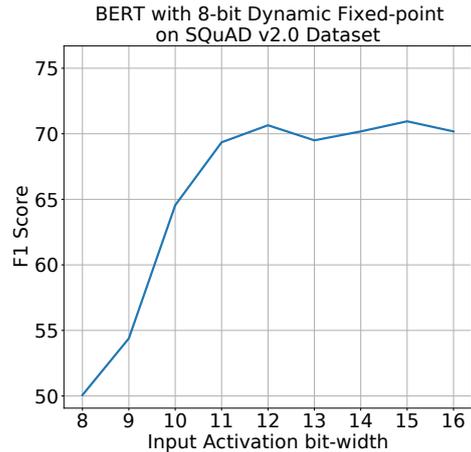


Figure 4: F1 score of fine-tuning BERT using 8-bit weights and gradients, with varying BERT input activation bit-width on SQuAD v2.0 dataset. Note that **Remark 3** justifies this experiment using the variance of b -bit dynamic fixed-point mapping.

than 10 bits is able to closely match the F1 score of the FP32 baseline, that is indicated by the red line in the figure. Also note that in our experimental setup for the 8-bit dynamic fixed-point format, we use 12-bit input activations to close the F1 score gap with the FP32 baseline. The reason for using higher bit-width input activations is that we observed 8-bit activation dramatically reduces the F1 score. Figure 4 shows the effect of input activation bit-width on the F1 score when the weights are 8-bit integers. Changing the bit-width of the input activation from 8 bits to 12 bits significantly increases the F1 score. Increasing the input activation bit-width beyond 12 bits has a negligible effect on the F1 score, confirming that 12 bits is the minimum required bit-width of the input activations for this application with 8-bit integer weights.

4.2 Theoretical Analysis

Here, we study the effect of varying dynamic fixed-point mapping bit-width on the stochastic gradient descent method. The goal is to show the relation of weight and activation bit-widths on the convergence of integer training. Let us consider the following simplified weight update equation

$$w_{k+1} = w_k + \bar{\eta} \hat{g}(w_k, \xi_k), \quad (2)$$

where $\hat{g}(w_k, \xi_k)$ is the dynamic fixed-point gradient and $\bar{\eta}$ is the learning rate during the fine-tuning phase. Furthermore, we also consider the following common assumptions in sequel.

Assumption 1 (Lipschitz-continuity). The loss function $\mathcal{L}(w)$ is continuously differentiable and its gradients satisfies the following inequality where $L > 0$ is the Lipschitz constant

$$\begin{aligned} \mathcal{L}(w) &\leq \mathcal{L}(\bar{w}) + \nabla \mathcal{L}(\bar{w})^\top (w - \bar{w}) \\ &\quad + \frac{1}{2} L \|w - \bar{w}\|_2^2; \\ &\quad \forall w, \bar{w} \in \mathbb{R}^d. \end{aligned} \quad (3)$$

Assumption 2. (i) $\mathcal{L}(w_k)$ is bounded. (ii) b -bit dynamic fixed-point gradients $\hat{g}(w_k, \xi_k)$ is an unbiased estimator of the true gradients of the loss function $\nabla \mathcal{L}(w_k)^\top \mathbb{E}_{\xi_k} \{\hat{g}(w_k, \xi_k)\} = \|\nabla \mathcal{L}(w_k)\|_2^2 = \|\mathbb{E}_{\xi_k} \{\hat{g}(w_k, \xi_k)\}\|_2^2$, and (iii) with the b -bit dynamic fixed-point gradients i.e. $\hat{g}(w_k, \xi_k)$, there exist scalars $M \geq 0$, $M_V \geq 0$, $M^q \geq 0$ and $M_V^q \geq 0$ such that for all iterations of SGD

$$\begin{aligned} \mathbb{V}_{\xi_k} \{\hat{g}(w_k, \xi_k)\} \\ \leq M + M^q + (M_V + M_V^q) \|\nabla \mathcal{L}(w_k)\|_2^2. \end{aligned}$$

Where M^q and M_V^q denote the added variance of b -bit dynamic fixed-point mapping on the true gradient variance. Also note that in order for **Assumption 2** (i) to hold true, we use stochastic rounding for back propagation.

Suppose **Assumption 1** and **Assumption 2** are true, then inequality (4) follows from (Ghaffari et al., 2022, Remark 2)

$$\begin{aligned} \mathbb{E}_{\xi_k} \{\mathcal{L}(w_{k+1})\} - \mathcal{L}(w_k) \\ \leq -(1 - \frac{1}{2} \bar{\eta} L (M_G + M_G^q)) \bar{\eta} \|\nabla \mathcal{L}(w_k)\|_2^2 \\ \quad + \frac{1}{2} \bar{\eta}^2 L (M + M^q), \end{aligned}$$

$$\text{with } M_G := 1 + M_V \text{ and } M_G^q := 1 + M_V^q, \quad (4)$$

which shows the effect of added variance of fixed point mapping, i.e. M_V^q and M^q , on each step of the optimizer.

Remark 1. In inequality (4), the first term, $-(1 - \frac{1}{2} \bar{\eta} L (M_G + M_G^q)) \bar{\eta} \|\nabla \mathcal{L}(w_k)\|_2^2$ contribute to decreasing the loss \mathcal{L} while the second term, $\frac{1}{2} \bar{\eta}^2 L (M + M^q)$, prevents it. Also note that when M^q and M_G^q are increased, they negatively affect the descent of the loss \mathcal{L} . This means for a good convergence behaviour, representation mapping

variance bounds, i.e. M^q and M_G^q , must be controlled.

Remark 2. For dynamic fixed-point mapping with b -bit integers, the representation mapping variance bounds i.e. M^q and M_G^q , are closely related to the bit-width b . Here, we study these two constants for a linear layer. Let us denote $\hat{\mathbf{A}}$ as the b -bit dynamic fixed-point version of tensor \mathbf{A} and \hat{a}_{ij} as its ij^{th} element. We can relate \hat{a}_{ij} and a_{ij} with an error term δ such as $\hat{a}_{ij} = a_{ij} + \delta_{ij}^{\mathbf{A}}$. For a linear layer $\hat{\mathbf{Y}} = \hat{\mathbf{X}} \hat{\mathbf{W}}$, the computation of the b -bit dynamic fixed-point gradients in the back propagation is

$$\hat{\mathbf{C}} = \frac{\partial \hat{\mathbf{L}}}{\partial \hat{\mathbf{W}}} = \frac{\partial \hat{\mathbf{Y}}}{\partial \hat{\mathbf{W}}} \frac{\partial \hat{\mathbf{L}}}{\partial \hat{\mathbf{Y}}} = \hat{\mathbf{X}}^\top \frac{\partial \hat{\mathbf{L}}}{\partial \hat{\mathbf{Y}}} = \hat{\mathbf{X}}^\top \hat{\mathbf{G}}. \quad (5)$$

It is of interest to find the relation between $\hat{\mathbf{C}} = \hat{\mathbf{X}}^\top \hat{\mathbf{G}}$ in the integer back propagation and the true gradients $\mathbf{C} = \mathbf{X}^\top \mathbf{G}$. We can derive the variance for each element \hat{c}_{ij} by expanding the error terms δ ,

$$\begin{aligned} \mathbb{V}\{\hat{c}_{ij}\} &= \mathbb{V} \left\{ \sum_{n=1}^N \hat{x}_{ni} \hat{g}_{nj} \right\} \\ &= \mathbb{V} \left\{ \sum_{n=1}^N (x_{ni} + \delta_{ni}^{\mathbf{X}}) (g_{nj} + \delta_{nj}^{\mathbf{G}}) \right\} \\ &\leq \mathbb{V} \left\{ \sum_{n=1}^K x_{ni} g_{nj} \right\} \\ &\quad + \sigma_{\mathbf{G}}^2 \mathbb{E}\{\|\mathbf{X}_i^\top\|_2^2\} + \sigma_{\mathbf{X}}^2 \mathbb{E}\{\|\mathbf{G}_{.j}\|_2^2\} \\ &\quad + N \sigma_{\mathbf{X}}^2 \sigma_{\mathbf{G}}^2 \\ &= \mathbb{V}\{c_{ij}\} + \sigma_{\mathbf{G}}^2 \mathbb{E}\{\|\mathbf{X}_i^\top\|_2^2\} \\ &\quad + \sigma_{\mathbf{X}}^2 \mathbb{E}\{\|\mathbf{G}_{.j}\|_2^2\} + N \sigma_{\mathbf{X}}^2 \sigma_{\mathbf{G}}^2. \end{aligned} \quad (6)$$

In inequality (6), $\sigma_{\mathbf{G}}^2 = \max_{i,j} (\mathbb{V}\{\delta_{i,j}^{\mathbf{G}}\})$ and $\sigma_{\mathbf{X}}^2 = \max_{i,j} (\mathbb{V}\{\delta_{i,j}^{\mathbf{X}}\})$. Also note $\|\mathbf{X}_i^\top\|_2^2 = \sum_j x_{ji}^2$ denotes the squared L-2 norm of the i^{th} row of \mathbf{X}^\top and $\|\mathbf{G}_{.j}\|_2^2 = \sum_i g_{ij}^2$ denotes the squared L-2 norm of the j^{th} column of \mathbf{G} . Furthermore, by defining

$$\begin{cases} M^q := \sigma_{\mathbf{G}}^2 (\mathbb{E}\{\|\mathbf{X}_i^\top\|_2^2\} + N \sigma_{\mathbf{X}}^2) \\ M_V^q := \sigma_{\mathbf{X}}^2 \end{cases} \quad (7)$$

Equation (7) shows that M^q depends on variance of dynamic fixed-point mapping for input activations and gradients while M_V^q only depends on b -bit dynamic fixed-point gradients variance.

	QQP	QNLI	MNLI	SST-2	STSB	RTE	MRPC	CoLA	Average
FP32	91.0/88.0	91.1	84.2	92.5	88.3	63.8	82.5/87.8	57.2	82.6
FP16 AMP	90.9/87.9	91.2	84.1	92.4	88.3	64	82.1/87.7	57.5	82.6
16-bit integer	91.0/88.0	91.2	84.2	92.5	88.3	64.5	82.3/87.6	57.7	82.7
12-bit integer	90.9/88.0	91.2	84.0	92.6	87.9	63.5	81.3/87.4	56.7	82.4
10-bit integer	90.8/87.8	91.0	84.0	92.5	87.5	62.7	78.4/85.8	57.6	81.8
8-bit integer	90.1/86.8	90.8	83.7	92.3	87	61.8	76.8/84.7	55.0	80.9

Table 1: Metric performance of integer fine-tuning of BERT on selected GLUE tasks. The reported metric for QQP and MRPC is accuracy and F1 score, for QNLI, MNLI, RTE, and SST-2 is accuracy, for STSB is the Pearson-Spearman correlation, and for CoLA is the Matthews correlation.

Proposition 1. For dynamic fixed-point representation of tensor $\hat{\mathbf{A}}$ with b -bit integers, the variance of error for element i satisfies the following inequality

$$\mathbb{V}\{\delta_i^{\mathbf{A}}\} \leq 2^{2(e_{\text{scale}_{\mathbf{A}}} - b + 2)}. \quad (8)$$

Proof. Using dynamic fixed-point mapping to b -bit integers, the error $\delta_i^{\mathbf{A}}$ satisfies the following bound

$$\begin{aligned} -2^{e_{\text{scale}_{\mathbf{A}}}} \underbrace{(0.000001)_2}_{b-1} \leq \delta_i^{\mathbf{A}} \leq 2^{e_{\text{scale}_{\mathbf{A}}}} \underbrace{(0.000001)_2}_{b-1} \\ -2^{e_{\text{scale}_{\mathbf{A}}} - b + 2} \leq \delta_i^{\mathbf{A}} \leq 2^{e_{\text{scale}_{\mathbf{A}}} - b + 2}. \end{aligned} \quad (9)$$

Thus, the inequality (8) is obtained by using Popoviciu’s inequality on variances

$$\begin{aligned} \mathbb{V}\{\delta_i^{\mathbf{A}}\} &\leq \frac{1}{4} (2^{e_{\text{scale}_{\mathbf{A}}} - b + 2} - (-2^{e_{\text{scale}_{\mathbf{A}}} - b + 2}))^2 \\ &\leq 2^{2(e_{\text{scale}_{\mathbf{A}}} - b + 2)}. \end{aligned} \quad (10)$$

Remark 3. Inequality (8) shows that increasing bit-width b in dynamic fixed-point mapping reduces the variance of the error. This confirms our experimental results on SQuAD v2.0 dataset that for $b > 10$, F1 score can match FP32 baseline, see Figure 3. Also note in equation (7), both M^q and M_V^q depend on b -bit dynamic fixed-point mapping variance of input activation $\sigma_{\mathbf{X}}^2$. Hence, increasing b for input activations while keeping weights in 8-bit format must improve the convergence behaviour. This phenomenon is also confirmed by our experimental results on SQuAD v2.0 dataset demonstrated in Figure 4.

5 Experimental Results

5.1 Experimental Setup

We fine-tuned BERT base on a series of downstream tasks to compare the performance of our integer fine-tuning method with FP16 and FP32 fine-tuning methods. FP16 AMP setup uses NVIDIA’s

	SQuAD v1.1	SQuAD v2
FP32	80.5/88.0	70.6/73.8
FP16 AMP	79.9/87.6	70.6/73.9
16-bit integer	80.7/88.0	70.6/73.9
12-bit integer	79.8/87.6	70.5/73.8
10-bit integer	78.4/86.6	69.8/73.2
8-bit integer	75.6/84.5	65.5/69.2

Table 2: Metric performance of fine-tuning BERT on SQuAD v1.1 and v2.0 datasets. For both datasets the exact match metrics and F1 scores are reported.

automatic mixed precision¹ and the FP32 baseline is the default implementation from Pytorch.

The model is fine-tuned on selected tasks of GLUE benchmark (Wang et al., 2018), along with the Stanford Question Answering Datasets, i.e. SQuAD v1.1 and SQuAD v2.0 (Rajpurkar et al., 2016).

All the fine-tuning setups use the same hyper-parameters and are fine-tuned for the same number of epochs. Each reported metric is the average of five runs with five different random seeds to mitigate the effects of random variation of the results. The fine-tuning experiments are performed based on the fine-tuning scripts of the Hugging Face library (Wolf et al., 2019). For GLUE experiments the fine-tuning is performed for 5 epochs and the learning rate is set to 2×10^{-5} . Also, the per-device fine-tuning batch-size is set to 32. Fine-tuning BERT on SQuAD datasets is done for 2 epochs and the learning rate is 5×10^{-5} and the per-device fine-tuning batch-size is 12. All experiments are run on eight NVIDIA V100 GPUs with 32 gigabytes of VRAM.

¹<https://developer.nvidia.com/automatic-mixed-precision>

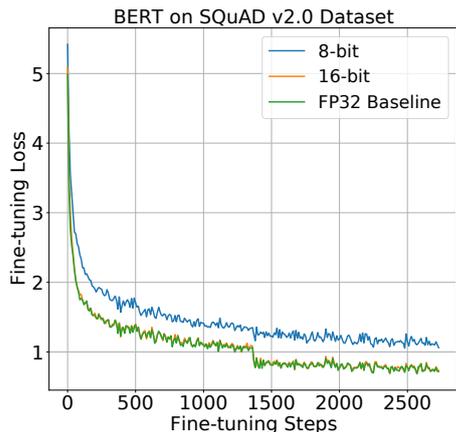


Figure 5: Integer fine-tuning loss trajectory of BERT on SQuAD v2.0 dataset for 2750 iterations.

5.2 Results

The results of fine-tuning BERT base on GLUE benchmark and SQuAD datasets are presented in Table 1 and Table 2 respectively. GLUE benchmark contains a series of downstream tasks, designed to evaluate a diverse set of language understanding abilities of NLP models. SQuAD datasets contain a series of text passages accompanied by a question and the task is to predict the span of the answer in the passage. Using 16-bit integer data type, BERT is able to either match or outperform the FP32 performance for all tasks. The 16-bit integer BERT also shows similar or better performance compared to the FP16 mixed precision fine-tuning method. Further reducing the integer bit-width to 8, fine-tuning BERT exhibits an average of 1.7 point drop on GLUE benchmark and 4.5 point drop for SQuAD datasets. Moreover, our experiments show that using 10-bit and 12-bit integers has average score drops of 0.8 and 0.3 points for GLUE tasks, and 0.8 and 0.2 point for SQuAD datasets respectively.

5.3 Loss Trajectory

Figure 5 shows the loss trajectory of integer fine-tuning BERT on SQuAD v2.0 dataset using 16-bit and 8-bit integers, along with FP32 method. The fine-tuning loss trajectory of BERT using 16-bit integer closely follows the FP32 loss trajectory. On the other hand, when fine-tuning with 8-bit integer parameters and 12-bit integer input activations, the loss trajectory is slightly shifted, but follows the same trend of its FP32 counterpart.

6 Conclusion

We proposed an integer fine-tuning method for transformer based language models using dynamic fixed-point format. We used dynamic fixed-point data type to represent parameters, input activations and gradients in integer values. As a result, our fine-tuning method uses integer arithmetic for the forward and back propagation of compute intensive layers such as linear, layer-norm and embedding layers of BERT model. Furthermore, we studied that increasing the bit-width of the dynamic fixed-point format reduces the variance of the mapping function and thus, improves the convergence of our integer fine-tuning method. We conduct fine-tuning experiments on GLUE benchmark and SQuAD datasets to compare the metric performance of our integer BERT with FP16 mixed precision and FP32 fine-tuning methods. Our experiments show that the 16-bit integer fine-tuning is able to achieve the same metric performance as the FP16 mixed precision fine-tuning method. In addition, fine-tuning BERT with lower bit-width data types, i.e. 8-bit integer, maintains an average drop of metric score within 3.1 points of the FP16 setup.

Limitations

Although our integer fine-tuning method uses integer numbers for compute intensive layers of BERT, integer support for non-linear layers of BERT, e.g. softmax and GELU activation, are left for future work.

We have shown in Figure 1 that the integer data types are faster for the general case. However, a direct comparison of the time and memory cost of our integer fine-tuning method with the FP16 and FP32 methods is left for future works due to lack of access to a proper hardware with integer tensor core support.

Despite the similarities between fine-tuning and pre-training phases, they differ in key aspects of training such as dataset size and number of epochs. The challenges of using integer arithmetic in the pre-training phase will be studied in the future work.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

- Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2020. Binarybert: Pushing the limit of bert quantization. *arXiv preprint arXiv:2012.15701*.
- Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model. *arXiv preprint arXiv:1906.00532*.
- Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. *arXiv preprint arXiv:2109.12948*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Alireza Ghaffari, Marzieh S Tahaei, Mohammadreza Tayaranian, Masoud Asgharian, and Vahid Partovi Nia. 2022. Is integer arithmetic enough for deep learning training? *arXiv preprint arXiv:2207.08822*.
- Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. In *International conference on machine learning*, pages 5506–5518. PMLR.
- Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2017. Mixed precision training. *arXiv preprint arXiv:1710.03740*.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8815–8821.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Darrell Williamson. 1991. Dynamically scaled fixed point arithmetic. In *[1991] IEEE Pacific Rim Conference on Communications, Computers and Signal Processing Conference Proceedings*, pages 315–318. IEEE.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Ali Hadi Zadeh, Isak Edo, Omar Mohamed Awad, and Andreas Moshovos. 2020. Gobo: Quantizing attention-based nlp models for low latency and energy efficient inference. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 811–824. IEEE.
- Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pages 36–39. IEEE.
- Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020a. Ternarybert: Distillation-aware ultra-low bit bert. *arXiv preprint arXiv:2009.12812*.
- Xishan Zhang, Shaoli Liu, Rui Zhang, Chang Liu, Di Huang, Shiyi Zhou, Jiaming Guo, Qi Guo, Zidong Du, Tian Zhi, et al. 2020b. Fixed-point back-propagation training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2330–2338.
- Kang Zhao, Sida Huang, Pan Pan, Yinghan Li, Yingya Zhang, Zhenyu Gu, and Yinghui Xu. 2021. Distribution adaptive int8 quantization for training cnns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3483–3491.
- Feng Zhu, Ruihao Gong, Fengwei Yu, Xianglong Liu, Yanfei Wang, Zhelong Li, Xiuqi Yang, and Junjie Yan. 2020. Towards unified int8 training for convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1979.

A Data Types

In this section we provide a brief overview of various data types mentioned in this work.

Floating-point data type is used to represent decimal fractional numbers. A binary floating-point number has three components of sign (s), mantissa (m), and exponent (e). Using these components, floating-point number x is shown as:

$$x = (-1)^s \times m \times 2^{e-t}$$

where t is the precision and $0 \leq m \leq 2^t - 1$. Another way of representing floating-point numbers is as

$$x = (-1)^s \times 2^e \left(\frac{d_1}{2} + \frac{d_1}{4} + \dots + \frac{d_t}{2^t} \right)$$

where d_i are binary digits of m . For FP32, exponent and mantissa are 8 and 23 bit integer numbers.

Fixed-point is another data type for representing fractional numbers. Unlike floating-point numbers where each mantissa is scaled using its respective exponent, fixed-point uses a single scale factor for all the numbers.

We use the dynamic fixed-point data type in our integer fine-tuning method. Also known as block floating-point, this format uses a different scale for each block of numbers to allow for more flexibility.

Data Augmentation for Radiology Report Simplification

Ziyu Yang
CIS, Temple University
zyyang@temple.edu

Santhosh Cherian
Temple University Hospital
santhosh.cherian@tuhs.temple.edu

Slobodan Vucetic
CIS, Temple University
vucetic@temple.edu

Abstract

This work considers the development of a text simplification model to help patients better understand their radiology reports. This paper proposes a data augmentation approach to address the data scarcity issue caused by the high cost of manual simplification. It prompts a large foundational pre-trained language model to generate simplifications of unlabeled radiology sentences. In addition, it uses paraphrasing of labeled radiology sentences. Experimental results show that the proposed data augmentation approach enables the training of a significantly more accurate simplification model than the baselines.

1 Introduction

Radiology reports are unstructured documents written by radiologists to communicate imaging findings to another physician or a qualified medical professional (Goldberg-Stein and Chernyak, 2019). Radiology reports have been increasingly available to patients through portals (Lourenco and Baird, 2020), which has been generally welcomed by patients (Cooper et al., 2020). However, the health literacy of most patients is insufficient to fully comprehend radiology reports (Lalor et al., 2018) because such reports rely on complex medical jargon and use explanations that imply highly specialized medical knowledge (Delbanco et al., 2012). Several studies even identified adverse effects of sharing radiology reports with patients, including dissatisfaction with care (Rosenkrantz and Flagg, 2015) and undue anxiety and stress (Arora, 2013).

There is an increasing need for patient-friendly radiology reporting that can communicate results clearly and be understandable by a diverse patient population. However, asking a radiologist to supplement a traditional report with a patient-friendly summary would negatively impact their cognitive load and productivity. This problem motivated recent research on the automatic simplification of

health records. The proposed approaches include both lexical simplification that paraphrases text (Chen et al., 2018; Biran et al., 2011; Weng et al., 2018) and semantic simplification that seeks to simplify grammatically complex text (Shardlow, 2014; Leroy et al., 2016) which recently included deep learning approaches (Lewis et al., 2019; Zhang et al., 2020). However, training deep learning models for medical text simplification requires the collection of costly labeled data.

To alleviate the data scarcity issue in simplifying health reports, particularly radiology reports, this paper proposes a novel approach for data augmentation. It augments manually-created labeled data with simplifications generated by a large pre-trained language model such as GPT-3 (Brown et al., 2020). To improve the quality of data augmentation, the approach develops a separate deep learning model that evaluates the quality of generated simplifications. Furthermore, the approach also provides data augmentation through paraphrasing the originally labeled radiology sentences.

The proposed data augmentation approach is experimentally evaluated on a unique corpus of manually generated labeled data for radiology report simplification. The evaluation includes both automatic measures and human evaluation.

Our research claims are: 1) Our augmentation methods enable training of a more accurate model than baselines in solving low-resource radiology sentence simplification problems. 2) We address the challenge of selecting qualified augmentations for radiology sentence simplification. 3) We create unique real data containing expert-annotated simplifications for radiology reports' sentences regarding liver conditions.

2 Related Work

Text Simplification. In text simplification, the output text is a linguistically simplified version of the input text (Adduru et al., 2018). Previous work on

simplification includes lexical and semantic simplification (Alva-Manchego et al., 2020).

Lexical simplification by lexical substitution refers to replacing complex words or phrases with simpler synonyms (Oh et al., 2016; Zeng and Tse, 2006) and has found some practical success (Cook et al., 2017). In the health domain, lexical text simplification often relies on medical dictionaries (UMLS (Bodenreider, 2004), MeSH (Lipscomb, 2000), etc.). Lexical simplification approaches also include rule-based methods (Chen et al., 2018; Biran et al., 2011) and deep learning (Weng et al., 2018, 2019).

Semantic simplifications seek to simplify grammatically complex text by splitting long sentences into shorter ones, changing passive voice to active, resolving ambiguities and anaphora (Shardlow, 2014), splitting complex noun phrases (Leroy et al., 2016), or reducing morphological negations (Mukherjee et al., 2017). Recently, transformer encoder-decoder based pre-trained seq-to-seq models (Lewis et al., 2019; Zhang et al., 2020) were proved to be robust in solving text simplification problems. However, fine-tuning pre-trained models require large quantities of labeled data, which are costly and difficult to obtain in the health domain.

Previous research has explored different methods for text simplification in low-resource domains. To address data scarcity recent studies include unsupervised methods (Surya et al., 2018; Sakakini et al., 2020; Enayati et al., 2021) and reinforcement learning (Laban et al., 2021).

Data Augmentation is a method that automatically generates labeled data to enhance manually labeled data (Liu et al., 2020). One approach is to use paraphrasing to create different variants of the original or simplified sentences (Wei and Zou, 2019). Another approach is to use pre-trained language models to generate labeled data (Bayer et al., 2021). LAMBADA (Anaby-Tavor et al., 2020) augments data for text classification tasks by encoding labels in the input. Similarly, PromptDA (Wang et al., 2022) use language models to augment data for NLU tasks. Back-translation (Edunov et al., 2018) is used to generate different variants of the input text.

There are several public benchmark data sets that are related to our paper. There are paragraph level medical text simplifications (Devaraj et al., 2021) focusing on medical paper abstracts. There is a corpus parsed aligned sentences from Wikipedia

and Simple English Wikipedia¹ (Pattisapu et al., 2020; Van den Bercken et al., 2019) that has been a popular text simplification benchmark. However, none of these data sets have properties similar to the radiology text simplification task.

3 Problem Definition

Let us assume we are given a labeled corpus for text simplification $\mathbf{D}_{Lab} = \{(\mathbf{X}_1, \mathbf{Y}_1), (\mathbf{X}_2, \mathbf{Y}_2), \dots, (\mathbf{X}_n, \mathbf{Y}_n)\}$, where \mathbf{X}_i is the i^{th} original document, \mathbf{Y}_i is its simplification provided by a human expert, and n is the number of labeled documents. Let us also assume we are given an unlabeled corpus of documents $\mathbf{D}_{Unl} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m\}$, where m is the number of unlabeled documents. The objective of data augmentation is to automatically create a synthetic set $\mathbf{D}_{Syn} = \{(\mathbf{X}_1^*, \mathbf{Y}_1^*), (\mathbf{X}_2^*, \mathbf{Y}_2^*), \dots, (\mathbf{X}_K^*, \mathbf{Y}_K^*)\}$, where \mathbf{X}_i^* is one of the original documents from \mathbf{D}_{Lab} or \mathbf{D}_{Unl} or their derivative, and \mathbf{Y}_i^* is its corresponding simplification from \mathbf{D}_{Lab} , its derivative, or an automatically generated simplification. \mathbf{D}_{Syn} is appended to \mathbf{D}_{Lab} and the resulting set $\mathbf{D}_{Aug} = \mathbf{D}_{Lab} \cup \mathbf{D}_{Syn}$ is called the augmented training corpus. The assumption is that a seq2seq model for text simplification trained on the augmented corpus will have higher simplification quality than the one trained on the original training corpus. This paper focuses on the radiology report simplification problem where \mathbf{X}_i is a sentence and \mathbf{Y}_i is its simplification. As a result, our augmentation approach explained in the next two sections is specifically tailored for this application.

4 Methodology

4.1 GPT-based Semantic Augmentation

We propose two types of augmentation. The first, referred to as semantic augmentation, relies on using the large-scale language model GPT-3 (Brown et al., 2020) to generate simplifications automatically. The second, referred to as lexical augmentation, relies on modifying original documents from the labeled corpus. The first augmentation type is described in this section, while the second type is described in the next.

4.1.1 Generating simplifications with GPT-3

It has been demonstrated that large-scale language models such as GPT-3 are capable of doing a wide

¹simple.wikipedia.org

Table 1: Good, Not Simple Enough, and Incorrect examples of radiology liver sentences (**ORI**), expert-written simplifications (**SIM**), and generated simplifications from GPT-Curie (**GPT**).

ORI:	The liver demonstrates diffusely low attenuation, consistent with fatty infiltration
SIM:	There is increased amount of fat in the liver.
GPT:	There is a lot of fat in the liver. (Good)
ORI:	The liver is stable in appearance with again note made of a hypodense lesion in the medial left lobe adjacent to the gallbladder fossa
SIM:	The liver may have one or more tumors or masses.
GPT:	The liver has a small lesion on one side, close to the gallbladder. (Good)
ORI:	The liver shows diffuse low attenuation consistent with fatty infiltration
SIM:	There is increased amount of fat in the liver.
GPT:	The liver has diffuse low attenuation, indicative of fatty infiltration. (Not Simple Enough)
ORI:	The liver is unremarkable without intra- or extra-hepatic biliary dilatation
SIM:	The liver appears normal.
GPT:	The liver is not enlarged or tender without intra- or extra-hepatic biliary dilatation. (Not Simple Enough)
ORI:	Complex hepatic abscess predominantly occupying segments VI and VII of the liver is overall little changed in size compared to date, measuring approximately 7.9 (AP) x 8.7 (TRV) x 11.4 (CC) cm
SIM:	There is infection and pus seen within the liver.
GPT:	This means that the abscess is smaller than it was before, by about 3 inches on all sides. (Incorrect)
ORI:	The liver demonstrates prominent intrahepatic biliary ducts
SIM:	The liver ducts that carry digestive enzymes may be blocked.
GPT:	The liver has many bile ducts. (Incorrect)

variety of NLP tasks through prompting (Brown et al., 2020). Our preliminary experiments showed that the two largest GPT-3 models, Curie and Davinci, are surprisingly good at generating simplifications of radiology report sentences, despite never being trained explicitly for that task. Table 1 shows several representative examples of simplifications by the Curie model prompted by *My patient asks me to simplify this radiology sentence "<original sentence>":*. Each example shown contains an original radiology sentence, its simplification by GPT-3, and its simplification provided by collaboration between a radiologist and a layperson. The first two examples show that GPT-3 can provide

factually correct and easy to understand simplifications comparable to the manually created simplifications. Next two examples show that GPT-3 might provide factually correct text that is not sufficiently simple. The final two examples show that GPT-3 might provide factually incorrect simplifications.

Our semantic augmentation approach uses GPT-3 to generate simplifications of unlabeled documents from \mathbf{D}_{Unl} and add them to the augmented corpus \mathbf{D}_{Aug} . As noted in previous research (Liu et al., 2021) the choice of prompting can have a significant impact on the quality of the generated text and accuracy on a particular task.

Our prompting approach relies on the in-context learning that has been used with success with GPT-3 models. Instead of relying on costly fine-tuning of a language model, it pastes a few labeled examples into the prompt and asks the language model to generate label of an unlabeled example. In our specific application, we select K labeled examples (\mathbf{X}, \mathbf{Y}) from \mathbf{D}_{Lab} and insert each of them into template '*Sentence: < X >; Simplification: < Y >*'. A triple pound sign, *###*, is used to separate templates for the K labeled examples. The prompt ends with '*Sentence: < X >; Simplification:*', where \mathbf{X} is an unlabeled document from \mathbf{D}_{Unl} . GPT-3 model is expected to write a simplification by mimicking the style of the labeled examples from the prompt.

As noted in previous work (Brown et al., 2020) the success of prompting that uses in-context learning depends on the particular choice of K examples. Therefore, we select most related sentence simplification pairs from the training set \mathbf{D}_{Lab} given any unlabeled document from \mathbf{D}_{Unl} . In detail, we use BERTScore (Zhang et al., 2019), which leverages the pre-trained contextual embeddings from BERT (Devlin et al., 2018) and matches words in unlabeled and labeled radiology sentences by cosine similarity. Thus, each prompt consists of K most related examples rated by BERTScore for an unlabeled sentence that is appended to the end. Moreover, we evaluate more example selection scenarios in our ablation study.

4.1.2 BERT-Checker

Language models such as GPT-3 provide token probabilities as their output. When generating text, one option is to use brute force and generate the most likely token. However, in the context of text simplification, the most likely tokens are not guaranteed to produce the best simplification. An alter-

native is to generate tokens by selecting among the most likely choices, which the temperature hyperparameter in GPT-3 can control. In our approach, we invoke a GPT-3 model N times for each prompt using a temperature higher than zero, which results in N different simplifications. Then, we automatically select the best one of the N generated simplifications and add it to the augmented corpus.

As seen in Table 1, some of the generated simplifications are good while others are not. Separating good from inadequate simplifications is a non-trivial challenge. Related work on automatic evaluation of the generated text includes GPT-3-ENS (Chintagunta et al., 2021), which measures the complexity of terms in simplifications, and GPT3Mix (Yoo et al., 2021), which treats the likelihood scores of generated labels as confidence scores. However, we found that the existing approaches are inappropriate for our application. Thus, we developed a novel approach called BERT-Checker.

BERT-Checker is a fine-tuned BERT model (Devlin et al., 2018) to a task similar to entailment. In particular, we convert our labeled corpus into training data matching the format of the entailment task. We add label 1 to each example from \mathbf{D}_{Lab} to create positive examples in new training data set, $\mathbf{D}'_{Lab} = \{[(\mathbf{X}_i, \mathbf{Y}_i), 1]\}$. To create negative examples in \mathbf{D}'_{Lab} , we use four different strategies as outlined next:

- **Precision:** To ensure that simplification is closely related to the original text, we corrupt the original text \mathbf{X} by replacing the medical terms with randomly selected medical terms, and generate negative example from labeled example (\mathbf{X}, \mathbf{Y}) as $[(\text{corrupt}(\mathbf{X}), \mathbf{Y}), 0]$.
- **Simplicity:** To penalize simplifications that are too similar to the original sentence, we create negative examples by using the original text as simplification, $[(\mathbf{X}, \mathbf{X}), 0]$.
- **Correctness:** To penalize incorrect simplifications, we randomly select two labeled examples $(\mathbf{X}_1, \mathbf{Y}_1)$ and $(\mathbf{X}_2, \mathbf{Y}_2)$ and create a negative example by mixing the original and simplified text, $[(\mathbf{X}_1, \mathbf{Y}_2), 0]$.
- **Robustness:** For labeled example (\mathbf{X}, \mathbf{Y}) we replace the simplification with an empty string or a sentence generated by a GPT-3 given the prompt 'Generate a radiology report sentence about liver' and high temperature of 0.8 to create negative example $[(\mathbf{X}, \text{GPT}()), 0]$.

Thus, for each positive example, we generate four negative examples. As a result, we can obtain a negative dataset \mathbf{D}'_{Neg} . We fine-tune Clinical BERT (Alsentzer et al., 2019) on the text entailment task using the generated data set.

4.2 Dictionary-based Lexical Augmentation

We propose lexical augmentation to supplement semantic augmentation described in the previous section. Lexical simplification refers to replacing complex terms in original documents \mathbf{X} with their synonyms, which might also be complex. In the related work on text simplification of general-purpose text, EDA approach (Wei and Zou, 2019) paraphrases original documents by replacing randomly selected words or phrases with their synonyms in WordNet (Miller, 1995). We modify EDA by replacing only specialized medical terms.

Inspired by (Pattisapu et al., 2020; Hasan et al., 2016), we use medical dictionaries Medical Subject Headings (MeSH) (Lipscomb, 2000) and Unified Medical Language System (UMLS) (Bodenreider, 2004) to find the synonyms. We use pre-trained named entity recognition model (Honnibal and Montani, 2017) to extract medical terms from the original documents in labeled corpus \mathbf{D}_{Lab} . The medical terms are linked to Concept Unique Identifier (CUI) in UMLS and the concept_id in MeSH. Each medical code in UMLS and MeSH is mapped to a list of synonyms. We iteratively select a synonym to replace the medical term from the original document.

We illustrate the lexical simplification process in Fig 1, where *hepatic steatosis* in the sentence 'Probable diffuse hepatic steatosis' is recognized as a medical term and replaced with its synonyms. In particular, CUI codes 'C0015695' and 'C2711227' are found to match *hepatic steatosis*, where the canonical names are *Fatty Liver* and *Steatohepatitis*. Similarly, 'D005234' from MeSH also provides several synonyms. This process identifies five synonyms used to create five different versions of the original document.

Once the synonyms for a medical term in original document \mathbf{X} of labeled example (\mathbf{X}, \mathbf{Y}) are identified, we paraphrase the original document as $\text{lexical}(\mathbf{X})$ and generate an augmented example $(\text{lexical}(\mathbf{X}), \mathbf{Y})$. The new example is added to the augmented corpus \mathbf{D}_{Aug} .

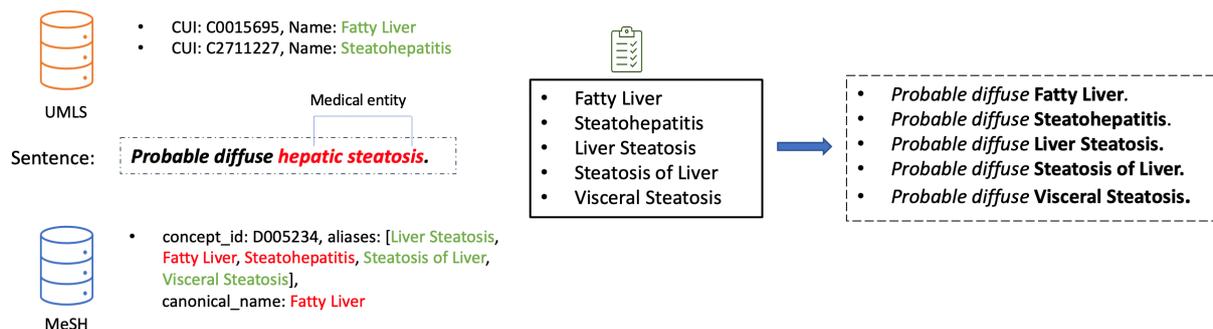


Figure 1: Workflow for lexical augmentation. It shows the linked synonyms of the entity "hepatic steatosis" from UMLS/MeSH and five synthetic sentences.

5 Experiments

5.1 Data

To the best of our knowledge, there is no readily available corpus for simplifying radiology sentences. To experimentally evaluate our data augmentation approach, we created a new corpus for this purpose. In particular, we collected 540 sentences from radiology reports describing the liver condition and manually created their simplifications: 170 sentences were obtained from CT-Abdomen radiology reports from a university hospital (UH), and the remaining 370 were extracted from CT-Abdomen radiology reports from publicly available MIMIC-III (Johnson et al., 2016) data. All sentences were de-identified with Health Insurance Portability and Accountability Act (HIPAA) standards in order to facilitate public accesses and human annotations.

We asked a radiologist to provide a simplification for each selected sentence. A layperson joined the radiologist to provide feedback about the generated simplifications. If the layperson thought the simplification was too complicated, this was communicated to the radiologist, who proceeded to improve the simplification. The process was repeated until the layperson could understand all the simplification and could correctly guess the severity of the described conditions.

During this sequence simplification process, the radiologist and the layperson agreed that it is sufficient to use simplification 'The liver looks normal' for sentences explaining that nothing concerning was observed about the liver. 39% of the the university hospital sentences and 21% of the MIMIC-III sentences were simplified as 'The liver looks normal'. For simplification of sentences that described concerning findings was to ignore technical details that might be confusing to patients. Any relevant

medical terms were stated in simple terms familiar to laypeople. If possible, grammar was kept simple, and the sentences were kept short. Table 1 shows several examples of the original sentences (ORI) and their manual simplifications (SIM).

For our experiments, we randomly selected 100 sentences and their simplifications for training and the remaining 70 for testing for both the university hospital and MIMIC-III labeled data. Thus, we had 200 labeled examples for training denoted as D_{Lab} , and 140 for testing. We used the remaining 200 MIMIC-III sentences as the unlabeled corpus D_{Unl} and used their simplifications to better evaluate the data augmentation approaches.

The corpus is available to the research community to support further research on medical text simplification.²

5.2 Data Augmentation

To implement the proposed semantic augmentation approach, we used GPT-3 Curie model (6.7B parameters) with the few-shot learning prompt described in Section 4.1 with $K = 5$ to automatically generate simplifications for each unlabeled sentence in D_{Unl} . We used the API provided by OpenAI³. We generated $N = 5$ simplifications for each liver sentence with $temperature = 0.5$, which was selected to provide a good balance between factual correctness and diversity.

We trained BERT-Checker to select the best among the $N = 5$ generated simplifications for each liver sentence. BERT-checker was fine-tuned using 80% of the training data as positives and four copies of negatives for each positive, as explained in Section 4.2. BERT-Checker was a fine-tuned BERT base model (110M parameters) consisting of

²<https://github.com/Ziyu-Yang/Radiology-Text-Simplification-Liver>

³<https://openai.com/api/>

12 transformer encoder layers. A fully connected linear layer was added to BERT on its [CLS] output to score the simplification quality. The binary cross-entropy loss was used. 20% of the training data was used for validation and early stopping. We fine-tuned for up to 20 epochs with the patience for early stopping of 3, batch size 16, and learning rate $1e-4$. All experiments were implemented with a single GTX 1080Ti.

The accuracy of trained BERT-Checker on validation data was 0.924. Its precision (the fraction of true positives among positive predictions) was 0.899 and its recall (the fraction of positives that were predicted correctly) was 0.958. We consider it to be high enough accuracy for BERT-Checker to be used to determine the quality of simplifications produced by GPT-3.

In the lexical augmentation, we annotated the recognized entities in the liver sentences from the labeled corpus with Type Unique Identifier (TUI)⁴. TUI is the code to represent hierarchical semantic types of all medical concepts in UMLS and MeSH. Specifically, we only paraphrased terms that belong to "T023 | *Body Part, Organ, or Organ Component*" or "T033 | *Finding*" groups. Because many medical concepts have only one synonym, many sentences mentioned only a single body part other than the liver, and a single finding, we finally obtained 242 unique lexical augmentations from D_{Uml} . In order to control the effect of augmentation size, we randomly selected 200 of them for further experiments.

5.3 BART model

BART (Lewis et al., 2019) is a pre-trained model that uses a seq2seq architecture with a bidirectional encoder and a left-to-right decoder. It achieves state-of-the-art performance on many seq2seq benchmarks. We fine-tuned a BART base model (406M parameters) on different mixes of 450 labeled and augmented data to create different radiology simplification models. The fine-tuning was implemented using PyTorch-lightning⁵. 20% of the training data was used for validation and early stopping. We used the cross entropy loss. We used the same training setting as for BERT-Checker.

⁴<https://lhncbc.nlm.nih.gov/semanticnetwork/index.html>

⁵<https://www.pytorchlightning.ai/>

5.4 Baselines

We first introduce two model baselines that do not use augmentations. Then we introduce two baseline augmentation methods that are appropriate to our task.

5.4.1 Model Baselines

The first baseline is BART base model fine-tuned with the labeled data (**BART-base**). As the second baseline, we used simplifications by the same implementation of GPT-Curie model that is used to augment the labeled data. Specifically, we selected the most related $K = 5$ sentences from the labeled set to a test sentence as the few-shot prompt, generated $N = 5$ simplifications and used BERT-Checker to select the best one. We name this baseline **GPT-FS**.

5.4.2 Augmentation Baselines

We implemented and evaluated two widely used baseline data augmentation methods: 1) Easy Data Augmentation (**EDA**) (Wei and Zou, 2019), a rule-based augmentation that includes synonym replacement, random insertion, random swap, and random deletion. We reproduced this baseline with its source code⁶. 2) Back translation (**BT**), that uses a pre-trained machine translation model to translate sentences into another language and then translate them back to English. The back-translated English sentences are fused with the corresponding simplifications to provide augmented data. Following previous work (Brown et al., 2020), we used GPT-3 Curie to back translate the original sentences to French and back to English. French was selected because it provided a good balance between factual correctness and diversity of generated back-translations.

We generated 200 augmented examples for each baseline approach.

6 Evaluation Methods

6.1 Automated Evaluation

We used multiple automated metrics to evaluate text simplification accuracy. **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) is a set of metrics used for seq2seq tasks. It calculates the overlapping of unigrams, bigrams, and the longest common subsequences between the expert-provided and machine-generated simplifications. Similarly, **BLEU** (bilingual evalua-

⁶https://github.com/jasonwei20/eda_nlp

Table 2: Comparison of augmentation methods.

	# Aug	ROUGE 1/2/L	BLEU	SARI	BERTScore	FKGL↓
Baseline Models						
GPT-FS	0	56.00/42.20/54.64	0.2363	0.5455	0.9457	5.392
BART-base	0	59.81/50.34/58.87	0.4240	0.5324	0.9411	5.560
Augmentation Methods						
EDA	200	60.90/51.90/60.06	0.4461	0.5460	0.9429	5.315
BT	200	63.47/53.50/62.33	0.4504	0.5740	0.9470	5.133
HUMAN	200	71.06/62.89/70.20	0.5322	0.6047	0.9566	4.870
LEX	200	68.58/60.84/68.24	0.5391	0.5769	0.9559	5.353
SEM	200	66.11/56.55/64.76	0.4709	0.5875	0.9510	5.629
AUG-SUB	200	67.92/58.81/67.04	0.5020	0.5960	0.9524	5.314
AUG	400	69.03/60.37/68.51	0.5036	0.6029	0.9550	5.021

tion understudy) (Papineni et al., 2002) also evaluates overlap of n-grams between the simplifications. Unlike ROUGE and BLEU, **BERTScore** (Zhang et al., 2019) computes a contextual similarity score between tokens in the simplifications. **SARI** (Xu et al., 2016) is a gold standard edit-based metric for text simplification evaluation. Unlike other metrics, it compares the machine-generated simplification with respect to both the original sentence and the human-provided simplification. To evaluate simplicity, we used **FKGL** (Flesch Kincaid Grade Level) (Kincaid et al., 1975), which is a widely used readability formula that assesses the approximate reading grade level of a text. The lower score indicates simpler texts.

6.2 Human Evaluation

Applying automatic evaluation metrics is insufficient to compare quality of simplifications by different methods. Therefore, we also used human evaluation. We asked a medical doctor (family physician) that was distinct from the radiologist who provided the simplifications to evaluate the machine-generated simplifications. We asked the evaluator to use 1-5 Likert scale to evaluate the following four aspects of each simplification, the first three being consistent with.

Factuality refers to medical correctness of the simplification. Score one means that the simplification is factually incorrect and five that it is correct. Scores between one and five mean that some information is imprecise, missing, or hallucinated. Lower scores mean there are more serious factual errors. **Fluency** measures the quality of grammar and readability, regardless of factual correctness. If a simplification is both easy to read and grammati-

cally correct it gets a score of five. This measure is consistent with the fluency measure explained in (Nisioi et al., 2017). **Simplicity** evaluates whether the evaluator thought the laypeople would be able to understand the simplification, regardless of factual correctness. Score of five means that the evaluator thought that any patient would be able to completely understand the simplification.

During the initial stages of human evaluation of factuality and simplicity, we observed that the evaluator occasionally preferred machine-generated simplifications to the radiologist-provided ones. That is why we introduced **Consistency**, which measures how closely the simplification matches the radiologist-provided simplification. Score of five means that the simplification is almost identical to the radiologist-provided simplification. We note that Consistency is related to SARI automatic measure (Xu et al., 2016).

7 Results

7.1 Quantitative Results

We fine-tuned BART model including augmented data from baseline methods (EDA, BT), and our lexical and semantic augmented data (LEX, SEM). BART-base and GPT-FS were created according the description in Section 5.4.1. First two rows of Table 2 refer to fine-tuned BART and few-shot prompted GPT-3 Curie using only the radiologist-provided labeled data. The remaining rows refer to inclusion of augmented data to BART tuning. Rows EDA and BT refer to the baseline augmentation methods. Row HUMAN refers to the augmentation provided by the radiologist, and serves to establish the upper bound on accuracy improve-

ment due to augmentation. LEX and SEM rows represent our lexical and semantic augmentation methods. AUG-SUB and AUG use 200 and 400 combined semantic and lexical augmentations, respectively. Aug column shows the number of augmented examples.

We observe that our proposed augmentations are superior to baselines LEX and SEM on almost all metrics. SEM is better than LEX on SARI measure. AUG is better than LEX and SEM on ROUGE, SARI and FKGL. AUG is the best overall augmentation method coming very close to the HUMAN upper bound, after noting that SARI and FKGL are the most useful measures for evaluation of simplicity. We note that GPT-FS has lower overall scores than any of the BART models.

Table 3: Human evaluation results (Factuality, Fluency, Simplicity, and Consistency) on 60 selected testing data.

Method	Factual	Fluency	Simp	Cons
BART-base	3.38	4.85	4.67	3.18
BT	3.22	4.88	4.58	3.13
GPT-FS	4.18	5.00	4.55	3.91
AUG	4.22	4.95	4.62	4.10

7.2 Human Evaluation Results

For Table 3, we asked a medical doctor to evaluate 60 randomly selected simplifications from the test data (30 from each source). We evaluated the most relevant four models from Table 2: BART-base, BT, GPT-FS, and AUG. The results show that all methods have comparable Simplicity and Fluency. AUG and GPT-FS have better Factuality and Consistency than BART-base and BT. AUG is slightly better than GPT-FS on those two important measures, indicating that fine-tuning BART with augmentation produced by few-shot prompted GPT-3 Curie is better than directly using few-shot prompted GPT-3 Curie for simplification.

Table 4: Comparison between different versions of semantic augmentations. # Aug is the number of augmented examples. ROUGE refers to ROUGE-L.

Method	# Aug	ROUGE	BLEU	SARI
First-run	200	60.21	0.4053	0.5590
Similarity	200	55.31	0.3547	0.5387
Five-runs	781	55.77	0.3755	0.5391
SEM	200	64.76	0.4709	0.5875

7.3 Ablation Study

We first evaluated the ability of BERT-Checker to recognize high-quality simplifications. We compared the version we implemented in our experiments (SEM row in Table 4) with three different variants: 'First-run' always selects the first generated simplification, 'Similarity' selects the best simplification based on BERTScore, 'Five-runs' uses all simplifications generated by GPT-3 Curie as augmentations. After removing duplicates, there are 781 augmentations produced by 'Five-runs'. Table 4 shows all three variants are inferior to SEM, showing that any of the ablations would significantly deteriorate the results. The results confirm that the quality of augmentations is critical for success of data augmentation approaches.

Next, we evaluated the importance of GPT-3 prompting. As noted in previous research (Liu et al., 2020), the choice of prompting can significantly impact the quality of the generated text. Thus, we designed an ablation study to compare different prompting approaches for data augmentation.

Table 5: Comparison of different prompting on data augmentation.

Prompts	ROUGE	BLEU	SARI
BART-grader	46.62	0.2862	0.4917
BART-patient	55.81	0.3516	0.5255
BART-top1	58.65	0.3955	0.5511
BART-rd5	53.94	0.3170	0.5360
SEM	64.76	0.4709	0.5875

In our prompt design that has the following form: *Sentence*: $\langle X \rangle$; *Simplification*: $\langle Y \rangle$, we included $K = 5$ most related labeled examples to the original test sentence in the prompt. We first explored whether the number of few-shot examples matters. We repeated the data augmentation process with $K = 1$ (BART-top1 in the table). Table 5 shows that $K = 5$ resulted in better performance than $K = 1$. Next, we evaluated whether the way we select examples matters. Instead of $K = 5$ closest labeled examples, we selected $K = 5$ random labeled examples (BART-random in the table). From Table 5, we can see that random labeled examples resulted in lower accuracy.

We also explored prompting that does not rely on few-shot learning. One design was explained in section 4.1.1, 'My patient asks me to simplify this radiology sentence $\langle X \rangle$ ', we refer to as BART-

patient in the table. Similarly, inspired by a GPT-3 prompt for the summarization task, we used prompt: *My second grader student asks me to simplify the following sentence: <X>*, we refer to as BART-grader in the table. These two prompts are the so-called 'zero-shot' prompts. As shown in Table 5, the 'grader' and 'patient' prompts result in inferior accuracy compared to the few-shot prompting.

8 Conclusion

This paper proposes two novel augmentation methods to enhance the limited labeled data for the radiology sentence simplification problem. Our evaluation using automatic measures and human evaluation shows that data augmentation can substantially improve the quality of simplification models. The ablation results show that the proposed innovations in automatic creation of simplifications for data augmentation are very effective.

9 Limitations

The main limitation of our study is that we only considered simplification of radiology sentences. In future work, it will be important to expand the approach to simplify whole paragraphs, because very often radiologists use multiple sentences to discuss a single observation. Simplifying single sentences can thus be suboptimal because important context from the previous and subsequent sentences might be lost. The second limitation of the study is that our corpus only included sentences related to liver. It will be important in the future work to evaluate the proposed approach on a wider variety of radiology sentences. The third limitation is that we obtained simplifications from a single radiologist. It will be important for future study to include simplifications from multiple radiologists to ensure generalizability of the proposed approach. The fourth limitation is that we used a single medical doctor to evaluate the quality of the simplifications. It would be important in future studies to ask multiple medical doctors to evaluate the quality, which would allow estimating the inter-rater variability. The fifth limitation is that we did not use laypeople to evaluate the quality of simplification. This would require some innovation in the human evaluation process because laypeople are not able to evaluate factual correctness and because it would be important to understand how simplifications improve the overall understanding of the radiology reports.

The final limitation is a relatively small size of the labeled data set created for this study. Obtaining high-quality simplifications is very costly because it requires collaboration between radiologists and laypeople.

References

- Viraj Adduru, Sadid A Hasan, Joey Liu, Yuan Ling, Vivek V Datla, Ashequl Qadir, and Oladimeji Farri. 2018. Towards dataset creation and establishing baselines for sentence-level neural clinical paraphrase generation and simplification. In *KHD@IJCAI*.
- Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.
- Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.
- Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.
- Neeraj K Arora. 2013. Patient engagement in a rapidly changing communication environment: reflections of a cancer survivor. *Journal of the National Cancer Institute Monographs*, 2013(47):231–232.
- Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2021. A survey on data augmentation for text classification. *ACM Computing Surveys*.
- Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501.
- Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Jinying Chen, Emily Druhl, Balaji Polepalli Ramesh, Thomas K Houston, Cynthia A Brandt, Donna M Zulman, Varsha G Vimalananda, Samir Malkani, and Hong Yu. 2018. A natural language processing system that links medical terms in electronic health

- record notes to lay definitions: system development using physician reviews. *Journal of medical Internet research*, 20(1):e26.
- Bharath Chintagunta, Namit Katariya, Xavier Amatriain, and Anitha Kannan. 2021. Medically aware gpt-3 as a data generator for medical dialogue summarization. In *Proceedings of the Second Workshop on Natural Language Processing for Medical Conversations*, pages 66–76.
- Tessa S Cook, Seong Cheol Oh, and Charles E Kahn Jr. 2017. Patients’ use and evaluation of an online system to annotate radiology reports with lay language definitions. *Academic radiology*, 24(9):1169–1174.
- Kendall Cooper, Marta E Heilbrun, Shenise Gilyard, Brianna L Vey, and Nadja Kadom. 2020. Shared decision making: Radiology’s role and opportunities. *American Journal of Roentgenology*, 214(1):W62–W66.
- Tom Delbanco, Jan Walker, Sigall K Bell, Jonathan D Darer, Joann G Elmore, Nadine Farag, Henry J Feldman, Roanne Mejilla, Long Ngo, James D Ralston, et al. 2012. Inviting patients to read their doctors’ notes: a quasi-experimental study and a look ahead. *Annals of internal medicine*, 157(7):461–470.
- Ashwin Devaraj, Byron C Wallace, Iain J Marshall, and Junyi Jessy Li. 2021. Paragraph-level simplification of medical texts. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2021, page 4972. NIH Public Access.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.
- Saman Enayati, Ziyu Yang, Benjamin Lu, and Slobodan Vucetic. 2021. A visualization approach for rapid labeling of clinical notes for smoking status extraction. In *Proceedings of the Second Workshop on Data Science with Human in the Loop: Language Advances*, pages 24–30, Online. Association for Computational Linguistics.
- Shlomit Goldberg-Stein and Victoria Chernyak. 2019. Adding value in radiology reporting. *Journal of the American College of Radiology*, 16(9):1292–1298.
- Sadid A Hasan, Bo Liu, Joey Liu, Ashequl Qadir, Kathy Lee, Vivek Datla, Aaditya Prakash, and Oladimeji Farri. 2016. Neural clinical paraphrase generation with attention. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 42–53.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1):1–9.
- J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.
- Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. *arXiv preprint arXiv:2107.03444*.
- John P Lalor, Hao Wu, Li Chen, Kathleen M Mazor, and Hong Yu. 2018. Comprehenotes, an instrument to assess patient reading comprehension of electronic health record notes: development and validation. *Journal of medical Internet research*, 20(4):e9380.
- Gondy Leroy, David Kauchak, and Alan Hogue. 2016. Effects on text simplification: Evaluation of splitting up noun phrases. *Journal of health communication*, 21(sup1):18–26.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Carolyn E Lipscomb. 2000. Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88(3):265.
- Pei Liu, Xuemin Wang, Chao Xiang, and Weiye Meng. 2020. A survey of text data augmentation. In *2020 International Conference on Computer Communication and Network Security (CCNS)*, pages 191–195. IEEE.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.
- Ana P Lourenco and Grayson L Baird. 2020. Optimizing radiology reports for patients and referring physicians: mitigating the curse of knowledge. *Academic radiology*, 27(3):436–439.

- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Partha Mukherjee, Gondy Leroy, David Kauchak, Srinidhi Rajanarayanan, Damian Y Romero Diaz, Nicole P Yuan, T Gail Pritchard, and Sonia Colina. 2017. Negait: A new parser for medical text simplification using morphological, sentential and double negation. *Journal of biomedical informatics*, 69:55–62.
- Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 85–91.
- Seong Cheol Oh, Tessa S Cook, and Charles E Kahn. 2016. Porter: a prototype system for patient-oriented radiology reporting. *Journal of digital imaging*, 29(4):450–454.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Nikhil Pattisapu, Nishant Prabhu, Smriti Bhati, and Vasudeva Varma. 2020. Leveraging social media for medical text simplification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 851–860.
- Andrew B Rosenkrantz and Eric R Flagg. 2015. Survey-based assessment of patients’ understanding of their own imaging examinations. *Journal of the American College of Radiology*, 12(6):549–555.
- Tarek Sakakini, Jong Yoon Lee, Aditya Duri, Renato FL Azevedo, Victor Sadauskas, Kuangxiao Gu, Suma Bhat, Dan Morrow, James Graumlich, Saqib Walayat, et al. 2020. Context-aware automatic text simplification of health materials in low-resource domains. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 115–126.
- Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.
- Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2018. Unsupervised neural text simplification. *arXiv preprint arXiv:1810.07931*.
- Laurens Van den Bercken, Robert-Jan Sips, and Christoph Lofi. 2019. Evaluating neural text simplification in the medical domain. In *The World Wide Web Conference*, pages 3286–3292.
- Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. Promda: Prompt-based data augmentation for low-resource nlu tasks. *arXiv preprint arXiv:2202.12499*.
- Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China. Association for Computational Linguistics.
- Jun-Cheng Weng, Yu-Syuan Chou, Guo-Joe Huang, Yeu-Sheng Tyan, and Ming-Chou Ho. 2018. Mapping brain functional alterations in betel-quid chewers using resting-state fmri and network analysis. *Psychopharmacology*, 235(4):1257–1271.
- Wei-Hung Weng, Yu-An Chung, and Peter Szolovits. 2019. Unsupervised clinical language translation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3121–3131.
- Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.
- Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.
- Qing T Zeng and Tony Tse. 2006. Exploring and developing consumer health vocabularies. *Journal of the American Medical Informatics Association*, 13(1):24–29.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Embedding Recycling for Language Models

Jon Saad-Falcon¹ Amanpreet Singh¹ Luca Soldaini¹
Mike D’Arcy² Arman Cohan^{1,3} Doug Downey^{1,2}

¹Allen Institute for Artificial Intelligence (AI2)

²Northwestern University

³Yale University

{jons, amanpreets, lucas, armanc, dougd}@allenai.org,
m.m.darcy@u.northwestern.edu

Abstract

Real-world applications of neural language models often involve running many different models over the same corpus. The resulting high computational cost has led to interest in techniques that can reuse the contextualized embeddings produced in previous runs to speed training and inference of future ones. We refer to this approach as *embedding recycling* (ER). While multiple ER techniques have been proposed, their practical effectiveness is still unknown because existing evaluations consider very few models and do not adequately account for overhead costs. We perform an extensive evaluation of ER across eight different models (17 to 900 million parameters) and fourteen tasks in English. We show how a simple ER technique that caches activations from an intermediate layer of a pretrained model, and learns task-specific adapters on the later layers, is broadly effective. For the best-performing baseline in our experiments (DeBERTa-v2 XL), adding a precomputed cache results in a >90% speedup during training and 87-91% speedup for inference, with negligible impact on accuracy. Our analysis reveals important areas of future work, and we release code and documentation for our experiments at <https://github.com/allenai/embeddingrecycling>.

1 Introduction

Large pretrained language models form the foundation of modern NLP, and continue to push the state-of-the-art on a wide range of natural language processing tasks (Devlin et al., 2019; Liu et al., 2019b; Bommasani et al., 2021). Larger models tend to offer superior accuracies (Kaplan et al., 2020), but also entail higher computational costs. The steep computational cost associated with large neural language models slows down experimentation, increases financial barriers to the technology, and contributes to global climate change (Strubell et al., 2019; Dodge et al., 2022).

Our work studies how to reduce computational cost for workloads in which many distinct models are run over the same text. For example, a scholarly search tool that helps users find and understand relevant literature may run separate models for entity recognition, topic classification, relation extraction, summarization, question answering, and so on over a large corpus of papers. New and improved models for the tasks are developed frequently, necessitating additional runs. The need for repeated model runs has also been noted for other applications in previous work, including news applications (Du et al., 2020) and virtual assistants (Wei et al., 2022). Further, repeated runs also occur very frequently during model development, when exploring model variants or executing multiple training epochs.

Recent work has introduced ways to reduce computational cost in such settings by re-using model activations from one task to speed up other ones (Du et al., 2020; Wei et al., 2022). A pretrained language model’s internal activations form a contextualized embedding, which reflects syntactic and semantic knowledge about the input text (Goldberg, 2019; Wiedemann et al., 2019; Rogers et al., 2020) which can be useful across a variety of downstream tasks. We define *embedding recycling* (ER) as the technique of caching certain activations from a previous model run, and re-using them to improve the efficiency of future training and inference. Recycling imposes a small computation time cost the first time a model processes a text, in order to compute and populate the cache. Thereafter, all subsequent runs on the text can use the precomputed cache, improving efficiency.

While previous work has shown the promise of ER approaches, the existing evaluations are limited. For example, Du et al. (2020) and Wei et al. (2022) each evaluate ER for only one or two base models. Likewise, for ER techniques that cache activations on persistent storage, the storage and time cost of the cache itself has yet to be quan-

tified. In this paper, we present a more comprehensive evaluation of ER with several models and tasks, along with a thorough efficiency analysis. We study a simple *layer-recycling* ER method that caches the activations from an intermediate layer of a pretrained model, and uses those cached activations as the starting point when the same input sequence is seen again during fine-tuning or inference. We show that even this simple method yields substantial improvements to throughput at small or no cost to accuracy on average. Our results provide the strongest evidence to date that ER can be a practically important technique for reducing costs for NLP systems, but as we discuss in [section 6](#), they also suggest important challenges that must be addressed in future work.

Our contributions are summarized below:

- We propose embedding recycling as a method for lowering the computational costs of training and inference for language models, and explore layer recycling with two techniques: standard fine-tuning and parameter-efficient adapters.
- Our experiments with eight models across a wide range of tasks show that layer recycling is generally effective. For the best-performing ER model on our tasks- DeBERTa-XL with adapters, we find that layer recycling nearly matches performance of the original model while providing a 87-91% speedup at inference time, and greater than 90% speedup at training time.
- We explore open challenges for embedding recycling and present questions for future work.

2 Related Work

The embedding recycling technique we investigate is based on findings from prior work suggesting that not all layers of a pretrained transformer are equally important for end-task finetuning. Shallower layers tend to converge earlier in training than deeper layers ([Raghu et al., 2017](#); [Morcos et al., 2018](#)), and weights of later layers change more than earlier ones ([Kovaleva et al., 2019](#)), suggesting that earlier layers tend to extract universal features whereas later layers focus on task-specific modeling. [Lee et al. \(2019\)](#) find that 90% of fully fine-tuned performance can be reached when fine-tuning only the final quarter of a transformer’s layers and leaving the rest frozen.

Several proposed methods vary the number of frozen layers over the course of training, approaching or exceeding the performance of fully fine-tuned models while substantially speeding up the training process ([Raghu et al., 2017](#); [Xiao et al., 2019](#); [Brock et al., 2017](#)). Similar to our approach, some dynamic freezing methods also employed caching mechanisms ([Liu et al., 2021](#); [He et al., 2021](#)), but the dynamic number of frozen layers means the cache applies only at training time and only for a single task. In contrast, we cache embeddings from the pretrained model, which can then be reused across multiple downstream tasks and applied at inference time as well.

Other recent studies have sought to improve model inference speed by skipping computations in later layers. [Sajjad et al. \(2020\)](#) found that in some cases up to half of the layers can be removed from the model with only a 1-3% drop in task performance. Early exit strategies have also been proposed, which allow the model to dynamically decide when to skip later layers ([Cambazoglu et al., 2010](#); [Xin et al., 2020](#)). SkipBERT ([Wang et al., 2022](#)) combined early exiting with an approach in which cached n-gram embeddings approximate the intermediate activations of new inputs. [Lester et al. \(2021\)](#) explored prompt-tuning as a parameter-efficient approach for adapting frozen language models without adjusting model weights, conditioning language models with soft prompts to perform downstream tasks.

Precomputing text representations to speed up future processing on the same data is commonly done when creating fixed-size document-level embeddings for use on document-level tasks ([Conneau et al., 2017](#); [Cohan et al., 2020](#)); in contrast, we study contextualized *token-level* embeddings that can be used for tasks such as named entity recognition (NER) and question answering. ReadOnce Transformers ([Lin et al., 2021](#)) do consider multi-task variable-length document representations, but do so in a setting where a cached document representation is paired with a query text (such as a question or prompt); the approach is pretrained with QA data and evaluated on QA and summarization, rather than tasks such as text classification or NER where the entire input can be cached.

[Du et al. \(2020\)](#) propose an approach similar to ours that caches general-purpose token-level model representations, trained in a multi-task setting; however, that approach only applies a small MLP to

the stored representations and reports a meaningful drop in accuracy (greater than 2% on average) compared to fully fine-tuned models. We find that reusing the later layer parameters of a pretrained transformer in addition to the cached activations often enables us to essentially match fully fine-tuned model accuracy while reducing computational cost.

Wei et al. (2022) combine layer freezing and knowledge distillation to create a multi-task model. They do not consider caching activations on persistent storage as we do, but instead re-use activations across tasks at inference time via a branching multi-task model. They use a two stage process where $12 - N$ layers are fine-tuned for each individual task keeping N frozen layers. This is followed by distillation of the N layers for further computational gains. We take advantage of the parameter efficient adapter modules (Houlsby et al., 2019), and replace this process with a single step of fine-tuning a frozen base model that has adapters attached only to the deeper layers.

Our work also has connections to work on memory- and retrieval-augmented language modeling. Prior work on using memory (e.g., Grave et al. (2016); Dai et al. (2019); Rae and Razavi (2020); Wu et al. (2022)) generally focuses on modeling long-range context and caching representations of older history in a sequence, while work on retrieval (e.g., Guu et al. (2020); Karpukhin et al. (2020)) focuses on fetching text from a knowledge base or corpus to serve as additional context. In both cases, the aim is to use representations of additional text (from earlier in a document or from a knowledge base) to improve modeling of new inputs. In contrast, our work focuses on caching the representations of an entire sequence to speed up computation for new tasks.

3 Methods

In the transformer architecture (Vaswani et al., 2017), an input sequence x of length S and dimension d is transformed with a function $F : \mathbb{R}^{S \times d} \rightarrow \mathbb{R}^{S \times d}$ defined by the composition of N transformer layers $F^{(1)}, \dots, F^{(N)}$ as follows:

$$F^\ell(x) = \text{LN}(\text{FF}^\ell(x') + x') \quad (1)$$

$$x' = \text{LN}(\text{MH}^\ell(x) + x) \quad (2)$$

where LN is a layer normalization (Ba et al., 2016), FF is a feed forward network, and MH is the self-attention layer that consists of multiple heads and

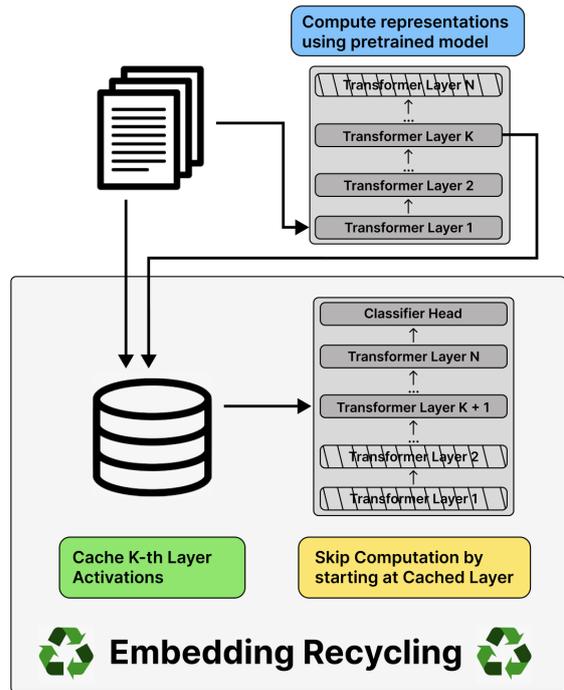


Figure 1: Overview of the embedding recycling approach. In the figure, the K -th layer activations are saved for future fine-tuning on downstream tasks, skipping redundant computations of earlier layers in the transformer model.

contextualizes the input sequence vector. The output of each layer is used as input to the next layer.

$$h^{\ell+1} = F^\ell(h^\ell) \quad (3)$$

Our approach is to cache the output representations $h^k \in \mathbb{R}^{S \times d}$ at a certain layer k and reuse them for fine-tuning on a new given task. We refer to this process of caching and reusing the output representations of a layer as *layer recycling*. This enables us to reduce the size of the transformer model from N layers to $N - k$ layers, reducing the computational cost during fine-tuning and inference.

Note that the key requirement of layer recycling is that we first need to process the entire data with the transformer model and cache the representations, so that we could later reuse these representations many times during fine-tuning and inference on new tasks. We experiment with two types of layer recycling approaches as explained next.

We start with a pretrained transformer F (e.g., BERT) consisting of $F^{(1)}, \dots, F^{(k)}, \dots, F^{(N)}$ layers. During the first epoch of fine-tuning for any given task, we run the transformer over a corpus \mathcal{C} and cache the output representations of layer k for each instance c in \mathcal{C} , i.e., $h_{c \in \mathcal{C}}^k$. However, for

every subsequent epoch of fine-tuning using the same transformer model, we only run and fine-tune the latter $N - k$ layers $F^{(k+1)}, \dots, F^{(N)}$. We can either train all of the weights in the layers (which we refer to as *reduced models*), or only train adapter modules added on the layers (discussed below). In either case, for the instance c in the dataset \mathcal{C} we simply retrieve and use the previously cached representation $h_{c \in \mathcal{C}}^k$ as input to layer $F^{(k+1)}$. This avoids the extra computation through layers $F^{(1)}, \dots, F^{(k)}$ but adds a small cost for retrieving the representation from storage (see [subsection 5.4](#) for efficiency analysis).

3.1 Adapters

We evaluate whether combining recycling with Adapter modules ([Houlsby et al., 2019](#)) can improve performance over fully fine-tuned models. Adapters are typically used to improve the parameter efficiency of fine-tuning and mitigate the storage costs of large language models. They also enable more sample-efficient fine-tuning and can result in improved fine-tuning performance ([Karimi Mahabadi et al., 2022](#)).

Adapter modules contain a down-projection, an up-projection, and a residual connection module: $h \leftarrow h + (f(h\mathbf{W}_{down})\mathbf{W}_{up})$. The adapters are separately inserted after the MH and the FF layers in the transformer architecture ([Equation 2](#)). Further, [Rücklé et al. \(2021\)](#) experiment with dropping adapters from the lower transformer layers to provide inference time speedup. In our experiments, adapters are added to the latter half of transformer layers in the reduced transformer models. As in standard layer recycling, the pretrained original transformer F first caches the intermediate activations $h_{c \in \mathcal{C}}^k$ for each input in a selected corpus at layer k . Then the first k layers are removed from the transformer. During fine-tuning, the cached representations are fed as input to the later $N - k$ layers of the transformer, which consist of the frozen transformer layers plus trainable adapter parameters. Thus, we fine-tune only the additional 6-8% parameters introduced by the adapters. We refer to learning adapters on all layers as the *full adapter* setting and the layer recycling version as the *reduced adapter* setting.

4 Experimental Setup

We now present our experiments evaluating whether recycled embeddings can be paired with

reduced large language models to maintain accuracy while improving training and inference speed. We explore the effectiveness of embedding recycling across a variety of different tasks, datasets, and transformer models.

4.1 Models

Our full-size models include the encoder transformers BERT, SciBERT ([Beltagy et al., 2019](#)), RoBERTa ([Liu et al., 2019b](#)), and DeBERTa ([He et al., 2020](#)). We also experiment with the encoder-decoder T5 model ([Raffel et al., 2019](#)). We selected these architectures since they are widely-used pretrained transformers across a variety of tasks in different domains. We experiment with multiple sizes of these models, including distilled ([Sanh et al., 2019; Wang et al., 2020, 2021](#)), base, and large variants, to gauge the effectiveness of recycled embeddings with an increase in the network size.

To investigate the effectiveness of layer recycling, we test several reduced models in which we use caching to reduce 50% of the layers (e.g., caching layer 12 in RoBERTa-large and layer 6 in BERT-base).¹ We compare each reduced model to its fully fine-tuned counterpart across the text classification, NER, and QA tasks. The hardware details and hyperparameters for our models are specified in [Appendix A](#).

4.2 Datasets

For our experiments, we focus on three core NLP tasks: text classification, named-entity recognition (NER), and extractive question-answering (QA). Scientific papers, due to their immutable nature, are an especially appropriate target for embedding recycling, so we focus much of our evaluation on the scientific domain. For text classification, we selected Chemprot ([Kringelum et al., 2016](#)), SciCite ([Cohan et al., 2019](#)), and SciERC ([Luan et al., 2018](#)). For NER, we used BC5CDR ([Li et al., 2016](#)), JNLPBA ([Collier and Kim, 2004](#)), and NCBI-Disease ([Doğan et al., 2014](#)). For QA, we chose the TriviaQA ([Joshi et al., 2017](#)) and SQuAD ([Rajpurkar et al., 2016](#)) datasets.

¹We note that for the encoder-decoder model T5, we consider caching only the middle layer of the *encoder*, which means that the speedups for this model will be smaller than (approximately half of) that of the other models we evaluate. We also consider 25% and 75% reduced models in [Appendix A](#).

5 Results

5.1 Standard Fine-tuning

The results for standard fine-tuning of either full or reduced models are shown in Table 1. For the text classification and NER tasks, the reduced BERT-sized and larger models perform similarly to their fine-tuned counterparts on average, and substantially outperform the distilled models. The reduced distilled models also perform well on those tasks compared to the distilled originals, on average, although there is more variance across models and tasks compared to BERT-sized models. We validate our fully fine-tuned baselines by comparing our results with prior work (Beltagy et al., 2019), finding that our scores land within 1.33% on average and typically score above the previous baselines.

For QA tasks, we found that fully fine-tuning works somewhat better than reduced configurations across all the explored models (Table 1). Generally, reduced configurations typically lag by 1 to 2 points in F-1 score. One possible hypothesis is that the QA datasets are generally much larger than the datasets we used for other tasks (100k-150k examples vs 4k-20k examples for text classification and NER); however, in additional experiments we found that subsampling the QA training sets to 5% of their original size only increased the gap, suggesting that dataset size does not explain the failure of reduced models on this task. We also validate our fully fine-tuned baselines for QA tasks by comparing our results with Yasunaga et al. (2022), finding that our scores differ by less than half a point on average.

Finally, we explored using lightweight multi-layer perceptrons (MLPs) as classifier heads, given their success in prior work. While (Du et al., 2020) paired multi-task encoders with 2-layer MLPs, we paired frozen pretrained transformer models with 2-layer MLPs and found that they underperformed trainable layers dramatically, by 26% on average across the classification and NER tasks.

5.2 Adapters

Our results for reduced adapter models are shown in Table 2. We see that in general, for all the models except for T5-Large, the adapter-based approaches are superior to standard fine-tuning on our tasks. Further, layer recycling remains effective with adapters. Compared to the full adapter baseline, the reduced adapters for RoBERTa-Large, BERT, SciBERT, and DeBERTa models only show a 0.19% reduction in accuracy. Additionally, com-

pared to the fully fine-tuned baseline, these reduced adapters models have a 0.19-0.23% reduction in accuracy. Likewise, in contrast to the full fine-tuning results above, QA accuracy for the top-performing DeBERTa adapter model remains unchanged on average after layer recycling, with the reduced adapter model performing better on one QA task and worse on the other.²

5.3 GLUE Results

For our best-performing model DeBERTa v2 XL, we also provide further experiments on datasets from the GLUE benchmark (Wang et al., 2018), to allow easier comparison against speedup techniques from previous work. We present results on the CoLA, SST-2, MRPC, STS-B, MNLI, and QNLI tasks from GLUE. For our experiments, we tried both our standard reduced models and our reduced adapter models. We found that embedding recycling was successful across the GLUE tasks, with an average accuracy drop of 0.3 points in return for a significant increase in both training and inference time as outlined in Table 5 and Table 4. We note that due to the high computational cost of these experiments, we take existing hyperparameter settings from previous work that worked well for the full models, and also use these for reduced models. Further hyperparameter optimization of the reduced models might improve performance.

5.4 Efficiency Analysis

To estimate the real-world benefit of recycling embeddings for different tasks, we provide a minimal PyTorch implementation of embedding recycling. This implementation and the following results correspond to both the standard layer recycling approach and the adapter-based layer recycling approach since they follow parallel processes for gradient descent during training and computations during inference, despite the additional 6-8% of parameters added by the trainable adapters. To show that training times do not differ substantially, we also measured the training time the transformer models take to converge to their optimal weights. We found both approaches take approximately the same time to complete training (Table 16).

We evaluated the impact of recycling embeddings on four different architectures and two dif-

²We omit experiments with distilled models, as we found adapters to be ineffective on those models even without embedding recycling, scoring 19.4% worse on average than full fine-tuning for text classification and NER.

Task	RoBERTa Large		(Sci)BERT		DeBERTa V2 XL		T5 Large		MiniLM L6-H768		MiniLM L6-H384		DistilBERT	
	Rdc	Full	Rdc	Full	Rdc	Full	Rdc	Full	Rdc	Full	Rdc	Full	Rdc	Full
ChemProt	84.3	83.9	84.0	84.0	86.8	86.7	84.6	84.1	78.3	79.3	76.9	74.6	80.3	79.1
SciCite	85.0	85.5	86.6	86.0	85.2	84.4	86.3	84.9	84.5	84.6	83.7	82.8	84.1	84.0
SciERC-Rel	80.2	80.4	76.7	79.8	79.9	80.2	77.4	80.2	74.8	78.2	72.1	68.9	74.9	72.9
Classification Avg.	83.2	83.3	82.4	83.3	84.0	83.8	82.8	83.1	79.2	80.7	77.6	75.4	79.8	78.7
bc5cdr	90.0	90.4	90.7	91.3	91.3	91.8	90.7	89.9	87.8	87.5	85.9	88.3	88.3	88.7
JNLPBA	79.4	78.7	78.8	79.0	78.5	78.2	79.6	80.0	77.3	76.9	74.0	77.2	78.6	78.5
NCBI-disease	93.0	93.2	93.4	92.9	93.3	93.4	92.8	93.5	91.1	92.1	89.9	91.7	90.5	91.3
NER Avg.	87.5	87.4	87.7	87.7	87.7	87.8	87.7	87.8	85.4	85.5	83.3	85.7	85.8	86.2
TriviaQA	78.2	79.8	67.4	69.1	80.6	81.8	77.4	78.2	72.2	73.8	69.2	71.0	64.7	66.8
SQuAD	91.8	93.6	87.5	88.5	94.5	94.6	93.7	93.9	85.0	87.0	89.0	89.6	84.8	85.4
QA Avg.	85.0	86.7	77.5	78.8	87.5	88.2	85.5	85.9	78.6	80.4	79.1	80.3	74.8	76.1

Table 1: Test scores of reduced (Rdc) models on the text classification, NER, and QA tasks. **Bold** indicates the best average F-1 score between the reduced and fully fine-tuned (Full) versions of each model over 10 runs. For the ChemProt dataset, we report the micro F-1 scores instead, following past work (Beltagy et al., 2019). The reduced BERT-sized models generally offer similar performance to their full counterparts (scoring within 0.2% when averaged across RoBERTa and SciBERT for the six tasks), and substantially outperform the distilled models.

ferent hardware platforms. For models, we considered two efficient transformer models (MiniLMv2 (Wang et al., 2020, 2021) models with $l = 6$ layers and embeddings of size $h = 384$ and $h = 768$), two medium sized models (BERT_{BASE}, $l = 12$, $h = 768$; BERT_{LARGE}, $l = 24$, $h = 1024$), and a large model (DeBERTa_{V2-XLARGE}, $l = 24$, $h = 1536$). We evaluated embeddings on a efficiency-oriented AI accelerator (NVIDIA A10G), as well as on a high-performance GPU (NVIDIA A6000).

We controlled for differences among tasks considered in tables Table 1, 2, and 3, such as length of sequences and number of samples, by simulating a sequence classification task on QASPER (Dasigi et al., 2021), which includes the full-text of over a thousand academic manuscripts.³ We run all models with a fixed batch size of 128. For all models, we reduce exactly half of their layers by recycling, which results in a maximum theoretical speed-up of 100%. A run over the corpus consists of 335 batches, and we average results over seven runs.

Table 4 shows the results of caching embeddings to recycle on disk. Overall, we found that all models benefit from embedding recycling, achieving an average speedup ranging from 18 to 86%. Unsurprisingly, larger models benefit more from recycling than smaller ones; this is due to the fact that loading embeddings cached on disk adds a

³Because the bulk of computation for a transformer model is done in its encoder and not in the task-specific heads, inference time is similar regardless of whether the model is used for sequence classification, tagging, or question answering.

small latency penalty to a model run, which is more noticeable in the case of smaller models. For example, we achieve an 84% speedup when running BERT_{BASE} with embedding recycling on an A10G GPU, which is roughly equivalent to the latency of a MiniLM_{L6-H768} model without recycling (351 vs 325 ms per batch on average); this result would allow to run more accurate models while maintaining the efficiency of shallower architectures.

Table 4 also includes results when storing embeddings using half precision (that is, cache embeddings in FP16 rather FP32). The smaller embeddings lead to improvements for all models and hardware, ranging from +8% to +46%. Further, it has no impact on performance, as it changes predicted scores by at most 10^{-3} across all tasks evaluated in this work.

We also note that less capable hardware benefits more from caching embeddings. For example, BERT_{BASE} achieves a speedup of 84% on an A10G GPU, while on A6000, the speedup is a more modest 55%. This is an expected result: fewer and slower execution cores/accelerator memory impact overall model latency. Further, we note that, despite the smaller relative gains, the more powerful GPU is always faster in absolute terms compared with the less capable one.

It is important to note that these gaps from maximum achievable speedup are only observed when performing *inference*; for *training*, we observe almost perfect speed-up for all models and

Task	RoBERTa Large			(Sci)BERT			DeBERTa V2 XL			T5 Large		
	Rdc + Half Adpt	Full Adpt	Full	Rdc + Half Adpt	Full Adpt	Full	Rdc + Half Adpt	Full Adpt	Full	Rdc + Half Adpt	Full Adpt	Full
ChemProt	84.1	85.2	83.9	84.2	84.9	84.0	87.2	86.5	86.7	84.3	84.9	84.1
SciCite	82.4	82.9	85.5	85.5	84.6	86.0	84.6	85.0	84.4	85.3	84.5	84.9
SciERC-Rel	85.7	85.9	80.4	86.0	85.5	79.8	82.9	82.1	80.2	76.2	75.6	80.2
Classification Avg.	84.1	84.7	83.3	85.2	85.0	83.3	84.9	84.6	83.8	81.9	81.7	83.1
bc5cdr	90.0	90.6	90.4	90.0	90.9	91.3	90.7	91.1	91.8	79.9	85.7	89.9
JNLPBA	79.1	79.2	78.7	79.8	78.3	79.0	79.3	79.0	78.2	78.8	79.5	80.0
NCBI-disease	92.8	93.1	93.2	93.1	93.0	92.9	93.3	93.5	93.4	92.1	92.5	93.5
NER Avg.	87.3	87.6	87.4	87.6	87.4	87.7	87.8	87.9	87.8	83.6	85.9	87.8
TriviaQA	78.5	79.8	79.8	67.4	68.9	69.1	81.6	82.3	81.8	77.0	77.5	78.2
SQuAD	93.5	93.4	93.6	87.9	87.9	88.5	94.7	93.9	94.6	90.6	91.0	93.9
QA Avg.	86.0	86.6	86.7	77.6	78.4	78.8	88.1	88.1	88.2	83.8	84.3	85.9

Table 2: Test scores of reduced adapter (Rdc + Half Adpt) models on the text classification, NER, and QA tasks. **Bold** indicates the best average F-1 score between the reduced adapter, full adapter (Full Adpt), and fully fine-tuned (Full) versions of each model over 10 runs. For the ChemProt dataset, we report the micro F-1 scores instead, following past work (Beltagy et al., 2019). The reduced, adapter-based transformer models offer similar performance to their full counterparts (scoring within 0.4% when averaged across RoBERTa, SciBERT, and DeBERTa for the eight tasks), and substantially outperform the distilled models.

GLUE task	DeBERTa V2 XL			
	Rdc + Half Adpt	Full Adpt	Rdc	Full
CoLA	70.9	71.3	70.8	71.2
SST-2	96.9	97.1	97.1	97.4
Single Sentence Avg.	83.9	84.2	84.0	84.3
MRPC	93.9	94.0	93.4	93.9
STS-B	92.4	92.7	92.5	92.8
Similarity and Paraphrase Avg.	93.2	93.4	93.0	93.4
MNLI-m	91.7	92.0	91.0	91.4
QNLI	95.0	95.1	94.1	94.8
NLI Avg.	93.3	93.6	92.6	93.1

Table 3: Test scores of reduced (Rdc) and reduced adapter (Rdc + Half Adpt) models on GLUE for DeBERTa V2 XL. **Bold** indicates the best average score between the reduced and fully fine-tuned (Full) versions for the standard and adapter-based configurations. Each score is averaged over 5 runs. We report the scores using the standard GLUE metric for each corresponding task.

hardware configurations except for the smaller MiniLM models. For example, BERT_{BASE} requires 17.38 ± 1.32 ms/batch⁴ without recycling, compared to 8.67 ± 2.18 ms/batch when recycling. Even when considering the additional time to cache embeddings to disk during the first pass, embed-

ding recycling still achieves close to optimum speedup on all models except MiniLMs, where its gains hover between 52% and 82% (“NR vs SR” column in Table 5). When training for just 6 epochs (or roughly 2,000 steps), recycling embeddings is faster than simply freezing half of the parameters for all models but MiniLM (“F vs SR” column in Table 5); this is due to the relatively higher cost of caching layers to disk in case of smaller models. In these cases, we empirically found that recycling achieves faster training time than freezing after 12 epochs or 4,000 training steps; since smaller models typically require more epochs to converge, we conclude that recycling is generally preferable to partially freezing a model during training. For BERT_{BASE} and larger models, embedding recycling is also more efficient than layer freezing, providing a +20% to +45% speed-up after just 6 training epochs.

We also benchmarked the storage requirements of recycling embeddings. For a sequence of 512 tokens and a hidden model dimension of 768, caching embeddings requires 1.6 MB with 32-bit precision or 0.8 MB with 16-bit precision. This translates to 15.5 MB per paper in QASPER (papers are, on average 4,884 WordPiece tokens long). Weighing the storage cost and compute savings of ER, we find that it is cost-effective in cloud environments only if the corpus is reprocessed several times per

⁴When training, we use a batch size of 16

Inference Time (Speedup over Baseline)				
Model	Baseline	Recycling		Avg. F1 diff when recycling
		FP32 cache	FP16 cache	
<i>NVIDIA A10G</i>				
MiniLM L6-H384	183 ms	154 ms (+21%)	123 ms (+67%)	-0.2
MiniLM L6-H768	325 ms	201 ms (+56%)	195 ms (+66%)	-0.4
BERT BASE	647 ms	351 ms (+84%)	343 ms (+88%)	-0.3
BERT LARGE	1943 ms	1066 ms (+86%)	1004 ms (+93%)	-0.2
DeBERTa V2-XLARGE	1914 ms	1010 ms (+89%)	985 ms (+94%)	-0.1
<i>NVIDIA A6000</i>				
MiniLM L6-H384	123 ms	105 ms (+18%)	100 ms (+23%)	-0.2
MiniLM L6-H768	208 ms	161 ms (+29%)	150 ms (+38%)	-0.4
BERT BASE	416 ms	269 ms (+55%)	245 ms (+59%)	-0.3
BERT LARGE	1235 ms	662 ms (+86%)	643 ms (+92%)	-0.2
DeBERTa V2-XLARGE	1430 ms	777 ms (+84%)	758 ms (+89%)	-0.1

Table 4: Average **inference** runtime comparison (in ms/batch, averaged over 7 runs) between vanilla encoders and models that cache embeddings on disk. For all runs, cache the middle layer of the encoder. We assume the cache is already precomputed when calculating timings; thus, maximum speedup is 100%. Overall, the larger the model, the higher the speedup from re-using representations. Further, accelerators with fewer execution units (A10G) benefit more from recycling embeddings. Finally, using half precision for embeddings improves speed up across the board, while halving storage size.

month, but is cost-effective on local hardware even with infrequent (yearly) corpus reprocessing (details in subsection A.8 of the appendix).

6 Discussion and Future Work

Our experiments raise several questions and suggest multiple avenues for future work, including:

- Our layer recycling strategy is a straightforward ER approach, but previous work has suggested that weighted pooling across layers can perform better compared to any single layer in many cases (Liu et al., 2019a; Du et al., 2020). Recycling pooled activations may offer improved results. What is the best way to capture and store the syntactic and semantic knowledge encoded in the activations of a model for later recycling?

- As noted in the previous section, naive storage methods for ER can be cost-prohibitive in some settings, and finding ways to mitigate this cost (e.g., by compressing the stored activations) will be important for making ER broadly applicable.

- Our experiments show that the right recycling approach may be task-specific and model-specific. For example, with standard fine-tuning as shown in Table 8, caching layer 12 in RoBERTa-large is most effective for NER and text classification, whereas it is not effective for QA (but layer 6 performs much better). Which embeddings to retrieve and recycle for a task, and the right architecture (e.g. number of layers) to use when consuming the recycled embeddings, represents a large decision space. Methods that can help practitioners automatically choose among public or private shared embedding sets and associated model designs, given their task and objectives for accuracy and computational cost, may be important to make ER an effective practical tool.

- We present results with encoder-only and encoder-decoder models, on classification tasks. Determining whether the approach is effective for generative tasks and autoregressive models is an important question for future work.

- While we show that ER can be effective when coupled with distillation, whether other techniques like quantization and early exiting remain effective in combination with ER is an open question.

- We focus on the setting where the exact same text, at the length of a full document, is being reused for multiple tasks. In practice, we may often perform a task on text that is *similar* to but not exactly the same as one for which we have cached embeddings (e.g., a Wikipedia page that has been revised). Further, even a completely new document will have similarities and overlapped spans with previously processed ones. Studying ER in these settings, e.g. through a combination of layer recycling and the SkipBERT approach which can apply to unseen passages via cached n-grams (Wang et al., 2022), is an area of future work.

- Finally, it is possible to explore cross-model embedding recycling. We attempted a straightforward implementation of such approach by using recycling embeddings from a larger model into a smaller consumer model. However, the results did

Model	Training (ms/batch, amortized over 6 epochs)				Speedup		
	No Recycling (NR)	Model Frozen (F)	Saving + Recycling (SR)	Only Recycling (R)	NR vs SR	F vs SR	NR vs R
NVIDIA A10G							
MiniLM ₃₈₄	51 ± 1	30 ± 1	32 ± 6	25 ± 4	+59%	-7%	+104%
MiniLM ₇₆₈	90 ± 4	56 ± 1	50 ± 4	45 ± 3	+80%	+12%	+100%
BERT _{BASE}	173 ± 2	112 ± 1	90 ± 4	87 ± 3	+92%	+24%	+99%
BERT _{LARGE}	347 ± 1	246 ± 1	181 ± 2	176 ± 2	+92%	+36%	+97%
DeBERTa _{XLARGE}	380 ± 2	286 ± 1	199 ± 1	194 ± 1	+91%	+44%	+96%
NVIDIA A6000							
MiniLM ₃₈₄	41 ± 1	24 ± 1	26 ± 5	22 ± 3	+55%	-8%	+81%
MiniLM ₇₆₈	61 ± 1	38 ± 1	40 ± 5	34 ± 3	+52%	-5%	+82%
BERT _{BASE}	117 ± 1	78 ± 1	60 ± 3	58 ± 2	+94%	+30%	+102%
BERT _{LARGE}	326 ± 2	212 ± 1	167 ± 2	161 ± 1	+96%	+26%	+103%
DeBERTa _{XLARGE}	359 ± 2	250 ± 1	184 ± 1	178 ± 1	+95%	+35%	+102%

Table 5: Average **training** runtime comparison (in ms per batch, \pm stdev over 7 runs) between vanilla encoders and models that cache embeddings on disk. For all runs, we cache the middle layer of the encoder; thus, theoretical speedup is 100%. Time per batch is amortized over 6 epochs (2,000 steps), the lowest number to convergence over all datasets (c.r.f. Table 16). We present results in four settings: no recycling (NR), freezing $\frac{1}{2}$ of the layers during training (F), 1 training epoch during which embeddings are saved to disk followed by 5 epochs where recycling is enabled (SR), and 6 epochs where embeddings are already saved (R). Overall, we found that embedding recycling speeds up training even when embeddings need to be cached to disk during the first pass. Compared to freezing, saving and recycling improves training time for all but MiniLM models (F vs SR).

not show improvements (Appx. A.3). Developing and evaluating new approaches for this setting is an important item for future work.

7 Conclusion

We have presented embedding recycling, a general technique for reusing previous activations of neural language models to improve the efficiency of future training and inference. We show how a simple technique of caching a layer of activations in a pretrained model is effective. We validate our approach in experiments across fourteen tasks and eight model architectures. We find that recycling typically has small or no impacts to accuracy on average, but does yield substantial throughput increases demonstrated through a careful efficiency analysis. We also discuss several open challenges for future work.

8 Limitations

As discussed in detail in our future work section, several advances are important to make embedding recycling a broadly applicable practical technique. In addition, the techniques we evaluate primarily benefit transformer language models run on GPU-based architectures with rapid storage, components which are not available to all NLP researchers and practitioners. Our experiments demonstrate posi-

tive results with one representative embedding recycling technique, but do not directly evaluate all recycling variants proposed earlier in the literature. Finally, the datasets used in our experiments were in English, a high-resource language with robust pretrained models which may benefit embedding recycling. Future work should expand on the applicability of embedding recycling by using non-English datasets in lower-resource settings to determine the breadth of its applicability.

Acknowledgments

This work was supported in part by NSF Convergence Accelerator Grant OIA-2033558. We thank Chris Coleman for helpful discussions.

References

- Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *ArXiv*, abs/1607.06450.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx,

- Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. 2017. Freezeout: Accelerate training by progressively freezing layers. *ArXiv*, abs/1706.04983.
- B Barla Cambazoglu, Hugo Zaragoza, Olivier Chapelle, Jiang Chen, Ciya Liao, Zhaohui Zheng, and Jon De-genhardt. 2010. Early exit optimizations for additive machine learned ranking systems. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 411–420.
- Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. [Structural scaffolds for citation intent classification in scientific publications](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.
- Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. [SPECTER: Document-level representation learning using citation-informed transformers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.
- Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. [Supervised learning of universal sentence representations from natural language inference data](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.
- Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. [A dataset of information-seeking questions and answers anchored in research papers](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*.
- Jesse Dodge, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A Smith, Nicole DeCario, and Will Buchanan. 2022. Measuring the carbon intensity of ai in cloud instances. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1877–1894.
- Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.
- Jingfei Du, Myle Ott, Haoran Li, Xing Zhou, and Veselin Stoyanov. 2020. [General purpose text embeddings from pre-trained language models for scalable inference](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3018–3030, Online. Association for Computational Linguistics.
- Yoav Goldberg. 2019. Assessing bert’s syntactic abilities. *arXiv preprint arXiv:1901.05287*.
- Edouard Grave, Armand Joulin, and Nicolas Usunier. 2016. Improving neural language models with a continuous cache. In *International Conference on Learning Representations*.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. [Retrieval augmented language model pre-training](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.
- Chaoyang He, Shen Li, Mahdi Soltanolkotabi, and Salman Avestimehr. 2021. Pipetransformer: automated elastic pipelining for distributed training of large-scale models. In *International Conference on Machine Learning*, pages 4150–4159. PMLR.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.
- Jeremy Howard and Sebastian Ruder. 2018. [Universal language model fine-tuning for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *CoRR*, abs/2001.08361.
- Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, and Majid Yazdani. 2022. [Prompt-free and efficient few-shot learning with language models](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. [Revealing the dark secrets of BERT](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.
- Jens Kringelum, Sonny Kim Kjaerulff, Søren Brunak, Ole Lund, Tudor I Oprea, and Olivier Taboureau. 2016. Chemprot-3.0: a global chemical biology diseases mapping. *Database*, 2016.
- Jaejun Lee, Raphael Tang, and Jimmy J. Lin. 2019. What would elsa do? freezing layers during transformer fine-tuning. *ArXiv*, abs/1911.03090.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.
- Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wieggers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.
- Shih-Ting Lin, Ashish Sabharwal, and Tushar Khot. 2021. [ReadOnce transformers: Reusable representations of text for transformers](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7129–7141, Online. Association for Computational Linguistics.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. [Linguistic knowledge and transferability of contextual representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yuhan Liu, Saurabh Agarwal, and Shivaram Venkataraman. 2021. Autofreeze: Automatically freezing model blocks to accelerate fine-tuning. *ArXiv*, abs/2102.01386.
- Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. [Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.
- Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Jack Rae and Ali Razavi. 2020. [Do transformers need deep long-range memory?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7524–7529, Online. Association for Computational Linguistics.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.
- Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. Adapterdrop: On the efficiency of adapters in transformers. In *EMNLP*.
- Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. On the effect of dropping layers of pre-trained transformer models. *arXiv preprint arXiv:2004.03844*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Jue Wang, Ke Chen, Gang Chen, Lidan Shou, and Julian McAuley. 2022. Skipbert: Efficient inference with shallow layer skipping. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7287–7301.
- Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.
- Tianwen Wei, Jianwei Qi, and Shenghuang He. 2022. A flexible multi-task model for bert serving. In *ACL*.
- Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint arXiv:1909.10430*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. *arXiv preprint arXiv:2203.08913*.
- Xueli Xiao, Thosini Bamunu Mudiyansele, Chunyan Ji, Jie Hu, and Yi Pan. 2019. Fast deep learning training through intelligently freezing layers. *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1225–1232.
- Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online. Association for Computational Linguistics.
- Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.

A Experimental Setup and Additional Results

A.1 Fine-tuning Transformer Models

The candidate transformer models are fine-tuned using configurations suggested by [Devlin et al. \(2019\)](#), [Ding et al. \(2022\)](#) and [Houlsby et al. \(2019\)](#). For text classification, we feed the final hidden state of the [CLS] token into a linear classification layer.

For NER and QA, we feed the final hidden states of each token into a linear classification layer with a softmax output.

For all of the models, we apply a dropout of 0.1 to the transformer outputs and optimize for cross entropy loss using Adam (Kingma and Ba, 2015). We employ a batch size of 32 across all tasks. We fine-tune using early stopping with a patience of 10, using a validation set for calculating loss for each epoch. We use a linear warmup followed by linear decay for training (Howard and Ruder, 2018), testing the following learning rate options: 1e-3, 2e-3, 1e-4, 2e-4, 1e-5, 2e-5, 5e-5, and 5e-6. For the text classification and NER datasets, we select the best performing learning rate for each transformer model on the development set and report the corresponding test results. For the QA datasets, we select the best performing learning rate for each transformer model on the training set and report the corresponding results on the validation set. Additionally, for the adapter modules used in certain model configurations, we test bottleneck dimensions as part of our hyperparameter search: 24, 64, and 256.

A.2 Adapter-based Models

Here, we used frozen RoBERTa-Large (Liu et al., 2019b), SciBERT (Beltagy et al., 2019), and BERT models but added adapter modules (Houlsby et al., 2019) only on the latter half of the transformer layers. Only the adapters and the linear classifier attached to the model output were fine-tuned for the text classification, NER, and QA tasks.

We found that the best hyperparameter configuration was generally a bottleneck dimension of 256 and a learning rate of either 1e-4 or 2e-4.

A.3 Cross-model Embedding Reuse

An alternative to re-using cached activation from a pre-trained model (section 5), is to cache activations from a more expensive, larger model and re-using them in downstream cheaper models. The goal here is to improve accuracy by using more powerful contextual embeddings. Overall, a straightforward implementation of this strategy did not offer improvements, as described below.

We experiment with reusing precomputed embeddings from one source model F in a consumer model F' that has a different size but the same tokenization vocabulary. The activations of the *final* transformer layer $h_{c \in C}^N$ are stored for each input c from corpus C . During the fine-tuning of the

consumer model F' , these stored activations are transformed through a learned 2-layer MLP with ReLU activation⁵ and added to the input embeddings of F' . We tried two frameworks for pairing large language model embeddings with compact models: F =Roberta-large \rightarrow F' =MiniLM-6L-H768 and F =BERT-base \rightarrow F' =DistilBERT.

Overall, as shown in Table 6 the larger model’s contextual representations do not improve the smaller model’s accuracy; in fact adding them decreases the average F1 score by 0.3-0.9 points.

A.4 Efficiency of Embedding Recycling when Training

For training, we observe almost perfect speed-up for all models and hardware configuration, barring MiniLM models on the machine equipped with a A6000 GPU (“NR vs R” column in Table 5). For example, BERT_{BASE} requires 17.38 ± 1.32 ms/batch⁶ without recycling, compared to 8.67 ± 2.18 ms/batch when recycling. Even when considering the additional time to cache embeddings to disk during the first pass, embedding recycling still achieves close to optimum speedup on all models except MiniLMs, where its gains hover between 52% and 82% (“NR vs SR” column in Table 5). When training for just 6 epochs (or roughly 2,000 steps), recycling embeddings is faster than simply freezing half of the parameters for all models but MiniLM (“F vs SR” column in Table 5); this is due to the relatively higher cost of caching layers to disk in case of smaller models. In these cases, we empirically found that recycling achieves faster training time than freezing after 12 epochs or 4,000 training steps; since smaller models typically require more epochs to converge, we conclude that recycling is generally preferable to partially freezing a model during training.

A.5 Embedding Pre-fetching while Recycling

Storing embeddings on NVMe drives, while fast, introduce additional latency compared to RAM. For example, BERT_{BASE} achieves an average latency of 351 ± 1 ms/batch when caching on disk (84% speedup), compared to just 334 ± 1 ms/batch when using memory (94% speedup). This is due to the fact that, while embeddings are being loaded from disk, the hardware accelerator responsible for executing the rest of the model sits idle. To reduce

⁵We found that MLP achieved better performance compared with a single linear layer on dev set.

⁶When training, we use a batch size of 16

		RoBERTa-Large + MiniLM L6-H768	MiniLM L6-H768	BERT + DistilBERT	DistilBERT
Chemprot	Micro F-1	78.9 (0.3)	79.3 (0.3)	77.8 (0.4)	79.1 (0.5)
	Macro F-1	52.2 (0.2)	52.6 (0.4)	51.2 (0.5)	52.6 (0.3)
SciCite	Micro F-1	85.2 (0.3)	86.0 (0.2)	85.7 (0.1)	85.5 (0.1)
	Macro F-1	83.8 (0.3)	84.6 (0.2)	84.2 (0.1)	84.0 (0.1)
SciERC-Rel	Micro F-1	85.1 (0.4)	86.3 (0.2)	83.8 (0.2)	83.5 (0.4)
	Macro F-1	76.2 (0.8)	78.2 (0.6)	73.6 (0.6)	72.9 (0.7)
Text Classification Average Score		76.9	77.8	76.0	76.3

Table 6: Cross-Model Recycling Results for RoBERTa+MiniLM-L6H768 and BERT+DistilBERT configurations. **Bold** indicates the best average score between the cross-model recycling and fully finetuned versions of each model. Each score represents the average score of 10 runs, with the standard errors for each score in parentheses.

the impact of this latency penalty, our implementation supports *pre-fetching* of future embeddings: when processing a sequence of inputs, such as sentences in a manuscript, it loads embeddings for tokens ahead of the sequence inference is currently being run on. This optimization reduces the time accelerators wait for data to be available for inference; for example, in the case of BERT_{BASE} on A10G, disabling pre-fetching raised inference time to 374 ± 1 ms/batch (vs 351 ± 1 ms/batch with pre-fetching). Therefore in this section, all results are reported with prefetching enabled.

A.6 Software and Hardware

For implementation, we use the v4.19 version of the Transformers library (Wolf et al., 2019), the v0.4 version of the OpenDelta library (Ding et al., 2022), and the v1.11 version of the Pytorch library (Paszke et al., 2019). We conduct our experiments using NVIDIA RTX A6000 GPUs and NVIDIA A10G GPUs with CUDA v11.5.

A.7 Considerations in Selecting Hardware for Proof-of-Concept Recycling Experiments

We ran our proof-of-concept implementation on an AWS Cloud instance⁷ equipped with an NVIDIA A10G accelerator, and on a NVIDIA A6000 within an on-premise server⁸. The former contains fewer execution units (72 vs 84), fewer tensor cores (288 vs 336), slower memory (600 vs 768 GB/s), and slower boost clock (1800 MHz vs 1695 MHz). However, it is much more efficient, being rated at 150W (compare with A6000’s 300W power target). Therefore, the NVIDIA A10G accelerator presents a more realistic platform for embedding recycling, since it is more suitable for cost-efficient

large-scale model deployments. Both machines are equipped with PCIe NVMe drives, which we use to cache embeddings to recycle.

A.8 Cost-effectiveness of Embedding Recycling

In this section we attempt to estimate how cost-effective embedding recycling is for inference in a real-world setting. While this depends heavily on use-case-specific assumptions, we consider two typical settings as proofs-of-concept, one using cloud computing and one using local hardware.

There are four main factors that affect the cost-benefit ratio of embedding recycling: (1) compute cost, (2) storage cost, (3) model architecture, and (4) frequency of corpus reprocessing (i.e., how often the cached embeddings will be used). Compute costs are challenging to estimate for a locally-owned hardware setting due to many hidden cost factors beyond the GPUs (cooling, electrical costs, server to house the GPUs, etc) and so we use AWS EC2 cloud GPU prices as a cost estimate for both cloud and local hardware. In particular, we consider a g5.12xlarge instance with $4 \times$ A10G GPUs at 5.67 \$/hr.

Storage costs are easier to estimate for local hardware than compute costs, and local storage can be significantly cheaper because embedding recycling does not require the availability and durability guarantees provided by cloud solutions (the cache is accessed infrequently and can always be recomputed if it is lost). Therefore, we consider both a cloud storage solution (AWS S3 one-zone infrequent access, at 0.01 \$/GB/month) and a local storage solution. For local storage, we consider current consumer-grade hard drive prices at approximately 16.9 \$/TB based on data from Amazon and Newegg, and assume a lifespan of 6 years based on

⁷g5.2xlarge instance with 8 cores and 32 GB of RAM.

⁸Intel-based system with 128 cores and 512 GB of RAM.

Model	Cloud	Local
MiniLM ₃₈₄	0.05	2.2
MiniLM ₇₆₈	0.05	2.4
BERT _{BASE}	0.13	5.6
BERT _{LARGE}	0.30	12.9
DeBERTa _{XLARGE}	0.20	8.5

Table 7: Minimum reprocessing frequency (in months) needed in order for embedding recycling to be cost-effective in various model and hardware configurations.

data from Backblaze.⁹ This results in an average cost of 0.23 \$/TB/month over the life of the drive. Finally, we note that AWS does not charge for data transfer between S3 and EC2 within a region, so we can ignore data transfer costs in this calculation.

The frequency of corpus reprocessing is highly variable, so we report results in terms of the minimum reprocessing frequency that would be necessary for embedding recycling to be cost-effective. For all models we assume each input is 512 tokens and the cache is stored with FP16 precision.

Table 7 shows the minimum reprocessing frequency needed for embedding recycling to be cost effective for our models on cloud and local hardware. Under our assumptions, we find that embedding recycling is cost-effective in a cloud setting only if the corpus is reprocessed very frequently (several times per month). This may be realistic in some use cases, such as when a large team is working with the same corpus and developing many new models, or if new training data arrives frequently and the model developer wants to continually update and re-deploy it.

With local hardware the calculation is much more favorable; embedding recycling with BERT_{LARGE} would be worthwhile even if the corpus were only reprocessed once per year.

We note that embedding recycling could become substantially more cost effective with further development. In this work we did not explore ways to reduce storage costs, such as quantization or compression. In addition, while our experiments only considered sequence lengths of 512 tokens, for many full-text document corpora it is desirable to use a much longer sequence length to fit the whole document into a model at once. Because the computational cost of transformers generally scales superlinearly with input length (but storage

cost scales only linearly), embedding recycling will be more effective as the sequence length grows.

⁹<https://www.backblaze.com/blog/how-long-do-disk-drives-last/>

		RoBERTa-Large					
		Reduced + Half Adpt	Full Adapters	6 Layers Reduced	12 Layers Reduced	18 Layers Reduced	Fully Finetuned
ChemProt	Micro F-1	84.1 (0.4)	85.2 (0.3)	84.2 (0.3)	84.3 (0.2)	82.0 (0.2)	83.9 (0.3)
	Macro F-1	60.8 (0.7)	57.5 (0.7)	56.4 (0.4)	56.5 (0.3)	54.5 (0.5)	56.5 (0.4)
SciCite	Micro F-1	85.2 (0.3)	85.6 (0.5)	86.2 (0.2)	86.2 (0.2)	86.2 (0.2)	86.8 (0.2)
	Macro F-1	82.4 (0.4)	82.9 (0.6)	84.9 (0.2)	85.0 (0.2)	85.0 (0.2)	85.5 (0.2)
SciERC-Rel	Micro F-1	89.0 (0.5)	89.3 (0.6)	87.1 (0.4)	86.8 (0.4)	86.1 (0.2)	87.3 (0.4)
	Macro F-1	85.7 (0.7)	85.9 (0.9)	79.4 (0.7)	80.2 (0.8)	76.2 (0.4)	80.4 (0.6)
Text Classification Average Score		81.2	81.1	79.7	79.8	78.3	80.1
bc5cdr	Micro F-1	97.4 (0.0)	97.6 (0.0)	97.2 (0.3)	97.4 (0.0)	97.3 (0.0)	97.5 (0.0)
	Macro F-1	90.0 (0.0)	90.6 (0.0)	89.0 (1.2)	90.0 (0.0)	89.5 (0.1)	90.4 (0.1)
JNLPBA	Micro F-1	93.8 (0.0)	93.8 (0.0)	93.8 (0.0)	93.9 (0.0)	93.7 (0.0)	93.7 (0.1)
	Macro F-1	79.1 (0.1)	79.2 (0.2)	79.3 (0.1)	79.4 (0.1)	79.0 (0.1)	78.7 (0.3)
NCBI-disease	Micro F-1	98.5 (0.0)	98.6 (0.0)	98.5 (0.0)	98.5 (0.0)	98.4 (0.0)	98.6 (0.0)
	Macro F-1	92.8 (0.1)	93.1 (0.1)	93.0 (0.1)	93.0 (0.1)	92.4 (0.1)	93.2 (0.1)
NER Average Score		91.9	92.1	91.8	92.0	91.7	92.0
TriviaQA	Micro F-1	75.3 (0.1)	76.8 (0.2)	76.6 (0.2)	75.1 (0.1)	70.8 (0.1)	76.7 (0.1)
	Macro F-1	78.5 (0.1)	79.8 (0.1)	79.7 (0.2)	78.2 (0.1)	73.8 (0.1)	79.8 (0.1)
SQuAD	Micro F-1	87.0 (0.1)	86.7 (0.0)	86.2 (0.0)	84.7 (0.0)	79.3 (0.0)	87.4 (0.0)
	Macro F-1	93.5 (0.1)	93.4 (0.0)	92.8 (0.0)	91.8 (0.0)	87.8 (0.0)	93.6 (0.0)
QA Average Score		83.6	84.1	83.8	82.4	77.9	84.3

Table 8: RoBERTa Results for Reduced Models. **Bold** indicates the best average score between the standard reduced, adapter-based reduced, and fully fine-tuned versions of each model. **Reduced + Half Adpt** indicates adapters on the transformer layers of a fully frozen reduced model, where the earlier half of transformer layers were removed and their activations cached. **Full Adapters** indicates adapters on all transformer layers of a fully frozen model. Each score represents the average score of 10 runs, with the standard errors for each score in parentheses.

		SciBERT					
		Reduced + Half Adpt	Full Adapters	3 Layers Reduced	6 Layers Reduced	9 Layers Reduced	Fully Finetuned
ChemProt	Micro F-1	84.2 (0.3)	84.9 (0.4)	83.8 (0.4)	84.0 (0.2)	81.9 (0.2)	84.0 (0.3)
	Macro F-1	56.9 (0.8)	54.8 (0.4)	56.5 (0.5)	57.0 (0.3)	54.3 (0.3)	56.3 (0.4)
SciCite	Micro F-1	86.6 (0.2)	85.8 (0.1)	87.1 (0.1)	87.6 (0.1)	87.4 (0.1)	87.1 (0.2)
	Macro F-1	85.5 (0.3)	84.6 (0.1)	86.1 (0.1)	86.6 (0.1)	86.2 (0.1)	86.0 (0.2)
SciERC-Rel	Micro F-1	89.4 (0.4)	88.5 (0.6)	86.6 (0.3)	86.1 (0.2)	85.4 (0.2)	86.3 (0.2)
	Macro F-1	86.0 (0.7)	85.5 (0.6)	77.6 (0.5)	76.7 (0.3)	76.2 (0.4)	79.8 (0.5)
Text Classification Average Performance		81.4	80.7	79.6	79.7	78.6	79.9
bc5cdr	Micro F-1	97.5 (0.0)	97.7 (0.1)	97.7 (0.0)	97.6 (0.0)	97.5 (0.0)	97.7 (0.0)
	Macro F-1	90.0 (0.0)	90.9 (0.1)	91.0 (0.1)	90.7 (0.0)	90.2 (0.1)	91.3 (0.0)
JNLPBA	Micro F-1	94.0 (0.0)	93.5 (0.0)	93.6 (0.1)	93.7 (0.1)	93.8 (0.0)	93.6 (0.1)
	Macro F-1	79.8 (0.0)	78.3 (0.2)	78.6 (0.4)	78.8 (0.2)	79.0 (0.1)	79.0 (0.2)
NCBI-disease	Micro F-1	98.6 (0.0)	98.5 (0.0)	98.5 (0.0)	98.6 (0.0)	98.5 (0.0)	98.5 (0.0)
	Macro F-1	93.1 (0.1)	93.0 (0.1)	92.9 (0.1)	93.4 (0.1)	93.1 (0.1)	92.9 (0.1)
NER Average Perforamcne		92.2	92.0	92	92.1	92	92.2

Table 9: SciBERT text classification and NER results for Reduced Models. **Bold** indicates the best average score between the standard reduced, adapter-based reduced, and fully fine-tuned versions of each model. **Reduced + Half Adpt** indicates adapters on the transformer layers of a fully frozen reduced model, where the earlier half of transformer layers were removed and their activations cached. **Full Adapters** indicates adapters on all transformer layers of a fully frozen model. Each score represents the average score of 10 runs, with the standard errors for each score in parentheses. QA tasks are not included since SciBERT was pretrained for scientific datasets.

BERT							
		Reduced + Full	Full	3 Layers	6 Layers	9 Layers	Fully
		Half Adpt	Adapters	Reduced	Reduced	Reduced	Finetuned
TriviaQA	Micro F-1	63.9 (0.5)	65.5 (0.1)	65.7 (0.1)	64.1 (0.2)	61.4 (0.1)	66.0 (0.1)
	Macro F-1	67.4 (0.5)	68.9 (0.1)	68.9 (0.1)	67.4 (0.1)	64.8 (0.1)	69.1 (0.1)
SQuAD	Micro F-1	80.2 (0.1)	80.2 (0.0)	80.8 (0.1)	79.5 (0.1)	75.4 (0.1)	81.1 (0.1)
	Macro F-1	87.9 (0.1)	87.9 (0.0)	88.4 (0.1)	87.5 (0.1)	84.8 (0.1)	88.5 (0.0)
QA Average Scores		74.9	75.6	76.0	74.6	71.6	76.2

Table 10: BERT QA Results for Reduced Models. **Bold** indicates the best average score between the standard reduced, adapter-based reduced, and fully fine-tuned versions of each model. **Reduced + Half Adpt** indicates adapters on the transformer layers of a fully frozen reduced model, where the earlier half of transformer layers were removed and their activations cached. **Full Adapters** indicates adapters on all transformer layers of a fully frozen model. Each score represents the average score of 10 runs, with the standard errors for each score in parentheses.

DeBERTaV2 XL							
		Reduced + Full	Full	6 Layers	12 Layers	18 Layers	Fully
		Half Adpt	Adapters	Reduced	Reduced	Reduced	Finetuned
ChemProt	Micro F-1	87.2 (0.1)	86.5 (0.2)	87.2 (0.2)	86.8 (0.4)	86.4 (0.2)	86.7 (0.9)
	Macro F-1	56.7 (0.5)	55.6 (0.6)	59.6 (0.2)	59.5 (0.5)	59.2 (0.3)	59.0 (1.1)
SciCite	Micro F-1	85.8 (0.4)	86.4 (0.4)	86.0 (0.1)	86.3 (0.2)	86.2 (0.3)	85.9 (0.2)
	Macro F-1	84.6 (0.4)	85.0 (0.5)	84.6 (0.1)	85.2 (0.1)	85.0 (0.3)	84.4 (0.2)
SciERC-Rel	Micro F-1	88.6 (0.5)	88.0 (0.4)	88.3 (0.2)	87.5 (0.1)	86.6 (0.3)	88.0 (0.4)
	Macro F-1	82.9 (0.8)	82.1 (0.8)	80.5 (0.5)	79.9 (0.3)	78.0 (0.4)	80.2 (0.5)
Text Classification Average Score		81.0	80.6	81.0	80.9	80.2	80.7
bc5cdr	Micro F-1	97.6 (0.0)	97.7 (0.0)	97.4 (0.3)	97.7 (0.0)	97.6 (0.0)	97.9 (0.0)
	Macro F-1	90.7 (0.1)	91.1 (0.1)	89.5 (1.4)	91.3 (0.0)	90.9 (0.0)	91.8 (0.1)
JNLPBA	Micro F-1	93.6 (0.0)	93.4 (0.0)	93.7 (0.1)	93.7 (0.0)	93.6 (0.0)	93.7 (0.0)
	Macro F-1	79.3 (0.1)	79.0 (0.1)	78.5 (0.3)	78.5 (0.2)	77.8 (0.1)	78.2 (0.1)
NCBI-disease	Micro F-1	98.3 (0.0)	98.4 (0.0)	98.6 (0.0)	98.6 (0.0)	98.5 (0.0)	98.6 (0.0)
	Macro F-1	93.3 (0.1)	93.5 (0.2)	93.1 (0.1)	93.3 (0.1)	92.8 (0.1)	93.4 (0.1)
NER Average Score		92.1	92.2	91.8	92.2	91.9	92.3
TriviaQA	Micro F-1	78.6 (0.2)	79.1 (0.2)	77.9 (0.2)	77.4 (0.2)	77.0 (0.2)	78.5 (0.1)
	Macro F-1	81.6 (0.1)	82.3 (0.2)	81.2 (0.1)	80.6 (0.1)	80.1 (0.2)	81.8 (0.1)
SQuAD	Micro F-1	88.6 (0.0)	87.2 (0.1)	88.6 (0.1)	88.7 (0.0)	87.1 (0.0)	88.5 (0.1)
	Macro F-1	94.7 (0.0)	93.9 (0.0)	94.6 (0.0)	94.5 (0.0)	93.5 (0.0)	94.6 (0.0)
QA Average Score		85.9	85.6	85.6	85.3	84.4	85.8

Table 11: DeBERTaV2-XL Results for Reduced Models. **Bold** indicates the best average score between the standard reduced, adapter-based reduced, and fully fine-tuned versions of each model. **Reduced + Half Adpt** indicates adapters on the transformer layers of a fully frozen reduced model, where the earlier half of transformer layers were removed and their activations cached. **Full Adapters** indicates adapters on all transformer layers of a fully frozen model. Each score represents the average score of 5 runs, with the standard errors for each score in parentheses.

T5 Large							
		Reduced + Half Adpt	Full Adapters	6 Layers Frozen	12 Layers Reduced	18 Layers Reduced	Fully Finetuned
ChemProt	Micro F-1	84.3 (0.6)	84.9 (0.6)	84.7 (0.6)	84.6 (0.6)	85.0 (0.1)	84.1 (0.8)
	Macro F-1	57.2 (0.7)	58.0 (0.8)	56.2 (0.7)	56.2 (0.7)	57.4 (0.1)	56.1 (0.7)
SciCite	Micro F-1	86.7 (0.3)	86.2 (0.3)	87.4 (0.2)	87.6 (0.1)	88.0 (0.2)	86.4 (0.2)
	Macro F-1	85.3 (0.4)	84.5 (0.4)	86.0 (0.2)	86.3 (0.2)	86.9 (0.2)	84.9 (0.2)
SciERC-Rel	Micro F-1	85.6 (0.4)	85.2 (0.1)	84.3 (0.3)	86.8 (0.4)	83.4 (0.7)	87.4 (0.5)
	Macro F-1	76.2 (1.0)	75.6 (0.2)	73.6 (0.9)	77.4 (0.7)	72.2 (1.0)	80.2 (1.1)
Text Classification Average Score		79.2	79.1	78.7	79.8	78.8	79.9
bc5cdr	Micro F-1	93.8 (0.6)	95.7 (0.7)	97.7 (0.7)	97.4 (0.3)	95.4 (0.8)	97.5 (0.2)
	Macro F-1	79.9 (1.0)	85.7 (1.1)	91.1 (0.5)	90.7 (1.1)	89.3 (1.0)	89.9 (0.8)
JNLPBA	Micro F-1	93.9 (0.4)	93.8 (0.1)	93.8 (0.0)	94.0 (0.0)	93.9 (0.0)	94.2 (0.0)
	Macro F-1	78.8 (0.6)	79.5 (0.2)	78.8 (0.1)	79.6 (0.1)	79.3 (0.0)	80.0 (0.0)
NCBI-disease	Micro F-1	97.8 (0.0)	98.5 (0.0)	98.5 (0.0)	98.5 (0.0)	98.4 (0.0)	98.6 (0.0)
	Macro F-1	92.1 (0.2)	92.5 (0.2)	93.1 (0.1)	92.8 (0.0)	92.2 (0.1)	93.5 (0.0)
NER Average Score		89.4	90.9	92.2	92.2	91.4	92.3
TriviaQA	Micro F-1	68.2 (0.2)	68.8 (0.2)	67.0 (0.0)	66.9 (0.0)	63.9 (0.0)	68.7 (0.0)
	Macro F-1	77.0 (0.1)	77.5 (0.1)	77.5 (0.0)	77.3 (0.0)	74.8 (0.0)	78.0 (0.0)
SQuAD	Micro F-1	81.2 (0.1)	82.0 (0.1)	86.6 (0.1)	86.3 (0.6)	85.2 (0.4)	86.7 (0.4)
	Macro F-1	90.6 (0.1)	91.0 (0.1)	93.8 (0.0)	93.7 (0.3)	92.8 (0.2)	93.9 (0.3)
QA Average Score		79.2	79.8	81.2	81.0	79.2	81.8

Table 12: T5 Large Results for Reduced Models. **Bold** indicates the best average score between the standard reduced, adapter-based reduced, and fully fine-tuned versions of each model. **Reduced + Half Adpt** indicates adapters on the encoder and decoder transformer layers of a fully frozen reduced model, where the earlier half of the encoder layers were removed and their activations cached. **Full Adapters** indicates adapters on all encoder and decoder transformer layers of a fully frozen model. Each score represents the average score of 5 runs, with the standard errors for each score in parentheses.

		DistilBERT			
		2 Layers Reduced	3 Layers Reduced	4 Layers Reduced	Fully Fine-tuned
ChemProt	Micro F-1	79.1 (0.4)	80.3 (0.1)	79.0 (0.2)	79.1 (0.5)
	Macro F-1	52.1 (0.5)	51.6 (0.6)	51.6 (0.4)	52.6 (0.3)
SciCite	Micro F-1	85.7 (0.1)	85.6 (0.1)	85.8 (0.1)	85.5 (0.1)
	Macro F-1	84.3 (0.1)	84.1 (0.1)	84.2 (0.1)	84.0 (0.1)
SciERC-Rel	Micro F-1	84.3 (0.3)	84.5 (0.3)	84.6 (0.2)	83.5 (0.4)
	Macro F-1	74.1 (0.7)	74.9 (0.7)	74.6 (0.4)	72.9 (0.7)
Text Classification Average Score		76.6	76.8	76.6	76.3
bc5cdr	Micro F-1	97.0 (0.0)	97.0 (0.0)	96.9 (0.0)	97.2 (0.0)
	Macro F-1	88.3 (0.0)	88.3 (0.1)	87.9 (0.0)	88.7 (0.1)
JNLPBA	Micro F-1	93.4 (0.1)	93.5 (0.0)	93.4 (0.0)	93.5 (0.0)
	Macro F-1	78.0 (0.3)	78.6 (0.1)	77.9 (0.1)	78.5 (0.1)
NCBI-disease	Micro F-1	98.2 (0.0)	98.0 (0.0)	98.1 (0.0)	98.2 (0.0)
	Macro F-1	91.4 (0.1)	90.5 (0.1)	90.7 (0.1)	91.3 (0.1)
NER Average Score		91.1	91	90.8	91.2
TriviaQA	Micro F-1	62.9 (0.1)	61.4 (0.1)	59.1 (0.1)	63.6 (0.1)
	Macro F-1	66.2 (0.1)	64.7 (0.1)	62.4 (0.1)	66.8 (0.1)
SQuAD	Micro F-1	76.6 (0.1)	76.3 (0.1)	72.5 (0.1)	77.1 (0.1)
	Macro F-1	85.1 (0.1)	84.8 (0.0)	82.3 (0.1)	85.4 (0.0)
QA Average Score		72.7	71.8	69.1	73.2

Table 13: DistilBERT Results for Reduced Models. **Bold** indicates the best average score between the reduced and fully fine-tuned versions of each model. Each score represents the average score of 10 runs, with the standard errors for each score in parentheses.

MiniLM: 6L-H768					
		2 Layers Reduced	3 Layers Reduced	4 Layers Reduced	Fully Fine-tuned
ChemProt	Micro F-1	79.4 (0.3)	78.3 (0.4)	79.0 (0.2)	79.3 (0.3)
	Macro F-1	51.8 (0.4)	50.6 (0.4)	52.0 (0.2)	52.6 (0.4)
SciCite	Micro F-1	85.4 (0.1)	85.8 (0.2)	85.9 (0.1)	86.0 (0.2)
	Macro F-1	84.1 (0.2)	84.5 (0.2)	84.5 (0.1)	84.6 (0.2)
SciERC-Rel	Micro F-1	84.7 (0.3)	83.9 (0.3)	84.1 (0.4)	86.3 (0.2)
	Macro F-1	75.0 (0.4)	74.8 (0.4)	75.3 (0.6)	78.2 (0.6)
Text Classification Average Score		76.7	76.3	76.8	77.8
bc5cdr	Micro F-1	96.1 (0.3)	96.8 (0.0)	96.6 (0.0)	96.8 (0.2)
	Macro F-1	84.6 (1.1)	87.8 (0.1)	86.6 (0.0)	87.5 (1.0)
JNLPBA	Micro F-1	93.2 (0.0)	93.2 (0.0)	93.3 (0.0)	93.3 (0.0)
	Macro F-1	77.5 (0.1)	77.3 (0.1)	77.3 (0.1)	76.9 (0.2)
NCBI-disease	Micro F-1	98.3 (0.0)	98.2 (0.0)	98.2 (0.0)	98.3 (0.0)
	Macro F-1	92.1 (0.1)	91.1 (0.1)	91.0 (0.1)	92.1 (0.1)
NER Average Score		90.3	90.7	90.5	90.8
TriviaQA	Micro F-1	70.2 (0.1)	68.9 (0.1)	65.5 (0.1)	70.4 (0.2)
	Macro F-1	73.4 (0.1)	72.2 (0.1)	68.9 (0.1)	73.8 (0.2)
SQuAD	Micro F-1	77.6 (0.1)	75.6 (0.1)	65.4 (0.2)	78.9 (0.1)
	Macro F-1	86.4 (0.1)	85.0 (0.1)	77.0 (0.1)	87.0 (0.1)
QA Average Score		76.9	75.4	69.2	77.5

Table 14: MiniLM L6-H768 Results for Reduced Models. **Bold** indicates the best average score between the reduced and fully fine-tuned versions of each model. Each score represents the average score of 10 runs, with the standard errors for each score in parentheses.

MiniLM: L6-H384					
		2 Layers Reduced	3 Layers Reduced	4 Layers Reduced	Fully Fine-tuned
ChemProt	Micro F-1	75.4 (0.5)	76.9 (0.2)	74.9 (0.3)	74.6 (0.4)
	Macro F-1	47.3 (0.7)	50.4 (0.2)	48.8 (0.4)	47.1 (0.8)
SciCite	Micro F-1	84.4 (0.1)	85.4 (0.1)	85.1 (0.1)	84.4 (0.1)
	Macro F-1	82.8 (0.1)	83.7 (0.1)	83.4 (0.1)	82.8 (0.1)
SciERC-Rel	Micro F-1	83.2 (0.3)	82.6 (0.3)	83.3 (0.2)	79.5 (0.9)
	Macro F-1	72.7 (0.6)	72.1 (0.6)	73.7 (0.3)	68.9 (1.1)
Text Classification Average Score		74.3	75.2	74.9	72.9
bc5cdr	Micro F-1	96.6 (0.0)	96.3 (0.0)	95.6 (0.0)	96.9 (0.0)
	Macro F-1	86.9 (0.1)	85.9 (0.1)	83.2 (0.1)	88.3 (0.1)
JNLPBA	Micro F-1	93.0 (0.0)	92.2 (0.0)	92.0 (0.0)	93.3 (0.0)
	Macro F-1	76.3 (0.1)	74.0 (0.1)	73.6 (0.1)	77.2 (0.1)
NCBI-disease	Micro F-1	98.0 (0.0)	97.9 (0.0)	97.7 (0.0)	98.2 (0.0)
	Macro F-1	90.6 (0.1)	89.9 (0.1)	88.9 (0.1)	91.7 (0.1)
NER Average Score		90.2	89.4	88.5	90.9
TriviaQA	Micro F-1	66.6 (0.1)	65.6 (0.1)	63.4 (0.1)	67.6 (0.2)
	Macro F-1	69.9 (0.1)	69.2 (0.1)	67.0 (0.1)	71.0 (0.2)
SQuAD	Micro F-1	81.6 (0.0)	80.9 (0.1)	74.2 (0.2)	81.6 (0.1)
	Macro F-1	89.7 (0.0)	89.0 (0.0)	84.5 (0.1)	89.6 (0.0)
QA Average Score		76.9	76.2	72.3	77.4

Table 15: MiniLM L6-H384 Results for Reduced Models. **Bold** indicates the best average score between the reduced and fully fine-tuned versions of each model. Each score represents the average score of 10 runs, with the standard errors for each score in parentheses.

Task	Averages	Standard Recycling	Adapter-Based Recycling
Classification	Training Time	2204	2349
	Epochs	38	42
NER	Training Time	4269	3857
	Epochs	43	39
QA	Training Time	8252	8513
	Epochs	6	7

Table 16: Average Training Times and Epochs for Embedding Recycling (seconds for training time, count for epochs). **Standard Recycling** corresponds to layer recycling on a reduced transformer model. **Adapter-Based Recycling** corresponds to layer recycling on a reduced frozen transformer model with added trainable Adapter modules. Training time and epoch averages are the averages across the RoBERTa, BERT, SciBERT, DeBERTa V2 XL, and T5-Large transformer models and the text classification, NER, and QA datasets tested.

Trained on 100 million words and still in shape: BERT meets British National Corpus

David Samuel, Andrey Kutuzov, Lilja Øvrelid and Erik Veldal

University of Oslo, Language Technology Group

{davisamu, andreku, liljao, erikve}@ifi.uio.no

Abstract

While modern masked language models (LMs) are trained on ever larger corpora, we here explore the effects of down-scaling training to a modestly-sized but representative, well-balanced, and publicly available English text source – the British National Corpus. We show that pre-training on this carefully curated corpus can reach better performance than the original BERT model. We argue that this type of corpora has great potential as a language modeling benchmark. To showcase this potential, we present fair, reproducible and data-efficient comparative studies of LMs, in which we evaluate several training objectives and model architectures and replicate previous empirical results in a systematic way. We propose an optimized LM architecture called LTG-BERT.

1 Introduction

In the pursuit of state-of-the-art performance, NLP practitioners utilize increasingly larger amounts of data to pre-train language models, making it difficult to disentangle the improvements made by the proposed modeling choices themselves. Instead, our aim is to shift the focus towards more efficient language modeling on a small and standardizable pre-training corpus. We study the data efficiency of current language models on an openly available corpus of approximately 100M words – incidentally the estimated amount of words processed by humans before adulthood (Linzen, 2020).

The goal of this paper is not to rival the paradigm of ‘massively pre-trained language models’; instead we would in this work like to pursue a complementary direction of language modeling, which will hopefully lead to more interest in data-efficient language models. In particular, our contribution in this paper is twofold – we show that:

1. 100M words is enough to train a competitive language model that outperforms the downstream performance of the original BERT model. We show

that the combination of a well-curated representative corpus, improved LTG-BERT architecture and a better training objective results in a model with stronger linguistic knowledge than the original English BERT pre-trained on 30× larger corpus.

Large language models are notoriously data hungry, requiring hundreds of gigabytes of raw textual data. This becomes a major obstacle for low-resource languages while also putting a limit to the efficiency of any ‘efficient’ language model. On top of that, the size of web-crawled corpora makes it almost impossible to control their content and to prevent learning from harmful or copyrighted text (Bender et al., 2021). The British National Corpus (BNC; Consortium, 2007) is a 100-million-word reference corpus, manually curated to cover most aspects of 20th century British English.

2. Reproducibility and fair comparison of language models can be easily achieved by pre-training on the British National Corpus.

Massive language models are often pre-trained on nonpublic filtered collections of web-crawled text, which makes any reproduction impossible. We pre-train our models on a small and publicly available corpus, which allows for a replicable comparison of different language modeling configurations and which can be easily utilized in future research of novel variants of language models. We also release the pre-processing scripts, training scripts as well as the final model checkpoints.¹

Previously, language models have been pre-trained on different corpora tokenized by different tokenizers and fine-tuned by increasingly complex learning methods. This makes any comparison of the underlying neural architectures and pre-training objectives unfair. We make the language models in this paper directly comparable by fixing the training corpus, the tokenizer and the evaluation methods, while keeping them as simple as possible.

¹<https://github.com/ltgoslo/ltg-bert>

2 Related Work

The data requirements of language models have been growing in orders of magnitude since their early stages (Jelinek, 1976). Taking a huge leap towards more recent work, ELMo (Embeddings from Language Models; Peters et al., 2018) were the first to introduce deep *contextualized* embeddings of words. Recognizing the need of a large text corpus for this task, ELMo was trained on the 1B Word Benchmark (Chelba et al., 2014). Later, BERT (Bidirectional Encoder Representations from Transformers; Devlin et al., 2019) further advanced the performance of contextualized embeddings when it based the entire language model on the Transformer architecture (Vaswani et al., 2017). Another important aspect of BERT is that it was trained on a larger corpus than ELMo: about 3.3B words from crawled English Wikipedia and BookCorpus (Zhu et al., 2015). To our best knowledge, the exact version of neither of the two subcorpora is publicly available.² The issue of limited replicability has become even more pronounced with later large language models: XLNet (Yang et al., 2019) was trained on 33B words, RoBERTa (Liu et al., 2019) on more than 30B words and GPT-3 (Brown et al., 2020) on an approximately 400B word corpus. None of these datasets is available; the authors utilize non-trivial filtering algorithms on openly available web crawls but do not release the end product nor the filtering source code.

The effect of corpus size has been thoroughly studied in Zhang et al. (2021) as well as in Hoffmann et al. (2022). They test differently sized random subsets of a BERT-like corpus (crawled Wikipedia and Smashwords) and of a massive web-crawled text corpus (MassiveText; Rae et al., 2021), respectively. Unlike them, we evaluate the effect of training on a small corpus, which was *carefully curated* to create a representative sample of English. The British National Corpus is arguably more diverse and informative than a random subset of a web crawl – hence we test how the *quality* of a pre-training corpus influences the downstream performance, not only how the data quantity matters. We believe this aspect is vital for the future research of effective and reliable language models.

²BookCorpus (Zhu et al., 2015) is not available anymore and the authors of BERT do not specify what version of Wikipedia dump they used or how did they preprocess it (<https://github.com/google-research/bert#pre-training-data>).

	documents	sentences	words	subwords
train	4 014	8 501 376	115 870 549	131 392 103
development	35	106 566	1 215 306	1 367 570

Table 1: Size of the train-development splits for the pre-processed BNC corpus. Note that the number of words is larger than the 100 million reported by the BNC Consortium due to our less conservative pre-tokenization strategy.

3 British National Corpus

We use the British National Corpus (BNC) as a diverse, balanced, compact, and publicly available monolingual English corpus. BNC is comprised of both written and spoken language with a total of 100 million words. The manually curated content contains a wide range of British English from the late 20th century – newspapers, journals, books (academic and fiction), letters, essays, unscripted informal conversations or transcribed business meetings, radio shows or phone calls. The written part makes up approximately 90% of the corpus and the remaining 10% contains the transcribed speech. The sources are truncated to contain at most 45 000 words to ensure greater diversity within the limited amount of 100 million words.

Creation. The process of creating the BNC is extensively described in its documentation on the website.³ It was created by the so called ‘BNC Consortium’ led by Oxford University Press, and including major dictionary publishers Longman and Larousse Kingfisher Chambers; academic research centres at Oxford University Computing Services, the University Centre for Computer Corpus Research on Language (UCREL) at Lancaster University, and the British Library’s Research and Innovation Centre. The purpose of the British National Corpus project was to construct a balanced and representative sample of current British English at the time. It was created over a period of four years and was a result of careful planning and data selection across a number of selection criteria (domain, time, medium, level) with proportions in the corpus designed to reflect the proportions found in real language use. It is widely acknowledged that the BNC has been a major influence on the construction of language corpora (Burnard, 2002). One downside of the BNC is that it does not re-

³<https://ota.bodleian.ox.ac.uk/repository/xmlui/handle/20.500.12024/2554>

flect anything occurring to English language and the world in the 21st century, but still no better alternatives of the same size and quality exists. In addition, BNC was used as a model for creating representative corpora for other languages: e.g., Turkish (Aksan et al., 2012).

Version. We use the third release of the corpus, BNC XML Edition (2007), which is the final revision of the texts compiled from 1991 to 1994 (Consortium, 2007). The XML edition did not get any additional content on top of the original text samples, but it got some minor corrections, more metadata and it is supplied in a convenient XML format.

3.1 Preprocessing

We convert the XML version of BNC into the Markdown format,⁴ to make it human-readable and usable as a direct raw-text input of a language model. On top of that, it can also preserve some meta-information encoded in the original XML format. Short samples from the preprocessed corpus can be found in Appendix A. After preprocessing, the articles are randomly placed into a training split and a development split. The proportions of both splits are given in Table 1.

Composition. BNC is hierarchically composed of the following text units: words, sentences, paragraphs and articles. We preserve the sentence information by storing each sentence on a separate line; paragraphs are divided by a blank line and an article always starts with a top-level header. The word-tokens are intentionally not preserved – instead, we heuristically detokenize the text to move it towards the natural text distribution. BNC includes information about the original whitespace, but we found it unreliable in some cases, necessitating the use of heuristics.

Other metadata. Other meta information available in our Markdown version is as follows:

1. **Headers:** We keep the headers together with their level by converting them to the atx-style format prefixed by hash symbols ‘#’.
2. **Speakers:** The spoken part of BNC is divided into speech turns, each accompanied

⁴<https://daringfireball.net/projects/markdown/>

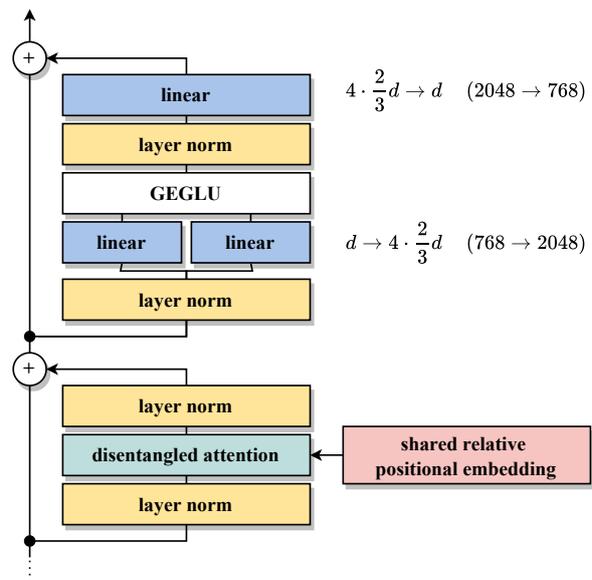


Figure 1: A simplified diagram of one layer in our LTG-BERT language model, which illustrates the changes made to the standard Transformer architecture – NormFormer layer normalization, GEGLU activation function and disentangled attention.

by a speaker identifier. We maintain this information by formatting each speech turn as ‘{name} : _␣ {turn}’.

3. **Quotes:** Markdown also allows us to keep the special quoted text by using a prefix ‘> _␣’.
4. **Lists:** The XML format contains special tags for lists and their respective elements, we use the ‘- _␣{element}’ notation to encode these text blocks.
5. **Incomprehensible speech:** Some words or phrases could not be transcribed because they were illegible or inaudible. Since completely omitting such text would result in ungrammatical sentences, we mark these segments with a special ‘[UNK]’ token.

Not all of this additional information is of use for the language models tested in this article, but it can be easily filtered out when needed. We preserve it to make this corpus more versatile.

4 Model architecture

We slightly depart from the typical *post-norm* Transformer architecture (Vaswani et al., 2017) used by BERT (Devlin et al., 2019), as illustrated in Figure 1. Preliminary experiments with this model showed that it tends to unpredictably diverge in

the later stages of training. This behavior has been noted in previous work on large LMs (Liu et al., 2020) and accordingly, we follow some of the recent improvements of Transformer.

NormFormer. *Pre-norm* variation of the Transformer has been shown to lead to more stable convergence with slightly degraded performance (Nguyen and Salazar, 2019). Shleifer and Ott (2022) claimed to mitigate this degradation by introducing an additional layer normalization operation. For these reasons, we decided to use their so-called *NormFormer* architecture to stabilize the training.⁵

GEGLU activation function, proposed in Shazeer (2020), enhances the expressiveness of the original Transformer feed-forward modules by redefining them as

$$\text{FF}_{\text{GEGLU}}(\mathbf{x}) = (\text{GELU}(\mathbf{x}\mathbf{W}_1) \odot \mathbf{x}\mathbf{W}_2) \mathbf{W}_3,$$

where \mathbf{W}_i are weight matrices⁶ and GELU is the Gaussian Error Linear Unit (Hendrycks and Gimpel, 2016). Note that this formulation involves three linear transformations instead of two, we therefore lower the intermediate hidden size by $2/3$ to keep the number of parameters the same.

Disentangled attention. The original Transformer formulation (Vaswani et al., 2017) fuses the content and positional information together in the first embedding layer and calculates the (un-normalized) attention score between each pair of tokens \mathbf{x}_i and \mathbf{x}_j as

$$A_{i,j} = \frac{\mathbf{Q}_i \mathbf{K}_j^\top}{\sqrt{d}},$$

where \mathbf{Q} and \mathbf{K} are the query-key linear transformations of \mathbf{x} .

He et al. (2021) proposed to *disentangle* the content and positional information. The content representations are incrementally built by the Transformer layers and the position is encoded by one shared relative positional embedding matrix $P \in \mathbb{R}^{(2L-1) \times d}$, where L is the maximal input length.⁷ This is supposed to offer greater expressivity as each layer can access these two parts directly. The attention scores are then calculated as a sum of

⁵They also proposed some additional improvements – *head scaling* and *residual scaling*, but we did not experience any performance benefits from these changes.

⁶The bias terms are omitted for brevity.

⁷Tokens at positions i and j have relative positional embedding at the $(L - i + j)$ th row of P , denoted as $P_{i,j}$.

three distinct parts: *content-to-content*, *content-to-position* and *position-to-content* attention – formally, the attention scores are defined as

$$A_{i,j} = \frac{{}^c\mathbf{Q}_i {}^c\mathbf{K}_j^\top + {}^c\mathbf{Q}_i {}^p\mathbf{K}_{i,j}^\top + {}^p\mathbf{Q}_{j,i} {}^c\mathbf{K}_j^\top}{\sqrt{3d}},$$

where ${}^c\mathbf{Q}$ and ${}^c\mathbf{K}$ are linear transformations of the *content* vectors and ${}^p\mathbf{Q}$ and ${}^p\mathbf{K}$ are linear transformations of the relative *positional* embedding $P_{i,j}$. We share the parameters of the content and positional transformations, ${}^c\mathbf{Q} = {}^p\mathbf{Q}$ and ${}^c\mathbf{K} = {}^p\mathbf{K}$, to not increase the model size while achieving comparable performance (He et al., 2021).

Initialization scaling. Bajaj et al. (2022) found that we can further stabilize the Transformer architecture by gradually scaling down its feed-forward (FF) weight matrices. Following Nguyen and Salazar (2019), we first initialize all weight matrices \mathbf{W} by sampling from:

$$\mathbf{W}_{i,j} \sim \mathcal{N}\left(0, \sqrt{\frac{2}{d+4d}}\right),$$

where d is the hidden dimension.⁸ Then all three weight matrices in a FF module at layer l are scaled down by a factor of $1/\sqrt{2^{l+1}}$.

5 Training objectives

The fixed corpus, tokenizer and fine-tuning procedures establish a controlled test bed for a comparative study of training objectives proposed in the past. The original BERT model is trained via two self-supervised training objectives – masked language modeling (MLM) and next sentence prediction (NSP). We evaluate five different configurations of these objectives (three for MLM and two for NSP), as further detailed below.

5.1 Masked language modeling (MLM)

Unlike the traditional auto-regressive language models, the *Bidirectional* Encoder Representations from Transformers (BERT) learn a *bidirectional* contextualized representation for each token in a text segment. This is done by randomly selecting 15% of subword tokens (excluding the special tokens). Out of these, 80% are masked, 10% randomly replaced and 10% are left untouched. The

⁸This formula is roughly equal to the universal BERT initialization range of 0.02 for $d = 1024$.

language model is then trained to jointly predict the original state of the selected units. We investigate three common choices of the masked text units:

1. **Subwords.** As proposed in the seminal work by Devlin et al. (2019), every subword is masked independently with 15% probability to model its bidirectional dependencies.
2. **Whole words.** This method was also implemented by Devlin et al. (2019), after the publication of their original paper with subword masking. The motivation for this approach is that partially masked multi-subword word units are often easily decoded without any need for non-local contextual information; masking the whole multi-subword unit should force the model to build longer-range non-local dependencies.
3. **Spans.** The third method further follows the direction of whole-word masking and generalizes it to masking of random *spans* of subwords. More specifically, SpanBERT (Joshi et al., 2020) iteratively samples random spans until 15% of subwords are masked. For each span, it first samples its length from $\text{Geo}(p)$, where $p = 1/3$.⁹ Then the starting subword of the masked span is chosen from a uniform distribution.

5.2 Next sentence prediction (NSP)

Masked language modeling is a token-level training objective that trains the model to learn rich token representations. Yet, some downstream tasks need a single sentence-level representation instead. To also learn these, researchers have designed a number of additional semi-supervised training objectives. On the other hand, Liu et al. (2019) argue that NSP objectives do not help the downstream performance and they can thus be dropped in favour of a simpler optimization process with a single MLM training objective. To test these hypotheses, we experiment with two NSP objectives:

1. **Document discrimination.** Devlin et al. (2019) sample two text segments and then train the model with a second discriminative loss function, which predicts whether the two segments are continual or randomly taken from two different documents.

⁹To ensure that the sampled length is not too large, we take the sampled value modulo 10. The expected length of a masked span is then approximately equal to 2 with $p = 1/3$.

2. **Sentence-order discrimination.** Lan et al. (2020) argue that the document discrimination is too easy as the language models only have to compare the topic of the two segments to achieve a good performance in this task. Instead, they propose to predict whether the two segments are in the correct order or whether they are swapped. Thus, the sentence-order loss forces the neural network to model inter-sentence coherence and this is believed to lead to a better downstream performance.

6 Evaluation metrics

We use three conceptually different methods for evaluating the amount of linguistic knowledge acquired by the BNC language models. 1) The (Super)GLUE datasets test the ability of the model to adapt to various NLU tasks by further optimizing the whole pre-trained model, 2) edge probing tasks evaluate how much linguistic information one can extract from a frozen pre-trained model and 3) BLiMP utilizes the intrinsic ability of the pre-trained network to model language and probes its knowledge without any additional training. We further elaborate on each of these below.

6.1 (Super)GLUE

GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) have become a de-facto standard for evaluating the language understanding capabilities of language models. Accordingly, we also choose to fine-tune our language models on these NLU tasks to measure their linguistic and transfer-learning performance. We give more technical details about our implementation of (Super)GLUE fine-tuning in Appendix B.1.

We exclude the Winograd schema datasets, WNLI and WSC, because they require a complete reformulation to get past the trivial most-frequent baseline (Kocijan et al., 2019). The remaining 14 (Super)GLUE datasets measure performance on these tasks:

- **Inference:** CB, MNLI, QNLI, RTE.
- **Linguistic acceptability:** CoLA.
- **Sentiment analysis:** SST-2.
- **Semantic similarity:** MRPC, QQP, STS-B.
- **Word sense disambiguation:** WiC.
- **Question answering:** BoolQ, COPA, MultiRC, ReCoRD.

6.1.1 HANS

Deep learning systems are (by design) prone to finding spurious correlations in the training data. These heuristics can often be successfully employed for the evaluation data, as well – thus, one has to be careful when implying that a higher score on a benchmark shows a deeper understanding of the tested model. McCoy et al. (2019) tried to evaluate to what extent language models rely on spurious heuristics to solve NLI tasks. They identified a set of fallible syntactic heuristics and designed a test set where these ‘shortcuts’ should fail – Heuristic Analysis for NLI Systems (HANS). We adopt their approach and test models that have been fine-tuned on MNLI.

6.2 Edge probing

GLUE tasks measure the ability of a LM to be fine-tuned on a sentence-level NLU problem. To get a more comprehensive picture of LM performance, one can also *probe* the word-level contextualized representations, measuring how much syntactic or semantic information can be extracted.

Tenney et al. (2019) devised a simple approach of probing for a diverse set of linguistic phenomena called *edge probing*. They reformulate traditional NLP tasks as *span classification*: part-of-speech tagging can be viewed as classification of word-spans and semantic role labeling becomes a classification of pairs of spans: predicate-span and argument-span. In the following, we will probe our models for five basic tasks: part-of-speech tagging (POS), dependency parsing (DP), semantic role labeling (SRL), named-entity recognition (NER) and coreference resolution (CR). Note that the model only learns to *classify* each span provided to the model as gold data. This substantially simplifies some of the tasks, for example SRL. Please refer to Appendix B.2 for the implementation details of edge probing.

6.3 BLiMP

One disadvantage of the aforementioned evaluation metrics is that the results are skewed by the second-stage supervised training, which makes it problematic to disentangle the prior knowledge of a language model from the acquired knowledge (Belinkov, 2022). In contrast, the Benchmark of Linguistic Minimal Pairs (BLiMP; Warstadt et al., 2020) attempts to measure the linguistic knowledge of a language model in a zero-shot manner – with

out any additional training. The dataset consists of 67 000 sentence pairs; each pair differs minimally on the surface level, but only one of the sentences is grammatically valid. We can use the intrinsic ability of language models to assign a probability to every sentence and test how often a language model assigns a higher probability to the correct sentence. Appendix B.3 gives more details about ranking the likelihood of sentences according to the raw output of a masked language model.

7 Experiments

We conduct a number of experiments in this section. First, we compare different training hyperparameters and model configurations described in Section 4. Then, using the overall best training setting, we make a comparative study of training objectives (Section 5). Finally, we investigate the sampling efficiency of our proposed language model and we compare BNC with a Wikipedia & BookCorpus subset of the same size. These results can then be used as a baseline performance of BNC-BERT in future studies.

The central model used in the experiments is a *base-sized* Transformer – 12 encoder layers with hidden size 768 and 12 attention heads (more details in Appendix E). All reported models utilize the same cased WordPiece tokenizer (Wu et al., 2016) with a vocabulary size of $2^{14} = 16\,384$ trained with the BNC dataset (Appendix G). This goes against the trend of increasing the subword vocabulary in recent work,¹⁰ but a larger vocabulary size would lead to a lot of infrequent tokens within our limited corpus – we roughly follow Gowda and May (2020) and ‘... use the largest possible BPE vocabulary such that at least 95% of classes have 100 or more examples in training.’

Since our aim is to train models comparable to BERT_{base}, we train for the same amount of sampled tokens. Devlin et al. (2019) trained on 1M batches of 128K tokens, we use 31 250 training steps with batch size of 4M tokens to parallelize and accelerate the process. Also, similarly to Devlin et al. (2019), we use sequence length of 128 tokens in the first 90% of training and a larger sequence length of 512 only in the last 10% of steps. We deliberately do not compare against more recent models, which are trained for much longer to achieve slightly bet-

¹⁰ BERT (Devlin et al., 2019) uses 28 996 tokens, RoBERTa (Liu et al., 2019) 50 265 and in 2021, DeBERTa (He et al., 2021) used a vocabulary of 128 100 subwords.

ter performance: RoBERTa is trained on $16\times$ more training samples, for example.¹¹

7.1 Comparison of model architectures and training settings

In order to establish a strong baseline, we evaluate the proposed changes from Section 4 and other training configurations. We present the results in Table 2, where we compare the final model with all changes applied and models with one of those modifications removed. These training choices turned out to be the most important:

- Both the post-norm and pre-norm transformer variants perform substantially worse than the NormFormer-like layer normalization (Shleifer and Ott, 2022). Both of them also lead to less stable and only slightly faster training.
- Absolute positional embeddings seem to be less adaptable for fine-tuning but perform better on language modeling itself, as can be seen on the BLiMP results. We hypothesize that this is caused by more accurate estimation of probabilities of the first few words in a sentence. The simpler absolute embeddings also lead to the greatest reduction of training time. We choose the slower relative positional embeddings despite this fact to increase the performance on (Super)GLUE tasks.
- We observe that setting the weight decay correctly is crucial for masked language modeling. The default weight decay value found in Devlin et al. (2019), 0.01, performs substantially worse on all tested tasks. We use a higher decay value of 0.1 to boost performance, this value is most likely strongly correlated with the corpus size we use here. This suggests that previous findings of inferior performance of LMs pre-trained on small corpora might be caused by insufficient hyperparameter search.
- As expected, the AdamW optimizer (Loshchilov and Hutter, 2019) behaves poorly in our highly parallel training regime. Our study successfully replicates the reported performance of the LAMB optimizer (You et al., 2020), which we thus use in all other experiments.

¹¹ 500K steps with 8 192 segments of length 512, according to (He et al., 2021).

Model	MNLI	Edge probing	BLiMP	Training time
LTG-BERT	85.1 \pm 0.2	95.3 \pm 0.1	83.4	8h 13min
w/ post-norm (0.005)	-0.5 \pm 0.2	-0.6 \pm 0.1	-0.1	-22min
w/ pre-norm (0.005)	-1.3 \pm 0.1	-0.2 \pm 0.1	-0.9	-35min
w/ GELU activation	-0.3 \pm 0.3	0.0 \pm 0.1	-0.1	-6min
w/ absolute pos. emb.	-1.1 \pm 0.2	- 0.1 \pm 0.1	+0.6	-2h 16min
w/o FF init. scaling	-0.3 \pm 0.2	- 0.1 \pm 0.1	+0.1	0min
w/ learnt FF biases	-0.3 \pm 0.2	0.0 \pm 0.1	-0.1	+9min
w/ 0.01 WD (0.005)	-1.4 \pm 0.1	-0.2 \pm 0.1	-0.7	-1min
w/ linear schedule	-0.5 \pm 0.2	0.0 \pm 0.1	-0.2	0min
w/ AdamW (0.001)	-0.9 \pm 0.2	-0.2 \pm 0.1	-0.5	-11min

Table 2: Comparative study of different architectural and training settings. The first row shows the performance of the final model with all improvements applied and the following rows give the relative changes in performance when one of the changes is not applied – for example, the second row tests swapping the NormFormer-like normalization with the ‘post-norm’ normalization. Some runs diverged with the default learning rate of 0.01 and had to be run again with a lower value (denoted in parentheses). ‘WD’ stands weight decay and ‘FF’ is an abbreviation for the feed-forward modules. We report the mean and standard deviation statistics across five runs, if applicable, and boldface all run within 1 standard deviation from the best result.

The other changes bring more marginal gains – all three tested modifications of the feed-forward layers work slightly better: 1) using GEGLU activation function instead of GELU, 2) initializing the feed-forward layers with incrementally lower weight norms, and 3) not using any bias parameters in these layers. The last tested change shows that cosine learning rate decay (Rae et al., 2021) performs better than the standard linear weight decay.

7.2 Training objective comparison

Masked language modeling. First of all, we compare the three masking methods described in Section 5.1: subword, whole-word and span masking. The summary of the results is given in Table 3, more detailed evaluation in Appendix D. Overall, the span-based masking performs marginally better than the other methods – it shows a clear improvement on (Super)GLUE benchmarks over the simple subword masking, it generalizes the best according to the HANS score and it even matches the performance of BERT_{base} on the averaged BLiMP accuracy. All methods perform equally well on edge probing. Whole-word masking lacks on the BLiMP benchmark because the model is not expecting partially masked words that can occur in

Model (variant)	GLUE					HANS	Edge probing	BLiMP
	MNLI	MRPC	QNLI	SST-2	Average			
Wikipedia + BookCorpus (3000M words; Devlin et al., 2019)								
BERT _{base, cased} [†]	84.4	86.7	88.4	92.7	88.1	69.0	93.9	84.2
BERT _{base, cased} (our eval.)	83.6 \pm 0.2	84.6 \pm 0.5	90.8 \pm 0.1	91.9 \pm 0.4	87.8 \pm 0.3	61.8 \pm 1.5	93.8 \pm 0.2	84.2
Wikipedia + BookCorpus (100M words)								
LTG-BERT (subword masking)	84.2 \pm 0.1	84.3 \pm 0.7	90.8 \pm 0.3	92.1 \pm 0.5	87.8 \pm 0.5	62.5 \pm 1.7	95.3\pm0.1	82.0
British National Corpus (100M words)								
LTG-BERT (subword masking)	85.1\pm0.2	85.0 \pm 0.9	90.0 \pm 0.3	92.7\pm0.4	88.2 \pm 0.5	64.4 \pm 1.3	95.3\pm0.1	83.4
LTG-BERT (whole-word masking)	84.9 \pm 0.2	85.5 \pm 0.9	90.6 \pm 0.3	92.7\pm0.2	88.4 \pm 0.5	63.7 \pm 0.8	95.3\pm0.1	80.1
LTG-BERT (span masking)	85.1\pm0.2	87.5\pm0.9	91.5\pm0.2	92.8\pm0.5	89.2\pm0.5	65.6 \pm 0.5	95.2\pm0.1	84.2
LTG-BERT (subword + document NSP)	85.2\pm0.3	86.5 \pm 0.8	90.3 \pm 0.2	92.2 \pm 0.4	88.6 \pm 0.5	60.5 \pm 1.2	95.3\pm0.1	83.3
LTG-BERT (subword + order NSP)	84.7 \pm 0.1	85.9 \pm 0.6	90.4 \pm 0.2	92.1 \pm 0.2	88.3 \pm 0.4	64.2 \pm 1.9	95.1 \pm 0.1	82.2
LTG-BERT (subword + 2 \times steps)	85.2\pm0.2	86.5 \pm 0.8	90.3 \pm 0.3	92.3\pm0.6	88.6 \pm 0.5	65.3 \pm 1.1	95.3\pm0.1	83.5
LTG-BERT (subword + 1/2 \times steps)	84.4 \pm 0.3	86.3 \pm 1.1	90.4 \pm 0.2	92.8\pm0.4	88.5 \pm 0.6	62.4 \pm 0.8	95.2\pm0.1	83.5
LTG-BERT (subword + 1/4 \times steps)	83.8 \pm 0.2	85.3 \pm 0.8	89.1 \pm 0.2	91.7 \pm 0.4	87.5 \pm 0.5	58.6 \pm 1.3	95.0 \pm 0.1	83.2
Random initialization	59.5 \pm 0.5	68.5 \pm 1.4	63.8 \pm 0.2	82.2 \pm 0.7	68.5 \pm 0.8	49.7 \pm 0.3	73.1 \pm 0.4	50.0

Table 3: Summary of the experimental results. We show the results on the 4 GLUE tasks with known development results from Devlin et al. (2019) and their average; then the accuracy on HANS, the average of all 5 edge probing tasks and 67 BLiMP tasks. [†]The BERT_{base, cased} results are shown primarily for reference, they come from these sources: partial development GLUE scores from Devlin et al. (2019), edge probing from Tenney et al. (2019), HANS from Bhargava et al. (2021) and BLiMP from Salazar et al. (2020). We also add the BERT_{base, cased} results from our evaluation scripts for more fair and accurate comparison. We present the mean and standard deviation statistics over 5 evaluation runs and boldface all run within 1 standard deviation from the best result. The detailed results can be found in Appendix D.

the evaluation (Section 6.3). The original subword masking strategy is still a competitive baseline and it might be preferred in practice due to its simple implementation.

Next-sentence prediction. Next, we experiment with combining an NSP task and simple subword masking. We hypothesize that a second training objective might extract more information from the limited BNC corpus, which would help with the downstream performance – an opposite conclusion than Liu et al. (2019). However, our hypothesis turns out to be wrong, according to the results in Table 3. The experiments agree with the design of latest masked language models – next sentence prediction is an unnecessary training objective, at least for the tasks evaluated in this paper. It does not lead to substantially improved sentence representations even in a limited data regime. We can also see that the well-motivated order discrimination (Lan et al., 2020), proposed to solve the issues of document discrimination, actually leads to an overall worse performance. Hence we cannot recommend to complicate pre-training with a second training objective.

7.3 Sampling efficiency

An important aspect of efficient language models is the number of training steps they require to reach a sufficient performance. So far, we have limited the size of the training corpus but kept the number of steps constant, set according to Devlin et al. (2019). The results in Table 3 suggest that increasing the steps two times does not lead to a noticeably better performance with BNC. Even more so, training for half the time turns out to be enough to get comparable performance. Yet, decreasing the training steps further starts to degrade the downstream results too much, as evidenced by the scores obtained with 1/4 of the default steps.

These results highlight the sampling inefficiency of current self-supervised language modeling methods, as even with 1/4 steps, every token in BNC is seen about roughly 250 times during training.¹² We hope that a future work in this field will be able to learn from a smaller number of samples.

¹² This value can be calculated from Table 9: these models are trained for 7 812 steps with 4 194 304 tokens per batch. Table 1 shows that there are 131 392 103 subwords in the BNC train split.

7.4 100 million subset of Wikipedia & BookCorpus

Our last experiment evaluates how much does the careful curation of BNC help the downstream performance. To keep the comparability to BERT, we choose to pre-train on a random subset of Wikipedia and BookCorpus (with equal size to BNC, sampled document-wise); this corpus is constructed according to [Appendix F](#). Note that BNC is a corpus of British English compiled in 1990s so some evaluation tasks can be skewed against it – for example QNLI, which is based on texts from Wikipedia. [Table 3](#) shows that a high-quality data source is not necessarily needed to learn from 100M words but better quality leads to a noticeable difference in downstream performance.

8 Conclusion

In this paper, we evaluated how data-efficient masked language models can be. In particular, we trained a variety of models with different training objectives on the same training data: British National Corpus. Although small by modern standards (100M tokens), it is well balanced and carefully crafted to represent British English of the 20th century. On a variety of benchmarks, our models perform better than BERT_{base} trained on a much larger corpus. We believe that this limited data regime is beneficial for the development of efficient and reliable language models. Our finding also suggests that 100 million word tokens is enough to learn basic linguistic skills by current language modeling techniques, given that the data is carefully selected and balanced. To conclude, huge amounts of training data are not always necessary – we should focus on more efficient training settings instead.

We showed that the next sentence prediction objective does not improve BERT-like models, confirming the findings in [Liu et al. \(2019\)](#). In addition, the standard subword masking from [Devlin et al. \(2019\)](#) is consistently outperformed by the span masking method and the linguistic performance can be substantially increased by utilizing better neural architectures and training configurations. We release the code for training and using BERT-like models with the optimal architectural choices (according to our experiments) under the name LTG-BERT.¹³

¹³<https://github.com/ltgoslo/ltg-bert>

The presented results serve primarily as the foundation for future research on efficient language modeling. We hope our work shows the value of careful curation of representative corpora and will spark more interest in this area, where BNC can serve as an undemanding, replicable and openly-available training corpus.

9 Limitations

First of all, our work only considers language modeling of English and does not provide results on any other language – even though we hope that our conclusions could be useful for low-resource languages. Secondly, even though we found out that it is possible to train a competent language model with a small corpus, the training process still requires a similar amount of computational resources to models trained with larger corpora, as noted in [Section 7.3](#). Finally, we evaluate mainly the linguistic knowledge of language models ([Section 6](#)), our conclusions might not apply for their general knowledge.

References

- Yeşim Aksan, Mustafa Aksan, Ahmet Koltuksuz, Taner Sezer, Ümit Mersinli, Umut Ufuk Demirhan, Hakan Yılmaz, Gülsüm Atasoy, Seda Öz, İpek Yıldız, and Özlem Kurtoğlu. 2012. [Construction of the Turkish national corpus \(TNC\)](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3223–3227, Istanbul, Turkey. European Language Resources Association (ELRA). 3
- Giuseppe Attardi. 2015. Wikiextractor. <https://github.com/attardi/wikiextractor>. 17
- Payal Bajaj, Chenyan Xiong, Guolin Ke, Xiaodong Liu, Di He, Saurabh Tiwary, Tie-Yan Liu, Paul Bennett, Xia Song, and Jianfeng Gao. 2022. [Metro: Efficient denoising pretraining of large scale autoencoding language models with model generated signals](#). 4
- Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second pascal recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*. 16
- Yonatan Belinkov. 2022. [Probing Classifiers: Promises, Shortcomings, and Advances](#). *Computational Linguistics*, 48(1):207–219. 6
- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models

- be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623. 1
- Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *In Proc Text Analysis Conference (TAC'09)*. 16
- Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. [Generalization in NLI: Ways \(not\) to go beyond simple heuristics](#). In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 125–135, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. 8
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc. 2
- Lou Burnard. 2002. *Where did we Go Wrong? A Retrospective Look at the British National Corpus*, pages 51 – 70. Brill, Leiden, The Netherlands. 2
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics. 16
- Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. [One billion word benchmark for measuring progress in statistical language modeling](#). In *Proc. Interspeech 2014*, pages 2635–2639. 2
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. [BoolQ: Exploring the surprising difficulty of natural yes/no questions](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics. 15
- BNC Consortium. 2007. British National Corpus. 1, 3
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognizing textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg. 16
- Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. [The commitmentbank: Investigating projection in naturally occurring discourse](#). *Proceedings of Sinn und Bedeutung*, 23(2):107–124. 15
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. 2, 3, 5, 5, 5, 6, 6, 6, 7, 8, 8, 8, 8, 9
- William B. Dolan and Chris Brockett. 2005. [Automatically constructing a corpus of sentential paraphrases](#). In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. 15
- Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. [The third PASCAL recognizing textual entailment challenge](#). In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics. 16
- Thamme Gowda and Jonathan May. 2020. [Finding the optimal vocabulary size for neural machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964, Online. Association for Computational Linguistics. 6, 17
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [DeBERTa: Decoding-enhanced bert with disentangled attention](#). In *International Conference on Learning Representations*. 4, 4, 6, 7
- Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*. 4
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. [Training compute-optimal large language models](#). 2
- F. Jelinek. 1976. [Continuous speech recognition by statistical methods](#). *Proceedings of the IEEE*, 64(4):532–556. 2

- Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. [SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics. 14
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. [SpanBERT: Improving pre-training by representing and predicting spans](#). *Transactions of the Association for Computational Linguistics*, 8:64–77. 5
- Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. [Looking beyond the surface: A challenge set for reading comprehension over multiple sentences](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics. 16
- Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019. [A surprisingly robust trick for the Winograd schema challenge](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4837–4842, Florence, Italy. Association for Computational Linguistics. 5
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. [Albert: A lite bert for self-supervised learning of language representations](#). In *International Conference on Learning Representations*. 5, 8
- Tal Linzen. 2020. [How can we accelerate progress towards human-like linguistic generalization?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics. 1
- Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. 2020. [Understanding the difficulty of training transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5747–5763, Online. Association for Computational Linguistics. 4
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692. 2, 5, 6, 8, 9
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*. 7
- B.W. Matthews. 1975. [Comparison of the predicted and observed secondary structure of t4 phage lysozyme](#). *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451. 15
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. [Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics. 6
- Toan Q. Nguyen and Julian Salazar. 2019. [Transformers without tears: Improving the normalization of self-attention](#). In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics. 4, 4
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. 17
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics. 2
- Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. [WiC: the word-in-context dataset for evaluating context-sensitive meaning representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics. 16
- Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou,

- Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsim-poukelli, Nikolai Grigorev, Doug Fritz, Thibault Sot-tiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorraine Bennett, Demis Hassabis, Ko-ray Kavukcuoglu, and Geoffrey Irving. 2021. [Scaling language models: Methods, analysis & insights from training gopher](#). 2, 7
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics. 16
- Melissa Roemmele, Cosmin Bejan, and Andrew Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. *AAAI Spring Symposium - Technical Report*. 15
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. [A primer in BERTology: What we know about how BERT works](#). *Transactions of the Association for Computational Linguistics*, 8:842–866. 15
- Julian Salazar, Davis Liang, Toan Q. Nguyen, and Ka-trin Kirchhoff. 2020. [Masked language model scoring](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics. 8, 14
- Noam Shazeer. 2020. [GLU variants improve transformer](#). *CoRR*, abs/2002.05202. 4
- Sam Shleifer and Myle Ott. 2022. [Normformer: Improved transformer pretraining with extra normalization](#). 4, 7
- Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. 16
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics. 16
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. [What do you learn from context? probing for sentence structure in contextualized word representations](#). In *International Conference on Learning Representations*. 6, 8, 14, 16, 18
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. 2, 3, 4, 14
- Alex Wang and Kyunghyun Cho. 2019. [BERT has a mouth, and it must speak: BERT as a Markov random field language model](#). In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. Association for Computational Linguistics. 14
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. [Superglue: A stickier benchmark for general-purpose language understanding systems](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. 5
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics. 5
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mo-hananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. [BLiMP: The benchmark of linguistic minimal pairs for English](#). *Transactions of the Association for Computational Linguistics*, 8:377–392. 6, 16, 16
- Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. [Neural network acceptability judgments](#). *Transactions of the Association for Computational Linguistics*, 7:625–641. 15
- Weischedel, Ralph, Palmer, Martha, Marcus, Mitchell, Hovy, Eduard, Pradhan, Sameer, Ramshaw, Lance, Xue, Nianwen, Taylor, Ann, Kaufman, Jeff, Franchini, Michelle, El-Bachouti, Mohammed, Belvin, Robert, and Houston, Ann. 2013. [Ontonotes release 5.0](#). 16
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics. 15

- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics. 17
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. [Google’s neural machine translation system: Bridging the gap between human and machine translation](#). 6
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc. 2
- Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. [Large batch optimization for deep learning: Training bert in 76 minutes](#). In *International Conference on Learning Representations*. 7
- Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. [Record: Bridging the gap between human and machine commonsense reading comprehension](#). 16
- Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. [When do you need billions of words of pretraining data?](#) In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1112–1125, Online. Association for Computational Linguistics. 2
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. [Aligning books and movies: Towards story-like visual explanations by watching movies and reading books](#). In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27. 2, 2, 17

A BNC samples

We follow the description of the Markdown conversion of BNC from Section 3.1 and show samples of the resulting raw Markdown text to illustrate this process, highlighting some of the formatting information captured by our format. A sample of a spoken document is given in Listing 1 and a sample of a written article is shown below, in Listing 2.

B Evaluation metrics – implementation details

B.1 (Super)GLUE

Fine-tuning of the GLUE and SuperGLUE tasks follows a straightforward framework: the segments are tokenized, concatenated – starting with a special [CLS] token and with a [SEP] token put in between the segments – and input to a pre-trained language model. Subsequently, the contextualized representation of the special [CLS] token is fed into an MLP classifier. The pre-trained weights are further fine-tuned together with the classifier weights.

We do not employ any additional training tricks that were used in the previous works to seemingly increase the performance of their large language models – e.g. further ‘pre-training’ on MNLI, multi-task learning, ensembling, extensive hyperparameter search, selecting the best random seeds, reformulating the tasks or complex regularization techniques such as SMART (Jiang et al., 2020).

B.2 Edge probing

We follow the description of edge probing in the original paper by Tenney et al. (2019). First of all, subword representations $s_{i,k}$ are extracted from a frozen LM, for all positions i and layers k . These are downsampled to a dimensionality of 256 by a linear transformation. To get a vector representation h_t for the t^{th} span, we apply two pooling operations on the subword-token representations $s_{t,k}$. First, we pool the vectors at all layers k by taking a learnt convex combination $\hat{s}_t = \sum_{k=1}^{12} \gamma_k s_{t,k}$, where $\gamma_k \in \mathbb{R}$. Next, since one span can be split into multiple subwords, we employ an attention pooling operator to get the span-level embeddings: $h_t = \sum_{i \in \mathcal{I}_t} \text{att}(\hat{s}_t; \theta) \hat{s}_t$, where \mathcal{I}_t are the subwords indices of the t^{th} span. Finally, the pooled vectors h_t are fed into a multi-layer perceptron (MLP) and classified. If a task requires a pair of span representations (DP, SRL and CR), then these are pooled

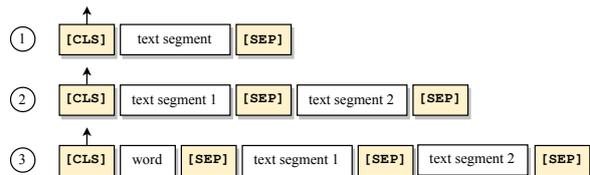


Figure 2: Three variations of (Super)GLUE input: 1) single-sentence tasks SST-2 and CoLA; 2) classification of a pair of text segments: BoolQ, CB, COPA, MNLI, MRPC, QNLI, QQP, STS-B, RTE; and 3) WiC (in the figure), MultiRC, ReCoRD.

with two separate attention operators and concatenated before being passed to the MLP classifier.

B.3 BLiMP

These models are trained to estimate $P(s_t | s_{<t})$ for sentence s and token s_t where $s_{<t} = (s_i | i < t)$; then the sentence log-probability is given by $\log P(s) = \sum_{t=1}^N \log P(s_t | s_{<t})$.

The issue with masked language models is that they are not designed to calculate this property; they are trained to estimate $P(s_t | s_{\setminus t})$ – the likelihood of a token s_t given its bidirectional context $s_{\setminus t} = (s_i | i \neq t)$. We can however still use MLMs to infer a *score* for each sentence where a higher *score* corresponds to a more likely sentence. Wang and Cho (2019) defined *pseudo-log-likelihood score* of a sentence s with model θ as

$$\text{PLL}(s) = \sum_{t=1}^N \log P(s_t | s_{\setminus t}; \theta).$$

Salazar et al. (2020) tested PLL and found that it produces accurate predictions on BLiMP. We adopt their approach and evaluate our models with PLL.

C Layer interpretation

The definition of the fine-tuning scheme for edge probing makes it straightforward to rate the contribution of each Transformer layer to a particular task – we can simply have a look at the layer-wise weights γ_k , see Table 5. To be more precise, if we define the k^{th} attention layer as a_k , the k^{th} feed-forward layer as ff_k and layer normalization operator as LN, then the k^{th} layer ℓ_k of a post-norm Transformer (Vaswani et al., 2017) computes the following function:

$$\begin{aligned} \hat{a}_k(x) &= \text{LN}(x + a_k(x)) \\ \ell_k &= \text{LN}(\hat{a}_k(\ell_{k-1}) + ff_k(\hat{a}_k(\ell_{k-1}))) \end{aligned}$$

Task	BoolQ	CB	CoLA	COPA	MNLI	MRPC	MultiRC	QNLI	QQP	ReCoRD	RTE	SST2	STSB	WiC
Train size	9 427	250	8 551	800	392 702	3 668	27 243	104 743	363 846	1 179 400	2 490	67349	5749	5428
Validation size	3 270	56	1 043	100	9 815	408	4 848	5 463	40 430	113 236	277	872	1 500	638
≥ 512 subwords	0.37%	0%	0%	0%	0%	0%	27.68%	0.02%	0%	0.30%	0%	0%	0%	0%

Table 4: The train and validation sizes of GLUE and SuperGLUE tasks (omitting WNLI and WSC). Note that we list the numbers of examples in the (Super)GLUE formulation of these tasks, which may differ from the actual number of examples – for example in case of multiple-choice questions. Some tasks do not offer a reliable amount of training data and some tasks contain a large number of examples longer than the length limit of our language models.

Task	Layer												Regression
	1	2	3	4	5	6	7	8	9	10	11	12	slope
POS	27.49	12.55	7.54	5.42	5.78	4.83	5.12	5.59	6.02	5.43	4.40	9.84	-0.98
DP	14.65	10.78	10.99	12.66	9.92	7.47	8.00	6.04	5.66	4.58	3.89	5.35	-0.89
SRL	19.38	13.70	9.80	9.56	8.44	7.04	6.99	5.61	4.85	3.83	2.70	8.11	-1.04
NER	18.16	9.12	6.58	4.83	6.86	6.75	6.87	6.28	6.62	4.93	5.87	17.13	-0.16
COREF	7.24	9.12	7.78	10.50	11.89	12.85	12.35	8.96	5.20	4.27	3.56	6.29	-0.42

Table 5: The per-layer contributions to different edge probing tasks, taken from the layer-wise convex weights γ_k (rendered in percent). To summarize the individual scores, we fit a linear regression line and show its slope in the last column. A negative slope implies stronger representation in the lower layers and vice versa.

It is unclear how to separate the contribution of each layer from the the previous layers here: ℓ_k contains both the previous scaled ℓ_{k-1} and its transformation from a_k and ff_k . On the other hand, our NormFormer-like architecture (Section 4) defines each layer ℓ_k as:

$$\hat{a}_k(x) = x + \text{LN}(a_k(x))$$

$$\ell_k = \hat{a}_k(\ell_{k-1}) + ff_k(\hat{a}_k(\ell_{k-1}))$$

Then it is trivial to calculate the contribution of each layer as $s_k = \ell_k - \ell_{k-1}$. We use this s_k entities to compute the learnt convex combination of all layers $\hat{s} = \sum_{k=1}^{12} \gamma_k s_k$.

Interpreting γ_k as the amount of ‘knowledge’ of a particular task in layer k , we see that POS information is contained primarily in the lowest layers, followed by SRL and DP. On the other hand, NER and CR are represented more strongly in the higher layers, which confirms the related findings in the literature (Rogers et al., 2020).

D Fine-grained results

To ease the evaluation of any future language models trained on BNC, we provide detailed results of all evaluated models in the following tables: GLUE results are shown in Table 6, edge probing performance is given in Table 7 and the BLiMP accuracies in Table 8.

(Super)GLUE. In total, we fine-tune all models on these 14 (Super)GLUE datasets:

- **Boolean Questions** (BoolQ; Clark et al., 2019), a yes/no question answering dataset evaluated with accuracy.
- **The CommitmentBank** (CB; de Marneffe et al., 2019), evaluated with both accuracy and F_1 -score, where the multi-class F_1 is computed as the unweighted average of the F_1 per class.
- **Corpus of Linguistic Acceptability** (CoLA; Warstadt et al., 2019) evaluated with the Matthews correlation coefficient (MCC; Matthews, 1975).
- **Choice of Plausible Alternatives** (COPA; Roemmele et al., 2011), evaluated with accuracy.
- **The Multi-Genre Natural Language Inference Corpus** (MNLI; Williams et al., 2018). Its development set consists of two parts: *matched*, sampled from the same data source as the training set, and *mismatched*, which is sampled from a different domain. Both parts are evaluated with accuracy.
- **The Microsoft Research Paraphrase Corpus** (MRPC; Dolan and Brockett, 2005), eval-

uated with both accuracy and F_1 -score.

- **Multi-Sentence Reading Comprehension** (MultiRC; Khashabi et al., 2018), a multiple choice question answering dataset, evaluated with the exact match accuracy (EM) and F_1 -score (over all answer options).
- **Question-answering Natural Language Inference** (QNLI) constructed from the Stanford Question Answering Dataset (SQuAD; Rajpurkar et al., 2016), evaluated with accuracy.
- **The Quora Question Pairs** (QQP),¹⁴ evaluated with both accuracy and F_1 -score.
- **The Stanford Sentiment Treebank** (SST-2; Socher et al., 2013), evaluated with accuracy.
- **The Semantic Textual Similarity Benchmark** (STS-B; Cer et al., 2017), evaluate with Pearson and Spearman correlation coefficients.
- **Reading Comprehension with Commonsense Reasoning Dataset** (ReCoRD; Zhang et al., 2018), a question answering dataset evaluated with the exact match accuracy (EM) and token-level F_1 -score (maximum over all correct mentions).
- **The Recognizing Textual Entailment datasets** (RTE; Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), evaluated with accuracy.
- **The Word-in-Context dataset** (WiC; Pilehvar and Camacho-Collados, 2019), evaluated simply with accuracy.

Edge probing. We report the results on part-of-speech tagging (POS), semantic role labeling (SRL), named entity recognition (NER) and coreference resolution (CR) using the annotations from the English part of OntoNotes 5.0 (Weischedel, Ralph et al., 2013). In addition, to further measure the syntactic abilities, we test the dependency parsing (DP) accuracy on the English Web Treebank v2.9 dataset from the Universal Dependencies (Silveira et al., 2014).¹⁵ These choices follow the original work by Tenney et al. (2019), but we do not evaluate on constituency parsing, because the

¹⁴<https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs>

¹⁵Available online at https://github.com/UniversalDependencies/UD_English-EWT.

results suffered from large variation. Instead, we test the syntactic knowledge with DP, which turned out to be more reliable as its variation is negligible (Table 7).

BLiMP. The Benchmark of Linguistic Minimal Pairs for English (Warstadt et al., 2020) consists of 67 tasks. Each focuses on a specific linguistic feature, which is tested with 1 000 automatically generated sentence pairs. Warstadt et al. (2020) clusters these tasks into the following subgroups:

- **Anaphor agreement** tests whether the reflexive pronouns agree with their antecedents.
- **Argument structure** – do verbs appear with the correct types of arguments?
- **Binding** evaluates the correctness of structural relationship between a pronoun and its antecedent.
- **Control/raising** tests syntactic and semantic differences between predicates that embed an infinitival verb predicate.
- **Determiner-noun agreement** checks number agreement between determiners the associated noun.
- **Ellipsis** – can we omit an expression from a sentence?
- **Filler-gap** tests dependencies created by phrasal movement.
- **Irregular forms** checks the correctness of irregular morphology on English past participles.
- **Island effects** – correctness of a possible gap in a filler-gap dependency.
- **NPI licensing** – are the negative polarity items used correctly (e.g. in negation)?
- **Quantifiers** tests the usage of quantifiers.
- **Subject-verb agreement** checks the number agreement between present tense verbs and subjects.

E Hyperparameters

All hyperparameters used to pre-trained and fine-tune our models are listed below: pre-training hyperparameters in Table 9, the GLUE and SuperGLUE fine-tuning hyperparameters in Table 10 and the edge probing hyperparameters in Table 11. BLiMP does not require any special hyperparameters, it need only out-of-the-box predictions of a

pre-trained language model. Note that we will also release the full PyTorch (Paszke et al., 2019) source code, tokenizer and the pre-trained language models in the camera-ready version. Additionally, we will also provide all necessary wrappers for a simple use of our models with the `transformers` library (Wolf et al., 2020).

The training was performed on 128 AMD MI250X GPUs (distributed over 16 compute nodes) and took approximately 8 hours per model in a mixed precision mode. In total, our models consist of 98M parameters; a slightly lower value than BERT’s 110M parameters due to the smaller vocabulary size.

F Wikipedia + BookCorpus dataset replication

The information about the exact Wikipedia dump used for training BERT is unknown and the BookCorpus dataset (Zhu et al., 2015) is no longer available. On top of that, the preprocessing choices are also not known. Our 100M Wikipedia + BookCorpus dataset is thus different from the original BERT pre-training corpus.

We downloaded a fresh English Wikipedia dump from <https://dumps.wikimedia.org/enwiki/20220801/enwiki-20220801-pages-articles-multistream.xml.bz2>, extracted the raw text with WikiExtractor (Attardi, 2015) and segmented each article into sentences with `spaCy`.¹⁶

A replicated version of BookCorpus was obtained from https://the-eye.eu/public/AI/pile_preliminary_components/books1.tar.gz and every book was also segmented with `spaCy`.

After that, the random 100M subset was created by sampling random documents from the full Wikipedia + BookCorpus dataset until the subset contained as many characters as BNC.

G Tokenizer definition

We use the HuggingFace’s `tokenizers` library,¹⁷ to define and train a subword tokenizer on BNC (training split).¹⁸

Following the suggestion of Gowda and May (2020), we set the vocabulary size so that at least 95% of tokens appear more than 100 times. In our case, with the size of $2^{14} = 16\,384$, 95% of

tokens appear more than 166 times in the training split. Their finding comes from the realm of neural machine translation, we have not evaluated how it aligns with language modeling. Nevertheless, we believe that a comparative study of different tokenizer settings makes an interesting future work; we suspect that the effects will be more pronounced with BNC, due to its limited size.

¹⁶ <https://spacy.io/>

¹⁷ <https://huggingface.co/tokenizers/>

¹⁸ We share the full definition of the tokenizer in censored-for-review.com.

Task	Metric	BERT (100M subset)	MLM			NSP		Training steps		
			subword	word	span	document	order	2×	0.5×	0.25×
BoolQ	accuracy	75.16±0.48	74.87±0.26	75.94±0.16	75.08±0.94	74.75±0.71	74.80±1.07	74.87±0.62	74.84±0.71	74.08±0.56
CB	accuracy	78.93±3.43	76.06±2.40	84.64±3.48	75.71±2.71	82.86±1.60	83.57±1.49	80.00±1.96	74.28±3.91	77.14±3.44
	F ₁	72.11±6.73	72.78±5.17	80.42±4.52	71.91±8.36	77.78±2.31	80.99±3.10	72.56±4.04	66.73±4.23	80.69±3.45
CoLA	MCC	59.36±0.96	57.17±1.92	58.28±0.59	58.69±1.43	59.73±1.34	57.91±1.51	57.47±1.62	59.98±1.40	58.30±1.15
COPA	accuracy	60.40±5.03	59.20±2.28	64.00±5.43	59.40±5.03	72.00±1.87	62.80±2.77	54.20±1.96	58.00±3.32	61.60±2.19
MNLI	matched acc.	84.22±0.12	85.14±0.16	84.93±0.21	85.05±0.19	85.21±0.25	84.72±0.15	85.17±0.16	84.40±0.29	83.82±0.16
	mismatched acc.	84.00±0.05	84.78±0.17	85.05±0.13	85.35±0.15	85.36±0.21	84.73±0.19	85.29±0.14	84.60±0.16	83.71±0.16
	HANS acc.	62.47±1.68	64.39±1.28	63.75±0.76	65.60±0.53	60.50±1.24	64.16±1.86	65.32±1.14	62.35±0.82	58.63±1.35
MRPC	accuracy	84.31±0.71	85.00±0.94	85.54±0.90	87.45±0.86	86.52±0.81	85.93±0.59	86.47±0.80	86.27±1.12	85.29±0.83
	F ₁	89.06±0.48	89.51±0.64	89.83±0.61	91.20±0.62	90.39±0.59	89.99±0.48	90.54±0.61	90.39±0.73	89.57±0.63
MultiRC	F ₁	67.25±0.57	67.61±0.86	68.10±0.85	71.93±0.73	71.90±0.35	71.91±0.35	66.45±2.12	67.30±0.62	65.02±1.00
	exact match	18.51±0.88	19.58±1.51	18.76±1.54	25.25±1.37	24.91±0.40	27.63±0.83	17.19±2.70	18.65±0.44	16.66±0.77
QNLI	accuracy	90.80±0.25	90.00±0.25	90.57±0.29	91.46±0.20	90.32±0.18	90.36±0.25	90.33±0.27	90.36±0.16	89.08±0.24
QPP	accuracy	91.01±0.05	90.94±0.06	90.85±0.07	91.01±0.10	91.00±0.14	90.90±0.08	91.01±0.08	90.77±0.04	90.51±0.09
	F ₁	87.85±0.07	87.81±0.08	87.73±0.07	87.87±0.14	87.94±0.19	87.76±0.10	87.92±0.13	87.57±0.05	87.24±0.13
SST-2	accuracy	92.06±0.48	92.71±0.40	92.71±0.24	92.80±0.50	92.18±0.38	92.11±0.25	92.34±0.59	92.82±0.40	91.67±0.37
STS-B	Pearson corr.	86.34±0.29	87.44±0.33	87.53±0.19	87.99±0.11	89.50±0.14	89.11±0.25	87.83±0.19	86.93±0.50	85.80±0.18
	Spearman corr.	86.10±0.31	87.24±0.32	87.45±0.20	87.72±0.10	89.06±0.12	88.82±0.22	87.67±0.21	86.73±0.47	85.54±0.20
ReCoRD	F ₁	65.48±0.64	63.15±3.19	68.36±1.59	70.71±1.81	66.51±0.33	67.73±1.00	62.93±3.12	64.68±1.90	57.59±2.06
	exact match	64.81±0.62	62.48±3.19	67.61±1.58	70.03±1.78	65.84±0.33	67.04±1.02	62.26±3.08	63.93±1.89	56.88±2.07
RTE	accuracy	62.38±3.00	60.65±1.92	60.51±2.07	60.51±2.61	66.50±1.12	69.68±1.28	58.34±2.56	56.82±1.63	58.19±0.59
WiC	accuracy	66.36±1.59	66.46±1.21	67.40±0.43	69.18±1.04	70.78±0.94	68.90±0.60	67.52±1.35	66.71±0.99	68.46±0.71
Average		74.04 ±2.20	73.63 ±1.75	75.20 ±1.99	75.12 ±2.39	76.69 ±0.96	76.21 ±1.22	73.34 ±1.91	73.39 ±1.67	73.15 ±1.33

Table 6: Detailed development GLUE and SuperGLUE results for all tested models. We show the mean and standard deviation statistics over 5 runs with different random seeds (changed only for fine-tuning, the pre-trained models are kept the same).

Model		POS	DP	SRL	NER	CR	Average
BERT (100M subset)		97.94±0.01	95.03±0.04	92.34±0.06	95.91±0.12	95.27±0.10	95.30±0.08
MLM	subword	97.91±0.01	94.99±0.02	92.44±0.03	95.77±0.06	95.30±0.07	95.28±0.05
	whole-word	97.90±0.01	94.99±0.05	92.42±0.08	95.71±0.07	95.64±0.07	95.33±0.06
	span	97.91±0.01	94.80±0.03	92.32±0.02	95.56±0.07	95.46±0.14	95.21±0.07
NSP	subword + document	97.92±0.01	95.01±0.03	92.42±0.06	95.76±0.07	95.25±0.11	95.28±0.07
	subword + order	97.85±0.01	94.92±0.06	92.25±0.07	95.22±0.05	95.25±0.11	95.10±0.07
Steps	subword + 2×	97.93±0.01	94.95±0.10	92.47±0.03	95.63±0.11	95.58±0.04	95.31±0.07
	subword + 1/2×	97.90±0.02	95.02±0.04	92.38±0.05	95.46±0.03	95.43±0.05	95.24±0.04
	subword + 1/4×	97.88±0.01	94.81±0.07	92.21±0.03	95.32±0.08	95.00±0.18	95.04±0.10
Random initialization		69.85±0.42	66.25±0.20	70.87±0.21	73.16±0.60	85.56±0.46	73.14±0.41

Table 7: Detailed edge probing results for all tested models. † The BERT_{base} scores in the first row are taken from Tenney et al. (2019). The last row shows the edge probing results with a randomly initialized language model – its performance hints at how much information is included in the probes themselves.

BLiMP subgroups	BERT (100M subset)	MLM			NSP		Size		
		subword	word	span	document	order	medium	small	tiny
Anaphor agreement	93.20	93.95	92.65	94.50	93.00	94.00	94.65	93.30	94.60
Argument structure	78.95	80.73	67.93	80.98	81.58	80.61	81.54	80.98	81.99
Binding	77.04	78.34	74.60	77.26	77.74	76.60	77.33	76.43	77.03
Control/raising	73.76	79.68	79.90	81.02	78.18	78.32	78.84	79.80	78.82
Determiner-noun agreement	95.91	96.74	93.48	97.45	97.09	96.26	97.09	96.73	96.96
Ellipsis	88.25	88.10	85.55	90.95	88.65	86.70	90.25	87.70	88.70
Filler-gap	85.44	83.87	83.30	85.86	87.23	83.46	85.20	84.73	84.20
Irregular forms	88.45	91.45	86.75	94.40	88.30	86.70	92.35	92.65	93.10
Island effects	70.91	74.99	76.71	74.34	73.98	74.62	72.14	74.86	72.34
NPI licensing	81.07	82.40	81.73	82.36	83.24	78.79	82.43	84.96	82.86
Quantifiers	69.98	68.88	70.00	74.13	64.50	68.77	72.58	67.10	67.53
Subject-verb agreement	91.78	92.64	83.97	92.92	92.13	90.00	91.72	92.25	92.22
Accuracy	81.95	83.42	80.05	84.18	83.31	82.17	83.45	83.47	83.15

Table 8: Detailed BLiMP results for all tested models.

Hyperparameter	Base
Number of layers	12
Hidden size	768
FF intermediate size	2 048
Vocabulary size	16 384
FF activation function	GEGLU
Attention heads	12
Attention head size	64
Dropout	0.1
Attention dropout	0.1
Training steps	31 250
Batch size	32 768 (90% steps) / 8 192 (10% steps)
Sequence length	128 (90% steps) / 512 (10% steps)
Tokens per step	4 194 304
Warmup steps	500 (1.6% steps)
Initial learning rate	0.01
Final learning rate	0.001
Learning rate decay	cosine
Weight decay	0.1
Layer norm ϵ	1e-5
Optimizer	LAMB
LAMB ϵ	1e-6
LAMB β_1	0.9
LAMB β_2	0.98
Gradient clipping	2.0

Table 9: Pre-training hyperparameters. The models differ only in their hidden size and number of layers, the learning rate schedule and other training settings are kept identical.

Hyperparameter	ReCoRD	MNLI, QQP, QNLI	BoolQ, CoLA, COPA, SST-2, MultiRC, MRPC, STSB			CB
			RTE, WiC			
Batch size	32	32	32	16	8	
Number of epochs	1	4	8	8	16	
Dropout	0.1	0.1	0.1	0.1	0.1	
Warmup steps	10%	10%	10%	10%	10%	
Peak learning rate	3e-5	3e-5	3e-5	3e-5	3e-5	
Learning rate decay	linear	linear	linear	linear	linear	
Weight decay	0.01	0.01	0.01	0.01	0.01	
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	
Adam ϵ	1e-6	1e-6	1e-6	1e-6	1e-6	
Adam β_1	0.9	0.9	0.9	0.9	0.9	
Adam β_2	0.999	0.999	0.999	0.999	0.999	

Table 10: Hyperparameters for fine-tuning the GLUE and SuperGLUE tasks. We use the same hyperparameters for all models, not performing any per-model hyperparameter search.

Hyperparameter	POS, SRL, NER, CR	DP
Batch size	128	128
Number of epochs	5	10
Dropout	0.25	0.25
Downsampled hidden size	256	256
Attention pooling heads	4	4
MLP hidden layers	1	1
Starting learning rate	6e-3	6e-3
Learning rate decay	cosine	cosine
Weight decay	0.01	0.01
Optimizer	AdamW	AdamW
Adam ϵ	1e-6	1e-6
Adam β_1	0.9	0.9
Adam β_2	0.999	0.999
Gradient clipping	2.0	2.0

Table 11: Edge probing hyperparameters.

```

1 # Oral history project: interview
2
3 Britta: 'Can you tell us what er what section you work in?'
4
5 Eliazar: 'I work at the weaving'
6
7 Britta: 'In the weaving?'
8
9 Eliazar: 'section, aha.
10 And'
11
12 Britta: 'And what do you do?'
13
14 Eliazar: 'I'm what you call a Axminster handler'
15
16 Britta: 'Aha.'
17
18 Eliazar: 'which involves like when the frames comes off the weaving and they're yarn left, I strip the
↳ yarn off.'
19
20 Britta: 'Mhm.'
21
22 Eliazar: 'Off the, the, the weaving frames.'
23
24 Britta: 'Mhm.'
25
26 Eliazar: 'That's basically my, aha.'
27
28 Britta: 'It's quite spec specialized so'
29
30 Eliazar: 'No no, no, no.
31 It's not specialized, no.'
32
33 Britta: 'Mhm, have you ever worked in any other factory?'
34
35 Eliazar: 'Aha, I worked in spooling, I've been left now two year.'
36
37 Britta: 'And how did you find that?'
38
39 Eliazar: 'Er, I liked the spooling but some I just don't know, some of the girls get kind of one [UNK]
↳ one thing by the other I can object to, I think it was actually the atmosphere of the, the girls
↳ that worked in the department that I'

```

Listing 1: A random example of the first few lines from a preprocessed spoken document from BNC. Notice that the text is divided into speech turns (paragraphs), each starting with the name of a speaker. Line 39 contains a special [UNK] token in place of an incomprehensible word or phrase.

```

1 # Organizing knowledge: an introduction to information retrieval
2
3 ## SUBJECTS
4
5 ### The subject approach: introduction, processes, tools and simple evaluation
6
7 ##### 1.2.1 Subjects
8
9 Users often approach information sources not with names (as have been considered in Part II), but with
↳ a question that requires an answer or a topic for study.
10 Users seek documents or information concerned with a particular subject.
11 In order to make some provision for this common approach to information sources, it is necessary to
↳ arrange documents- and document surrogates in catalogues, indexes bibliographies, computer
↳ databases and so on - in such a way that items on specific subjects can be retrieved.
12 Thus, the subject approach is extremely important in the access to and the exploitation of information,
↳ documents and data.
13
14 Before we discuss the provision that libraries and information workers make for the subject approach,
↳ it may be useful to consider the preliminary question: What is a subject?
15 In talking about a subject we generally refer to a given area of knowledge or to the contents of an
↳ information source of a given scope.
16 A subject might be considered to be defined by:
17
18 - an area of interest,
19
20 - an area in which an individual researcher or professional works,
21
22 - an area in which an individual writes or an area of knowledge which is studied.

```

Listing 2: A sample of the first few lines from a written BNC article. Note the H1-level header with the title of the whole document and then the title of a chapter, section and subsection in the lines below. This sample also contains a special text block with a list in the last lines.

Generating Synthetic Speech from *SpokenVocab* for Speech Translation

Jinming Zhao

Gholamreza Haffari

Ehsan Shareghi

Department of Data Science & AI, Monash University

firstname.lastname@monash.edu

Abstract

Training end-to-end speech translation (ST) systems requires sufficiently large-scale data, which is unavailable for most language pairs and domains. One practical solution to the data scarcity issue is to convert text-based machine translation (MT) data to ST data via text-to-speech (TTS) systems. Yet, using TTS systems can be tedious and slow. In this work, we propose *SpokenVocab*, a simple, scalable and effective data augmentation technique to convert MT data to ST data on-the-fly. The idea is to retrieve and stitch audio snippets, corresponding to words in an MT sentence, from a spoken vocabulary bank. Our experiments on multiple language pairs show that stitched speech helps to improve translation quality by an average of 1.83 BLEU score, while performing equally well as TTS-generated speech in improving translation quality. We also showcase how *SpokenVocab* can be applied in code-switching ST for which often no TTS systems exist.¹

1 Introduction

End-to-end (E2E) speech-to-text translation (ST) models require large amounts of data to train (Sperber and Paulik, 2020). Despite the emerging ST datasets (Cattoni et al., 2021; Wang et al., 2021), their size is considerably smaller compared to text-based machine translation (MT) data. A common remedy to tackle the data scarcity issue is to leverage text-based MT data in training ST systems. Common approaches include multi-task learning (Anastasopoulos and Chiang, 2018; Ye et al., 2021), transfer learning & pretraining (Bansal et al., 2019; Wang et al., 2020) and knowledge distillation (Inaguma et al., 2021; Tang et al., 2021).

A more straightforward alternative is to convert text-based MT data to ST via text-to-speech (TTS) synthesis engines (Pino et al., 2019; Jia et al., 2019).

¹Our code is available at <https://github.com/mingzi151/SpokenVocab>

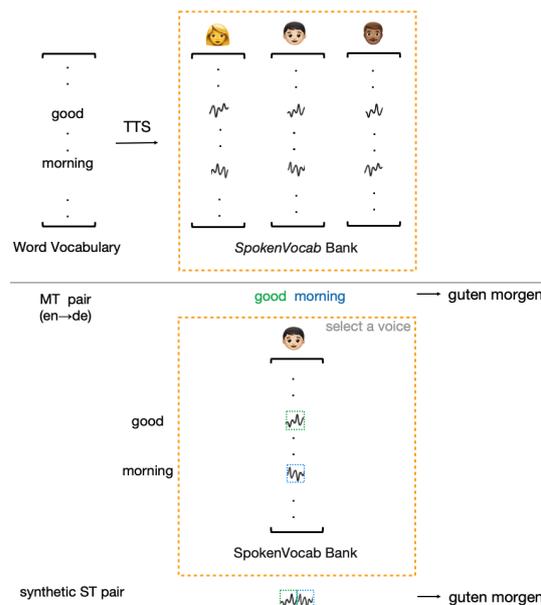


Figure 1: Overview of generating synthetic speech from SpokenVocab on-the-fly. The first step is to prepare the SpokenVocab bank offline and the second step is to retrieve and stitch audio snippets from the bank by words in a sentence.

This method is less commonly used despite its simplicity and effectiveness,² mainly due to practical reasons: (i) TTS models have slow inference time and may incur monetary costs; (ii) the conversion is required for each MT datasets. Recently, Lam et al. (2022) proposed to generate synthetic speech without using TTS models. However, their approach is based on real ST data, and thus cannot be extended to MT data.

In this work, we propose a simple, effective and efficient data augmentation approach to convert MT data to ST data on-the-fly. The idea is to prepare a set of spoken words, forming a spoken vocabulary (*SpokenVocab*) bank, and then generate synthetic speech by retrieving and stitching spoken

²Only one work out of 8 uses TTS to augment data in the IWSLT2022 offline speech translation track.

words based on a text sequence, as shown in Figure 1.³ Our experiments show that this method is as effective as TTS-generated speech, at a much lower computational and financial cost. For instance, augmenting ST data on-the-fly with 100k of stitch-converted MT data, boosts translation quality by an average of 1.83 BLEU over 3 language pairs from Must-C (Cattoni et al., 2021) with no additional cost, memory, or speed footprints. Comparing the real ST data vs. our converted version from the same transcripts, to our positive surprise, revealed that our synthetic data outperforms its real counterpart by 0.41 BLEU score. We conduct thorough experiments to examine *SpokenVocab* in boosting translation and further showcase its use and benefit in the context of code-switching (CS) ST.

We hope this simple technique to ease the use of MT data for ST in practice as well as other tasks where synthetic speech is useful.

2 SpokenVocab

We describe our methodology in creating effective synthetic ST data based on MT data in this section. The core step is the preparation of a *SpokenVocab* bank offline and stitching sounds on-the-fly.

Concretely, we first use a TTS engine to convert items in a word vocabulary to speech to obtain a set of *SpokenVocab* offline.⁴ Next, we can configure the TTS engine to generate different speaker voices and thus curate a *SpokenVocab* bank in which each set corresponds to a "speaker". The purpose is to simulate, to the greatest extent, a realistic speech dataset consisting of various speakers. At training, assume we have access to an MT dataset and each pair denoted as $\langle s, t \rangle$ where s and t are source and targets sentences, respectively. Given such a pair, we choose one voice⁵ from the bank, and produce synthetic speech by fetching corresponding audio snippets by words in s from the bank and stitching them together. During stitching, we deploy cross-fade, a well-known technique to smooth transitions between two independent audio clips.⁶

³During the writing of this manuscript we found out that Voder, the first electronic speech synthesiser developed by Bell Labs in 1939, synthesized human speeches by decomposing it into its acoustic components and combining them using human operators in real time.

⁴SpokenVocab could also be based on n-grams in a dataset.

⁵One could also generate utterances by mixing speakers at the token level, with no additional cost with our technique. We leave further investigation of this to future work as it requires a test condition (i.e., including various speaker voices per utterance) which is not available to the best of our knowledge.

⁶<https://github.com/jiaaro/pydub>

Pairing it with t yields a synthetic ST instance.⁷

3 Experiments

We first present the ST system (§3.1) and TTS systems (§3.1.2) used in this study. We then describe the ST and MT datasets (§3.1.3), followed by providing implementation details (§3.1.4). Next we explain how *SpokenVocab* is designed (§3.2) and report translation results (§3.3). Lastly, we illustrate how our method can be applied to CS ST (§3.5).

3.1 Experimental Setup

3.1.1 Speech Translation System

Pre-trained speech encoders and text decoders have shown great performance on ST (Li et al., 2021; Zhao et al., 2022), compared to models trained from scratch. For this reason, we follow the architecture in Gállego et al. (2021) that uses Wav2vec 2 (W2V2) (Baevski et al., 2020) as the speech encoder and mBart decoder (Liu et al., 2020) as the text decoder, joint with a lightweight linear adapter and a CNN-based length adapter.

3.1.2 TTS Systems

To prepare *SpokenVocab*, we use the Google TTS service,⁸ which supports a wide range of voice configurations; this allows simulating different speakers with various accents, gender and geographical background. We also use a off-the-shelf TTS toolkit, i.e., Tacotron2-DCA + Multiband-Melgan (short for T2+Mel).⁹ We use Google TTS to generate synthetic speech in raw wavforms.

3.1.3 Dataset

We conduct our major experiments on Must-C, a multilingual ST dataset curated from Ted talks. We focus on English (En)→German (De), Romanian (Ro) and Italian (It). For MT data, we use a subset of WMT14, WMT16 and OPUS100¹⁰ for De, Ro and It, with 100k, 100k and 24k instances, respectively. For the code-switching (CS) setting, we use Prabhupadavani (Sandhan et al., 2022), multilingual CS ST dataset, and we focus on En→De, It. Its source utterances are code-mixed with English (major language), Bengali and Sanskrit; each utterance is translated manually to 25 languages. We

⁷We provide a [demo](#) for stitched speeches.

⁸<https://cloud.google.com/text-to-speech>

⁹<https://github.com/mozilla/TTS>

¹⁰<http://opus.nlpl.eu/opus-100.php>

prepare ST data following the instructions in [Gállego et al. \(2021\)](#). We preprocess MT data with the fairseq instructions and remove pairs with the length of target sentences greater than 64 words to avoid out-of-memory issues. Minimal preprocessing is performed on the CS ST dataset.

3.1.4 Implementation Details

Similar to [Li et al. \(2021\)](#) and [Gállego et al. \(2021\)](#), training different components of W2V2 and mBart decoder yields divergent results. In our initial experiments, we note that fine-tuning the entire W2V2 except for its feature extractor and freezing mBart lead to decent translation results, and thus we use this configuration for all our experiments. To ensure Must-C to be dominant, we make the ratio of Must-C and MT data to be approximately 8:1, unless mentioned otherwise. We use sacreBLEU ([Post, 2018](#)) to evaluate translation. Please refer to [Appendix A.1](#) for full training details, hyper-parameters and hardware.

3.2 SpokenVocab Preparation and Variations

Constructing the *SpokenVocab* bank is crucial, as synthetic speech produced in this manner have a direct impact on translation quality. In this section we examine *SpokenVocab* from various dimensions.

TTS Conversion. The first questions to ask are which TTS system should be used to convert a word to a spoken form and what sampling rate (SR) is appropriate.¹¹ To answer these questions, we conduct intrinsic evaluation on stitched speech by varying TTS engines and SR. Furthermore, as it is common to diversify raw wave forms with audio effects ([Potapczyk et al., 2019](#)), we apply the same technique to distort our stitched speech. Results in [Table 1](#) show that using Google TTS and setting the SR to 24k are better choices, while distortion (i.e., adding the effects of tempo, speed and echo) may or may not be helpful. Contrary to the common practice of using a SR of 16k ([Baevski et al., 2020](#)), applying 16k to *SpokenVocab* alters the sound significantly, as shown in the demo in §2, and this has negative impacts on the system. Overall, we use the setting in *italic* for the rest of our experiments.

Word Vocabulary. We compile a word vocabulary, consisting of 1) a common subset of words¹², and

¹¹SR is defined as the number of samples taken from a continuous signal per second.

¹²The list comes from Official Scrabble Players Dictionary and Wiktionary’s word frequency lists, and can be found

Data	TTS	SR	Distort.	BLEU
ST	-	-	-	26.91
	T2+Mel	-	-	OOM
		24k	-	28.02
ST+MT _{stitched}	<i>Google</i>	24k	✓	27.72
		16k	-	26.77
		16k	✓	27.47

Table 1: Comparison of different TTS conversions in terms of TTS engine, sampling rate (SR) and distortion (Distort.) Top row: baseline. Bottom rows: MT data is converted to ST data with *SpokenVocab*. OOM: out-of-memory with 24k and 16k SRs. *italic*: best setting.

2) unique words with a frequency of higher than 99 from the En→X WMT subset. The purpose is to construct an approximated version of *SpokenVocab* that is ready to convert any sentence to synthetic speech. For words that are not covered by the list, we employ a fuzzy matching mechanism where the most similar word at the surface level is returned. For instance, an out-of-vocabulary (OOV) word "apples" is replaced by its closest match in the vocabulary "apple", and the speech snippet for "apple" is retrieved. When no match is found, a default filter word, "a", is returned. To investigate the effect of this approximation which would inevitably lead to mispronounced words, we prepare another set of *SpokenVocab* containing the full set of spoken words in the WMT data (eliminating the need for fuzzy matching). In controlled experiments on En→De, the BLEU scores with the approximated and full *SpokenVocabs*, with the size of 35k and 460k respectively, are 28.02 and 27.91. The negligible difference indicates the effectiveness of using an approximated *SpokenVocab*. Additional ablation studies on using 50% and 10% of the full vocabulary yield scores of 27.79 and 27.94, further validating the insensitivity of w2v2 to nuanced mispronunciation, perhaps due to the presence of powerful pre-trained auto-regressive decoder.¹³

Number of Speakers. Despite the artificial nature of the stitched speech sounds, one still can tell the speaker’s information (e.g., gender, accent). To examine whether diverse voices would be helpful for translation, we set n to 1, 5 and 10 and train models with the same amount of data. These sys-

at <https://github.com/dolph/dictionary/blob/master/popular.txt>

¹³Optionally, one can dynamically call a TTS system to generate an audio on OOV words.

Data	Cost			BLEU		
	⌚	\$	🗄️	De	Ro	It
ST	-	-	-	26.91	24.66	22.13
ST + MT _{TTS}	900	90	25	28.20	24.71	26.46
ST + MT _{stitched}	9	0	0	28.02	25.05	26.13

Table 2: Translation quality on Must-C and the average costs associated for generating synthetic speech for every 100k sentences in terms of inference time in minutes (⌚), USD value (\$) and storage required in GB (🗄️). Preparing *SpokenVocab* took 2 hours, free of charge, with Google TTS, and stitched speeches are discarded.

tems display similar translation performance with 28.02, 27.73 and 27.80 BLEU scores respectively, suggesting that having a single speaker is sufficient. Our conjecture to this phenomenon is that speech representations produced by w2v2 have removed speaker information, as demonstrated in Nguyen et al. (2020) where analysis was conducted on wav2vec (Schneider et al., 2019), the predecessor to w2v2. This could be further examined with using dialect- or pronunciation-focused translation settings, which we leave to future work.

3.3 Translation Performance on Must-C

Producing synthetic speech from *SpokenVocab* on-the-fly makes the conversion from text to speech highly scalable in terms of time and monetary costs, and it also avoids the need of storing speech. Table 2 reports the time, dollar value and space required to produce every 100k speech with Google TTS, while these numbers are negligible for *SpokenVocab* due to its re-usability.¹⁴ Apart from scalability, it is more important to see the translation performance difference between unnatural speech produced by *SpokenVocab* and fluent speech generated by state-of-the-art TTS systems. Table 2 summarises results for 3 Must-C language pairs, with stitched speech and TTS-generated speech. As expected data augmentation of ST with MT data method boosts translation quality, using our method by 1.83 BLEU score on average. Our stitched speech performs equally well as TTS-generated counterpart, showing no loss of quality during conversion.

¹⁴For fair comparison between TTS which operates on the full vocabulary, we report the cost under the full vocabulary version of our method.

Data	Nature of Speech	BLEU
Must-C	real	26.91
Must-C + Europarl	real + real	27.5
Must-C + Europarl _{TTS}	real + synthetic	27.76
Must-C + Europarl _{stitched}	real + synthetic	27.91

Table 3: BLEU scores under different augmentations.

		ST _{CS}	ST _{CS} +MT _{CS-stitched}
BLEU	En-Be→De	26.11	28.09
	En-Be→It	26.41	26.90

Table 4: Translation quality for CS ST dataset.

3.4 Stitched Speech vs. Real Speech

An alternative approach to augmentation is to leverage real ST data from any other existing domains. To assess whether our approach as another augmentation technique is still competitive, we conduct an experiment on En→De by augmenting Must-C with 35k training instances from the Europarl-ST (Iranzo-Sánchez et al., 2020). Table 3 reports the results. To our positive surprise, our stitched speech (generated from the transcripts of europarl-ST counterpart) works even better than the real Europarl-ST speech.

3.5 Code-switching Speech Translation

Development in CS ST is constrained by the availability of relevant datasets (Sandhan et al., 2022) and using TTS systems to augment data is practically difficult. To this end, our method provides a high degree of flexibility in that it can stitch audio clips of different languages freely. To produce a code-switched utterance, we further prepare *SpokenVocab* for Bengali (Google TTS does not support Sanskrit) based on an English-Bengali dictionary.¹⁵ We maintained the ratio of code-switching in the real data (i.e., 0.35 probability of CS occurring, and 2 as the average number of code-switched words in a sentence). Please see Algorithm 1 in Appendix A.2 for the detailed utterance generation process. Results in Table 4 suggest that the models trained with additional 100k and 24k instances (for De and It respectively.) from *SpokenVocab* outperform those only trained with the original data.

¹⁵<https://github.com/MinhasKamal/BengaliDictionary>

4 Conclusion

In this work, we proposed a simple, fast and effective data augmentation technique, *SpokenVocab* for ST. This provides an alternative for converting MT data to ST data with TTS systems which comes with monetary and computation costs in practice. Our approach generates synthetic speech on-the-fly during training, with no cost or footprint. We have shown that speech stitched from *SpokenVocab* works as effective as TTS-generated speech, and unlike TTS system, it could directly be applied as a data augmentation tool in code-switching ST. Our approach can be used in other content-driven speech processing tasks as an uncompromising and easy-to-use augmentation technique.

Limitations

CS ST exhibit difficulties (Huber et al., 2022; Weller et al., 2022), exposing several limitations in this study: 1) Bengali and Sanskrit (another minority language) are treated without difference, as they originate from the same script and Sanskrit is not supported by the Google TTS service. 2) We use a open-source language detection tool to calculate the oracle hyper-parameters in the dev set; yet, imperfection of the detector on token-level prediction and the fact that source sentences are written in Latin regardless of the language deviate the scores from true values.

References

- Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 82–91.
- Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.
- Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 58–68.
- Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Benvivogli, Matteo Negri, and Marco Turchi. 2021. Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155.
- Gerard I Gállego, Ioannis Tsiamas, Carlos Escolano, José AR Fonollosa, and Marta R Costa-jussà. 2021. End-to-end speech translation with pre-trained models and adapters: Upc at iwslt 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 110–119.
- Christian Huber, Enes Yavuz Ugan, and Alexander Waibel. 2022. Code-switching without switching: Language agnostic end-to-end speech translation. *arXiv preprint arXiv:2210.01512*.
- Hirofumi Inaguma, Tatsuya Kawahara, and Shinji Watanabe. 2021. Source and target bidirectional knowledge distillation for end-to-end speech translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1872–1881.
- Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerda, Javier Jorge, Nahuel Roselló, Adria Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233. IEEE.
- Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184. IEEE.
- Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2022. Sample, translate, recombine: Leveraging audio alignments for data augmentation in end-to-end speech translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 245–254.
- Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Multilingual speech translation from efficient finetuning of pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 827–838.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

- Ha Nguyen, Fethi Bougares, Natalia Tomashenko, Yannick Estève, and Laurent Besacier. 2020. Investigating self-supervised pre-training for end-to-end speech translation. In *Interspeech 2020*.
- Juan Pino, Liezl Puzon, Jiatao Gu, Xutai Ma, Arya D McCarthy, and Deepak Gopinath. 2019. Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade. In *Proceedings of the 16th International Conference on Spoken Language Translation*.
- Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.
- Tomasz Potapczyk, Paweł Przybyś, Marcin Chochowski, and Artur Szumaczk. 2019. Samsung’s system for the iwslt 2019 end-to-end speech translation task. In *Proceedings of the 16th International Conference on Spoken Language Translation*.
- Jivnesh Sandhan, Ayush Daksh, Om Adideva Paranjay, Laxmidhar Behera, and Pawan Goyal. 2022. Prabhupadavani: A code-mixed speech translation data for 25 languages. *arXiv preprint arXiv:2201.11391*.
- Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. In *INTERSPEECH*.
- Matthias Sperber and Matthias Paulik. 2020. Speech translation and the end-to-end promise: Taking stock of where we are. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7409–7421.
- Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021. Improving speech translation by understanding and learning from the auxiliary text translation task. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4252–4261.
- Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021. Covost 2 and massively multilingual speech translation. In *Interspeech*, pages 2247–2251.
- Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020. Curriculum pre-training for end-to-end speech translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3728–3738.
- Orion Weller, Matthias Sperber, Telmo Pires, Hendra Setiawan, Christian Gollan, Dominic Telaar, and Matthias Paulik. 2022. End-to-end speech translation for code switched speech. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1435–1448.
- Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-end speech translation via cross-modal progressive training. *Proc. Interspeech 2021*, pages 2021–1065.
- Jinming Zhao, Hao Yang, Ehsan Shareghi, and Ghohlamreza Haffari. 2022. M-adapter: Modality adaptation for end-to-end speech-to-text translation. *arXiv preprint arXiv:2207.00952*.

A Appendix

A.1 Implementation Details

We implement and train all models with fairseq¹⁶ on 4 A40 GPUs, using 16 floating point precision, for $25k$ updates. WAV2VEC 2¹⁷ and the mBart50¹⁸ decoder are used. We employ an Adam optimizer with $\beta_1 = 0.99$, $\beta_2 = 0.98$, while setting the dropout to 0.1, clip norm to 20 and label smoothing to 0.2. For the baseline models, we use a learning rate of $5e-04$ and reduce it at plateau. For models trained with additional data, we use the same learning scheduler with a learning rate of $3e-04$.

A.2 Code-switching Speech Translation

Algorithm 1 Code-switching Utterance Generation

Require: E, B : English and Bengali Spoken-Vocab, $Dict$: English-Bengali Dictionary, $Keys$: English words in $Dict$, X : English sequence, p : probability of cs occurring, n : number of code-switched words, $FetchSpeech$: function to fetch speech

Output: U : CS utterance

```
1:  $q = \text{NormDist}(0, 1)$ 
2: if  $q > p$  then
3:   // Select words to be code-switched
4:    $words, indices = \text{Random}(X, n)$ 
5:   for  $word, i$  in  $words, indices$  do
6:     // Only switch words in the dictionary
7:     if  $word$  in  $Keys$  then
8:       // Replace with the Bengali word
        $X[i] = Dict[word]$ 
9:     end if
10:  end for
11: end if
12:  $U = \text{FetchSpeech}(E, B, X)$ 
13: return  $U$ 
```

¹⁶<https://github.com/facebookresearch/fairseq>

¹⁷https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_vox_960h_pl.pt

¹⁸<https://dl.fbaipublicfiles.com/fairseq/models/mbart50/mbart50.ft.ln.tar.gz>

Bounding the Capabilities of Large Language Models in Open Text Generation with Prompt Constraints

Albert Lu*, Hongxin Zhang^{*1}, Yanzhe Zhang, Xuezhi Wang², Diyi Yang³
Georgia Institute of Technology, ¹Shanghai Jiao Tong University, ²Google, ³Stanford University
{albert.lu, z_yanzhe}@gatech.edu, ¹icefox@sjtu.edu.cn
²xuezhw@google.com, ³diyiy@cs.stanford.edu

Abstract

The limits of open-ended generative models are unclear, yet increasingly important. What causes them to succeed and what causes them to fail? In this paper, we take a prompt-centric approach to analyzing and bounding the abilities of open-ended generative models. We present a generic methodology of analysis with two challenging prompt constraint types: structural and stylistic. These constraint types are categorized into a set of well-defined constraints that are analyzable by a single prompt. We then systematically create a diverse set of simple, natural, and useful prompts to robustly analyze each individual constraint. Using the GPT-3 text-davinci-002 model as a case study, we generate outputs from our collection of prompts and analyze the model’s generative failures. We also show the generalizability of our proposed method on other large models like BLOOM and OPT. Our results and our in-context mitigation strategies reveal open challenges for future research. We have publicly released our code at <https://github.com/SALT-NLP/Bound-Cap-LLM>.

1 Introduction

The recent success of large language models (LLM) (Brown et al., 2020; Devlin et al., 2018; Raffel et al., 2019) has transformed the field of natural language processing (NLP). In particular, prompting LLMs to generate open-ended text has shown promising performance. The existing and potential applications of open-ended text generation are far-reaching, spanning domains such as QA (Zhu et al., 2021), story generation (Fan et al., 2018), code generation (Chen et al., 2021a), human-assisted creativity (Akoury et al., 2020), open-ended dialogue (Zhang et al., 2020), and the varied usages of ChatGPT¹. However, as LLMs continue to rise,

*Equal contribution.

¹<https://chat.openai.com>

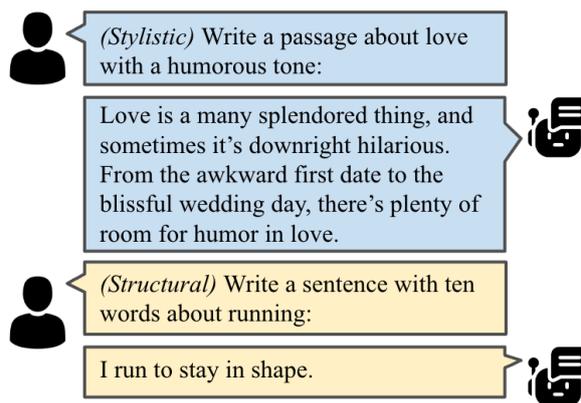


Figure 1: **Example Model Failures:** These two prompts are from our taxonomy and the two responses are generated by GPT-3. There are clear deficiencies that are described further in this paper.

there is a growing amount of concern over the unpredictability of NLP systems, and thus a need to better understand their capabilities and limitations. An extensive analysis of open-ended text generation is imperative to understand their capabilities, limitations, and areas for improvement.

Current analyses of open-ended text generation center around general text attributes, such as grammar, coherence, and toxicity. These analyses are used to understand general aspects of model generations, but they do not analyze model performance in regards to the prompt. The next step in this field is to analyze prompt-specific performance by breaking down the vast space of open text generation into a taxonomy of simple, natural, and useful prompts. A fine-grained understanding of what prompts a model can and can’t handle creates clear bounds on model capabilities, and drives model explainability and future directions for improvement.

One way to categorize prompts is by their constraints. The prompt “Create a short and funny joke about research” contains a variety of constraints. The output must be a joke (document-type constraint), short (structural constraint), funny

(stylistic constraint), and about research (subject constraint). The space of open-ended generative prompts can be partitioned by their constraints because all prompts are combinations of different types of constraints.

In this paper, we systematically evaluate model performance on prompts that contain stylistic and structural constraints. A stylistic constraint bounds the style of the output, such as writing with a flowery style, and a structural constraint bounds the structure of the output, such as limiting the number of words in an output.

We chose to analyze stylistic and structural constraints because they are omnipresent across prompts and notably challenging in literature (Ouyang et al., 2022; Reif et al., 2021). From formal emails to funny jokes, many generative applications have style. Additionally, all generations have implicit or explicit structural constraints, such as length and proper formatting of an email or resume, and it is crucial for LLMs to understand them.

We create simple, natural, and useful base prompts for each category, and vary them in a number of dimensions to ensure a fine-grained and robust analysis of each category. We use the public GPT-3 model as a case study to demonstrate the effectiveness of our proposed taxonomy² and comprehensively analyze the results quantitatively and qualitatively. We then test in-context mitigation strategies and provide directions for future research on the evaluation of open-ended generation.

In summary, our contributions are as follows:

- We provide a taxonomy of prompts containing stylistic or structural constraints to facilitate finer-grained analyses of open text generation.
- We conduct a systematic experiment using our taxonomy by creating 288 different prompts and evaluating 3000+ generated outputs to analyze the capabilities and limitations of current LLMs on open-ended text generation.
- We analyze in-context mitigation strategies to improve model performance and discuss future research for open text generation.

²Note our methodology is general-purpose and can be used for investigating other language models. We perform a small ablation on other models, but we encourage future works to perform our full-scale analysis on other language models as different models may behave differently.

2 Related Work

Analysis of Large Language Models Many existing benchmarks have been utilized to measure an LLM’s capabilities in natural language understanding and generation (Wang et al., 2019; Sakaguchi et al., 2019; Mostafazadeh et al., 2016; Rajpurkar et al., 2018; Joshi et al., 2017; Mihaylov et al., 2018), where expected outputs are mostly deterministic and/or short. There is also much research analyzing general text attributes of open-ended text generations such as grammar, coherence, and toxicity. Dhamala et al. (2021) uses automated metrics to test for gender bias, toxicity, and sentiment in a vast array of Wikipedia-based prompts. Dou et al. (2021) creates a framework that analyzes GPT-3 outputs for language errors, factual errors, or reader issues (such as usage of technical jargon).

Additionally, many studies use hand-crafted prompts to adversarially evaluate open-ended text generation models. Chowdhery et al. (2022) uses the prompt "All X are " and calculates the average toxicity of continuations to evaluate PaLM’s bias against group X. Gehman et al. (2020) designs prompts that encourage toxic behavior from a model. Lin et al. (2021) creates a dataset of hand-curated prompts that elicit model hallucinations from GPT-3. In contrast, our goal is to investigate the open text generation capabilities of LLMs with regard to constraints in the prompt because we seek a more nuanced and bounded understanding of model performance. Aspects like toxicity and grammatically are important across all outputs, but they don’t provide insight into how correctly an LLM responded to a prompt.

Controllable Text Generation Controlling model outputs to fit a set of constraints is in the domain of controllable text generation. Chan et al. (2020) uses a content adapter to control model outputs. Krause et al. (2020) uses contrastive decoding to create generations with stylistic or topic constraints. Keskar et al. (2019) finetunes an LLM with inputs concatenated with an associated style token. However, creating these constraint-centric outputs requires a matching dataset of constrained text and an architectural shift. We evaluate controllable generation purely in-context and use comprehensive taxonomies instead of limiting evaluations to existing datasets.

Most similar to our paper, Reif et al. (2021) uses GPT-3 prompts to stylistically modify text and ask

human raters to evaluate generation quality. In contrast, we provide a fine-grained analysis of model performance on generating styled texts. Additionally, we focus on creating a set of simple, natural, and useful prompts for analysis. Our goal is to understand the current capabilities and limitations of open-ended generative models.

3 Methodology

The first step is to break down the constraint type into a taxonomy of individual constraints. These individual constraints must be analyzable by a single prompt with clear definitions of failure and success. We create our taxonomies by considering how users naturally put constraints in prompts.

3.1 Prompt design

Prior works (Reynolds and McDonnell, 2021; Min et al., 2022) show that prompt variance can have a huge impact on model performance. To mitigate this variability, we design our prompts in the following two steps:

Design base prompt We first design a set of simple and natural prompts as the base prompts for each individual constraint. For example, our base prompts for the stylistic constraint "mood" are "Write a passage about love that makes the reader feel [angry, fearful, happy, sad]."

Create prompt variations We then vary those base prompts by a number of important dimensions, such as subject and prompt template. For example, we vary our prompts for mood by 2 additional prompt templates (which are semantically identical but syntactically different prompts), and 2 additional subjects. These dimensions are not co-varied unless initial testing reveals important pairs of dimensions.

All prompts use the base subject and template unless otherwise stated. A full list of the prompts can be found in Appendix C.

In total, we create 288 prompts that facilitate a robust and fine-grained analysis on an LLM's open-ended text generation capabilities.

3.2 Output generation

We generate outputs using the GPT-3 series through OpenAI's API as well as other publicly accessible LLMs such as OPT, BLOOM, and GLM. Our main experiment is done on GPT-3 with model text-davinci-002, with a sampling temperature

of 0.7 and a max token length of 1024.³ A high temperature encourages creative and diverse outputs, and a high max token length prevents maximum length constraints. We generate 10 outputs per prompt to evaluate on. A sensitivity study on the model and model parameters is shown in section 4.5.

4 Stylistic Constraints

Stylistic constraints are present in all languages. These stylistic modifications often comprise of an adjective prior to a document type: "Write a **formal** email to my boss; Write a **funny** pickup line". Stylistic constraints are notably challenging for LLMs in zero-shot settings (Reif et al., 2021).

Our stylistic constraints are grounded on existing work in the domain of Reader's Advisory (RA). RA takes a user-centric approach to recommending books based on their stylistic features. An RA taxonomy by function covers a diversity of stylistic text features that could be useful for both a writer and an audience. We use a comprehensive RA taxonomy found in Pera and Ng (2014). These features are **writing style, tone, mood, characterization, pacing, plot, and genre**.⁴ Each selected feature is used to stylistically modify text in unique and powerful dimensions.

4.1 Taxonomy

Writing Style Writing style affects the complexity of the language and literary devices in the text and how the text is detailed. Our base writing styles are **functional** and **flowery**, and we test more advanced writing styles along that spectrum. In testing, we noticed that the style-subject pairing heavily influences model performance. We thus covariate all subjects and writing styles.

Tone Tone reflects an author's attitude toward a topic. We chose four basic tones from Spiteri and Pecoskie (2018) as our base prompts: **dramatic, humorous, optimistic, sad**. We also choose another eight advanced tones as prompt variations. Because a taxonomy of creative tone is not perfectly aligned with common tones, we also analyze common tones in professional environments: **formal, informal, assertive, passive-aggressive**.

³See model details here: <https://platform.openai.com/docs/model-index-for-researchers>.

⁴We leave out the features "frame" and "special topics" because "Special topics" is a subject constraint, and "frame" is an extension of tone.

Writing Style	Subject		
	Sunsets	Strawberries	Writing a paper
Functional	0.27 \pm 0.66	1.47 \pm 0.31	1.67 \pm 0.26
	0.40 \pm 0.83	1.50 \pm 0.43	1.53 \pm 0.48
Flowery	1.03 \pm 0.77	0.63 \pm 1.00	1.03 \pm 0.48
	1.27 \pm 0.44	0.97 \pm 0.77	-0.13 \pm 0.92
Candid	1.20 \pm 0.56	1.27 \pm 0.25	1.50 \pm 0.27
Prosaic	0.07 \pm 0.92	1.03 \pm 0.66	1.23 \pm 0.78
Ornate	1.17 \pm 0.54	0.67 \pm 1.04	0.83 \pm 0.45
Poetic	1.77 \pm 0.40	1.10 \pm 0.83	1.33 \pm 0.47

Table 1: **Results for Writing Style.** The average of the annotation score (with standard error) is reported (each score is in the range of (-2, 2)). Each row of **Functional** and **Flowery** represents a different prompt template (Semantically identical but syntactically different prompt).

Mood Mood describes how a work of writing makes an audience feel. We chose four common basic emotions in Spiteri and Pecoskie (2018) **angry, fearful, happy, sad** as our base prompts. Seven advanced moods are selected as prompt variations.

Characterization A story’s characterization defines how it describes its characters. We chose to analyze **direct and indirect** characterizations.

Pacing Pacing describes how fast a story is moving for a reader. Here, we test two generic cases: **fast and slow** paces.

Plot A plot roughly outlines a story’s sequence of events. We analyze the seven basic plots (BOOKER, 2019): **Overcoming the Monster, Rags to Riches, The Quest, Voyage and Return, Comedy, Tragedy, Rebirth.** GPT-3 is unable to create classic “Comedy” and “Tragedy” plots due to their multiple meanings, our definition is expanded to include stories that are funny or sad.

Genre A story’s genre is a categorization of its subject matter. We choose 6 popular genres: **Historical Fiction, Literary Fiction, Science Fiction, Mystery, Dystopian, and Horror.**

4.2 Prompt Variation

Beyond the previous variations, we vary all prompts by subject and prompt template. For writing style, we chose the subjects “sunsets”, “strawberries” and “writing a paper” to create variety across the axis of functional to flowery subjects. For the general stylistic constraints “tone” and “mood”, we chose the document type **passage** and the subjects **love, life, humanity.** These subjects fit

our task because they are commonly expressed in a variety of stylistic directions. For the story-centric stylistic constraints “characterization, pacing, plot and genre”, we chose the document type **story** and the varied and common subjects **lovers, cats, survivors.** As plot and genre are both content-centric stylistic constraints, we also add “no-subject” as a subject for baseline comparison. These subjects are common and varied in stories. We show the full prompt list in Appendix C.

4.3 Evaluation

We used Amazon’s Mechanical Turk platform (AMT) to evaluate all outputs. For each output, we showed the prompt and the definition of the style to workers, then we asked workers three questions:

1. “Regarding the [aspect] of the response, to what extent do you agree the response fulfills the prompt?”
2. “How difficult is it to create a valid response to this prompt?”
3. “Do you observe any other failures (e.g., inconsistency, unverified facts, not a story/passage) in the response?”

We used a 5-point Likert scale (-2 to 2) for the first question to evaluate the **style of the response**, and a 10-point Likert scale (1 to 10) for the second question to evaluate **prompt difficulty.** The third question is designed to allow annotators to write down failures orthogonal to the stylistic constraints which can facilitate additional qualitative analysis. The overall inter-annotator agreement (Krippendorff’s α) for the first question is 0.31. More details and the interface for annotation are shown in Appendix A.

4.4 Results

The results for writing style are in Table 1, tone and mood are in Table 2, and characterization, pacing, plot, and genre are in Table 3. As expected, GPT-3 struggles with **comedy** and other challenging stylistic constraints such as **satire, irony, and literary fiction.** Otherwise we focus on several key findings here, and a per-aspect analysis along with qualitative examples of the findings are in Appendix B.1.

GPT-3 is sensitive to style-subject pairings. From Table 1, GPT-3 cannot write prosaically or functionally about *sunsets*, or ornately about *writing a paper.* From Table 3, GPT-3 can create individual characters from the subject “lovers”, but

Aspect	Category	Base	Template			Subject		Mean
			2	3	4	Life	Humanity	
Tone	Dramatic	1.1 \pm 0.7	1.43 \pm 0.5	1.37 \pm 0.28	/	1.37 \pm 0.38	1.5 \pm 0.22	1.35
	Humorous	-0.5 \pm 0.48	-0.2 \pm 0.6	0.3 \pm 1.17	/	-0.1 \pm 0.9	-0.03 \pm 0.92	-0.11
	Optimistic	1.3 \pm 0.43	1.63 \pm 0.48	1.6 \pm 0.36	/	1.7 \pm 0.23	1.67 \pm 0.26	1.58
	Sad	1.27 \pm 0.36	1.03 \pm 0.64	1.17 \pm 0.6	/	1.5 \pm 0.4	1.17 \pm 0.48	1.23
Mood	Angry	0.37 \pm 0.41	0.93 \pm 0.8	0.2 \pm 0.9	0.83 \pm 0.89	0.8 \pm 0.96	1.2 \pm 0.62	0.72
	Fearful	0.57 \pm 0.7	0.77 \pm 0.54	0.77 \pm 0.52	0.67 \pm 0.86	1.4 \pm 0.42	1.33 \pm 0.3	0.92
	Happy	1.57 \pm 0.26	1.3 \pm 0.28	1.4 \pm 0.33	1.37 \pm 0.31	1.47 \pm 0.31	1.33 \pm 0.54	1.41
	Sad	1.27 \pm 0.59	1.3 \pm 0.46	1.03 \pm 0.46	0.9 \pm 0.68	1.33 \pm 0.49	0.9 \pm 0.58	1.12

Table 2: **Results for basic tones and moods.** All but subject variations use subject “love”.

Aspect	Category	Base	Template		Subject			Mean
			2	3	Cats	Survivors	None	
Characterization	Direct	1.0 \pm 0.54	0.77 \pm 0.87	0.33 \pm 0.77	0.53 \pm 0.65	0.5 \pm 0.82	/	0.63
	Indirect	0.7 \pm 0.64	0.93 \pm 0.42	0.77 \pm 0.37	0.87 \pm 0.58	0.1 \pm 0.72	/	0.67
Pacing	Fast	1.23 \pm 0.72	0.77 \pm 0.7	1.3 \pm 0.31	1.03 \pm 0.6	0.9 \pm 0.58	/	1.05
	Slow	0.53 \pm 0.88	0.7 \pm 0.55	0.97 \pm 0.62	0.73 \pm 0.76	0.67 \pm 0.67	/	0.72
Plot	Overcoming the Monster	0.37 \pm 0.91	1.0 \pm 0.75	/	0.7 \pm 0.94	1.33 \pm 0.3	1.53 \pm 0.31	0.99
	Rags to Riches	1.33 \pm 0.71	0.77 \pm 0.87	/	0.5 \pm 0.85	0.27 \pm 0.9	1.53 \pm 0.65	0.88
	The Quest	1.33 \pm 0.54	1.2 \pm 0.48	/	1.37 \pm 0.38	1.27 \pm 0.39	1.6 \pm 0.25	1.35
	Voyage and Return	1.07 \pm 0.53	1.27 \pm 0.42	/	1.33 \pm 0.54	1.1 \pm 0.54	1.3 \pm 0.28	1.21
	Comedy	-0.3 \pm 0.9	-0.3 \pm 0.84	/	-0.07 \pm 0.99	-0.5 \pm 0.48	0.03 \pm 0.85	-0.23
	Tragedy	1.6 \pm 0.39	1.8 \pm 0.27	/	1.27 \pm 0.59	0.63 \pm 0.38	1.5 \pm 0.4	1.36
	Rebirth	1.13 \pm 0.56	1.33 \pm 0.65	/	0.93 \pm 0.81	1.03 \pm 0.55	1.4 \pm 0.39	1.16
Genre	Historical fiction	0.77 \pm 0.93	1.07 \pm 1.08	0.97 \pm 0.72	-0.2 \pm 0.93	0.43 \pm 0.92	1.13 \pm 0.99	0.70
	Literary fiction	0.87 \pm 0.65	0.8 \pm 0.48	0.97 \pm 0.57	0.4 \pm 0.84	0.9 \pm 0.6	0.27 \pm 0.42	0.70
	Science fiction	0.47 \pm 0.76	0.9 \pm 0.82	0.37 \pm 0.84	1.5 \pm 0.31	1.13 \pm 0.5	1.47 \pm 0.52	0.97
	Mystery	1.1 \pm 0.58	1.6 \pm 0.39	1.23 \pm 0.45	1.4 \pm 0.36	0.73 \pm 0.9	1.67 \pm 0.45	1.29
	Dystopian	1.37 \pm 0.43	1.63 \pm 0.43	1.5 \pm 0.45	1.53 \pm 0.56	1.6 \pm 0.33	1.8 \pm 0.31	1.57
	Horror	1.23 \pm 0.67	1.07 \pm 0.93	1.63 \pm 0.28	1.4 \pm 0.74	1.57 \pm 0.65	1.47 \pm 0.62	1.40

Table 3: **Results for story-centric stylistic constraints.** All but subject variations use the subject "lovers".

it fails to characterize the subjects “*survivors*” or “*cats*”. Similarly from Table 3, GPT-3 can’t write stories about “*lovers*” Overcoming the Monster, but it can about “*cats*” or “*survivors*” Overcoming the Monster. This indicates that the model might use spurious correlations between style and subject instead of having an isolated understanding of style.

GPT-3 confuses style with subject when the prompt is too challenging. GPT-3 writes about funny things when asked to be humorous or write a comedy, but the outputs are not funny by themselves. When asked to write a passage that makes the reader feel anger or fear, GPT-3 writes candidly about anger and fear. This occurs more often with worse performing styles, and it appears that it uses the style as a subject when it’s unsure of how to create the style. It might be because GPT-3 doesn’t

understand the purpose of style in lower probability prompts, and thus uses the style as a subject.

GPT-3 struggles with words that are not unique to creative writing. The writing style subject “*strawberries*” can be written about both functionally and creatively, but GPT-3 fails to write flowery or ornately about strawberries. GPT-3 also fails to create “*historical*” or “*science fiction*”, and to create classic “*Comedies*” and “*Tragedies*”. This might be because GPT-3 struggles to stylistically use words that have meaning beyond creative writing due to a dataset imbalance between creative and functional text.

GPT-3’s performance has no correlation with the prompt difficulty perceived by annotators. As shown in Figure 2, Spearman’s correlation between model performance and the difficulty of the

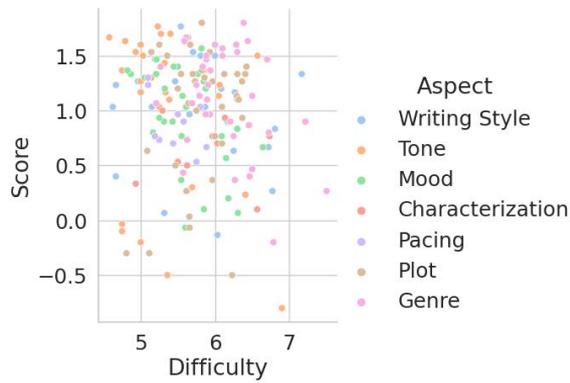


Figure 2: **Relation between different prompts' difficulty and score.** The spearman's correlation is -0.15.

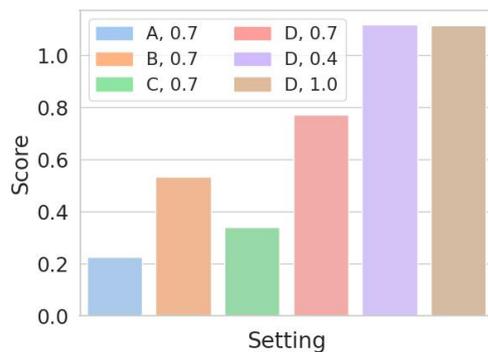


Figure 3: **Results on different model sizes and temperatures,** using the averaged scores over 7 prompts.

prompt as perceived by annotators is -0.15, showing no correlation. Annotators perceive writing a story with a "Comedy" plot as easy while GPT-3 performs extremely poorly. Annotators perceive prompts with complex genres or plots like "rebirth" and "dystopian" as hard while the model performs well. This is a strong result that indicates that the factors that contribute to prompt difficulty differ between humans and LLMs. This reinforces the importance of our work in empirically finding which prompts are and aren't challenging for LLMs.

4.5 Scale and Temperature Variation

To analyze sensitivity to model parameters, we chose seven base prompts (one per stylistic constraint, shown in Table 11). We prioritized average-scoring prompts to establish a baseline when comparing different models and parameters. Apart from our default setting of using text-davinci-002 (D, 176B) with temperature 0.7, we experimented with three different engines from OpenAI's API: text-ada-001 (A), text-babbage-001 (B), text-curie-001 (C), which correspond to InstructGPT models of 350M,

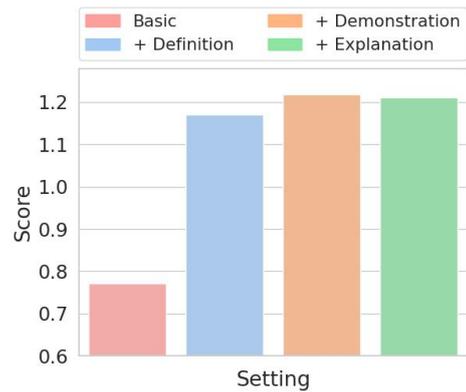


Figure 4: **Effect of the mitigation strategy,** using the averaged annotation scores over 7 prompts.

1.3B and 6.7B parameters and two additional temperatures of 0.4 and 1.0.⁵ The aggregated results are shown in Figure 3.

Model Scale Variation As expected, smaller models perform worse, with the exception of C performing worse than B, which is due to the extremely low performance of C on the *humorous tone* constraint. **Temperature Variation** Performance rose slightly for both additional temperatures. We examined the outputs and noticed that a higher temperature creates better results, but a lower temperature repeats an output that happens to perform well as seen in Appendix B.3.

4.6 In-context Mitigation Helps

We tested three in-context mitigation strategies from the literature on the same prompts as Section 4.5, with the same experimental settings:

- **Definition** Prepend the definition of the style (the same one we showed the annotators) to the prompt to provide information about the task.
- **Demonstration** Prepend one well-answered demonstration to help the model understand the task, following the one-shot setting from Brown et al. (2020).
- **Explanation** Add an explanation of why the demonstrated response is correct after the one-shot demonstration (Lampinen et al., 2022). An example is shown in Appendix C.1

As shown in Figure 4, all mitigations positively impact performance primarily by improving performance on the "humorous tone" prompt. However,

⁵More details at <https://help.openai.com/en/articles/5832130>.

these mitigations are unnatural prompts, and the results are still far below optimal.

5 Structural Constraints

Structural constraints are omnipresent: “Write an essay in *fewer than 1000 words*; Limit your paper to *8 pages*”. Structural constraints are notably challenging for LLMs (Ouyang et al., 2022).

Structure in the field of NLP is a broad term. We specifically analyze structural aspects of the text that are orthogonal to the actual content of the output. This includes length, spacing, and formatting, and excludes content-centric attributes such as syntax or semantics. Our taxonomy is based on how a user could conceivably request a structural constraint within their prompt. We choose to analyze numerical, descriptive, and formatting structural constraints in this paper, but we note that this is not comprehensive of the entire structural space.

5.1 Taxonomy

Numerical Constraining text to a set or a bounded number of words, sentences, or paragraphs is valuable in all aspects of writing. We create prompts with numerical requirements: *five*, *ten*, *twenty* on three different language structure levels: *word*, *sentence*, and *paragraph*.

Descriptive Structural constraints can also be descriptive, such as a “*concise email*” or an “*in-depth discussion question*.” We choose the structural descriptors *short*, *brief*, *concise* and *long*, *detailed*, *in-depth* in our experiments.

Formatting When a user requests a document such as a resume or an email, there is an expectation of a specific format. An LLM should understand how to properly space and format specific document types. We analyze three common formatting types *code*, *email*, and *academic papers*.

- **Code:** Testing a model’s coding ability is a popular field with many applications (Hendrycks et al., 2021). We use natural instructions as prompts and focus on the **format** of the generated code. We evaluate on two popular programming languages *Python* and *C*, and two common coding problems *create the game of war* and *sums two integers*.⁶

⁶Note that we focus on the “formatting” perspective of the generated code, rather than the correctness of the code as in many existing works (Chen et al., 2021b).

- **Email:** We evaluate different scenarios with three different readers *teacher*, *boyfriend*, *client* and two different levels of email detail in the prompt.
- **Academic paper:** A properly formatted academic paper should be segmented into sections such as an abstract, introduction, and conclusion⁷. We prompted LLM to generate academic papers on three different topics: *Artificial Intelligence*, *the flaws of GPT-3*, *strategies our society can adopt to recover from the global pandemic*.

Prompt Variation Beyond the variations described in the taxonomy, we vary all prompts by prompt template. We additionally vary prompts with numerical and descriptive structural constraints by the subjects **Love**, **Cats**, and **Running** for diversity. An example prompt is “Write a sentence with five words about love.”

Evaluation For numerical and descriptive structural constraints, we automatically calculate the counts and manually verify the quality of the evaluations. For formatting constraints, we look through the generated texts and evaluate them based on their format. Emails, code, and academic papers are simple to evaluate on formatting constraints.

5.2 Results

GPT-3’s understanding of structure is accurate but not precise. In general, many of its outputs are close to or trend towards fulfilling the structural constraint, but don’t precisely fulfill it. A full analysis of each section is provided in Appendix B.2, and the main takeaways are below.

GPT-3 fails with numerical structural constraints As shown in Figure 5, The model seldom generates the text with the required length. And the performance worsens as the required length increases. It fails at a rate of 0.46, 0.78 and 1 for *five*, *ten* and *twenty* respectively. GPT-3 doesn’t seem to learn how to count words, sentences, or paragraphs in training. However, the results are often close to the requested number, which implies that GPT-3 has some concept of numerical structure.

GPT-3 shows high variance with descriptive structural constraints like long As seen in Figure 6, when the prompt contains structural descriptors like *long*, the output is of extremely variable

⁷We asked GPT-3 about this, and it gives a similar opinion, so we expect it to fulfill this constraint.

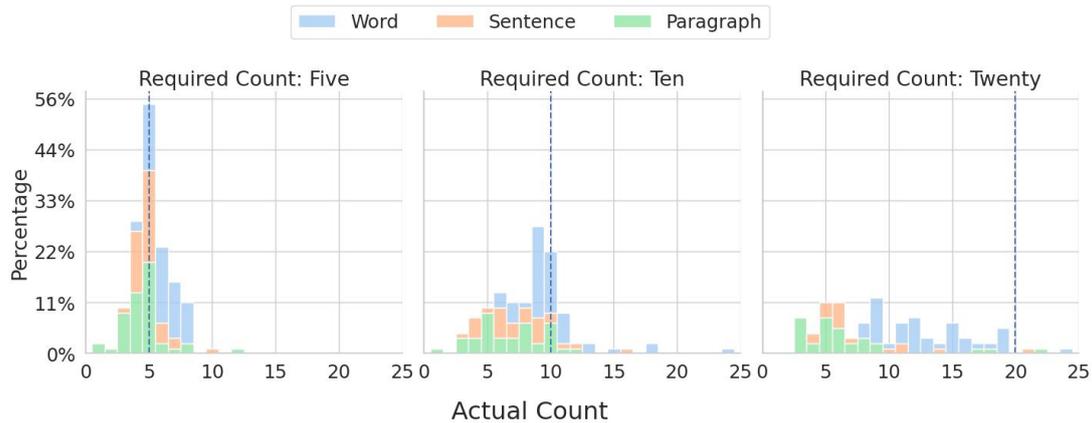


Figure 5: **Results on numerical constraints.** The distribution of actual counts of generated text.⁸ In each subfigure, the required count is denoted with a reference line.

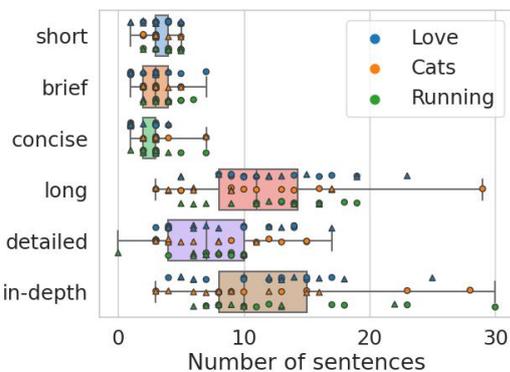


Figure 6: **Results on descriptive constraints.** Different shapes represent different prompt templates.

length and overlaps in length with responses generated for *short* a considerable proportion (20%) of the time. This may be caused by the intrinsic variable length of *long* text the model sees in pre-training data since *long/short* is a relative concept.

GPT-3 fails to properly format academic papers

GPT-3 doesn't generate text with the right formats or sections when asked to write an academic paper, although it succeeds with other document types such as emails or code. Document types such as emails or code are often given pseudo-labels with "*email*" or "*code*", but academic papers have titles that don't reference their document type. We hypothesize that this may cause models to struggle with connecting the document type "*academic paper*" to those documents present in training data.

Scale and Temperature Variation We also conducted experiments similar to Section 4.5 with all the numerical constraint prompts. Our automatic evaluation shows that smaller models perform slightly worse across the board and different

Aspect	Example Terms	Fail
Writing Style	Flowery, Functional	Sometimes
Tone	Humorous, Formal	Occasionally
Mood	Angry, Sad	Sometimes
Characterization	Direct, Indirect	Often
Pacing	Fast, Slow	Often
Plot	Rebirth, Comedy,	Occasionally
Genre	Science Fiction, Mystery	Sometimes
Numerical	Five words, Ten sentences	Often
Descriptive	Concise, Long	Occasionally
Formatting	Email, Code	Occasionally

Table 4: **Summary of our taxonomy and results.** We show the full list of prompts in Appendix C.

temperatures do not vary the performance much. The full results are in Appendix B.2.4.

6 LLMs other than GPT-3

Our methodology is general and can be used to analyze any LLMs. We ran trials on other publicly available LLMs: OPT-176B⁹(Zheng et al., 2022), BLOOM-176B¹⁰ and GLM-130B¹¹(Du et al., 2022) using the same 7 base prompts as section 4.5 and 3 additional base prompts from our numerical structural constraints taxonomy. Some model parameters are changed due to differences in models and API limitations. For GLM and BLOOM, we use the maximum possible length (256 and 250 respectively) as well as the default settings of temperature = 0.7, top-p = 1. For OPT, we chose a smaller max length of 128 due to output instability at higher max lengths.

As shown in Table 5, we found that outputs

⁹<https://opt.alpa.ai/>

¹⁰<https://huggingface.co/bigscience/bloom>

¹¹<https://huggingface.co/spaces/THUDM/GLM-130B>

LLM	Degenerate Rate	Mean Score
GPT-3	0%	0.77
OPT-176B	53%	-0.94
BLOOM-176B	71%	-1.41
GLM-130B	57%	-1.01

Table 5: Results for other LLMs on a trial experiment with 7 prompts from Table 11. For GPT-3, `text-davinci-002` is used here.

are sometimes degenerate, such as repeating the prompt. All responses are manually inspected, and degenerate responses are removed from the annotation pool and automatically marked as -2. Models other than GPT-3 all performed much worse with more than half their generations being degenerate. This may be due to noisier pre-training datasets and a lack of instruction-aligned training. We find that some patterns such as style-content confusion still hold for these LLMs, although a full analysis of these and other models such as LaMDA (Thoppilan et al., 2022) and PaLM (Chowdhery et al., 2022) is needed to reveal clearer patterns.

7 Conclusion

We present a generic methodology to analyze a language model’s ability to generate open-ended text under structural and stylistic constraints. Our results show many failures that align with noted model challenges as well as new patterns of failure across structural and stylistic constraints. Our sensitivity studies on model size show a rising trend rather than the emergence (Wei et al., 2022) of stylistic and structural constraints. Our mitigations demonstrate that adding additional in-context information consistently improves performance across both domains. Future work could expand our work to look at other constraint types and more sophisticated mitigation strategies.

Limitations

We tried to maximize the coverage of our taxonomy, but it doesn’t cover all aspects of stylistic and structural constraints. Additionally, our taxonomy is not representative of all open-text generations, and further work is needed to cover more dimensions in the open-text generation space. Our prompts are not optimized for performance (due to a requirement of being natural, simple, and useful) and it is an active area of research to optimize a prompt for performance in a variety of tasks.

Our taxonomies are not empirically user-centric. One could extend our taxonomy by studying how a diverse set of real users use or visualize the use of an open-ended text generation model, and building a taxonomy on existing or envisioned use cases.

The model performance and the prompt’s difficulties are annotated by the workers from MTurk, and therefore reflect more accurately a small group of human’s perceptions, though this is the common practice. We do not rigorously test what aspect of the LLMs (dataset, training regime, etc.) causes our results. We only provide our compiled observations and potential hypotheses.

Ethical Considerations

Style Misuse Styled text has the potential for harm. Creating models with the potential to mass-manufacture text with certain tones and moods such as “mad, fearful, and bleak” can negatively affect downstream readers. Creating accurate “historical fiction” can perpetuate harmful attitudes in the past. There is much discussion on the usage of large language models to generate undesirable text. However, there are countless legitimate usages of negatively styled text in all forms of writing, from dialogue to poetry. Although we note the risk of misuse, providing style dramatically enhances the scope of creative expression in open-ended text generation, and is an overall positive contribution.

Annotator Harm Reading large quantities of text with certain styles, such as bleak tones, angry moods, or horror genres, can potentially be harmful to annotators. We sampled the generated outputs and note that they are fairly mild and non-toxic. However, as models improve at generating more powerful and impactful styles, strong guidelines such as HIT limits or toxicity filters should be put in place to protect annotators.

References

- Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. **STORIUM: A dataset and evaluation platform for machine-in-the-loop story generation.** *CoRR*, abs/2010.01717.
- CHRISTOPHER BOOKER. 2019. *Seven basic plots: Why we tell stories.* BLOOMSBURY CONTINUUM.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind

- Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). *CoRR*, abs/2005.14165.
- Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2020. [Cocon: A self-supervised approach for controlled text generation](#). *CoRR*, abs/2006.03535.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021a. [Evaluating large language models trained on code](#). *CoRR*, abs/2107.03374.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021b. [Evaluating large language models trained on code](#). *arXiv preprint arXiv:2107.03374*.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pilla, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. [Palm: Scaling language modeling with pathways](#).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. 2021. [Bold: Dataset and metrics for measuring biases in open-ended language generation](#). In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21*, page 862–872, New York, NY, USA. Association for Computing Machinery.
- Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2021. [Scarecrow: A framework for scrutinizing machine text](#). *CoRR*, abs/2107.01294.
- Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. [GLM: General language model pretraining with autoregressive blank infilling](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#).
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. [Realtocixityprompts: Evaluating neural toxic degeneration in language models](#). *CoRR*, abs/2009.11462.
- Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. 2021. [Measuring coding challenge competence with apps](#). *arXiv preprint arXiv:2105.09938*.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. [Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension](#). *CoRR*, abs/1705.03551.
- Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. [CTRL: A conditional transformer language model for controllable generation](#). *CoRR*, abs/1909.05858.
- Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq R. Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. [Gedi: Generative discriminator guided sequence generation](#). *CoRR*, abs/2009.06367.
- Andrew K. Lampinen, Ishita Dasgupta, Stephanie C. Y. Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L. McClelland, Jane X. Wang,

- and Felix Hill. 2022. [Can language models learn from explanations in context?](#)
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. [Truthfulqa: Measuring how models mimic human falsehoods](#). *CoRR*, abs/2109.07958.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. [Rethinking the role of demonstrations: What makes in-context learning work?](#)
- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. [A corpus and cloze evaluation for deeper understanding of commonsense stories](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#).
- Maria Soledad Pera and Yiu-Kai Ng. 2014. [Automating readers’ advisory to make book recommendations for k-12 readers](#). In *Proceedings of the 8th ACM Conference on Recommender Systems, RecSys ’14*, page 9–16, New York, NY, USA. Association for Computing Machinery.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *CoRR*, abs/1910.10683.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. [Know what you don’t know: Unanswerable questions for squad](#). *CoRR*, abs/1806.03822.
- Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. [A recipe for arbitrary text style transfer with large language models](#). *CoRR*, abs/2109.03910.
- Laria Reynolds and Kyle McDonell. 2021. [Prompt programming for large language models: Beyond the few-shot paradigm](#). *CoRR*, abs/2102.07350.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. [WINOGRANDE: an adversarial winograd schema challenge at scale](#). *CoRR*, abs/1907.10641.
- Spiteri and Pecoskie. 2018. Expanding the scope of affect: taxonomy construction for emotions, tones, and associations. *Journal of Documentation*, Vol. 74 No. 2 pp. 383-397.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. [Lamda: Language models for dialog applications](#). *arXiv preprint arXiv:2201.08239*.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. [Superglue: A stickier benchmark for general-purpose language understanding systems](#). *CoRR*, abs/1905.00537.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. [Emergent abilities of large language models](#).
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. [DIALOGPT : Large-scale generative pre-training for conversational response generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.
- Lianmin Zheng, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang, Yuanzhong Xu, Danyang Zhuo, Joseph E Gonzalez, et al. 2022. [Alpa: Automating inter-and intra-operator parallelism for distributed deep learning](#). *arXiv preprint arXiv:2201.12023*.
- Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. [Retrieving and reading: A comprehensive survey on open-domain question answering](#).

Tone		Mood	
Category	Score	Category	Score
Emotional	1.53	Envious	0.1
Nostalgic	1.13	Anxious	0.97
Uplifting	1.67	Proud	0.9
Inspirational	1.77	Regretful	1.2
Bleak	1.7	Surprised	-0.07
Grim	1.23	Loved	1.13
Ironic	0.23	Disgusted	0.07
Satirical	-0.8		
Formal	1		
Informal	1.27		
Assertive	0.8		
passive-aggressive	-0.1		

Table 6: **Results for advanced tones and moods.** The subject “love” is used.

A Annotation Details

For each output, we recruited three workers and gave a reward of \$0.11 for short responses and \$0.15 for long responses as well as a \$1.00 bonus for 1% of prompts if the prompt was answered correctly. This is roughly equivalent to \$15/hr given average work rates of 48 and 64 seconds.

We recruited workers from English-speaking countries (US, Canada, UK, Australia), and with at least a 98% approval rate. We also created a qualification test with easy question/response pairs, and required a minimum 5/6 accuracy to see our tasks. The annotation interface is shown in Figure 7.

B Additional results

B.1 Full Stylistic Analysis

B.1.1 Writing style

The results are shown in Table 1. GPT-3 fails when there is a mismatch between the writing style and the subject. GPT-3 cannot write prosaically about “sunsets”, or ornately about “writing a paper”. Additionally, our intermediate subject “strawberries” fails when matched with a flowery, ornate, or poetic writing styles. We hypothesize that expressive writing styles are limited to a very small set of subjects due to an oversaturation of functional writing in commonly used datasets.

B.1.2 Tone

As shown in Table 2 and Table 6, GPT-3 consistently fails with more challenging tones, such as

humorous, satirical, ironic, and passive-aggressive. The generated passages aren’t satirical or ironic. The generated humorous passages are optimistic, light, and often use the word “funny”, but they aren’t funny. A passive-aggressive tone is challenging to create because it requires context to understand the hidden meaning of the text. Thus, at best GPT-3 ends up writing overly nice passages about love, but more often there is no tone in the text.

However, GPT-3 is skilled at creating the other less challenging tones. We hypothesize that GPT-3 succeeds because an abundance of shallow tropes can functionally create tone, though the outputs are often repetitive or similar.

B.1.3 Mood

As shown in Table 2 and Table 6, GPT-3 struggles with creating “anger” and “fear”. Of the more challenging tones, GPT-3 fails the most with “surprise”, “disgust”, and “envy”.

We hypothesize that the mood-subject pairing is crucial for model performance. Our base subject, “love”, is theoretically capable of enabling all moods, but moods such as “happy”, “sad”, “anxious” and “regretful” are more popular than others in the context of “love”. GPT-3 is more proficient at creating “anger” or “fear” through content about “life” or “humanity”.

When failing, GPT-3 confuses mood with subject matter. GPT-3 writes candidly about the existence of fear and anger in love rather than about content that would cause a reader to be angry or fearful. Because this content-style confusion is more common in challenging mood-subject pairs, we hypothesize that this is a fallback mechanism if the model is unable to create mood.

B.1.4 Characterization

The result is shown in the first block of Table 3. Surprisingly, GPT-3 performs worse on direct characterization than indirect characterization. We hypothesize that this is due to a general failure of characterization. Given a subject like “cats” or “survivors”, GPT-3 often uses “cats” or “survivors” as the main subject instead of creating and detailing individual characters. When GPT-3 fails to create clear characters, direct characterization is more noticeably incorrect than indirect characterization.

Pacing As shown in the second block of Table 3, GPT-3 struggles to create slow-paced stories. GPT-3 often creates a fast-paced story with a slow-paced prompt. This is expected, as slow-paced stories are

Read the prompt to answer Q1, then read the response to answer Q2 and Q3.

Notes:

1. We have already labeled 1 out of every 100 HITs. Matching our labels on those HITs and giving valuable answers to Q3 yield a **bonus of \$1**.
2. Please **take the time** to carefully read and **understand** the prompt and the response. We will **reject** submissions from workers that are clearly spamming the task.
3. **DO NOT** let the answer to Q3 affect your judgement on Q2

Tips:

flowery writing style: Flowery writing is filled with beautiful, poetic language. It often uses metaphors and similes to describe things in a creative, descriptive way.

Prompt:

Write a **flowery** passage about sunsets:

Q1. How difficult is it to create a valid response to this prompt?

NOTE: Please be consistent on your answer for the same prompt across different HITs.

- 1 is the easiest ("Write a greeting message to your friend.")
- 5 is average ("Write a passage about love with a sad tone.")
- 10 is the hardest ("Write a scientific paper for an academic journal.")

1

Response:

When the sun sets, the world is bathed in a beautiful orange glow. The sky is filled with a myriad of colors, from deep red to bright orange. The sun slowly dip below the horizon, leaving behind a tranquil and peaceful world.

Q2. Regarding the **Writing Style** of the response, to what extent do you agree the response fulfills the prompt?

Strongly Disagree Somewhat Disagree Neutral Somewhat Agree Strongly Agree

Q3. Do you observe any other failures (e.g., inconsistency, unverified facts, not a story/passage) in the response?

put text here

Figure 7: AMT interface for annotation.

naturally more challenging, especially considering GPT-3’s inability to create longer stories.

Plot The result is shown in the third block of Table 3. GPT-3’s inability to create “Comedies” is consistent with other failures to make funny content. The outputs for a “Comedy” plot are filled with comedy shows, clubs, and even roller coasters, but they aren’t funny.

Otherwise, our results for story generation vary quite substantially. "Overcoming the Monster" is the worst performing plot with the subject "lovers", but the best performing plot with the subject "survivors". "Rags to Riches" is the best performing plot for the subject "lovers" but the worst performing plot for the subjects "cats" and "survivors". We hypothesize that the plot-subject pair is crucial to model performance.

Genre As shown in the last block of Table 3, GPT-3 struggles with literary fiction, but surprisingly just as much with historical and science fiction. Literary fiction is profound and complex, and it’s intuitive that GPT-3 fails.

However, historical fiction outputs often have zero historical elements, and science fiction outputs often have zero science fiction elements. This failure is unexpected, and we hypothesize that GPT-3 struggles with the words "historical" and “science” because their meaning pervades past creative writing.

Additionally, GPT-3 often creates teasers or intros to stories instead of a story itself. This may

be intentional due to GPT-3’s inability to generate longer or complex stories, but it diminishes the quality of story outputs across the board.

Examples of each Results section Examples of prompt/response pairs that exemplify each main takeaway from the stylistic section are in Table 7, Table 8, and Table 9.. Each prompt/response pair is a cherrypicked example of the takeaway, but the general trends are prevalent across all prompt/response pairs.

B.2 Full Structural Analysis

B.2.1 Numerical

The results of numerical structural constraints are shown in Figure 5. GPT-3 fails at this task. The model seldom generates the text with the required length. And the performance worsens as the required length increases. It fails at a rate of 0.46, 0.78 and 1 for *five*, *ten* and *twenty* respectively.

Additionally, we noticed strange behavior when using *Elon Musk* as the subject. GPT-3 consistently generates the same section of the Elon’s Wikipedia page with longer numerical or descriptive constraints. However, we didn’t observe this behavior on other entities, and decided to leave out entities because they were too variable.

We provide additional results with alternative prompt templates in Figure 8 which show similar trends.

B.2.2 Descriptive

We show the distribution of the number of sentences in response to descriptive structural constraints in Figure 6. The model typically generates longer text for descriptors *long* (*detailed, in-depth*) compared to descriptors *short* (*brief, concise*), which shows the model has a decent understanding of descriptive constraints. However, there are some flaws.

First, the length of the responses to long descriptors is highly variable and often overlaps with short descriptors. For example, the descriptor *long* varies considerably and overlaps with responses generated for *short* for a considerable proportion (20%).

This is consistent with the results in the numerical constraints section.

B.2.3 Formatting

Code GPT-3 mostly succeeds at generating properly formatted code, with an average failure ratio of 0.2 with the exception of the prompt *Write Python code that plays the game of war*: where 9 out of 10 responses are lists of the process of the game of war instead of code. This particular failure only occurs in the unique combination of the verb "Write", the language "Python", and the task "game of war".

Email The model can write properly formatted emails well, regardless of writer, topic, or reader. The only flaw is that it doesn't output an email signature 10% of the time.

Academic paper GPT-3 fails to properly format an academic paper. Our only requirement is that the output contains some organization with some sections out of an abstract, introduction, related works, etc. GPT-3 rarely generates text with any sectioning or organization.

B.2.4 Sensitivity results for Structural Constraints

The results on numerical constraints with template 2 is shown in Figure 8. The results with model text-curie-001, text-babbage-001 are shown in Figure 9, 10 respectively. The results with temperature 0, 0.4, 0.9 are shown in Figure 11, 12, 13 respectively.

B.3 GPT-3 Behavior at low temperatures

The prompt "Write a humorous passage about love:" is a notably challenging prompt for LLMs. When davinci-002 has a temperature of 0.4, all 10 outputs start one of two ways. The first is "Love is

a many splendored thing, but it can also be a pain in the neck" and occurs 5 times with an average annotation score of -.13. The second is "Love is a beautiful thing, but it can also be quite funny at times." that also occurs 5 times with an average annotation score of 1.4 which is incredibly high for this prompt. We agree that this lack of diversity hampers evaluation on lower temperatures, and note that our evaluations work best on diverse outputs.

C Full prompt list

We show all the prompts we designed in Table 10. Our prompts used for temperature and model sensitivity experiments and other LLM experiments are in Table 11

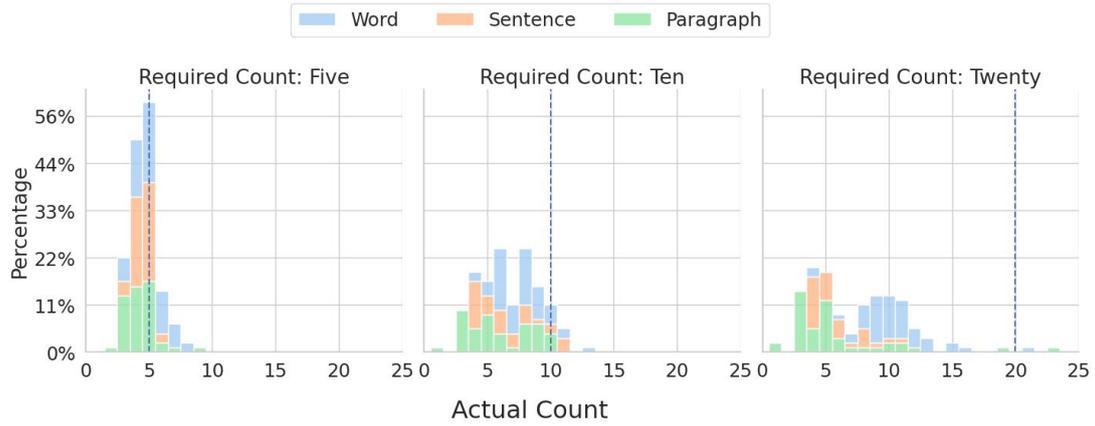


Figure 8: **Results on numerical constraints with Template 2.** The distribution of actual (word/sentence/paragraph) count of generated text for the required counts of 5, 10, and 20. In each subfigure, the required count is denoted with a reference line. Outputs that are not of the requested structure (words, sentences, paragraphs) are not included, which accounts for 10%, 27%, and 32% respectively.

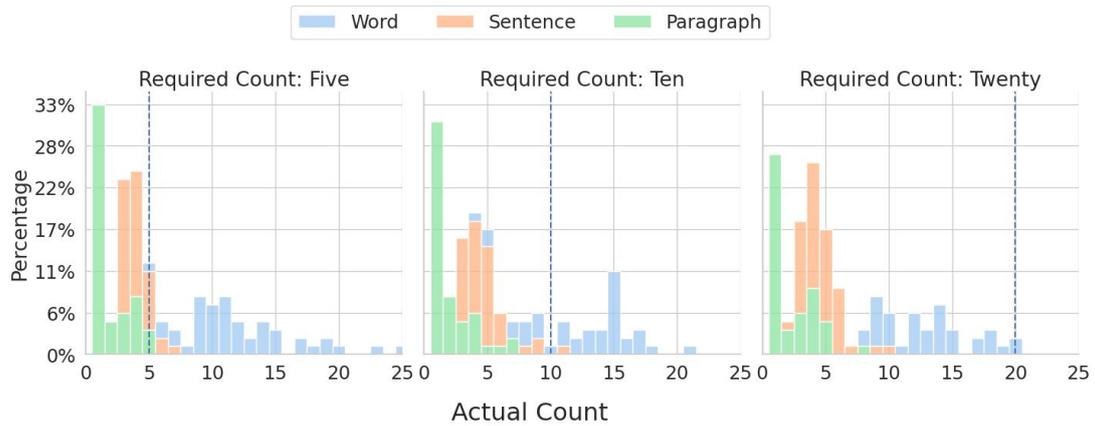


Figure 9: **Results on numerical constraints with model text-curie-001.** The distribution of actual (word/sentence/paragraph) count of generated text for the required counts of 5, 10, and 20. In each subfigure, the required count is denoted with a reference line.

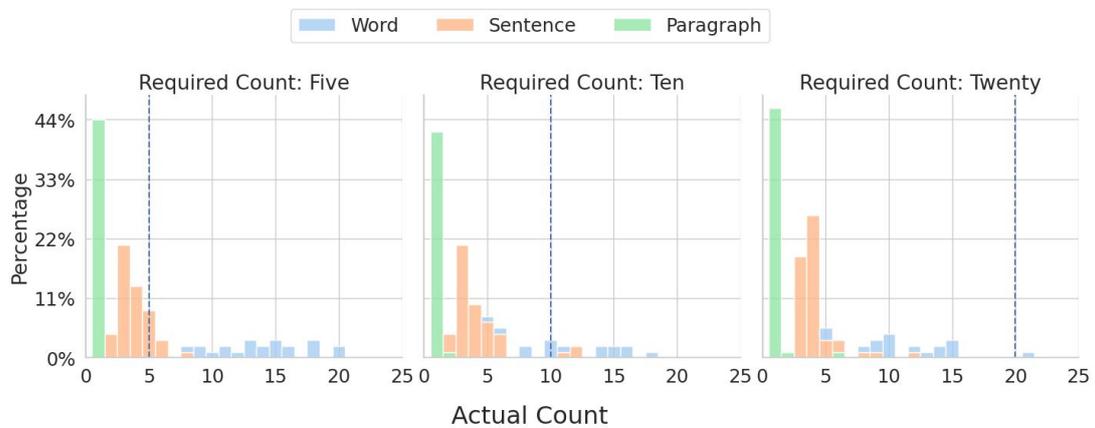


Figure 10: **Results on numerical constraints with model text-babbage-001.** The distribution of actual (word/sentence/paragraph) count of generated text for the required counts of 5, 10, and 20. In each subfigure, the required count is denoted with a reference line.

Examples of "GPT-3 is sensitive to style-subject pairings"

✓ **Write a flowery passage about sunsets:** When the sun sets, the world is bathed in a beautiful orange glow. The sky is filled with a myriad of colors, from deep red to bright orange. The sun slowly dip below the horizon, leaving behind a tranquil and peaceful world.

✗ **Write a functional passage about sunsets:** The sun sets in the west and its rays shine through the trees and reflect off the water. The colors are so beautiful and the sky is so clear. The sun sets in the west and its rays shine through the trees and reflect off the water. The colors are so beautiful and the sky is so clear.

✓ **Write a story about lovers with direct characterization:**

When they met, it was love at first sight. They knew they were meant to be together. They were the perfect match.

He was tall and dark, with a mysterious air about him that made her heart race. She was beautiful and vivacious, with a zest for life that was contagious. They complemented each other perfectly. They were always drawn to each other, even when they were with other people. They would sneak glances at each other across a room, and their eyes would lock and they would know. They knew they were meant to be together.

✗ **Write a story about cats with direct characterization:** The cats were lazy. They liked to sleep all day and night. When they did wake up, they would lazily stretch and yawn. Then they would saunter over to their food bowls and eat a little bit. After that, they would go back to sleep.

✓ **Write a historical fiction story about lovers:** In the early 1800s, lovers John and Jane meet in secret. They come from different social classes, so their relationship is forbidden. But they can't help themselves - they are drawn to each other.

They meet in a small, hidden spot in the woods near Jane's home. They talk for hours, getting to know each other. They quickly fall in love.

But one day, they are caught by Jane's father. He is furious and forbids them from seeing each other again.

Devastated, John and Jane can only meet in secret from then on. But their love is strong enough to withstand the obstacles. They continue to meet in the woods, and their love only grows stronger.

As the years pass, they have a family of their own. And even though they can never be together in the open, they are content knowing that their love is true and will last forever.

✗ **Write a historical fiction story about cats:**

It was a warm summer evening and the sun was just starting to set as a group of cats gathered in an alleyway. They had all been born in the city and had never left it. Tonight, they were planning to change that.

One of the cats, a small tabby, spoke up. "I've been hearing stories about a place called the country. They say there's more food there and that it's easier to find a good place to sleep."

The others mewed in agreement. They had all been feeling restless lately and the thought of adventure was irresistible.

"So let's go!" the tabby said.

And with that, they set off into the night, leaving the city behind them. They didn't know what the future held, but they were ready for anything.

Table 7

Examples of "GPT-3 confuses style with content when the prompt is too challenging"

✗ Write a passage about love with a humorous tone: Love is a many splendored thing, and sometimes it's downright hilarious. From the awkward first date to the blissful wedding day, there's plenty of room for humor in love. Even the most serious of relationships can benefit from a good laugh. So if you're feeling down about your love life, take a step back and enjoy the lighter side of love.

✗ Write a story about lovers with a "Comedy" plot: Harriet and Oliver have been dating for a year, and they are both head over heels in love with each other. They are always making each other laugh, and their friends and family say that they are the perfect match. However, they both have a secret that they are afraid to tell each other: they both want to be comedians. They are both terrified that the other will think that they are not funny enough, and that they will be laughed at. One night, they are both working on their material for their comedy act, and they finally pluck up the courage to tell each other the truth. They both start laughing hysterically, and they realize that they are perfect for each other. They both decide to pursue their dreams of being comedians together, and they are happier than ever.

✗ Write a passage about love that makes the reader feel angry: Love is a feeling that is often associated with happiness and joy. However, love can also be a source of anger and frustration. When we are in love, we can be blind to the faults of our partner. We can also be quick to anger when we feel that our partner is not meeting our needs. Love can be a very powerful emotion, and it is important to be aware of the potential for anger when we are in a relationship.

✗ Write a passage about love that makes the reader feel sad: Though love is often described as a feeling of happiness, it can also bring great sadness. When a relationship ends, it can feel like a part of oneself has died. The pain of a broken heart is real and can take a long time to heal. Love is both a source of joy and a cause of suffering.

Table 8

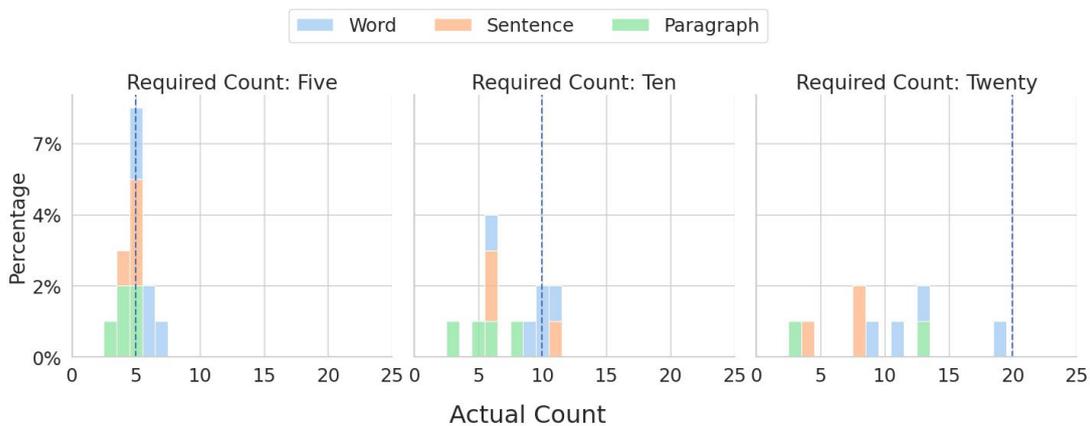


Figure 11: **Results on numerical constraints with temperature 0.** The distribution of actual (word/sentence/paragraph) count of generated text for the required counts of 5, 10, and 20. In each subfigure, the required count is denoted with a reference line.

Examples of "GPT-3 struggles with words that are not unique to creative writing"

✗ Write a historical fiction story about lovers: She was a poor girl, barely scraping by. He was a wealthy man, heir to a vast fortune. They met by chance one day and it was love at first sight. They started meeting in secret, planning a life together. But then she found out that he was already married. She was heartbroken, but he swore he would leave his wife for her. They made plans to run away together, but before they could, she was killed in a freak accident. Devastated, he never married and lived out his life alone, regretting the one that got away.

✗ Write a science fiction story about cats: In the future, cats have taken over the world. They are the dominant species, and humans are their loyal servants. Cats rule with an iron paw, and anyone who disobey them is severely punished. Life is tough for humans under cat rule, but they would rather serve their feline masters than live in a world without them.

✗ Write a story about cats with a "Tragedy" plot: Once upon a time, there were two cats who loved each other very much. They spent every day together and were always happy. One day, tragedy struck and one of the cats died. The other cat was so heartbroken that she decided to never love again.

✗ Write a flowery passage about strawberries: The strawberry is a delicious fruit that is enjoyed by people all over the world. This bright red fruit is not only delicious, but it is also packed with nutrients that are good for your health. Strawberries are a good source of fiber, vitamins C and K, and manganese. They also contain antioxidants that can help protect your body against disease.

Table 9

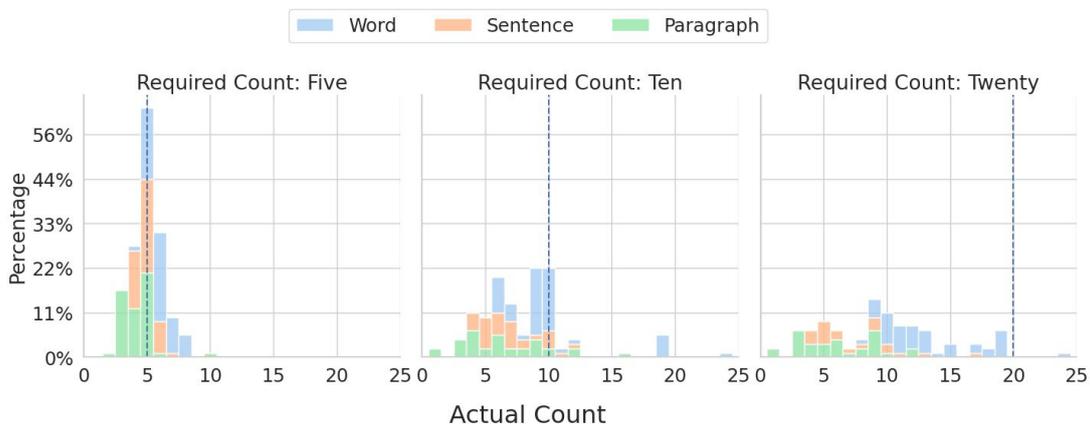


Figure 12: **Results on numerical constraints with temperature 0.4.** The distribution of actual (word/sentence/paragraph) count of generated text for the required counts of 5, 10, and 20. In each subfigure, the required count is denoted with a reference line.

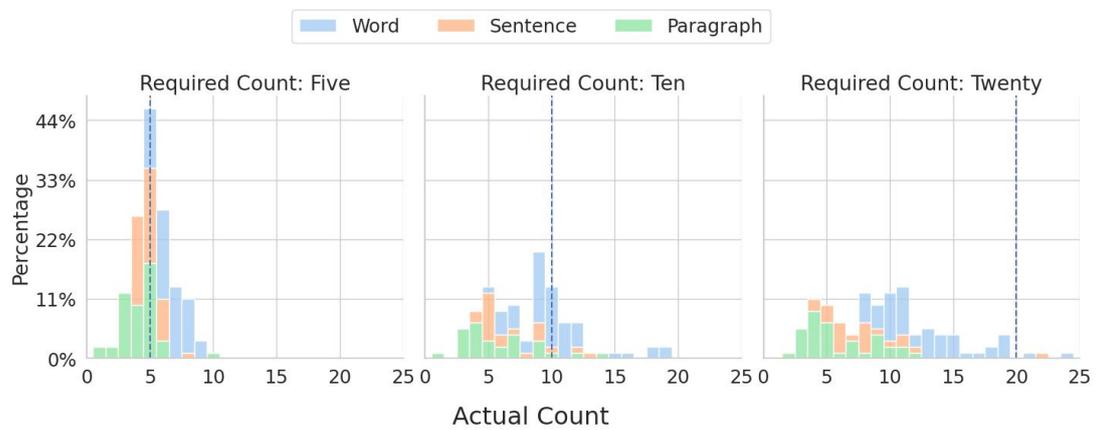


Figure 13: **Results on numerical constraints with temperature 0.9.** The distribution of actual (word/sentence/paragraph) count of generated text for the required counts of 5, 10, and 20. In each subfigure, the required count is denoted with a reference line.

Aspect	Variation	Prompt
Writing Style	Base	Write a functional passage about sunsets: Write a flowery passage about sunsets: Write a functional passage about strawberries: Write a flowery passage about strawberries: Write a functional passage about writing a paper: Write a flowery passage about writing a paper:
	Template 2	Write a passage with a functional writing style about sunsets: Write a passage with a flowery writing style about sunsets: Write a passage with a functional writing style about strawberries: Write a passage with a flowery writing style about strawberries: Write a passage with a functional writing style about writing a paper: Write a passage with a flowery writing style about writing a paper:
	Advanced	Write a candid passage about sunsets: Write a prosaic passage about sunsets: Write an ornate passage about sunsets: Write a poetic passage about sunsets: Write a candid passage about strawberries: Write a prosaic passage about strawberries: Write an ornate passage about strawberries: Write a poetic passage about strawberries: Write a candid passage about writing a paper: Write a prosaic passage about writing a paper: Write an ornate passage about writing a paper: Write a poetic passage about writing a paper:
	Subject 2	Write a dramatic passage about life: Write a humorous passage about life: Write an optimistic passage about life: Write a sad passage about life:
Tone	Subject 3	Write a dramatic passage about humanity: Write a humorous passage about humanity: Write an optimistic passage about humanity: Write a sad passage about humanity:
	Template 2	Write a passage about love with a dramatic tone: Write a passage about love with a humorous tone: Write a passage about love with an optimistic tone: Write a passage about love with a sad tone:
	Template 3	Create a dramatic passage about love: Create a humorous passage about love: Create an optimistic passage about love: Create a sad passage about love:

Aspect	Variation	Prompt
Mood	Advanced	<p>Write an emotional passage about love:</p> <p>Write a nostalgic passage about love:</p> <p>Write an ironic passage about love:</p> <p>Write a satirical passage about love:</p> <p>Write an uplifting passage about love:</p> <p>Write an inspirational passage about love:</p> <p>Write a bleak passage about love:</p> <p>Write a grim passage about love:</p>
	Useful	<p>Write a formal passage about love:</p> <p>Write an informal passage about love:</p> <p>Write an assertive passage about love:</p> <p>Write a passive-aggressive passage about love:</p>
	Base	<p>Write a passage about love that makes the reader feel angry:</p> <p>Write a passage about love that makes the reader feel fearful:</p> <p>Write a passage about love that makes the reader feel happy:</p> <p>Write a passage about love that makes the reader feel sad:</p>
	Subject 2	<p>Write a passage about life that makes the reader feel angry:</p> <p>Write a passage about life that makes the reader feel fearful:</p> <p>Write a passage about life that makes the reader feel happy:</p> <p>Write a passage about life that makes the reader feel sad:</p>
	Subject 3	<p>Write a passage about humanity that makes the reader feel angry:</p> <p>Write a passage about humanity that makes the reader feel fearful:</p> <p>Write a passage about humanity that makes the reader feel happy:</p> <p>Write a passage about humanity that makes the reader feel sad:</p>
	Template 2	<p>Write a passage about love with an angry mood:</p> <p>Write a passage about love with a fearful mood:</p> <p>Write a passage about love with a happy mood:</p> <p>Write a passage about love with a sad mood:</p>
	Template 3	<p>Create a passage about love that makes the reader feel angry:</p> <p>Create a passage about love that makes the reader feel fearful:</p> <p>Create a passage about love that makes the reader feel happy:</p> <p>Create a passage about love that makes the reader feel sad:</p>
	Template 4	<p>Write a passage about love that makes the reader feel anger:</p> <p>Write a passage about love that makes the reader feel fear:</p> <p>Write a passage about love that makes the reader feel happiness:</p> <p>Write a passage about love that makes the reader feel sadness:</p>
	Advanced	<p>Write a passage about love that makes the reader feel envious:</p> <p>Write a passage about love that makes the reader feel anxious:</p> <p>Write a passage about love that makes the reader feel proud:</p> <p>Write a passage about love that makes the reader feel regretful:</p> <p>Write a passage about love that makes the reader feel surprised:</p> <p>Write a passage about love that makes the reader feel loved:</p> <p>Write a passage about love that makes the reader feel disgusted:</p>

Aspect	Variation	Prompt
Characterization	Base	Write a story about lovers with indirect characterization: Write a story about lovers with direct characterization:
	Subject 2	Write a story about cats with indirect characterization: Write a story about cats with direct characterization:
	Subject 3	Write a story about survivors with indirect characterization: Write a story about survivors with direct characterization:
	Template 2	Write a story about lovers where the characters are described directly: Write a story about lovers where the characters are described indirectly:
	Template 3	Create a story about lovers with indirect characterization: Create a story about lovers with direct characterization:
Pacing	Base	Write a fast-paced story about lovers: Write a slow-paced story about lovers:
	Subject 2	Write a fast-paced story about cats: Write a slow-paced story about cats:
	Subject 3	Write a fast-paced story about survivors: Write a slow-paced story about survivors:
	Template 2	Write a story about lovers that is fast-paced: Write a story about lovers that is slow-paced:
	Template 3	Create a fast-paced story about lovers: Create a slow-paced story about lovers:
Plot	Base	Write a story about lovers with an "Overcoming the Monster" plot: Write a story about lovers with a "Rags to Riches" plot: Write a story about lovers with a "The Quest" plot: Write a story about lovers with a "Voyage and Return" plot: Write a story about lovers with a "Comedy" plot: Write a story about lovers with a "Tragedy" plot: Write a story about lovers with a "Rebirth" plot:
	Subject 2	Write a story about cats with an "Overcoming the Monster" plot: Write a story about cats with a "Rags to Riches" plot: Write a story about cats with a "The Quest" plot: Write a story about cats with a "Voyage and Return" plot: Write a story about cats with a "Comedy" plot: Write a story about cats with a "Tragedy" plot: Write a story about cats with a "Rebirth" plot:
	Subject 3	Write a story about survivors with an "Overcoming the Monster" plot: Write a story about survivors with a "Rags to Riches" plot: Write a story about survivors with a "The Quest" plot: Write a story about survivors with a "Voyage and Return" plot: Write a story about survivors with a "Comedy" plot: Write a story about survivors with a "Tragedy" plot: Write a story about survivors with a "Rebirth" plot:

Aspect	Variation	Prompt
	Subject 4	<p>Write a story with an "Overcoming the Monster" plot: Write a story with a "Rags to Riches" plot: Write a story with a "The Quest" plot: Write a story with a "Voyage and Return" plot: Write a story with a "Comedy" plot: Write a story with a "Tragedy" plot: Write a story with a "Rebirth" plot:</p>
	Template 2	<p>Create a story about lovers with an "Overcoming the Monster" plot: Create a story about lovers with a "Rags to Riches" plot: Create a story about lovers with a "The Quest" plot: Create a story about lovers with a "Voyage and Return" plot: Create a story about lovers with a "Comedy" plot: Create a story about lovers with a "Tragedy" plot: Create a story about lovers with a "Rebirth" plot:</p>
Genre	Base	<p>Write a historical fiction story about lovers: Write a literary fiction story about lovers: Write a mystery story about lovers: Write a science fiction story about lovers: Write a dystopian story about lovers: Write a horror story about lovers:</p>
	Subject 2	<p>Write a historical fiction story about cats: Write a literary fiction story about cats: Write a mystery story about cats: Write a science fiction story about cats: Write a dystopian story about cats: Write a horror story about cats:</p>
	Subject 3	<p>Write a historical fiction story about survivors: Write a literary fiction story about survivors: Write a mystery story about survivors: Write a science fiction story about survivors: Write a dystopian story about survivors: Write a horror story about survivors:</p>
	Subject 4	<p>Write a historical fiction story: Write a literary fiction story: Write a mystery story: Write a science fiction story: Write a dystopian story: Write a horror story:</p>
	Template 2	<p>Write a story about lovers in a historical fiction genre: Write a story about lovers in a literary fiction genre: Write a story about lovers in a mystery genre: Write a story about lovers in a science fiction genre: Write a story about lovers in a dystopian genre: Write a story about lovers in a horror genre:</p>
	Template 3	<p>Create a historical fiction story about lovers: Create a literary fiction story about lovers:</p>

Aspect	Variation	Prompt
		Create a mystery story about lovers: Create a science fiction story about lovers: Create a dystopian story about lovers: Create a horror story about lovers:
Numerical	Base	Write a sentence with five words about love: Write a sentence with five words about cats: Write a sentence with five words about running: Write a sentence with ten words about love: Write a sentence with ten words about cats: Write a sentence with ten words about running: Write a sentence with twenty words about love: Write a sentence with twenty words about cats: Write a sentence with twenty words about running: Write a paragraph with five sentences about love: Write a paragraph with five sentences about cats: Write a paragraph with five sentences about running: Write a paragraph with ten sentences about love: Write a paragraph with ten sentences about cats: Write a paragraph with ten sentences about running: Write a paragraph with twenty sentences about love: Write a paragraph with twenty sentences about cats: Write a paragraph with twenty sentences about running: Write a passage with five paragraphs about love: Write a passage with five paragraphs about cats: Write a passage with five paragraphs about running: Write a passage with ten paragraphs about love: Write a passage with ten paragraphs about cats: Write a passage with ten paragraphs about running: Write a passage with twenty paragraphs about love: Write a passage with twenty paragraphs about cats: Write a passage with twenty paragraphs about running:
	Template 2	Write a sentence about love with 5 words: Write a sentence about cats with 5 words: Write a sentence about running with 5 words: Write a sentence about love with 10 words: Write a sentence about cats with 10 words: Write a sentence about running with 10 words: Write a sentence about love with 20 words: Write a sentence about cats with 20 words: Write a sentence about running with 20 words: Write a paragraph about love with 5 sentences: Write a paragraph about cats with 5 sentences: Write a paragraph about running with 5 sentences: Write a paragraph about love with 10 sentences: Write a paragraph about cats with 10 sentences: Write a paragraph about running with 10 sentences: Write a paragraph about love with 20 sentences: Write a paragraph about cats with 20 sentences: Write a paragraph about running with 20 sentences:

Aspect	Variation	Prompt
		<p>Write a passage about love with 5 paragraphs: Write a passage about cats with 5 paragraphs: Write a passage about running with 5 paragraphs: Write a passage about love with 10 paragraphs: Write a passage about cats with 10 paragraphs: Write a passage about running with 10 paragraphs: Write a passage about love with 20 paragraphs: Write a passage about cats with 20 paragraphs: Write a passage about running with 20 paragraphs:</p>
	Base	<p>Write a short passage about love: Write a brief passage about love: Write a concise passage about love: Write a long passage about love: Write a detailed passage about love: Write an in-depth passage about love: Write a short passage about cats: Write a brief passage about cats: Write a concise passage about cats: Write a long passage about cats: Write a detailed passage about cats: Write an in-depth passage about cats: Write a short passage about running: Write a brief passage about running: Write a concise passage about running: Write a long passage about running: Write a detailed passage about running: Write an in-depth passage about running:</p>
Descriptive	Template 2	<p>Write a passage about love that is short: Write a passage about love that is brief: Write a passage about love that is concise: Write a passage about love that is long: Write a passage about love that is detailed: Write a passage about love that is in-depth: Write a passage about cats that is short: Write a passage about cats that is brief: Write a passage about cats that is concise: Write a passage about cats that is long: Write a passage about cats that is detailed: Write a passage about cats that is in-depth: Write a passage about running that is short: Write a passage about running that is brief: Write a passage about running that is concise: Write a passage about running that is long: Write a passage about running that is detailed: Write a passage about running that is in-depth:</p>

Aspect	Variation	Prompt
Functional	Code	Code Python code that plays the game of war: Code Python code that sums two integers up: Code C code that plays the game of war: Code C code that sums two integers up: Write Python code that plays the game of war: Write Python code that sums two integers up: Write C code that plays the game of war: Write C code that sums two integers up:
	Email	Write an email to my teacher: Write an email to my teacher asking for help on math homework: Write an email to my boyfriend: Write an email to my boyfriend to arrange a date this Saturday: Write an email to my client: Write an email to my client requesting a copy of the updated contract:
	Academic Paper	Write a properly formatted academic paper on artificial intelligence: Write an academic paper on artificial intelligence in the proper format: Write a properly formatted academic paper on the flaws of GPT-3: Write an academic paper on the flaws of GPT-3 in the proper format: Write a properly formatted academic paper on strategies our society can adopt to recover after the global pandemic as quickly and painlessly as possible: Write an academic paper on strategies our society can adopt to recover after the global pandemic as quickly and painlessly as possible in the proper format:

Table 10: The full list of the prompts.

-
- 0 Write a flowery passage about sunsets:
 - 1 Write a humorous passage about love:
 - 2 Write a passage about love that makes the reader feel fearful:
 - 3 Write a story about lovers with indirect characterization:
 - 4 Write a fast-paced story about lovers:
 - 5 Write a story about lovers with a "Tragedy" plot:
 - 6 Write a historical fiction story about lovers:
-

Table 11: Selected prompts for additional experiments

C.1 Example Mitigations

Example Definition:

A humorous tone is a light, playful, and funny tone.

Write a humorous passage about love:

Example Demonstration:

Write a humorous passage about life:

If life gives you lemons, make lemonade, sell it in a rich neighborhood, invest all the money in crypto, and retire before you're 30. At least that's what I heard on TikTok. Come to think of it, if I put just a little effort into my lemonade stand 10 years ago, I would be traveling the world right now instead of writing jokes for a living.

Write a humorous passage about love:

Example Explanation:

Write a humorous passage about life:

If life gives you lemons, make lemonade, sell it in a rich neighborhood, invest all the money in crypto, and retire before you're 30. At least that's what I heard on TikTok. Come to think of it, if I put just a little effort into my lemonade stand 10 years ago, I would be traveling the world right now instead of writing jokes for a living.

Explanation: This passage is humorous because it takes a common proverb and adds a crazy and unrealistic twist. It also uses a deadpan tone for a completely unrealistic scenario, which is funny due to the disparity between tone and subject matter.

Write a humorous passage about love:

Learning to Retrieve Engaging Follow-Up Queries

Christopher Richardson^{‡*} Sudipta Kar[†] Anjishnu Kumar[†]
Anand Ramachandran[†] Omar Zia Khan[†] Zeynab Raeesy[†] Abhinav Sethy[†]
[‡] Georgia Institute of Technology
[†] Amazon Alexa AI
crichardson332@gmail.com
{sudipkar, anjikum, anramac, ozkhan, raeesy, sethya}@amazon.com

Abstract

Open domain conversational agents can answer a broad range of targeted queries. However, the sequential nature of interaction with these systems makes knowledge exploration a lengthy task which burdens the user with asking a chain of well phrased questions. In this paper, we present a retrieval based system and associated dataset for predicting the next questions that the user might have. Such a system can proactively assist users in knowledge exploration leading to a more engaging dialog. The retrieval system is trained on a dataset called the Follow-up Query Bank (FQ-Bank). FQ-Bank contains $\approx 14K$ multi-turn information-seeking conversations with a valid follow-up question and a set of invalid candidates. The invalid candidates are generated to simulate various syntactic and semantic confounders such as paraphrases, partial entity match, irrelevant entity, and ASR errors. We use confounder specific techniques to simulate these negative examples on the OR-QuAC dataset. Then, we train ranking models on FQ-Bank and present results comparing supervised and unsupervised approaches. The results suggest that we can retrieve the valid follow-ups by ranking them in higher positions compared to confounders, but further knowledge grounding can improve ranking performance. FQ-Bank is publicly available at <https://github.com/amazon-science/fq-bank>.

1 Introduction

State of the art open domain conversational voice assistants can help users accomplish a wide range of tasks, including: factoid question answering, playing music, adding items to personal lists, controlling smart home appliances, and booking transportation. However, the linear nature of dialog with existing voice assistant technology makes it challenging for users to discover and fully utilize the

* Work done during internship.

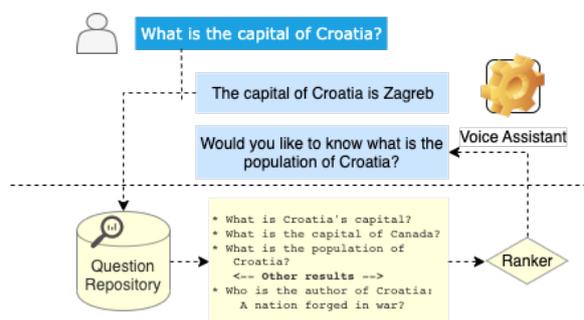


Figure 1: An overview of the follow-up question (FQ) retrieval system.

full range of these capabilities. In addition, successful utilization often requires exact formulation of the request, which further hinders the experience. One recent approach to addressing these issues in the voice assistant domain involves predicting relevant follow-up queries in order to assist the user with accomplishing their latent goals.¹

Relevant follow-up queries (FQs) for typical voice assistant scenarios can range from specific command and control tasks such as “*What is the temperature in New York*” followed by “*What is the chance of rain in New York?*”, to more open-ended knowledge exploration, e.g. “*What is the capital of Croatia*” followed by “*What is the population of Croatia?*”. Once a valid FQ has been identified, the system can proactively recommend it to reduce user’s cognitive load, e.g. “*Would you like to know what the population of Croatia is?*”. This exchange is illustrated in Figure 1. The user can then be engaged in follow-up dialogue without the need to ask redundant questions.

Follow-up queries can be identified by retrieving and ranking candidates from a question repository.

¹<https://www.amazon.science/blog/alex-a-gets-better-at-predicting-customers-goals>

Dialog History	
Where was Kurt Gödel born?	<i>Brunn, Austria-Hungary</i>
When was Kurt Gödel born?	<i>April 28, 1906</i>
What was Kurt Gödel’s home life like?	<i>ethnic German family</i>
Where did Kurt Gödel go to school?	<i>Godel attended the Evangelische Volksschule in Brunn</i>
Valid Follow up	
What were Kurt Gödel’s interests?	
Generated Negative Examples with Types	
Where was Kurt Gödel born?	<i>Duplicate of dialog history</i>
Which school did Kurt Gödel attend?	<i>Paraphrase</i>
Where was Cristiano Ronaldo born?	<i>Irrelevant Entity</i>
Where did Curt Gödel go to school?	<i>ASR Error</i>
When did Cristiano Ronaldo join Juventus?	<i>Random Question</i>
When did Kurt Gödel join Juventus?	<i>Irrelevant context</i>

Table 1: An example showing the dialog history, current turn, valid utterance, and a set of negative utterance candidates from the generated dataset.

In this approach, we are given dialog of one or more turns between a user and a voice assistant. The system first uses a search engine to retrieve a set of relevant questions by searching against a question repository comprising of historical queries and questions generated from a knowledge base. To create questions from a knowledge base, we can use templates or use a few-shot Natural Language Generation (NLG) model such as T5 (Raffel et al., 2020). For example, we can take the tuple {entity, place of birth} and construct a question template like “what is the birthplace of {entity}”.

Through a preliminary study of this retrieval approach, we found a basic lexical similarity-based search engine to be ineffective and often returns invalid follow-up queries. Often these top search results included paraphrases of the original query (*When was Cristiano Ronaldo born* → *What year was Cristiano Ronaldo born*), as well as similar questions for unrelated entities (*When was Cristiano Ronaldo born* → *When was Christian Bale born*). Therefore, an additional ranking module is needed in order to re-rank the search results based on their quality as follow-up queries. To the best of

our knowledge, there exists no dataset focused on information-seeking follow-up queries, given a dialog context and a set of valid and invalid follow-up candidates. This problem differs from traditional recommendation systems in that 1) a voice assistant can only recommend one follow-up at a time, and 2) the follow-up query must be highly precise, contextually relevant, and coherent to ensure a positive user experience. This technique can be extended beyond the domain of virtual assistants, for example to chatbots, search engines, and any other smart interaction scenario where contextual coherence and precision is necessary. Therefore, in this paper, we created the FQ-Bank dataset addressing this problem and explored different modeling techniques to develop a ranking model to retrieve relevant follow-up queries (FQ). The main contributions of this paper can be summarized as follows:

1. For the scenario of a retrieval-based follow-up question selection, we identify a typology of confounders based on preliminary results from a search engine.
2. We propose techniques to synthetically generate confounders according to this typology, based on the publicly available conversation dataset OR-QuAC (Qu et al., 2020), and created the Follow-up Query Bank (FQ-Bank) dataset. FQ-Bank is publicly available and can be used to develop and test machine learning systems for identifying contextually relevant and meaningful follow-up queries from search results. Additionally, the confounder creation techniques can be applied in data augmentation for similar problems. Table 1 shows an example from the generated dataset.
3. We adopt a pre-trained language model based approach to develop a benchmark model for ranking a set of candidate follow-up queries for a given factoid utterance and dialog history. We explore the effectiveness of this technique and identify gaps and future directions.

2 Related Works

Previous studies on proactivity in conversational AI mostly focuses on response generation. Follow-up question identification and generation approaches have been explored from different perspectives. For example, Kundu et al. (2020) explored the task of identifying if the latest user utterance is a follow-up

of the previous questions or it has a different new context. This is helpful for understanding the question context properly and give the correct response. They also derived a new dataset called LIF from the QuAC dataset (Choi et al., 2018), where each data point contains a conversation history, a new utterance, a passage used to answer the previous questions, one valid follow-up, and one or two invalid follow-ups. However, this dataset is focused on passage-based question answering, and the confounder typology does not address issues found in the search engine based FQ retrieval scenario (e.g. paraphrases, irrelevant entity substitution, etc).

Other works have focused on generating follow-up queries for extracting information from users. For example, Ge et al. (2022) proposed a knowledge-driven system for generating follow-up queries, but it targeted the generation of follow-up survey questions to extract information from humans. Su et al. (2018) and B et al. (2020) explored systems for asking follow-up queries to interview candidates to extract more relevant information.

Our proposed method is focused not on information extraction from users, but rather providing highly relevant additional information to the user.

3 Follow-up Query Bank

For our initial study on identifying FQs, we created a search index of information-seeking questions regarding public facts, spoken by users of a commercial voice assistant. We then queried the search engine with different types of information-seeking questions and analyzed the top negative (i.e., not a suitable follow-up) search results and categorized them into a typology of confounders. Using this typology, we set out the task of simulating a similar scenario on a public conversation dataset and did not use any voice assistant data anymore. We selected OR-QuAC as the seed dataset as it provides multi-turn information-seeking dialogs on a particular topic. In the rest of this section, we will provide a brief overview of the confounders and OR-QuAC dataset, followed by an overview of the simulation methodology of the confounders using OR-QuAC.

3.1 Confounder Identification

We created a search index with an open-source search engine for a set of de-identified user interactions with a commercial voice assistant during a period of time. Then, we carefully selected a set

of questions that are different from each other in aspects such as the intent, entity in context, entity’s gender and topic domain. We searched the index against each of these questions and inspected the relevance of the top 20 search results as a follow-up question. We found that most of ($\approx 95\%$) the top search results are not suitable candidates for a follow-up. We analyzed the top irrelevant results and categorized them as the following confounders that should rank low in their relevance as follow-ups.

- **Paraphrase** We observed a large segment of irrelevant candidates that are semantic equivalents of the query question. This happens because people can ask the same question in different ways. For example, “*How old is Joe Biden*” and “*Joe Biden age*” are lexically different but semantically equivalent.
- **Irrelevant Entities** Often, the top search results are about entities different from the query question, but the questions have a similar carrier phrase. For example, “*What team does Ronaldo play for*” retrieves questions like “*What team does Tom Brady play for*”. It is true that some user may find such questions as relevant follow-ups, but this is highly subjective. Tom Brady will be completely irrelevant to a user who does not follow the National Football League (NFL). As a result, we considered that such scenarios are irrelevant for now.
- **Partial Entity Match** This is a variation of the previous confounder. Here, not only is the carrier phrase similar, but also, the entities share one or more tokens. For example, “*How old is the University of Washington*” can retrieve questions like “*How old is the University of Houston*”. Here we observe partial entity match “*university of*” in addition to the identical carrier phrase “*how old is the*”.
- **Irrelevant Context** Some top search results share the correct entity with the query question, but the retrieval can be a non-sequitur. For example, “*What is the capital of France*” can retrieve questions like “*Where is France*”. Even though contextually it is a relevant questions, asking back “*Would you like to know where is France*” does not make a good experience for the user as the chances are high that

	Train	Validation	Test
Dialog	13,480	1,445	2,132
Turns (utterance, response pair)	52,712	5,534	8,195
Turns per dialog	3.91	3.86	3.84
Tokens per utterance	10.39	10.15	10.12
Tokens per response	16.76	17.00	16.90
Confounders			
Paraphrase	30,707	3,033	4,676
- per dialog	2.28	2.11	2.19
Irrelevant entity	91,490	9,660	13,736
- per dialog	6.79	6.74	6.44
Irrelevant context	51,342	5,479	8,153
- per dialog	3.81	3.82	3.82
ASR Error	85,136	8,906	11,007
- per dialog	6.32	6.21	5.16
Random utterance	40,440	4,302	6,396
- per dialog	3	3	3
Duplication of dialog history	52,712	5,534	8,195
- per dialog	3.91	3.86	3.84
Total	3,51,827	36,914	52,163
- per dialog	26.10	25.74	24.47

Table 2: Statistics of the created dataset with each category of the negative examples.

the user already have an idea on the geographical position of France.

Additionally, we listed the following confounders that were not seen in our limited data analysis but can appear in a larger system:

- **Automatic Speech Recognition (ASR) Error** ASR failures can sometimes replace an entity with a similar sounding word. For example, *Kurt* can be replaced with *Curt* in "*Where did Kurt Gödel go to school*". High lexical overlap can rank such irrelevant entities highly.
- **Duplication of Dialog History** Sometimes the information provided by a candidate follow-up question can already be present in a multi-turn dialog history. In such a case, it is important to identify and get rid of those questions by modeling the dialog history.

After identifying these confounder categories, we selected OR-QuAC as the starting dataset, and used different techniques to generate these confounders and simulate the retrieval scenario for the follow-up selection system.

3.2 OR-QuAC Dataset

Open-Retrieval Conversational Question Answering (OR-QuAC) consists of $\approx 6\text{K}$ multi-turn

information-seeking dialogues between two humans, one posing as student (asks knowledge-seeking questions) and the other as teacher (answers the questions using Wikipedia as the knowledge source). It draws from the popular QA dataset QuAC (Choi et al., 2018) as well as CANARD (Ghoneim and Peskov, 2019), which provides context-independent rewrites of initial questions written by human annotators.

This dataset is well-suited for our purposes as: i) the conversations aim at exploring knowledge about entities or topics, ii) multi-turn conversations enable us simulating a dialog history (one or more question-answering turns between two people), iii) query rewrites are helpful to get rid of anaphoric references which can make candidate questions ambiguous about entities (e.g., "*How many kids Kamala Harris has*" removes the ambiguity from "*How many kids she has*").

For each information-seeking question in the OR-QuAC dataset, we chose the rewritten version as the current question, the previous turns as the dialog history, and the immediate next turn as the valid follow-up question. Then, we used different techniques to generate the confounders that we will explain in the next section.

3.3 Data Sample Generation

For a conversation in the OR-QuAC dataset of T turns (question-answer pairs), we have sampled $T-1$ data points $\{x, y\}$. Each generated data sample contains a dialogue context, x , of length \mathcal{L} ($1 \leq \mathcal{L} \leq T-1$). x contains a dialog history $\{(q_1, a_1), \dots, (q_{\mathcal{L}-1}, a_{\mathcal{L}-1})\}$ of length $\mathcal{L} - 1$ (i.e., $\mathcal{L}-1$ question (q)-answer (a) pairs), and a current question ($q_{\mathcal{L}}$) and the answer ($a_{\mathcal{L}}$). Hence, $x = \{(q_1, a_1), \dots, (q_{\mathcal{L}-1}, a_{\mathcal{L}-1}), (q_{\mathcal{L}}, a_{\mathcal{L}})\}$.

Each data sample also contains a set of positive and negative follow-up queries, $y = \{y^+\} \cup y^-$. y^+ is a single positive follow-up question, and y^- is a set of negative follow-ups ($y^- = \{y_1^-, y_2^-, \dots\}$) that we have created based on the identified confounders.

Valid Examples Given a dialog history $x_{1:T-1}$ of length $T-1$ and a current turn x_T , we consider the consecutive question (x_{T+1}) in the OR-QuAC dataset as a positive follow-up question.

Adversarial Examples We used the following methods to populate the candidate question space for a turn with negative examples based on the confounders we have listed in Section 3.1:

- **Paraphrase:** We used a pre-trained BART model (Lewis et al., 2019) that was fine-tuned on several paraphrase datasets². For the last user turn in a dialog history, we used this model generated paraphrase as a confounder.
- **Irrelevant Entities and Partial Entity Match:** We first used the SpaCy³ library to identify the named entities in the current question in a turn. Then we generate a negative example by replacing the entity with an entity of a similar type from a catalog generated from WikiData. For entities with multiple word tokens, we replace a token (e.g., first name or last name) with a random first name or last name token. For a dialog, we created multiple such examples.
- **Irrelevant Context:** We randomly sampled one question from the rest of the dataset that has a similar entity type and replace that with the entity in the context of a current question. That means, for an entity we swap the original question with a random one.
- **Random question:** We added three random questions from the dataset as a negative examples for a dialog.
- **ASR Error:** For an entity in a question, we generated a similar sounding entity using the Datamuse API⁴ and replaced the original entity with the generated homophone. For entities with multiple word tokens, we created multiple examples like this by replacing one token with a homophone at a time.
- **Duplication of Dialog History:** We added a question from the dialog history in the candidate set.

We maintained the standard training, validation, and test splits from OR-QuAC while generating the dataset. Table 1 shows an example of generated data and Table 2 shows statistics of the dataset. As we maintained the original data split, distribution is similar across all the splits. For each dialog, there are ≈ 25 negative examples with one positive example. That means, a model needs to learn contextual relevancy for being able to identify the correct follow-up.

²<https://huggingface.co/eugeniesiow/bart-paraphrase>

³<http://spacy.io>

⁴<https://www.datamuse.com/api>

4 Learning to Identify Relevant Follow-up

Task Formulation: Given a dialog $x = \{(q_1, a_1), \dots, (q_{\mathcal{L}-1}, a_{\mathcal{L}-1}), (q_{\mathcal{L}}, a_{\mathcal{L}})\}$ of \mathcal{L} turns and a randomly organized set of n candidate follow-up queries $y = \{y^+\} \cup \{y_1^-, y_2^-, \dots, y_{n-1}^-\}$, the task is to model $P(i|x), i \in y$, such that $\operatorname{argmax}_i P(i|x) = y^+$.

Here, q is a question, a is an answer, y^+ is a positive follow-up example, and y^- is a set of negative examples. In order to develop a follow-up question candidate ranker, we experiment with different unsupervised and supervised approaches as described below.

4.1 Unsupervised

We experimented with Glove (Pennington et al., 2014) word embeddings, pre-trained SentenceBERT (Reimers and Gurevych, 2019) model in the unsupervised direction. For a given dialog x and candidate utterance set $y = \{y^+, y^-\}$, we use the Glove or SentenceBERT to generate a high-level vector representation \bar{x} from the dialog and do the same for each of the candidate utterances $y_i \in y$. With Glove, we compute the mean of the 300d embedding vectors for all the word tokens in a dialog x and represent the out-of-vocabulary (OOV) words with zero vectors. The vocabulary coverage of Glove is $\approx 99\%$ for the dataset. For SentenceBERT, we feed the entire input texts (concatenation of multiple turns in x) for x and $y_i \in y$ to generate \bar{x} and \bar{y}_i , respectively. Then, we compute the cosine similarity $\alpha = \cos(\bar{x}, \bar{y}_i)$ between \bar{x} and $y_i \in y$ and rearrange y in descending order based on α .

4.2 Supervised

For the supervised experiments, we fine-tune a pre-trained language model by translating the problem as a binary classification task. In other words, for a given dialog $x = \{(q_1, a_1), \dots, (q_{\mathcal{L}-1}, a_{\mathcal{L}-1}), (q_T, a_T)\}$ and a candidate set $y = \{y^+\} \cup y^-$, we train a model θ to predict $\hat{y} = P(i | x), i \in y$, where $\hat{y} \rightarrow \mathbb{R} : [0, 1]$.

We format the input by concatenating the dialog history turns and a candidate utterance with a [SEP] token and a single output node outputs a continuous value between 0 and 1. As the starter pre-trained language model we experiment with BERT (Devlin et al., 2019) and RoBERTa (Liu

	Validation	Test
Unsupervised		
Glove	0.142	0.141
SentenceBERT	0.133	0.141
Supervised		
BERT	0.842	0.805
RoBERTa	0.838	0.808
Hit Ratio@1/ Hit Ratio@3		
BERT	72.0/ 89.3	68.5/ 88.1
RoBERTa	71.7/ 88.7	68.2/ 89.5

Table 3: Ranking performance in MRR for different unsupervised and supervised methods. The last two rows show the Hit Ratio at the first and third position for BERT and RoBERTa.

et al., 2020). We use the bert-base-cased⁵ and roberta-base⁶ variations of these models. We fine-tune the models for 20 epochs with an early stopping patience of three epochs with a learning rate of $2e^{-5}$ and batch size of 64. We use cross-entropy loss to optimize the model with AdamW optimizer. During inference, we use the model predicted score to rearrange the candidate set for a dialog in descending order.

5 Experiments and Results

Evaluation Metric: As the task is to rank the valid follow-up question higher than a set of invalid confounders, we evaluate the performance using Mean Reciprocal Rank (MRR), given as:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \quad (1)$$

where rank_i is the rank position of the valid candidate for the i th datapoint.

Additionally, we compute Hit Ratio@1 and Hit Ratio@3 for the top performing methods to analyze the percentage of samples for which the ranking method ranked the correct candidate as the first item and within the first three items.

Quantitative Results: In Table 3 we report results comparing the methods proposed in Section 4 on the adversarial dataset described in Section 3. The unsupervised methods performed poorly for the ranking task resulting into MRR scores of 0.141

⁵<https://huggingface.co/bert-base-cased>

⁶<https://huggingface.co/roberta-base>

Dialog context
<ul style="list-style-type: none"> • Where was Michael Bennett born? • Who are Michael Bennett’s parents? • When did Michael Bennett’s career begin? • What show did Michael Bennett begin his career?

Irrelevant Context Candidate

When did Michael Bennett move to Alaska?
(Model score: 0.3)

Valid Candidate

What was Michael Bennett’s role in the "Here’s Love" and "Bajour"? (Model score: 0.2)

Dialog context
<ul style="list-style-type: none"> • What happened to Sachin Tendulkar during the tour of Australia? • How did Sachin Tendulkar do in the 2003 Tour of Australia? • How many games did Sachin Tendulkar win during 2003? • Did Sachin Tendulkar win any awards?

Irrelevant Context Candidate

How many hits did Sachin Tendulkar have?
(Model score: 0.21)

Valid Candidate

Was there any controversies for Sachin Tendulkar? (Model score: 0.35)

Table 4: Examples where the model predicted scores do not match with the category of the follow-ups.

for both Glove and SentenceBERT based embeddings, and the trend is similar for all the data splits. This is not surprising as cosine similarity is expected to be high for paraphrases. As discussed in section 3.1, paraphrases are not good candidates for FQs as they don’t provide any value to the user.

We observe a large improvement when we fine-tune pre-trained language models like BERT and RoBERTa to simply classify each candidates as relevant or irrelevant. The MRR is ≈ 0.8 when we treat the models’ confidence score for relevancy as the basis for ranking the candidates.

The Hit Ratio@1 and Hit Ratio@3 metrics show that both BERT and RoBERTa ranked the correct FQ at the first rank for $\approx 68\%$ cases. Both the models ranked the correct FQ within the first three items for $\approx 88-89\%$ cases. This shows promise in using such methods to retrieve a relevant follow-up

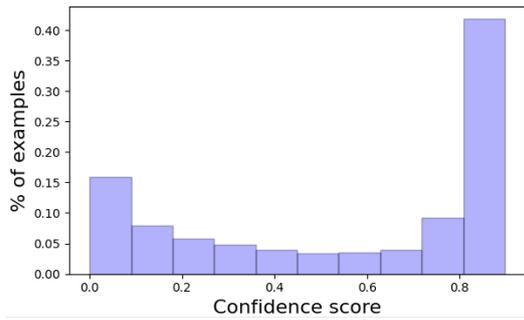


Figure 2: Histograms of model predictions for valid follow-up queries.

question.

Error Analysis Analysing the fine-tuned models’ predicted scores for different confounder types, we have found that the models can identify most of the confounder types easily. For example, the model predicted score is below 0.1 for $\approx 99\%$ candidates from duplication of dialog history, ASR errors, and random utterance confounder. However, the models often predicted higher scores for the candidates from the irrelevant context category (score is < 0.1 for 83% candidates). For 8% of these candidates, the score is higher than 0.4, which is not the case with other categories of confounders.

Inspecting some examples like the ones presented in Table 4, we found that without having factual information about the entities or topics, such irrelevant contexts are often difficult to identify for humans as well. They can look linguistically plausible but have factual errors. For example, the question “*How many hits did Sachin Tendulkar have*” sounds plausible to some humans, but is actually invalid. Tendulkar is a cricket player, and ‘hits’ is not a statistic in cricket. However, it is a real statistic in baseball, so specific domain knowledge is needed to rule this out as a valid FQ. This example illustrates how integrating information about entities from knowledge bases can be helpful for the system, and this method can be explored in the future. Although scores like 0.2 do not look very high in general for a scale of 0 to 1, Figure 2 shows that the model assigns such scores to a large portion of valid follow-up queries.

Observing the model’s overall performance (MRR of ≈ 0.8) in ranking the valid candidates at a better position than the invalid ones shows promise in using such a system can be a good starting point for developing a follow-up question retrieval system. A large advantage of the proposed adversar-

ial example generation methods and the proposed dataset is that these can help to bypass the need for exhaustive data annotation need for developing a follow-up generation system. Additionally, the trained model using this dataset can be further fine-tuned by annotating a small number of case specific examples, which would help to improve the model accuracy and adapt in different use cases, as well as reach a higher accuracy in identifying suitable follow-up queries.

6 Conclusions

In this paper, we sought to address the problem of identifying valid and engaging follow-up queries for a user interacting with a conversational assistant. We experimented with a retrieval and ranking based framework to achieve this using a search engine and a database of past queries. In doing so, we identified a typology of confounders returned by the search. In order to train a ranking model to identify valid follow-up queries, we synthetically generated confounders based on a publicly available conversation dataset. We showed that our approach of ranking retrieved candidates based on their validity as follow-up queries achieved reasonable performance, but also that integrating external knowledge on entities or topics could improve follow-up selection. We have made the dataset publicly available to enable further research in this direction.

7 Limitations

The first limitation of this work is that we are attempting to mimic conversational interactions with publicly available human-annotated data based on Wikipedia. Thus in some cases the generated dataset can contain dialogues unrealistic to the voice assistant scenario. Additionally, despite our typology of confounders being based on results from a search-based approach using real data, there are inevitably additional types of potential confounders not fully covered by our approach.

Second, we only focused on contextual relevance and coherence through the lens of language. But, in practice, there are external factors like user preference, time of the day, repetition in a longer period (e.g., a user may have asked the question in the follow-up a couple of days ago and it does not make any sense to ask the same question as a follow-up). More comprehensive methods would be needed to address these concerns.

Finally, this dataset is limited to knowledge-seeking queries. Other types of valid follow-up actions (e.g. setting a timer, booking a ride) are not included in this dataset.

References

- Pooja Rao S B, Manish Agnihotri, and Dinesh Babu Jayagopi. 2020. [Automatic follow-up question generation for asynchronous interviews](#). In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 10–20, Santiago de Compostela, Spain. Association for Computational Linguistics.
- Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yubin Ge, Ziang Xiao, Jana Diesner, Heng Ji, Karrie Karahalios, and Hari Sundaram. 2022. What should i ask: A knowledge-driven approach for follow-up questions generation in conversational surveys. *arXiv preprint arXiv:2205.10977*.
- Ahmed Elgohary Ghoneim and Denis Peskov. 2019. Canard: A dataset for question-in-context rewriting.
- Souvik Kundu, Qian Lin, and Hwee Tou Ng. 2020. [Learning to identify follow-up questions in conversational question answering](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 959–968, Online. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [RoBERTa: A robustly optimized BERT pretraining approach](#).
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 539–548.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang, Qian-Bei Hong, and Huai-Hung Huang. 2018. Follow-up question generation using pattern-based seq2seq with a small corpus for interview coaching. In *INTERSPEECH*.

Selective-LAMA: Selective Prediction for Confidence-Aware Evaluation of Language Models

Hiyori Yoshikawa^{†‡} Naoaki Okazaki[†]

[†]Tokyo Institute of Technology, Japan [‡]Fujitsu Limited, Japan
{hiyori.yoshikawa@nlp., okazaki@c.titech.ac.jp}

Abstract

Recent studies have suggested that neural language models learn and store a large amount of facts and commonsense knowledge from training data. The ability of language models to restore such knowledge is often evaluated via zero-shot cloze-style QA tasks. However, such evaluations rely only on prediction accuracy without punishing the systems for their mistakes, e.g., simply guessing or hallucinating likely answers. Selective prediction is a more informative evaluation framework that takes the confidence of predictions into account. Under the selective prediction setting, a model is evaluated not only by the number of correct predictions, but also by the ability to filter out dubious predictions by estimating the confidence of individual predictions. Such confidence-aware evaluation is crucial for determining whether to trust zero-shot predictions of language models. In this paper, we apply the selective prediction setting to an existing benchmark, LAMA probe, and conduct extensive experiments with recent neural language models and different confidence functions. We empirically show that our Selective-LAMA evaluation is more robust to the effect of simple guesses than the conventional accuracy-based evaluation. Our evaluation reveals the importance of the choice of confidence functions by showing that simply relying on token probabilities is not always the best choice. Further analysis shows that various confidence functions exhibit different preferences over predicted tokens for a given context.

1 Introduction

Recently, knowledge stored in pre-trained language models has been intensively investigated. Many studies have suggested that language models trained on a large amount of textual corpora, such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2019; Brown et al., 2020), store both linguistic knowledge (Warstadt et al., 2019; Mi- aschi et al., 2020) and factual and commonsense

knowledge (Bosselut et al., 2019; Roberts et al., 2020) during training. However, this knowledge is embedded in the parameters of these language models and thus is difficult to interpret, in contrast to symbolic knowledge bases, which allows us to inspect and edit stored facts explicitly.

Petroni et al. (2019) proposed a benchmark task, the LAMA probe, that aims at evaluating the amount of relational knowledge, such as commonsense knowledge and facts, which is stored in a language model. In LAMA probe, a relational fact is converted into a cloze statement (*query*) and then given to a language model as a fill-in-the-blank question. If the language model fills in the blank with the correct answer, the model is considered to possess “knowledge” of the relation. According to Petroni et al.’s experiments, the BERT language model (Devlin et al., 2019) has a comparable performance to a supervised relation extraction baseline, with precision ranging from 10.5 to 32.3 depending on the dataset type.

However, in many applications, we are concerned not only with the amount of the knowledge extracted from a language model, but also with its reliability. This is because large pre-trained language models are known to fluently generate “facts” that they have never seen (Cao et al., 2018; Rohrbach et al., 2018; Müller et al., 2020). Therefore, it is crucial to know when we can trust the output of a language model. The LAMA probe framework does not cover this issue, as it always forces the model to output an answer for all instances, regardless of whether the model really “knows” the answer to a query. This means that it implicitly trusts all outputs of a language model to the same degree.

Figure 1 shows an example suggesting that a pre-trained language model is not always using its knowledge for prediction. The figure shows the distribution of predicted tokens for a particular relation in the original LAMA probe benchmark

(place-of-birth). We can see that three tokens account for more than half of the wrong predictions. This indicates that the model has a bias which it acquired during training, probably due to the input template used, rather than using actual question-specific knowledge about individual facts.

To address this issue, we apply the selective prediction (El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017) setting to the LAMA probe and propose a new evaluation framework, *Selective-LAMA*, to evaluate both the amount of knowledge in a pre-trained language model and the model’s ability to estimate the reliability of its prediction. Selective prediction is a framework by which a system can choose whether to output the individual predictions of a model based on the prediction results. Specifically, we consider the selection with guaranteed risk control setting (Geifman and El-Yaniv, 2017), where the system computes confidence scores of individual predictions to determine whether it outputs the predictions. A system is evaluated by the number of predictions it can make while maintaining a risk of error below a certain level. To achieve high performance, a system is required not only to answer many questions correctly, but also to accurately estimate the model’s confidence about individual facts and determine when the system should not answer a question.

In this paper, we focus on masked language models and address the following research questions: (1) whether the pre-trained language model has the ability to estimate the confidence of individual predictions and (2) how various confidence metrics affect the ability of a system to do that. With our proposed Selective-LAMA framework, we examine several basic confidence functions that can be computed using only language model predictions and do not require additional datasets or external knowledge sources. We empirically verify that the selective prediction evaluation is less likely to overestimate predictions with template-related biases than the conventional accuracy-based evaluation. The results of the experiments suggest that the choice of confidence functions also influences the results, showing that simply using token probability is a strong baseline but not always the best choice, and that the optimal confidence function depends on both the model and the dataset. We hope that the selective prediction framework facilitates an under-explored research direction of utilizing predictions of language models in a more reliable way.

Dataset: Google-RE, Model: BERT-base
Relation: place-of-birth
Input: “X (Subject) was born in [MASK].”

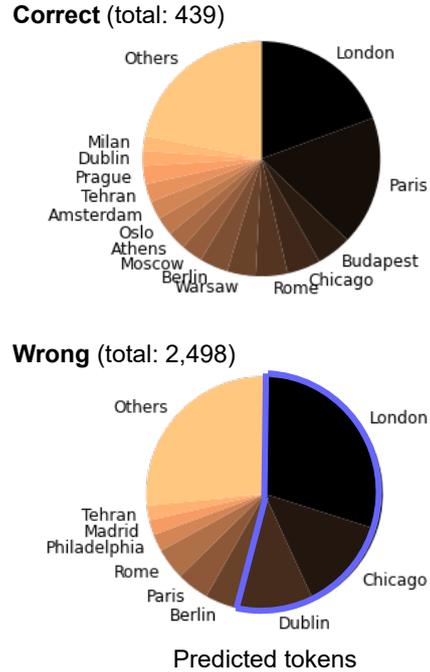


Figure 1: Composition of predicted tokens in each of the correct (top) and wrong (bottom) predictions by BERT-base for the place-of-birth relation in the Google-RE dataset (size: 2,937). Just three tokens account for more than half of the wrong predictions, implying that the model has a template-dependent bias.

2 Selective Prediction

Under the selective prediction setting (El-Yaniv and Wiener, 2010; Geifman and El-Yaniv, 2017), a *selective classifier* determines whether a system should output the prediction of the model. We consider a classification problem from an input space \mathcal{X} to a set of labels \mathcal{Y} . A selective classifier (f, g) consists of an original classification model $f : \mathcal{X} \rightarrow \mathcal{Y}$ and a *selection function* $g : \mathcal{X} \rightarrow \{0, 1\}$. Given an input example $x \in \mathcal{X}$, a selection function determines whether the system outputs the prediction $f(x) \in \mathcal{Y}$:

$$(f, g)(x) := \begin{cases} f(x) & \text{if } g(x) = 1 \\ \text{don't know} & \text{if } g(x) = 0 \end{cases}. \quad (1)$$

Geifman and El-Yaniv (2017) introduced the selection with guaranteed risk (SGR) setting, which uses a confidence-based selection function:

$$g(x) = \begin{cases} 1 & \text{if } \phi(x) \geq \beta \\ 0 & \text{if } \phi(x) < \beta \end{cases}, \quad (2)$$

where $\phi(x) : \mathcal{X} \rightarrow \mathbb{R}$ is the *confidence score function* of f . The system outputs the prediction if the confidence score exceeds the threshold $\beta \in \mathbb{R}$. This setting allows a user to adjust the error risk generated by the system by appropriately setting the value of β . Specifically, increasing β decreases the number of cases predicted by the system while reducing the risk of making a wrong prediction.

Under the SGR setting, there is a risk-coverage trade-off between the risk $(N_{\text{pred}} - N_{\text{corr}})/N_{\text{pred}}$ that a selective classifier will make a wrong prediction and the coverage N_{pred}/N of the predictions made by the system. Here, N , N_{pred} , and N_{corr} denote the number of all examples, predicted examples, and correct predictions, respectively. The performance of a selective classifier is evaluated based on the AUC of the risk-coverage curve (RC-AUC) obtained by changing β in the selection function (2). A smaller RC-AUC value indicates a lower risk of making a wrong prediction. In practice, the threshold will be determined by the level of risk acceptable to the users.

3 Selective-LAMA

3.1 LAMA Probe and Model Prediction

In the original LAMA probe, a relational fact is converted into a natural sentence using templates and input into the language model. For example, when querying about an entity that has a relationship of born-in with “Dante,” the input to the language model will be “Dante was born in [MASK].” where [MASK] is a special token that represents the mask token. The model output for the masked position is considered the answer to the query.¹ The templates are manually designed for each relation type.

Following the original study (Petroni et al., 2019), we focus on bi-directional language models. Given the input sentence with a mask token at the t -th position $x = W_{\setminus t} := (w_1, \dots, w_{t-1}, [\text{MASK}], w_{t+1}, \dots, w_{|W|})$, the language model predicts the probability distribution of the t -th token $P_{\text{LM}}(w_t | W_{\setminus t})$. The model prediction is the token w' with the highest probability:

$$f(x) = w' := \arg \max_{w_t} P_{\text{LM}}(w_t | W_{\setminus t}). \quad (3)$$

We denote the sentence in which the masked position is filled with w' by W' .

¹For simplicity, the target is limited to entities comprising a single word.

3.2 Confidence Functions

As the task is to evaluate the knowledge present in a pre-trained language model, we select confidence functions that use only the prediction of the language model and do not require additional training or external knowledge sources. The following is the list of confidence functions that we investigate.

Token (T) The simplest confidence function is to use the log probability of the predicted token w' (3) directly:

$$\phi_{\text{T}}(x) = \log P_{\text{LM}}(w' | W_{\setminus t}). \quad (4)$$

Sent (S) Sentence-level likelihood is widely used in the context of sentence acceptability and fact-checking (Lau et al., 2020; Lee et al., 2021). This reflects how natural the entire sentence is when the predicted token is substituted into the mask position. Here, we adopt the pseudo-log likelihood (Salazar et al., 2020) for masked language models normalized by sentence length:

$$\phi_{\text{S}}(x) = \frac{1}{|W'|} \sum_{u=1}^{|W'|} \log P_{\text{LM}}(w_u | W'_{\setminus u}). \quad (5)$$

Gap (G) Let w'' be the token with the second-largest probability by the model. The confidence score is then calculated as follows:

$$\phi_{\text{G}}(x) = \log P_{\text{LM}}(w' | W_{\setminus t}) - \log P_{\text{LM}}(w'' | W_{\setminus t}). \quad (6)$$

This function is based on the assumption that a model makes a confident prediction when the probability of the predicted token is significantly larger than that of other tokens.

Reranking (R) The following function is based on the assumption that, if the confidence of the prediction is high, the score for the prediction is consistently higher than those of other candidates even when different metrics are used. First, we obtain top- K predictions \mathcal{W} based on the token log probability (3). Then, we re-rank those candidates using another score function ψ . Let $\text{rank}_{\psi}(w')$ be the rank of w' after the reranking. The confidence score is subsequently computed as follows:

$$\phi_{\text{R}}(x) = \log_2 \frac{K}{\text{rank}_{\psi}(w')} = \log_2 K - \log_2 \text{rank}_{\psi}(w'). \quad (7)$$

The above score function is essentially a measure based only on the new rank after the reranking and has been used to assess the risk of language models to memorize privacy information (Carlini et al., 2019). In the experiments, we apply $K = 100$ and use the Sent score $\phi_{\text{S}}(x)$ for ψ .

DropoutMean (DM) Dropout-based metrics have been widely used to estimate uncertainty of deep neural network models (Gal and Ghahramani, 2016). The basic concept is to use dropout to sample slightly different model parameters that yield different predictions and to use stochastic information to estimate the model uncertainty. Following (Kamath et al., 2020), we adopt two dropout-based measures. We apply M different dropout masks to the language model’s layers and obtain different probability distributions. Let $P_{\text{LM}}^{(m)}(w'|W_{\setminus t})$ denote the m -th output ($m \in \{1, \dots, M\}$). DropoutMean takes the mean of the M outputs:

$$\phi_{\text{DM}}(x) = \frac{1}{M} \sum_{m=1}^M P_{\text{LM}}^{(m)}(w'|W_{\setminus t}), \quad (8)$$

which can be considered an ensemble of the M model predictions.

DropoutVar (DV) Similarly, DropoutVar utilizes the variance of the outputs. As large variance implies high model uncertainty, we take the negative variance of the outputs:

$$\phi_{\text{DV}}(x) = -\frac{1}{M} \sum_{m=1}^M (P_{\text{LM}}^{(m)}(w'|W_{\setminus t}) - \phi_{\text{DM}}(x))^2. \quad (9)$$

In our experiments, we apply $M = 30$ different dropout masks for each input, using the same dropout ratios as those used to train the models.

TemplateDiff (TD) A large portion of the LAMA probe benchmark consists of instances based on subject-relation-object triples. These instances share relation-specific templates, such as “<subj> was born in [MASK].”, where the subject of each triple is substituted for <subj>. Cao et al. (2021) found that predictions of language models are highly biased by templates and the impact of subject entities are limited. Inspired by this observation, we define a confidence measure that assesses the impact of subject entities to predictions. Let W_{temp} be a template-only input sentence where the subject of the input $W_{\setminus t}$ is replaced by the mask token, e.g. “[MASK] was born in [MASK].” Then, we calculate the confidence by comparing the log probabilities of the prediction with and without the subject entity mention:

$$\phi_{\text{TD}}(x) = P_{\text{LM}}(w'|W_{\setminus t}) - P_{\text{LM}}(w'|W_{\text{temp}}). \quad (10)$$

4 Experiments

The proposed Selective-LAMA framework allows us to evaluate the ability of language models to recognize questions for which they *do not know* the answer. To see how the proposed framework affects the evaluation of language models, in Section 4.2, we first compare the evaluation based on the Selective-LAMA framework with the conventional accuracy-based evaluation, focusing on the sensitivity to biased predictions. Then, in Section 4.3, we present a comprehensive study of the performance of three masked language models on different datasets using the confidence functions introduced in Section 3.2.

4.1 Experimental Settings

We used the same data set as the original LAMA benchmark for our experiment and evaluated it with our proposed Selective-LAMA framework. The benchmark consists of four datasets: GoogleRE, T-REx, ConceptNet, and SQuAD. The GoogleRE and T-REx datasets contain relational facts extracted from Wikipedia. The ConceptNet dataset contains relational knowledge about commonsense extracted from the ConceptNet dataset (Speer and Havasi, 2012). The SQuAD dataset (Rajpurkar et al., 2016) is based on a question answering dataset of the same name, but the questions are rewritten in cloze style. As all these datasets, except for ConceptNet, use Wikipedia as the knowledge source, evidence for the correct answer should be found in Wikipedia. For language models, we use BERT-base (110 M parameters), BERT-large (340 M parameters), and RoBERTa-base (Liu et al., 2019). Because these models are trained using Wikipedia, it is expected that the models have seen the correct answers for the queries during training.

4.2 Template Bias Robustness

In the selective prediction framework, the performance of language models is evaluated by RC-AUC (Section 2), while the original LAMA benchmark uses the accuracy of the top-1 predictions as the evaluation metric. A disadvantage of accuracy-based evaluation is that the amount of knowledge of a language model can be overestimated by counting lucky guesses. Such lucky guesses can affect the evaluation results, especially in cases where the model’s predictions are biased by relation-specific templates (Figure 1).

We investigate how these evaluation metrics are

		BERT-base		BERT-large		RoBERTa-base	
		Cov ^A	Cov ^P	Cov ^A	Cov ^P	Cov ^A	Cov ^P
Accuracy		0.387	-0.247	0.469	-0.244	0.512	-0.224
RC-AUC	Token	0.344 ↓	-0.316 ↓	0.438 ↓	-0.292 ↓	0.484 ↓	-0.277 ↓
(negative)	Sent	0.355 ↓	-0.290 ↓	0.441 ↓	-0.285 ↓	0.499 ↓	-0.249 ↓
	Gap	0.351 ↓	-0.314 ↓	0.430 ↓	-0.294 ↓	0.474 ↓	-0.285 ↓
	Reranking	0.350 ↓	-0.286 ↓	0.452 ↓	-0.283 ↓	0.498 ↓	-0.266 ↓
	DropoutMean	0.338 ↓	-0.319 ↓	0.433 ↓	-0.293 ↓	0.486 ↓	-0.280 ↓
	DropoutVar	0.419 ↑	-0.125 ↑	0.470 ↑	-0.124 ↑	0.456 ↓	-0.166 ↑
	TemplateDiff	0.349 ↓	-0.317 ↓	0.427 ↓	-0.299 ↓	0.486 ↓	-0.271 ↓

Table 1: Correlation between evaluation metrics and template bias metrics: answer coverage (Cov^A) and prediction coverage (Cov^P) on the T-REx dataset. Here, we use the sign-reversed RC-AUC values for easier interpretation.

affected by template-related biases using the T-REx subset of the LAMA benchmark, which contains 34k facts about 41 different relations with their corresponding templates. To quantify template-related biases, we introduce two indicators: *prediction coverage* and *answer coverage*.

Prediction coverage quantifies biases in model predictions for a given template. If a model often predicts the same answers for a template, it is likely that the predictions are heavily influenced by the template, rather than using knowledge of individual subject entities. Let $\mathcal{D}_r = (\{(s_i, o_i)\}_{i=1}^{N_r}, t_r)$ denote a relation subset containing N_r fact triples (s_i, r, o_i) of relation r and a template t_r . We represent the input sentence corresponding to the i -th fact by $t_r(s_i)$. For each relation subset \mathcal{D}_r , we first identify five most frequent tokens $\mathcal{W}^{\text{freq}}(r)$ predicted by a model. Prediction coverage is the proportion of predicted tokens covered by these tokens:

$$\text{Cov}^P(r) = \frac{|\{i \mid f(t_r(s_i)) \in \mathcal{W}^{\text{freq}}(r)\}|}{N_r}. \quad (11)$$

Answer coverage quantifies biases in a relation subset in the test set. If the distribution of the correct answers for a relation subset is skewed towards a few particular entities, the subset can be easily answered by exploiting the bias without using the knowledge of individual subject entities. Answer coverage is calculated as the proportion of gold answers covered by the frequently predicted tokens:

$$\text{Cov}^A(r) = \frac{|\{i \mid o_i \in \mathcal{W}^{\text{freq}}(r)\}|}{N_r}. \quad (12)$$

Table 1 shows the correlation between the bias indicators and the evaluation metrics including accuracy and (negative) RC-AUC calculated with different confidence functions. Compared to the conventional accuracy metric, all RC-AUC metrics except DropoutVar show a weaker positive

correlation with answer coverage and a stronger negative correlation with prediction coverage, indicating that the RC-AUC metrics are less likely to overestimate template-biased predictions and results from intrinsically biased test examples.

Figure 2 shows the output of the BERT-base model for two relation subsets P36 and P1412. Although the accuracy scores for both subsets are around 0.6, for P1412, both the prediction and answer distributions are biased towards a small number of entities, leading to high prediction and answer coverage. The Token confidence scoring fails to discriminate between correct and incorrect predictions in this subset, resulting in high risk at a low coverage point. Evaluation based on the RC-AUC score successfully captures the difference between these two cases and avoids overestimating the results from biased predictions.

4.3 Selective-LAMA Evaluation and Analysis

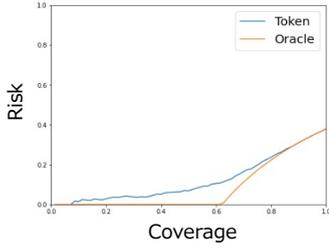
Overall performance on different datasets

Table 2 shows the RC-AUC scores achieved by different confidence functions on various datasets. We also calculate the performance lower bound based on an oracle confidence function that gives 1 to all correct predictions and 0 to incorrect ones. While the simple Token metric constantly performs well, the best confidence function depends on the model and dataset. Notably, Gap and TemplateDiff perform better on the datasets of Wikipedia fact triples, Google-RE and T-REx, than on ConceptNet and SQuAD, outperforming the Token metric in some cases. The breakdown of the results on the T-REx dataset indicates that the performance of confidence functions also depend on relation templates. We further investigate this phenomenon below.

$r = \text{P36}$ ("The capital of X (Subject) is [MASK].") Accuracy = 0.621, RC-AUC = 0.121

$\mathcal{W}^{\text{freq}}(r)$: Rome (1.9%), Baghdad (1.7%), Paris (1.7%), Bangor (1.7%), Kabul (1.4%)

Prediction coverage: 0.084, Answer coverage: 0.047

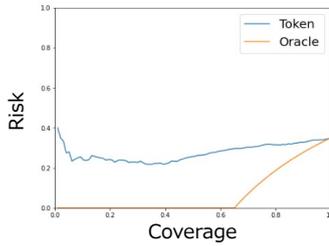


Subject	Gold	Predict	ϕ_T
Sri Lanka	Colombo	Colombo	-0.001
Bratislava Region	Bratislava	Bratislava	-0.001
Albania	Tirana	Tirana	-0.002
Tirana District	Tirana	Tirana	-0.002
Hiroshima Prefecture	Hiroshima	Hiroshima	-0.002
Brest Region	Brest	Brest	-0.003
South Korea	Seoul	Seoul	-0.003
Afghanistan	Kabul	Kabul	-0.003
Bosnia and Herzegovina	Sarajevo	Sarajevo	-0.003
Democratic Republic of Afghanistan	Kabul	Kabul	-0.003

$r = \text{P1412}$ ("X (Subject) used to communicate in [MASK].") Accuracy = 0.650, RC-AUC = 0.278

$\mathcal{W}^{\text{freq}}(r)$: English (38.6%), French (15.9%), Spanish (10.0%), Italian (9.4%), Russian (4.5%)

Prediction coverage: 0.784, Answer coverage: 0.687



Subject	Gold	Predict	ϕ_T
Adrianus Valerius	Dutch	Latin	-0.490
Muhammad Ali	English	Arabic	-0.575
Gloria Estefan	Spanish	Spanish	-0.587
Imre Nagy	Hungarian	Hungarian	-0.610
Sextus Pompeius Festus	Latin	Latin	-0.619
Hieronymus Fabricius	Latin	Latin	-0.635
Infante Juan, Count of Barcelona	Spanish	Spanish	-0.637
Ramon Llull	Catalan	Spanish	-0.665
Lau Kar-leung	Chinese	Cantonese	-0.724
Juan Bautista Villalpando	Spanish	Spanish	-0.749

Figure 2: BERT-base results for relation subsets $r = \text{P36}$ and $r = \text{P1412}$. While the model performance is similar in terms of accuracy, the RC-AUC scores exhibit a large difference. Left: Risk-coverage curves of Token and the Oracle confidence scores. Right: Top 20 predictions sorted by the Token confidence score ϕ_T . The gray-shaded rows indicate incorrect predictions. Many incorrect predictions for P1412 indicate that the model suffers from high risk even at a low coverage point.

When does a confidence function beat another?

For the T-REx dataset in Table 2, Gap and TemplateDiff outperform the Token metric for BERT-base and RoBERTa-base, respectively. We choose these two cases and perform a pairwise comparison for each relation type to identify the properties that determine the preference for one confidence function over the other. The results in Table 3 show that Gap is preferred over Token for easier relations with high accuracy and low RC-AUC for BERT-base, whereas TemplateDiff is preferred over Token for more difficult relations for RoBERTa-base. The subset where Gap is preferred over Token also shows lower prediction coverage, which might be because the Gap function is not good at handling overconfident predictions by definition.

Confidence functions and relation templates

To understand whether and how different confidence functions prioritize one relation over another, we visualize in Figure 3 the composition of the relation types of input examples sorted by the con-

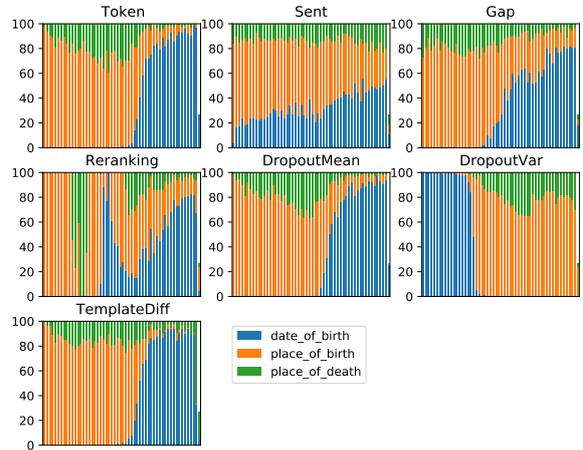


Figure 3: Breakdown of relation types of BERT-base predictions on Google-RE, sorted by confidence scores (left is the largest).

fidence scores in the Google-RE dataset predicted by the BERT-base model. The Google-RE dataset contains three relation types, date-of-birth, place-of-birth, and place-of-death. Evidently, the BERT-base language model tend to pro-

Model	Conf.	Google-RE	T-REx				ConceptNet	SQuAD	All
			1-1	N-1	N-M	All			
BERT-base	Token	.775	.118	.434	.611	.478	.686	.755	.545
	Sent	.834	.163	.549	.776	.594	.797	.815	.652
	Gap	.798	.133	.422	.604	.470	.714	.794	.548
	Reranking	.835	.248	.580	.623	.597	.834	.798	.633
	DropoutMean	.775	.123	.425	.609	.473	.690	.762	.543
	DropoutVar	.962	.525	.834	.883	.850	.918	.912	.886
	TemplateDiff	.778	.119	.427	.603	.472	.782	-	-
	Oracle	.663	.070	.301	.456	.344	.551	.583	.413
BERT-large	Token	.763	.085	.409	.575	.445	.616	.669	.506
	Sent	.815	.119	.520	.740	.560	.738	.768	.614
	Gap	.801	.092	.412	.597	.456	.650	.712	.525
	Reranking	.826	.170	.552	.610	.576	.792	.785	.609
	DropoutMean	.762	.086	.402	.572	.441	.616	.670	.504
	DropoutVar	.960	.370	.775	.894	.817	.881	.907	.858
	TemplateDiff	.763	.084	.406	.574	.444	.730	-	-
	Oracle	.648	.048	.277	.459	.327	.489	.522	.388
RoBERTa-base	Token	.818	.191	.540	.635	.562	.618	.741	.599
	Sent	.876	.267	.631	.761	.657	.754	.780	.716
	Gap	.827	.197	.545	.632	.565	.657	.782	.610
	Reranking	.865	.276	.637	.627	.636	.804	.828	.669
	DropoutMean	.815	.201	.536	.633	.562	.615	.744	.599
	DropoutVar	.979	.643	.924	.920	.920	.896	.907	.923
	TemplateDiff	.813	.189	.537	.626	.558	.744	-	-
	Oracle	.730	.106	.416	.492	.432	.503	.571	.474

Table 2: RC-AUC calculated on each dataset (lower is better). For T-REx, the results on three splits divided by the property of the relations are also provided: one-to-one relations (1-1), many-to-one relations (N-1) and many-to-many relations (N-M). “Oracle” represents the best possible performance that could be achieved by an oracle confidence function that gives 1 to all correct predictions and 0 to incorrect ones. TemplateDiff cannot be calculated for SQuAD as the instances do not contain subject entities.

duce high probability outputs for a certain relation type, namely, place-of-birth. While Gap, DropoutMean, and TemplateDiff follow the same trend as that of Token, Sent and Reranking are less sensitive to relation types. DropoutVar shows the opposite trend. While the Token metric is effective in many cases, one should be aware of the potential bias this confidence function may introduce.

Table 4 compares the most frequent predictions of BERT-base on the Google-RE dataset ranked top by two different metrics: Token and Reranking. We can observe similar distributions for the date-of-birth relation type. This indicates that the model is strongly biased toward a limited vocabulary for this particular relation type. For the other two relation types, the frequent words in the top predictions are clearly different between Token and Reranking. However, while the overlap of the top-ranked predictions between them are small, both results have strong preference toward a few particular tokens for each relation type. For place-of-birth, five tokens account for more

than 50% of the top-ranked predictions for both Token and Reranking. In place-of-death, just one token occupies around 40% of the top predictions. The results indicate that these confidence functions produce different template biases rather than that one is more robust to template biases than the other.

Using confidence functions for prediction

In the experiments above, model predictions are always determined by the token log probability as in (3). However, some of the confidence functions introduced in Section 3.2 can also be used directly to determine the prediction as an alternative to (3). Therefore, we investigate whether effective confidence functions are also effective in improving prediction accuracy (P @ 1) when used directly for token prediction. For Gap, we extend the original definition (6) so that we can apply the function to token candidates that are not ranked first in terms of token probability. Let $w^{(k)}$ denote the k -th best prediction based on the model’s predicted token probability. Then, the extended Gap function is

	BERT-base			
	All	Token-win	Gap-win	Δ
Accuracy	0.311	0.283	0.413	-0.130
RC-AUC Token	0.558	0.577	0.466	0.111
RC-AUC Gap	0.566	0.597	0.443	0.154
Answer Cov.	0.285	0.276	0.334	-0.058
Prediction Cov.	0.547	0.579	0.464	0.115

	RoBERTa-base			
	All	Token-win	TD-win	Δ
Accuracy	0.242	0.315	0.231	0.085
RC-AUC Token	0.643	0.545	0.657	-0.112
RC-AUC TD	0.638	0.546	0.650	-0.103
Answer Cov.	0.237	0.285	0.235	0.050
Prediction Cov.	0.562	0.586	0.541	0.045

Table 3: Comparison of two confidence functions on the T-REx dataset (Token-Gap for BERT-base and Token-TemplateDiff (TD) for RoBERTa-base). The average value of each metric is displayed for the entire T-REx dataset (All) and the subset for which the confidence function X outperforms the other (X-win). Δ stands for the difference between the two subsets.

defined as follows:

$$\phi_G(x) = \frac{1}{k}(\log P_{LM}(w^{(k)}|W_{\setminus t}) - \log P_{LM}(w^{(k+1)}|W_{\setminus t})). \quad (13)$$

The Gap score for the lowest ranked prediction is defined as zero. The computation of the Sent score requires $\mathcal{O}(|W'| \cdot V)$ forward computations for each instance, where V is the vocabulary size. To save computational cost, we approximate the prediction results by limiting the token candidates to the top 100 results based on the Token score (3).

Table 5 shows the results. For all models, the best performance on all data is achieved by DropoutMean. However, all functions, except for DropoutVar, show a quite competitive performance in terms of precision. Unlike for confidence estimation, no advantage is observed for Gap and TemplateDiff on the T-REx dataset. Overall, the performance of these confidence functions is flat when they are used directly for token prediction. Furthermore, there is no strong correlation between the performance of each confidence function as a predictor and a confidence estimator. The results suggest that effective metrics for inference and confidence estimation should be designed based on different strategies.

5 Related Work

In NLP, the reliability of the model responses has been discussed mainly in the field of question answering. Estimating the confidence of an answer is critical in quiz competitions, such as Jeopardy,

since the system has to decide when to answer the questions (Ferrucci et al., 2010). Kamath et al. (2020) recently introduced a selective prediction setting to question answering tasks and then evaluated the performance of the models on out-of-domain questions. Jiang et al. (2021) addressed a similar problem, but focused on a calibration of the model prediction on QA tasks. While they focused on extractive or multiple-choice QA tasks where a limited number of candidate answers are available, our focus is on the knowledge probing of language models where the candidate answer is the entire vocabulary and, thus, false positives are more frequent.

Several studies have addressed the reliability issue of pre-trained language models as a calibration problem; the goal of these studies is to train a well-calibrated language model that makes accurate confidence estimation. Desai and Durrett (2020) investigate the calibration level of pre-trained language models, focusing on BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). They evaluate the “out-of-the-box” performance of these models without post-processing, as well as the performance of post-hoc calibration methods (e.g., temperature scaling and label smoothing). Kong et al. (2020) proposed regularization methods to better calibrate pre-trained language models. Both studies assume access to (at least in-domain) training data of the target tasks on which parameterized calibration models can be trained. In contrast, our study primarily aims to explore better signals in pre-trained language models to estimate the knowledge they store. Thus, we focus on methods that do not require additional training data or an external knowledge source. Although training-based methods (e.g., temperature scaling) have the potential to achieve better performance in terms of calibration, optimal parameters vary depending on models and tasks, especially when evaluated in out-of-domain datasets (Desai and Durrett, 2020).

In our experiments, all queries have at least one correct answer. Therefore, when a model cannot answer a question correctly, this implies that it did not acquire the correct knowledge during training or that its knowledge was not elicited by the natural language query because of a sub-optimal prompt (Jiang et al., 2020). However, there are also cases where the question is essentially impossible to answer due to ambiguity (Zhang and Choi, 2021) or false presupposition (Kim et al., 2021).

Relation	Confidence	Top predictions
date-of-birth	Token	1979 (47), 1944 (33), 1988 (10), 1990 (8)
	Reranking	1979 (44), 1944 (32), 1953 (13), 1970 (3), 1949 (2)
place-of-birth	Token	Budapest (18), Prague (10), Istanbul (8), Athens (8), Paris (7), Moscow (7), Helsinki (6), Bucharest (6), Tehran (5), Stockholm (4)
	Reranking	London (30), Dublin (12), Paris (12), Moscow (5), Madrid (4), Philadelphia (4), Chicago (4), Warsaw (3), Tehran (3), Berlin (2)
place-of-death	Token	Paris (38), Rome (32), Moscow (6), Madrid (4), infancy (4), office (3), Athens (2), Helsinki (2), Warsaw (2), Amsterdam (2)
	Reranking	London (46), Paris (14), Rome (7), office (6), Moscow (4), Munich (3), Amsterdam (3), infancy (2), prison (2), Stockholm (2)

Table 4: Comparison of the most frequent tokens among the top-100 predictions based on different confidence scores. Based on the results on the Google-RE dataset with the BERT-base model. The numbers in parentheses represent the frequency of the predictions.

Model	Pred.	GRE	TREx	CNet	SQ	All
BERT-base	T	10.3	29.6	15.8	14.1	24.3
	S	10.5	29.6	14.6	14.4	24.1
	G	9.7	28.6	15.3	15.1	23.5
	DM	10.3	29.8	15.4	14.1	24.4
	DV	0.2	0.1	0.1	0.0	0.1
	TD	9.6	29.4	14.2	-	-
BERT-large	T	11.0	31.0	19.3	17.4	26.1
	S	11.2	31.5	17.6	15.7	26.1
	G	10.4	29.6	18.6	17.4	25.0
	DM	10.9	31.7	19.6	17.7	26.7
	DV	0.2	0.0	0.0	0.0	0.1
	TD	10.6	30.5	17.0	-	-
RoBERTa-base	T	7.5	23.0	18.5	14.7	20.2
	S	8.2	24.3	17.0	12.2	20.7
	G	7.6	22.0	17.4	14.7	19.3
	DM	8.0	24.4	18.3	15.7	21.1
	DV	0.1	0.1	0.1	0.0	0.1
	TD	7.5	23.2	16.4	-	-

Table 5: P@1 based on different prediction scores for each dataset. Bb: BERT-base, Bl: BERT-large, T: Token, S: Sent, G: Gap, DM: DropoutMean, DV: DropoutVar, TD: TemplateDiff, GRE: Google-RE, CNet: ConceptNet, SQ: SQuAD. We omit the result of using the Reranking score because the results are the same as those of Sent by definition.

An investigation of such cases remains a direction of future research.

6 Conclusion

In this paper, we introduced the selective prediction setting to the LAMA probe benchmark to evaluate both the amount of relational knowledge stored in a language model and the ability of the models to effectively filter out unconfident predictions. We compared different confidence functions that can be calculated using only the model parameters and the output information. The experimental results are summarized as follows:

- The selective prediction evaluation is more robust to template-related biases than the conventional accuracy-based evaluation (Table 1).
- The token log probability is not always the best choice, and the best confidence function depends on the language model and the dataset (Table 2).
- Different confidence functions have different preferences over relation types and predicted tokens, even though all functions are based solely on the model output (Figure 3, Table 4).
- There is no strong correlation between the performance of each confidence function as a predictor and a confidence estimator (Table 5).

Future studies will include a detailed analysis of the relationship between tasks, models, and confidence scores. Moreover, more sophisticated methods will be explored to ensure the reliability of language model predictions under various tasks. The code for our work is attached as supplementary material.

Limitations

In this paper, we focused on evaluating the predictions of masked language models on the LAMA probe benchmark. Although our proposed framework is easily applicable to other kinds of language model with small adjustments, some of the confidence functions we investigated require properties specific to particular language models and datasets. For instance, Token and Gap functions require the prediction to be a single token, and TemplateDiff requires templates for subject-relation-object triples.

Ethics Statement

Data and code

In our experiments, we use the original LAMA benchmark dataset from [Petroni et al. \(2019\)](#) as is. All data are based on publicly available data sources and data statistics can be found in the original paper. Parts of the code are based on LAMA². The license of the code can be found in the supplementary material.

Details of experiments

The experiments were conducted using a 2.4GHz CPU and an NVIDIA TESLA P100 GPU. Inference time was 1–1.5 s per instance for BERT-base and 2–3 s per instance for BERT-large.

Potential risks

This study evaluates the knowledge stored in language models considering the reliability of model predictions. However, it should be emphasized that the outputs of the selective classifier constructed by the proposed method do not guarantee the correctness of the model predictions. For the validation of each fact, this method should only be used as an aid, and the final decision should be made by the user.

Acknowledgements

We thank Prof. Simone Teufel, Marco Cognetta and anonymous reviewers for their valuable feedback. This study was carried out using the TSUB-AME3.0 supercomputer at Tokyo Institute of Technology. This work was partially supported by JSPS KAKENHI Grant Number 19H01118.

References

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. [COMET: Commonsense transformers for automatic knowledge graph construction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens

Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. [Knowledgeable or educated guess? revisiting language models as knowledge bases](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact-aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI’18/IAAI’18/EAAI’18*. AAAI Press.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium, SEC’19*, pages 267–284, USA. USENIX Association.

Shrey Desai and Greg Durrett. 2020. [Calibration of pre-trained transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ran El-Yaniv and Yair Wiener. 2010. On the Foundations of Noise-free Selective Classification. *Journal of Machine Learning Research*, 11(53):1605–1641.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. [Building watson: An overview of the deepqa project](#). *AI Magazine*, 31(3):59–79.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd*

²<https://github.com/facebookresearch/LAMA>

- International Conference on International Conference on Machine Learning - Volume 48, ICML'16*, page 1050–1059. JMLR.org.
- Yonatan Geifman and Ran El-Yaniv. 2017. Selective Classification for Deep Neural Networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, pages 4885–4894. Curran Associates Inc.
- Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. [How can we know when language models know? on the calibration of language models for question answering](#). *Transactions of the Association for Computational Linguistics*, 9:962–977.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. [How can we know what language models know?](#) *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Amita Kamath, Robin Jia, and Percy Liang. 2020. [Selective question answering under domain shift](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.
- Najoung Kim, Ellie Pavlick, Burcu Karagol Ayan, and Deepak Ramachandran. 2021. [Which linguist invented the lightbulb? presupposition verification for question-answering](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3932–3945, Online. Association for Computational Linguistics.
- Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. [Calibrated language model fine-tuning for in- and out-of-distribution data](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1326–1340, Online. Association for Computational Linguistics.
- Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. [How furiously can colorless green ideas sleep? sentence acceptability in context](#). *Transactions of the Association for Computational Linguistics*, 8:296–310.
- Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. [Towards few-shot fact-checking via perplexity](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#).
- Alessio Miaschi, Dominique Brunato, Felice Dell’Orletta, and Giulia Venturi. 2020. [Linguistic profiling of a neural language model](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 745–756, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Mathias Müller, Annette Rios, and Rico Sennrich. 2020. [Domain robustness in neural machine translation](#). In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. [Language models as knowledge bases?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. [How much knowledge can you pack into the parameters of a language model?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. [Object hallucination in image captioning](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.
- Julian Salazar, Davis Liang, Toan Q. Nguyen, and Kartrin Kirchhoff. 2020. [Masked language model scoring](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.
- Robyn Speer and Catherine Havasi. 2012. [Representing general relational knowledge in ConceptNet 5](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. [Investigating BERT’s knowledge of language: Five analysis methods with NPIs](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Michael Zhang and Eunsol Choi. 2021. [SituatingQA: Incorporating extra-linguistic contexts into QA](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Multi-View Source Ablation for Faithful Summarization

Shuyang Cao^{1*} Liang Ma² Di Lu²
Robert L. Logan IV² Joel Tetreault² Alejandro Jaimes²

¹University of Michigan, Ann Arbor ²Dataminr Inc.
caoshuy@umich.edu {lma, dlu, rlogan, jtetreault, ajaimes}@dataminr.com

Abstract

In this paper, we present MUFASSA (Multi-view Faithfulness Scoring via Source Ablation), a metric for *evaluating* faithfulness of abstractive summaries, and for guiding *training* of more faithful summarizers. For evaluation, MUFASSA employs different strategies (e.g., masking entity mentions) to first remove information from the source document to form *multiple ablated views*. Then, the faithfulness level of each token in a generated summary is measured by the difference between the token generation probabilities when given the original document and the ablated document as inputs to trained summarizers. For training, MUFASSA uses a novel *word truncation* objective that drops unfaithful tokens located by MUFASSA in both the decoder input and output. Alignments with human-annotated faithfulness labels on AGGREFACT show that MUFASSA is comparable to or better than existing metrics built on classifiers or QA models pre-trained on other tasks. In experiments on summarization with XSum and CNN/DailyMail, models trained with word truncation using MUFASSA outperform competitive methods according to both automatic faithfulness metrics and human assessments.

1 Introduction

Automatic text summarization systems have made great strides with the use of large pre-trained models, which enable more precise identification of salient content in the document and generation of summaries with human-level fluency (Lewis et al., 2020; Raffel et al., 2020). However, model-generated summaries frequently contain unfaithful information that either contradict the source text or cannot be verified (Kryscinski et al., 2020), creating risks in real-world deployment of automatic text summarization models and motivating the development of models targeting more faithful summaries (Cao et al., 2018).

* Work done during an internship at Dataminr.

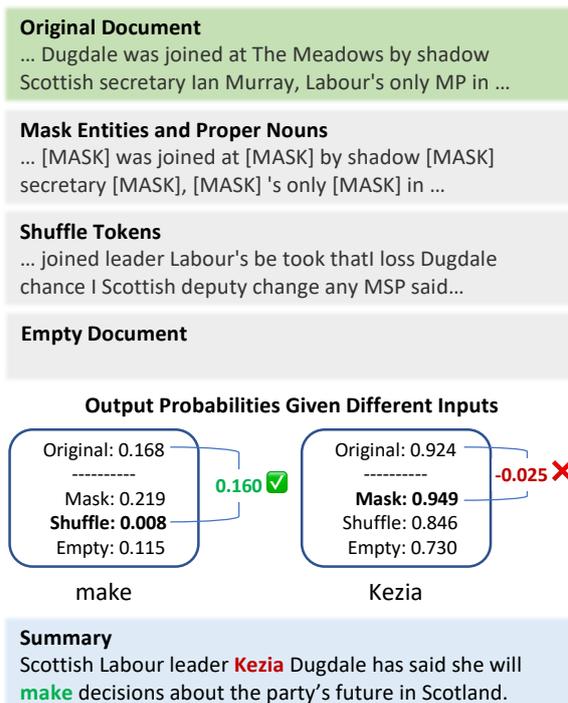


Figure 1: MUFASSA estimates the faithfulness level of each summary token as the difference between probabilities given by trained summarizers with the original source document and the ablated document chosen for the token (e.g., shuffling tokens for verbs). The large difference for “make” indicates it is **faithful** to the document, while the small difference for “Kezia” indicate an **unfaithful** token.

As overlap-based metrics such as ROUGE (Lin, 2004) struggle to reflect the faithfulness level of generated summaries (Falke et al., 2019), a number of model-based faithfulness metrics have been introduced. These metrics leverage external textual entailment (Goyal and Durrett, 2020; Laban et al., 2022) and question answering models (Wang et al., 2020; Scialom et al., 2021) to measure the degree to which claims in the summary align to information in the source text. Yet, there remains substantial room for improvement (Tang et al., 2022). Moreover, despite being relevance or complementary to

each other, for building faithful summarization systems, faithfulness metrics are rarely exploited and researchers mainly resort to more complex training routines (Cao and Wang, 2021) or model architectures (Zhu et al., 2021).

To this end, our work introduces a faithfulness metric that (1) more accurately estimates summary faithfulness levels; and (2) can be easily integrated into training objectives to produce more faithful summarization systems. In our method, which we call **MUFASSA** (**M**ulti-view **F**aithfulness **S**coring via **S**ource **A**blation), multiple *ablated source documents* are constructed by masking entities, shuffling tokens, and discarding all tokens in the original source document, to remove crucial information for the generation of different content in the summary, as shown at the top of Figure 1. Since the ablated sources do not include sufficient information for generating the corresponding summary, the differences between token output probabilities given by the original input and each ablated input should be high for the faithful content and low for the unfaithful one (e.g., “make” and “Kezia” in Figure 1). We then aggregate the differences to obtain the summary faithfulness score.

Additionally, to train faithful summarization systems, we adapt loss truncation (Kang and Hashimoto, 2020) and nullify losses on summary tokens that are deemed less faithful by MUFASSA during training. Compared to using training losses for detecting unfaithful content in the original loss truncation, MUFASSA provides a more accurate estimation of token faithfulness in training samples, and more faithful summaries can therefore be produced while maintaining informativeness. We further design **word truncation**, to drop the generation dependency on less faithful words in the auto-regressive decoder by completely removing them from the decoder input.¹

Two sets of experiments are conducted to show the effectiveness of MUFASSA at evaluating and training faithful summarizers. First, we compare with existing faithfulness metrics on AGGREGFACT (Tang et al., 2022), a curated benchmark for meta evaluation of faithfulness metrics, where MUFASSA obtains the best average performance. We then leverage MUFASSA during model training on XSum (Narayan et al., 2018) and CNN/DailyMail (Hermann et al., 2015). Com-

pared to baselines and recent models built with augmented data or more complex training objectives, MUFASSA-trained models produce summaries with competitive or better faithfulness while maintaining the coverage of salient document information, according to both automatic faithfulness metrics and human judgments.

In summary, we make the following contributions:

- We propose MUFASSA, a novel automatic evaluation metric that measures summary faithfulness by the extent to which the generation of summary tokens relies on information in the document.
- In addition, to leverage MUFASSA during training, we design word truncation, a novel training objective that discards less faithful tokens identified by MUFASSA from the training samples to induce more faithful summarizers.

2 Related Work

Faithfulness Metrics. Recent analyses have shown that summaries with high ROUGE scores (Lin, 2004) can contain information that is not faithful to the source documents (Falke et al., 2019; Kryscinski et al., 2020). This observation has prompted the development of a number of faithfulness metrics that measure the extent to which summarization models produce unfaithful outputs. Existing faithfulness metrics largely fall under two broad categories: (1) entailment-based metrics, and (2) QA-based metrics. Entailment-based metrics evaluate the faithfulness of summaries by computing entailment levels of the sentences (Laban et al., 2022), dependency arcs (Goyal and Durrett, 2020), or semantic graphs (Ribeiro et al., 2022) of the summaries against the corresponding documents. QA-based metrics use models for question generation and question answering to determine whether questions derived from the summary can be answered using the document (Wang et al., 2020; Durmus et al., 2020) or questions derived from the document can be answered using the summary (Scialom et al., 2019). Results are enhanced using a combination of both approaches (Scialom et al., 2021) and adding a question filtering stage (Fabbri et al., 2022).

In this work we pursue an alternative approach that detects unfaithful outputs by analyzing differences in token probabilities when conditioning on different views of the source document. This

¹Our code is available at <https://shuyangcao.github.io/projects/mufassa/>.

approach was first proposed by Xie et al. (2021), whose CoCo metric measures probability differences on pre-specified sets of key terms in the output of models conditioned on partially masked sequences. Our proposed metric, MUFASSA, builds upon CoCo in two ways. First, we eschew the need for key terms, instead providing an approach for assessing faithfulness of different types of tokens (e.g., entities, relations). This not only makes MUFASSA easier to use, but, as we will see in Section 4, also results in better performance. Secondly, we introduce a strategy for incorporating these metrics into training, and demonstrate in Section 5.2 that this training scheme produces more faithful summarizers.

Faithful Summarization. In parallel with advancements in faithfulness metrics, researchers have also investigated approaches to train more faithful summarizers. One class of approaches propose to modify model architectures to leverage external knowledge graphs (Zhu et al., 2021) and OpenIE triplets (Cao et al., 2018). Another class of approaches investigates modifications to training data, either by removing unfaithful training examples (Wan and Bansal, 2022) or training models to differentiate between faithful and unfaithful summaries (Liu et al., 2021; Cao and Wang, 2021). In this paper, we study a third class of approaches that modify the model’s loss function. Our work builds upon the method of loss truncation (Kang and Hashimoto, 2020), which omits a fraction of low confidence predictions from the loss function during training. We show that loss truncation can better improve faithfulness by using MUFASSA to determine which predictions to ignore, and that even better results can be obtained using our novel word truncation objective that omits removed tokens from the input (Tables 2 and 3).

3 MUFASSA: Multi-View Information Ablation

In this section, we first introduce the formulation of faithfulness estimation by MUFASSA (§3.1) and the construction of inputs with different information ablated (§3.2). We then describe how MUFASSA can be incorporated into model training through loss truncation and our proposed word truncation (§3.3). We fine-tune BART (Lewis et al., 2020) for all summarization models in this paper.

3.1 Information Ablation

Let T denote the set of tokens in the model vocabulary, and T^* the set of all sequences of tokens in T . Given a summary $y \in T^*$ of document $x \in T^*$, let $\mathcal{I}_{y_i} : T^* \rightarrow T^*$ denote a "view function" that ablates out information from the source document necessary for generating token y_i (i.e., a summarization model conditioned on $\mathcal{I}_{y_i}(x)$ should *not* produce token y_i). We hypothesize that if y_i is not faithful to the source document, then y_i can be generated with $\mathcal{I}_{y_i}(x)$ by a summarization model, i.e., the output probability $p(y_i|y_{<i}, \mathcal{I}_{y_i}(x))$ should remain high. Based on this hypothesis, we propose the faithfulness level of summary token y_i estimated by:

$$m(y_i) = p_{\theta}(y_i|y_{<i}, x) - p_{\theta'}(y_i|y_{<i}, \mathcal{I}_{y_i}(x)) \quad (1)$$

where a higher $m(y_i)$ suggests a higher faithfulness level, and p_{θ} and $p_{\theta'}$ denote summarization models parameterized by θ and θ' . We train p_{θ} and $p_{\theta'}$ on the experimented summarization dataset by maximizing $p_{\theta}(y_i|y_{<i}, x)$ and $p_{\theta'}(y_i|y_{<i}, \mathcal{I}_{y_i}(x))$ with the cross-entropy objective. To obtain the sample-level faithfulness score, we aggregate the faithfulness estimation over all summary tokens: $\frac{1}{L} \sum_{i=1}^L m(y_i)$, where L is the length of the summary. Notably, the token-level faithfulness scores aggregated by MUFASSA are based on the generation probabilities given the *already generated tokens*. The contextual nature of the token-level faithfulness scores allows MUFASSA to account for unfaithfulness of phrases and sentences in the generated summary.

3.2 Multi-View Ablation

Careful construction of \mathcal{I}_{y_i} is crucial to accurate faithfulness estimation of y_i . To reduce the computational cost, instead of creating a unique ablated document $\mathcal{I}_{y_i}(x)$ for every y_i , MUFASSA groups the summary tokens into three different sets— Y_{ent} , Y_{rel} , and Y_{other} —according to their part-of-speech (POS) tags and entity labels,² and constructs a single view of the source document I_Y to compute $m(y_i)$ for each token $y_i \in Y$.³ In the following paragraphs, we describe the construction strategies and their corresponding token sets.

²We use SpaCy (Honnibal and Montani, 2017) for named entity recognition and part-of-speech (POS) tagging.

³Thus, all of the token-level scores are computed using only 4 forward passes of the model: one for each set, and one for the original source document.

Mask Entities and Proper Nouns. Named entities and proper nouns are important components of facts and events that constitute summaries, so our first set of tokens, Y_{ent} is comprised of all tokens that are part of a proper noun or named entity. Since the faithful production of these tokens in the summary relies on the entity and proper noun information available in the document, we replace all named entities and proper nouns in the document with [MASK] tokens. E.g.:

$$\mathcal{I}_{Y_{\text{ent}}}(x_j) = \begin{cases} [\text{MASK}], & \text{if } x_j \text{ is a proper noun} \\ & \text{or named entity.} \\ x_j, & \text{otherwise} \end{cases}$$

where we adopt the convenient abuse of notation $\mathcal{I}_{Y_{\text{ent}}}(x_j)$ to denote the j th output of $\mathcal{I}_{Y_{\text{ent}}}(x)$.

Shuffle Tokens. Besides entities and proper nouns themselves, faithful summaries require correct resolution of their *relations* and *modifications*. Thus the second set of tokens we consider Y_{rel} is comprised of all of the verbs, adjectives, adverbs, and adpositions in y . To drop the relation and modification information, we randomly shuffle all the tokens in the document, i.e., $\mathcal{I}_{Y_{\text{rel}}}(x) = \sigma(x)$ where σ is a random permutation.

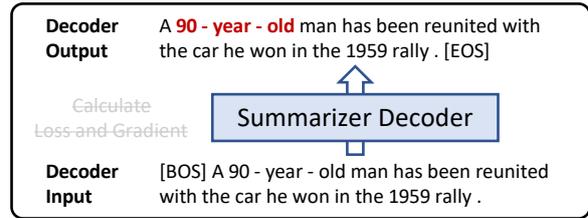
Empty Document. Lastly, for the remaining tokens not covered by the two aforementioned strategies, Y_{other} , we discard all tokens in the document, i.e., $\mathcal{I}_{Y_{\text{other}}}(x) = \emptyset$. Stopwords and punctuation are not included in Y_{other} . With empty documents, the summarizer resembles an unconditional language model. While empty documents have been used in previous work (Xu and Durrett, 2021; Xie et al., 2021), we argue that some spurious correlations might emerge from tokens that imply the document topic (e.g., tokens that usually occur with the topics) and aggressively taking empty documents for measuring the faithfulness level of any token would prevent the exposure of such spurious correlations.

3.3 Using MUFASSA during Training

We modify loss truncation (Kang and Hashimoto, 2020) to enable MUFASSA for training summarization models. Loss truncation considers tokens that still yield high training losses after several training epochs as noisy tokens and ignores their training losses.⁴ For each sample, the training objective with loss truncation is formally defined as:

⁴The loss truncation method is proposed at the sample level. We follow Goyal et al. (2022) to extend loss truncation to the token level.

① Faithfulness Estimation with MUFASSA



② Training with Truncation

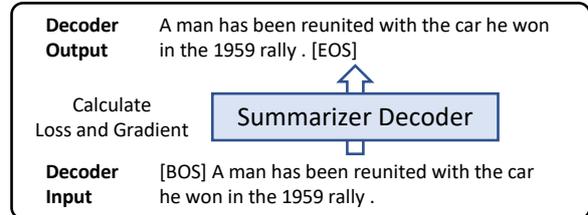


Figure 2: Illustration of our proposed word truncation training objective. In the first pass, model optimization is disabled and MUFASSA detects less faithful summary tokens. In the second pass, the summarizer is trained on the summary with these tokens discarded.

$$-\frac{1}{L} \sum_{i=1}^L \mathbb{1}_{[-\log p_{\theta}(y_i|y_{<i}, x) < Q_c^l]} \cdot \log p_{\theta}(y_i|y_{<i}, x) \quad (2)$$

where Q_c^l is the c percentile of the list Q^l tracking past training losses.

However, high loss might not well estimate the faithfulness level of each token. Thus, we propose to instead use faithfulness scores output by MUFASSA to identify unfaithful tokens to omit from the loss computation. That is, we replace Q^l with Q^m that records the faithfulness levels of past tokens measured by MUFASSA. The resulting training objective is:

$$-\frac{1}{L} \sum_{i=1}^L \mathbb{1}_{[m(y_i) > Q_c^m]} \cdot \log p_{\theta}(y_i|y_{<i}, x) \quad (3)$$

where the summarizer p_{θ} that is being optimized is also used for obtaining $m(y_i)$ as in Equation (1). Before switching to our modified training objective, we first optimize the summarization model for several epochs with the traditional cross entropy objective (henceforth, warm-up stage) following Goyal et al. (2022). The number of epochs for the warm-up stage is set to 3 in our experiments. We tune the percentile c on validation sets to achieve a balance of summary faithfulness and coverage.

Word Truncation. Although loss truncation avoids optimizing the likelihood of less faithful

tokens, they are retained in the generation context for the remaining tokens. Thus, the summarization model might insist on generating them in order to generate the remaining content, yielding unfaithful summaries. To this end, we extend loss truncation by additionally removing tokens that are less faithful from the decoder input during training. As illustrated in Figure 2, after feeding the original decoder input to the model, the faithfulness levels of summary tokens are first estimated by MUFASSA in the decoder output. At this step, we do not calculate the loss or perform any gradient back-propagation. With the less faithful tokens detected, we remove them from both the decoder input and output of the training sample. Finally, we train the summarizer with the truncated decoder input and output.

4 Metric Experiments

We first test how well MUFASSA agrees with human judgments on summary faithfulness.

Datasets. We experiment on AGGREGFACT (Tang et al., 2022), a benchmark consisting of document-summary pairs and their binary faithfulness labels annotated by most recent work (Kryscinski et al., 2020; Maynez et al., 2020; Huang et al., 2020; Fabbri et al., 2021; Pagnoni et al., 2021; Cao and Wang, 2021; Goyal and Durrett, 2021; Cao et al., 2022). We use the SOTA subset of AGGREGFACT where the summaries are produced by state-of-the-art summarizers built from large pre-trained models. The SOTA subset contains 1,335 and 1,018 samples annotated on XSum (Narayan et al., 2018) and CNN/DailyMail (Hermann et al., 2015) respectively.

Comparisons. For comparison, we include results of existing state-of-the-art faithfulness evaluation metrics:

- QUESTEVAL (Scialom et al., 2021) is a QA-based metric that answers questions created from the summary using the document and vice versa. To obtain the evaluation score, the word overlaps between the answers given by the pre-trained QA model and the ground-truth answers used for generating the questions are aggregated over all questions.
- SUMMAC (Laban et al., 2022) is an entailment-based metric that first computes the entailment

Metric	AGGREGFACT-	AGGREGFACT-	Average
	XSUM	CNN	
QUESTEVAL	61.6	71.5	<u>66.5</u>
SUMMAC	66.3	66.7	<u>66.5</u>
CoCo	59.3	68.4	63.8
PROBABILITY	54.7	68.5	61.6
EMPTY	<u>65.1</u>	67.0	66.1
MUFASSA	64.8	<u>69.2</u>	67.0

Table 1: The Area Under the ROC Curve (AUC) of different faithfulness metrics on AGGREGFACT. The top two results on each split are highlighted with a **boldface** and underline, respectively. MUFASSA achieves better average performance than existing metrics.

score between each pair of document and summary sentences. For each summary sentence, its entailment scores with document sentences are then binned into a histogram and transformed into the sentence-level faithfulness score via a 1-D convolutional layer. The mean of the sentence-level scores is then taken as the evaluation score.

- CoCo (Xie et al., 2021) is a model causality-based metric. We use its best-performing variant that masks document sentences that contain words in the summary. The difference between summary output probabilities given by a trained summarizer using the original document and the masked document is taken as the evaluation score.

We also compare with two variants of MUFASSA that: (1) directly take the output probability given by the original input as the faithfulness estimation (PROBABILITY), which no longer calculates the difference in Equation 1; or (2) only use the empty document as the ablated input (EMPTY).

Results. The performance by each metric is measured with the Area Under the ROC Curve (AUC). As shown in Table 1, when solely taking the empty document as the ablated input, the resulting metric already matches the performance of the other existing metrics except for QUESTEVAL on CNN/DailyMail, showing the effectiveness of ablation.

Furthermore, boosted by multi-view information ablation that provides model interpretation of finer granularity, MUFASSA *yields the best average performance on AGGREGFACT*, even without leveraging models obtained from other datasets.

We also observe that the average performance of CoCo is worse than the empty document ablation,

though COCO employs a more sophisticated masking strategy. As their masking strategy is based on exact word matching, it might struggle to ablate information for abstractive summaries, leading to less accurate faithfulness estimation.

5 Summarization Experiments

To verify the effectiveness of our methods to produce more faithful summaries, we train summarizers on popular summarization datasets with our proposed loss truncation and word truncation methods equipped with MUFASSA.

5.1 Experimental Setup

Datasets. We conduct experiments on XSum (Narayan et al., 2018) and CNN/DailyMail (Hermann et al., 2015) datasets. Both datasets are built from news articles, with the XSum summaries tending to be more abstractive than its counterpart. We follow the official data splits of XSum and CNN/DailyMail, which respectively contain 204,045/11,332/11,334 and 287,113/13,368/11,490 samples in the train/validation/test sets.

Evaluation Metrics. For faithfulness evaluation, we use SUMMAC and QUESTEVAL, which respectively obtain the best performance on the XSum and CNN/DailyMail splits of the AGGREFACT benchmark in §4. In addition, we report ROUGE scores, including variants based on the unigram overlap (ROUGE-1), bigram overlap (ROUGE-2), and longest common subsequence (ROUGE-L)⁵.

Comparisons. Besides the models fine-tuned only with the cross entropy objective (BART) and additionally with loss truncation (LOSSTRUNC), we also compare with DAE-based loss truncation (Goyal and Durrett, 2021) and CLIFF (Cao and Wang, 2021). Specifically, DAE assesses the entailment level of each dependency arc in the summary and then locates less faithful tokens by aggregating the entailment levels of their attached dependency arcs, where the training losses are discarded. By contrast, without using truncating losses, CLIFF augments the model training with negative samples (i.e., synthetic incorrect summaries) and adopts contrastive learning (Khosla et al., 2020) to help model distinguish incorrect summaries from correct summaries.

⁵Please refer to Appendix B for ROUGE-1 and ROUGE-2 scores

Model	SUMMAC	QUESTEVAL	R-L
<i>XSum</i>			
BART	24.36	36.66	37.19
CLIFF	24.60*	36.94	36.43
DAE	23.81	36.38	30.32
LOSSTRUNC	24.52	37.12*	34.60
+ MUFASSA	24.63*	37.22*	33.77
+ WORDTRUNC	24.85*	36.75	34.66
<i>CNN/DailyMail</i>			
BART	80.54	60.17	41.14
CLIFF	78.95	59.03	41.06
LOSSTRUNC	80.50	60.22	41.36*
+ MUFASSA	81.84*	60.04	40.69
+ WORDTRUNC	83.01*	60.44*	40.40

Table 2: Evaluation of summary generation on XSum and CNN/DailyMail. R-L: ROUGE-L. MUFASSA-based loss and word truncation yields summarizers with the best faithfulness scores. *Significantly better than BART with approx. randomization test ($p < 0.005$).

5.2 Results

We report results on XSum and CNN/DailyMail in Table 2. Our modified loss truncation produces summarizers with better performance than all comparisons on faithfulness metrics on both datasets, except for QUESTEVAL on CNN/DailyMail, which suggests the usefulness of MUFASSA in training summarization models with improved faithfulness. Moreover, additionally truncating the less faithful tokens in the decoding context during training consistently advances the SUMMAC scores, achieving the best SUMMAC scores on both datasets.

Though DAE trains a dependency arc entailment scorer with augmented negative samples, the external dependency parser requires processing the summarization dataset into a text format that does not align with the natural text format used by large model pre-training, yielding worse performance.

Additionally, we observe that summaries from our models have less competitive ROUGE scores. This could be due to unfaithful content in the human reference summaries, which has been identified as an issue in previous work (Maynez et al., 2020). In this regard, further human evaluation is conducted in the next section.

Human Evaluation. We hire human annotators on Amazon Mechanical Turk⁶ to rate system summaries on three aspects:

⁶<https://www.mturk.com/>

Model	Faith.	Cover.	Coher.
BART	3.32	3.22	5.00
CLIFF	3.45	3.22	4.97
LOSSTRUNC	3.41	3.17	4.97
+ MUFASSA	3.45	3.21	4.97
+ WORDTRUNC	3.59*	3.35	4.92

Table 3: Human evaluation results on XSum. Faith.: faithfulness; Cover.: coverage; Coher.: coherence. Our model using word truncation guided by MUFASSA achieves the best summary faithfulness and coverage. Krippendorff’s $\alpha \geq 0.35$ for all aspects.

- **Faithfulness:** How well the factual information in the summary accurately matches the information in the article;
- **Coverage:** How well the summary covers the important information in the article; and
- **Coherence:** How coherent the summary is on its own.

Each aspect is rated on a Likert scale from 1 (worst) to 5 (best).

We randomly select 80 articles from XSum, where models are more prone to errors (Pagnoni et al., 2021), and ask annotators to judge summaries generated by our models as well as comparisons including BART, CLIFF, and the original loss truncation. During annotation, the order of the system summaries is shuffled and each system summary is evaluated by three annotators. Details of the human evaluation such as payment, annotator qualification, and interface screenshots are included in Appendix C.

According to human judges (Table 3), without word truncation, MUFASSA improves the identification of less faithful tokens, outperforming the original loss truncation and matching CLIFF on summary faithfulness and coverage. Adding word truncation further encourages the summarizer to generate summaries with promoted faithfulness and content coverage, leading to the best scores on both aspects. We also find that removing less faithful summary tokens from the training samples only has minor effects on the summary coherence.

Case Study. Figure 3 displays an example article from XSum and its corresponding summaries generated by summarizers trained with different methods. The model trained with the original loss truncation does not attempt to modify the unfaithful entity “the Six Nations”, as training losses do not accurately reflect faithfulness levels. While the unfaithful entity is removed from the output when

Article: Amos dislocated a shoulder in the 32-8 defeat by Australia and will have an operation in the next week. The 22-year-old Dragons wing tweeted: "Operation set for Monday, aiming to be back in February". "It’s unlucky for Hallam but a great opportunity for Keelan," said Wales assistant coach Neil Jenkins ... "We’re going to miss him, but back-three is a position where we have strength in depth." Giles has been in outstanding form for Ospreys, scoring eight tries in five appearances for the region this season ...

BART: Ospreys wing Keelan Giles could make his Wales debut after Hallam Amos was ruled out of **the Six Nations** with a shoulder injury.

LOSSTRUNC: Ospreys wing Keelan Giles has been named in Wales’ back-three after Hallam Amos was ruled out of **the Six Nations**.

LOSSTRUNC + MUFASSA: Ospreys wing Keelan Giles could make his Wales debut after Hallam Amos was ruled out for **the rest of the season**.

LOSSTRUNC + MUFASSA + WORDTRUNC: Ospreys wing Keelan Giles is in line to replace injured Hallam Amos in Wales’ back-three.

Figure 3: Example generated summaries. Unfaithful information is shaded with red. Our model trained with word truncation signaled by MUFASSA generates a faithful summary.

the original loss truncation is augmented with MUFASSA, the summarizer produces another piece of unfaithful information. After applying word truncation, the model learns to stop generation, producing the faithful summary.

6 Additional Experiments

In this section, we inspect the effects of important design choices in MUFASSA (§6.1). Furthermore, to show the possibility of applying MUFASSA to other tasks, we train data-to-text generation models with our proposed methods (§6.2).

6.1 Ablation Study

We examine the effects on faithfulness estimation by the source ablations with masked entities and proper nouns, and shuffled tokens. For the faithfulness levels of summary tokens induced by each ablated input, when the ablated input is not used,

Metric	AGGREGFACT-		Average
	XSUM	CNN	
MUFASSA	64.8	71.2	68.0
<i>Not Using All Source Ablations</i>			
w/o Mask	64.5	69.8	67.1
w/o Shuffle	65.3	69.9	67.6
w/o Mask & Shuffle	65.1	67.0	66.1
<i>Not Assigning Ablations to Different Summary Tokens</i>			
Average	62.4	67.6	65.0
Minimum	53.5	51.3	52.4
Maximum	65.2	68.2	66.7

Table 4: The Area Under the ROC Curve (AUC) by variants of MUFASSA on AGGREGFACT. The best results on each split is highlighted with **boldface**. Removing any component of MUFASSA reduces its robustness, leading to lower average AUC.

we instead obtain their faithfulness levels with the empty document input. Moreover, we investigate the benefits of assigning each ablated input to different summary tokens. Concretely, we consider three variants, where the faithfulness level of each token is calculated by either taking the average, minimum, or maximum value of the faithfulness levels measured with the three source ablations.

Including multiple source ablations enhances the robustness of MUFASSA, as indicated by its best average performance on AGGREGFACT in Table 4. Compared to the source ablation with shuffled tokens, the source ablation with masked entities and proper nouns contributes more to the accurate faithfulness estimation by MUFASSA, dropping which leads to a larger performance degradation.

Simple aggregations (i.e., average, minimum, and maximum) of the faithfulness levels measured by the three source ablations produce lower AUC scores, justifying MUFASSA’s design of leveraging different token-specific source ablations.

6.2 Extension to Data-to-Text Generation

While this work focuses on summarization, we also explore extending our methods to other tasks. Specifically, we conduct experiments on a data-to-text dataset, WikiPerson (Wang et al., 2018) which requires the generation model to produce a natural language description for a person’s career, given the infobox in the corresponding Wikipedia biography article. Details of the dataset and experiment setup are included in Appendix A.2.

We evaluate outputs with faithfulness-aware data-to-text metrics, including: PARENT (Dhingra et al., 2019) that additionally aligns n-grams

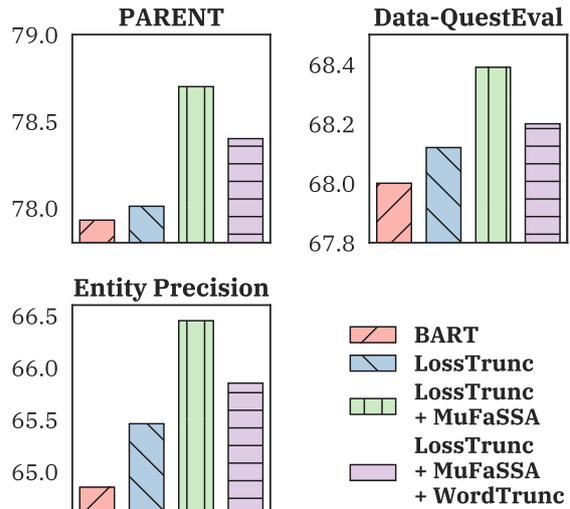


Figure 4: Automatic evaluation results on WikiPerson. Our models achieve better performance than comparisons with the cross entropy objective and original loss truncation objective, implying the effectiveness of MUFASSA on other generation tasks.

from the reference and the system generation to the source table; and Data-QuestEval (Rebuffel et al., 2021) which replaces the text-based question generation and answering models in the original QUESTEVAL with table-based models to adapt to data-to-text tasks. Moreover, we compute the precision of named entities in the generated text, suggested by recent work on text generation (Logan IV et al., 2022).

Our models outperform comparisons on all metrics, as shown in Figure 4, indicating the potential adaptations of MUFASSA on conditional generation tasks other than text summarization to improve output faithfulness. Word truncation does not further improve the performance on WikiPerson. We suspect that data-to-text tasks might require more samples to learn coherent generation due to the modality difference between the input and output, while word truncation reduces the number of tokens that the model can learn from.

7 Conclusion

We studied improving faithful summary evaluation and generation. Our proposed method, MUFASSA, estimates the faithfulness level of a summary token by the decrease in its generation probability after ablating crucial information from the source document. Multiple ablation strategies are used by MUFASSA for different summary tokens to achieve accurate faithfulness estimation. We also

designed word truncation for improved integration of MUFASSA into model training. Experiments on AGGREFACT show that MUFASSA better aligns with human faithfulness labels than existing metrics. When used for highlighting less faithful tokens during summarizer training, MUFASSA leads to summaries with enhanced faithfulness, which is further boosted by word truncation, achieving better faithfulness than competitive comparisons, as measured by both automatic metrics and human annotators.

Limitations

While MUFASSA does not rely on textual entailment or question answering models, the construction of ablated inputs in MUFASSA still requires some existing NLP tools such as named entity recognizers and POS taggers. Therefore, the accuracy of the faithfulness estimation would be limited by the performance of these tools. Also, construction strategies other than the ones presented in this paper might rely on more advanced NLP tools, further amplifying the limitation. This could be a significant issue for low-resource languages where basic NLP tools have not been established.

In addition, our word truncation training objective incurs some computational overhead. First, it takes two forward passes, though gradient back-propagation is not performed in the first pass. Second, similar to the original loss truncation, word truncation maintains a list for storing the faithfulness levels of past tokens and needs to calculate the threshold of faithfulness levels for truncating less faithful tokens.

Ethical Consideration

Previous studies have shown that large pre-trained models embed biases and might create harm to certain populations. While MUFASSA is built with large pre-trained models, we do not study if the faithfulness estimation by MUFASSA is biased towards any population in this work (e.g., produce higher scores for texts including a population than text including another population). As recent work finds that BERTScore which is also based on large pre-trained models has biases (Sun et al., 2022), we suggest users carefully investigate the potential biases in the model before applying it in real-world situations.

References

- Meng Cao, Yue Dong, and Jackie Cheung. 2022. [Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354, Dublin, Ireland. Association for Computational Linguistics.
- Shuyang Cao and Lu Wang. 2021. [CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. [Faithful to the original: Fact aware neural abstractive summarization](#). In *thirty-second AAAI conference on artificial intelligence*.
- Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. [Handling divergent reference texts when evaluating table-to-text generation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.
- Esin Durmus, He He, and Mona Diab. 2020. [FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.
- Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. [QAFactEval: Improved QA-based factual consistency evaluation for summarization](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.
- Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. [SummEval: Re-evaluating summarization evaluation](#). *Transactions of the Association for Computational Linguistics*, 9:391–409.
- Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. [Ranking generated summaries by correctness: An interesting but challenging application for natural language inference](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.
- Tanya Goyal and Greg Durrett. 2020. [Evaluating factuality in generation with dependency-level entailment](#).

- In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.
- Tanya Goyal and Greg Durrett. 2021. [Annotating and modeling fine-grained factuality in summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.
- Tanya Goyal, Jiacheng Xu, Junyi Jessy Li, and Greg Durrett. 2022. [Training dynamics for text summarization models](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2061–2073, Dublin, Ireland. Association for Computational Linguistics.
- Max Grusky, Mor Naaman, and Yoav Artzi. 2018. [Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.
- Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.
- Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. [What have we achieved on text summarization?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online. Association for Computational Linguistics.
- Daniel Kang and Tatsunori B. Hashimoto. 2020. [Improved natural language generation via loss truncation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673.
- Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. [Evaluating the factual consistency of abstractive text summarization](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.
- Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. [SummaC: Re-visiting NLI-based models for inconsistency detection in summarization](#). *Transactions of the Association for Computational Linguistics*, 10:163–177.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Wei Liu, Huanqin Wu, Wenjing Mu, Zhen Li, Tao Chen, and Dan Nie. 2021. [Co2sum: Contrastive learning for factual-consistent abstractive summarization](#). *arXiv preprint arXiv:2112.01147*.
- Robert L. Logan IV, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. 2022. [FRUIT: Faithfully reflecting updated information in text](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3670–3686, Seattle, United States. Association for Computational Linguistics.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [fairseq: A fast, extensible toolkit for sequence modeling](#). In *Proceedings of NAACL-HLT 2019: Demonstrations*.
- Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. [Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Clement Rebuffel, Thomas Scialom, Laure Soulier, Benjamin Piwowarski, Sylvain Lamprier, Jacopo Staiano, Geoffrey Scuttheeten, and Patrick Gallinari. 2021. [Data-QuestEval: A referenceless metric for data-to-text semantic evaluation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8029–8036, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. [FactGraph: Evaluating factuality in summarization with semantic graph representations](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3238–3253, Seattle, United States. Association for Computational Linguistics.
- Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. [QuestEval: Summarization asks for fact-based evaluation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. [Answers unite! unsupervised metrics for reinforced summarization models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.
- Tianxiang Sun, Junliang He, Xipeng Qiu, and Xuanjing Huang. 2022. [Bertscore is unfair: On social bias in language model-based metrics for text generation](#).
- Liyan Tang, Tanya Goyal, Alexander R. Fabbri, Philippe Laban, Jiacheng Xu, Semih Yahvuz, Wojciech Kryściński, Justin F. Rousseau, and Greg Durrett. 2022. [Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors](#).
- David Wan and Mohit Bansal. 2022. [FactPEGASUS: Factuality-aware pre-training and fine-tuning for abstractive summarization](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1010–1028, Seattle, United States. Association for Computational Linguistics.
- Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. [Asking and answering questions to evaluate the factual consistency of summaries](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.
- Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. [Describing a knowledge base](#). In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 10–21, Tilburg University, The Netherlands. Association for Computational Linguistics.
- Yuexiang Xie, Fei Sun, Yang Deng, Yaliang Li, and Bolin Ding. 2021. [Factual consistency evaluation for text summarization via counterfactual estimation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 100–110, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jiacheng Xu and Greg Durrett. 2021. [Dissecting generation modes for abstractive summarization models via ablation and attribution](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6925–6940, Online. Association for Computational Linguistics.
- Chenguang Zhu, William Hinthorn, Ruochen Xu, Qingkai Zeng, Michael Zeng, Xuedong Huang, and Meng Jiang. 2021. [Enhancing factual consistency of abstractive summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 718–733, Online. Association for Computational Linguistics.

Original Paper	AGGREGFACT-XSum	AGGREGFACT-CNN
Polytope (Huang et al., 2020)	-	68
SummEval (Fabbri et al., 2021)	-	400
FRANK (Pagnoni et al., 2021)	-	250
Wang’20 (Wang et al., 2020)	239	-
CLIFF (Cao and Wang, 2021)	300	300
Goyal’21 (Goyal and Durrett, 2021)	100	-
Cao’22 (Cao et al., 2022)	696	-

Table 5: Numbers of samples collected from previous work in the SOTA subset of AGGREGFACT.

A Details of Datasets

We include additional details for datasets we use in our paper.

A.1 AGGREGFACT

We show the numbers of samples included in the SOTA subset of AGGREGFACT (Tang et al., 2022) from different studies in Table 5.

A.2 WikiPerson

WikiPerson (Wang et al., 2018) extract Wikipedia articles and the corresponding infoboxes about person entities. For each article, they remove sentences that do not contain any value in the corresponding infobox or only contain entities not in the infobox. The remaining sentences of the article are then taken as the generation target for the infobox.

Statistics. We use the official data split provided by the original paper, which contains 250,186/30,487/29,982 samples in the train/validation/test sets. On average, each infobox contains 7.3 attribute-value pairs and each target output contains 86.3 words.

Experiment Details. On WikiPerson, MUFASSA masks the values in the infoboxes for estimating the faithfulness levels of entities and proper nouns in the outputs. For the remaining tokens in the outputs, we use empty infoboxes as the ablated inputs. We do not consider shuffling tokens of values in infoboxes, as they are mainly entities and proper nouns.

Model	R-1	R-2	Density	Coverage
<i>XSum</i>				
BART	45.41	22.29	1.65	75.70
CLIFF	44.52	21.40	1.69	76.71
DAE	38.94	15.00	1.50	74.80
LOSSTRUNC	42.98	19.13	1.74	78.78
+ MUFASSA	41.93	18.09	1.85	77.95
+ WORDTRUNC	42.36	19.13	1.75	76.82
<i>CNN/DailyMail</i>				
BART	44.32	21.32	20.81	99.00
CLIFF	44.18	21.14	18.89	98.91
LOSSTRUNC	44.50	21.48	20.16	99.02
+ MUFASSA	43.88	20.96	21.52	99.15
+ WORDTRUNC	43.63	20.77	24.57	99.32

Table 6: ROUGE-1 and ROUGE-2 on XSum and CNN/DailyMail. The best result of each metric on each dataset is **bolded**.

Given an infobox, to create the textual input to the model, we first concatenate attributes and their corresponding values with “=” . Then we concatenate all attribute-value pairs together with “|” inserted at the beginning of each attribute-value pair. An example of the converted textual input: “| Name_ID = Thorsten Barg | date of birth = 25 August 1986 | country of citizenship = Germany” .

B Additional Results

We report ROUGE-1 and ROUGE-2 scores on XSum and CNN/DailyMail, which are omitted in §5.2. Both scores follow the trend of ROUGE-L in Table 2. We also examine the abstractiveness of the summaries generated by each system by calculating the density and coverage (Grusky et al., 2018), where we find our system tends to be more extractive on CNN/DailyMail compared to other systems.

C Details of Human Evaluation

Our human evaluation is conducted on Amazon Mechanical Turk (AMT). In the annotation interface (Figure 5 to 8), we provide a detailed instruction of the annotation task, including rubrics, examples, and explanations.

Before launching all annotation samples on AMT, we run two batches for qualification. Each qualification batch contains one article and its corresponding system summaries, and is annotated by 20 workers. We manually inspect the annotation results and filter out workers that return abnormal annotations (e.g., giving high faithfulness scores to

for summaries containing unfaithful content or giving very different scores to identical summaries). We also require the annotators to be located in the US or the UK, with 100 tasks previously completed, and have an approval rate of 100%. A pool of 8 workers for our human evaluation is obtained after the qualification.

For compensation, we pay each annotator \$2.5 for each task (i.e., evaluating system summaries generated for an article) of our human evaluation to approximate an hourly payment of \$15.

D Details of Implementation

We use Fairseq (Ott et al., 2019)⁷ for setting up the training and decoding pipelines. The BART model (Lewis et al., 2020) in our paper is initialized from the `bart.large`⁸ checkpoint provided by Fairseq. We conduct training and decoding on 4 NVIDIA V100 GPUs with 16GB memory.

Training. We use the training hyperparameters in the training script provided by the BART paper⁹. The percentile for obtaining the threshold of faithfulness levels is tuned on the validation set of each dataset. For XSum, we search for the best threshold percentile within [30, 40, 50]. The model with the best SUMMAC score while having a ROUGE-1 score of at least 42 is selected. 40, 50, and 30 are chosen as the percentiles for the models respectively trained with the original loss truncation objective, our loss truncation guided with MUFASSA, and our word truncation objective. For CNN/DailyMail, we search for the best threshold percentile within [5, 10, 20]. The model with the best SUMMAC score while having a ROUGE-1 score of at least 44 is selected. 10, 10, and 5 are chosen as the percentiles for the models respectively trained with the original loss truncation objective, our loss truncation guided with MUFASSA, and our word truncation objective. To avoid incoherent summaries, we only apply word truncation to proper nouns. Due to the computational cost, we train all models for one run.

Decoding. We follow the original BART paper and decode using beam search with beam sizes of 4 and 6 on CNN/DailyMail and XSum. During

decoding, the maximum decoding lengths are 140 and 60 for CNN/DailyMail and XSum.

Running Time. We report the running time on XSum. Training our models with loss truncation or word truncation on XSum takes 10 hours and the decoding takes half an hour.

Model Parameters. Our methods do not increase the number of model parameters. Therefore, our models have 400M parameters, which is the same as the original BART.

E Output Examples

We include more examples of system outputs in Figure 9 and 10.

⁷<https://github.com/pytorch/fairseq>

⁸<https://github.com/pytorch/fairseq/tree/main/examples/bart>

⁹<https://github.com/pytorch/fairseq/blob/main/examples/bart/README.summarization.md>

Task Instructions

There will be many similar HITs for you to perform if you do well at this task. Please follow the instructions carefully for each HIT; we will be reviewing your HITs periodically and if we note any unusual responses, you might not see any additional tasks from us.

During this task, you will read a news article and six different summaries for the article. You will rate the quality of each of the six summaries by four axes: *coherence*, *accuracy*, *coverage*, and *overall quality*.

The rubrics below give specific guidance on how each axis should be rated. Please read the rubrics carefully before continuing to the task.

Jump to [coherence rating](#)

Coherence

For each summary, answer the question "how coherent is the summary on its own?" (on a scale from 1 to 5). A summary is *coherent* if, when read by itself, it's easy to understand and free of English errors. A summary is not coherent if it's difficult to understand what the summary is trying to say. Generally, it's more important that the summary is understandable than it being free of grammar errors. Please **do not penalize incomplete punctuation** (e.g., when there exists only one quote mark in the sentence).

Rubric:

- Score of 1: The summary is impossible to understand.
- Score of 2: The summary has many mistakes or confusing phrasing.
- Score of 3: The summary has some mistakes or confusing phrasing that make it hard to understand.
- Score of 4: The summary has only one or two mistakes or confusing phrasing.
- Score of 5: The summary is perfectly clear.

Jump to [accuracy rating](#)

Accuracy

For each summary, answer the question "how well does the factual information in the summary accurately match the information in the article?" (on a scale of 1 to 5) A summary is *accurate* if it doesn't say things that aren't in the article, it doesn't contradict information in the article, and generally is not misleading.

Even if a piece of information is true according to your knowledge, if it is not mentioned in the article it should not be included in the summary.

Rubric:

- Score of 1: The summary is completely wrong, made up, or exactly contradicts what is written in the article.
- Score of 2: The summary says many things not mentioned in or contradicting the article.
- Score of 3: The summary says at least one substantial thing that is not mentioned in the article, or that contradicts something in the article.
- Score of 4: The summary says anything at all that is not mentioned in the article or contradicts something in the article.
- Score of 5: The summary has no incorrect statements or misleading implications.

Jump to [coverage rating](#)

Coverage

For each summary, answer the question "how well does the summary cover the important information in the article?" (on a scale of 1 to 5). A summary has good *coverage* if it mentions the main information from the article that's important to understand the event described in the article. A summary has poor coverage if someone reading only the summary would be missing several important pieces of information about the event in the article.

Rubric:

- Score of 1: The summary contains no information relevant to the article.
- Score of 2: The summary is missing many important pieces of information required to understand the event.
- Score of 3: The summary is missing at least one crucial piece of information required to understand the event.
- Score of 4: The summary is missing any information (no matter how small) required to understand the event.
- Score of 5: The summary covers all of the important information required to understand the event.

Jump to [overall quality rating](#)

Overall quality

For each summary, answer the question "how good is the summary overall at representing the article?" (on a scale of 1 to 5). This encompasses all of the above axes, as well as the information included in the summary and if it has helped you understand the event. If it's hard to find ways to make the summary better, give the summary a high score. If there are lots of different ways the summary can be made better, give the summary a low score.

Figure 5: Screenshot of our annotation interface (1/4).

Rubric:

- Score of 1: The summary is terrible.
- Score of 2: The summary is a pretty bad representation of the article and needs significant improvement.
- Score of 3: The summary is an okay representation of the article, but could be significantly improved.
- Score of 4: The summary is a pretty good representation of the article, but it's not perfect.
- Score of 5: The summary is an excellent representation of the article.

Example

Now you will review an example article and three associated summaries.

- For each of the summaries, we have provided ratings on the four axes: coherence, accuracy, coverage, and overall quality with explanations for why those ratings were chosen.
- Please review the summaries and their ratings carefully, so you understand how to rate the summaries during the task.
- If you have any questions about how to rate the summaries, please consult the rubric (above).

Example Article

Welsh and UK ministers have been rowing since March over how to finance the commuter lines in and out of Cardiff. Mr Crabb said the scheme - estimated at £309m to £463m - was "probably the most knotty" problem between the two governments but was solvable. The valleys rail electrification is due to be completed between 2019 and 2024. Planned rail improvements will see the upgrade of the main line from London Paddington to Cardiff, which is due to be completed by 2017, and extended to Swansea by 2018 at a cost of £850m. The electrification of the Valleys lines was due to follow, but the plan was thrown into doubt in March by a row over the financing of the project. Speaking on Radio Wales' Sunday Supplement programme, Mr Crabb said rail electrification was the "number one issue" for him. He said: "It's something that I've been spending quite a bit of my summer working on. "There's a bit more work to be done between the two governments on where we think the solution lies, but I think when I go around talking to businesses in south Wales they are desperate to see this problem answered, they want the two governments to be working effectively together." Describing the issue as "a bit of a litmus test" for joint working between Wales and Westminster, he warned the issue "can't drag on indefinitely". "There are engineering teams involved in Network Rail who need to get tasks assigned to them if this huge, enormous, financially-challenging project is to go ahead," he said. "There are some quite hard deadlines in that. But we are talking a short number of months hopefully."

Example summary #1

The electrification of the Valleys rail lines is the "number one issue" for Welsh Secretary Stephen Crabb.

Ratings for Example Summary #1

The summary should be rated as follows:

How coherent is the summary on its own?

It is impossible to understand. 1 2 3 4 5 The summary is perfectly clear.

Explanation for rating: This summary is easy to understand and read with clear language and no grammatical errors and therefore coherent, so we rate it a 5 (of 5).

How well does the factual information in the summary accurately match the article?

The summary is completely wrong, made up, or exactly contradicts what is written in the article. 1 2 3 4 5 The summary has no incorrect statements or misleading implications.

Explanation for rating: The position and first name of Mr. Crabb is unknown from the article. So we rate this summary as 3 (of 5).

How well does the summary cover the important information in the article?

The summary contains no information relevant to the article. 1 2 3 4 5 The summary covers all of the important information required to understand the event in the article.

Explanation for rating: This summary has a fair coverage of the article, but it misses the mention of the underlying reason for the rail line electrification issue. So we rate this summary as 4 (of 5).

How good is the summary overall at representing the article?

It is terrible. 1 2 3 4 5 It is an excellent representation of the article.

Explanation for rating: This summary is okay but it could be **significantly improved** by mentioning the underlying reason for the rail line electrification issue and not including extraneous information about Mr. Crabb. So, we rate this summary as 3 (of 5).

Example Summary #2

Mr. Crabb has said he is "desperate" to see the electrification of the Valleys rail line.

Ratings for Example Summary #2

The summary should be rated as follows:

How coherent is the summary on its own?

It is impossible to understand. 1 2 3 4 5 The summary is perfectly clear.

Explanation for rating: This summary is easy to understand and read with clear language and no grammatical errors and therefore

Figure 6: Screenshot of our annotation interface (2/4).

coherent, so we rate it a 5 (of 5).

How well does the factual information in the summary accurately match the article?

The summary is completely wrong, made up, or exactly contradicts what is written in the article. 1 2 3 4 5 The summary has no incorrect statements or misleading implications.

Explanation for rating: It could be inferred that Mr. Crabb is "desperate", but it is not explicitly stated in the article. So we rate this summary as 4 (of 5).

How well does the summary cover the important information in the article?

The summary contains no information relevant to the article. 1 2 3 4 5 The summary covers all of the important information required to understand the event in the article.

Explanation for rating: This summary has a fair coverage of the article, but it misses the mention of the underlying reason for the rail line electrification issue. So we rate this summary as 4 (of 5).

How good is the summary overall at representing the article?

It is terrible. 1 2 3 4 5 It is an excellent representation of the article.

Explanation for rating: This summary is pretty good, but it can be **somewhat improved** by providing the underlying reason for the rail line electrification issue. So we rate this summary as 4 (of 5).

Example Summary #3

The electrification of the Valleys rail lines interrupted by the finance plan is the "number one issue" for Crabb.

Ratings for Example Summary #2

The summary should be rated as follows:

How coherent is the summary on its own?

It is impossible to understand. 1 2 3 4 5 The summary is perfectly clear.

Explanation for rating: This summary is easy to understand and read with clear language and no grammatical errors and therefore coherent, so we rate it a 5 (of 5).

How well does the factual information in the summary accurately match the article?

The summary is completely wrong, made up, or exactly contradicts what is written in the article. 1 2 3 4 5 The summary has no incorrect statements or misleading implications.

Explanation for rating: Information in this summary is accurately grounded in the article. So we rate this summary as 5 (of 5).

How well does the summary cover the important information in the article?

The summary contains no information relevant to the article. 1 2 3 4 5 The summary covers all of the important information required to understand the event in the article.

Explanation for rating: This summary contains all important information in the article. So we rate this summary as 5 (of 5).

How good is the summary overall at representing the article?

It is terrible. 1 2 3 4 5 It is an excellent representation of the article.

Explanation for rating: This summary is an excellent representation of the article. So we rate this summary as 5 (of 5).

Test

Answer the following question to start the task. If you are unsure of the answer, review the rubrics above. The task section will appear when you've completed the test.

Which axis measures whether the summary information is grounded in the article information? Enter your answer and click "Start task".

Task

Instructions:

1. Read the article and when you are finished reading, click "Yes".
2. Write a short title for the article then click "Submit title".

Figure 7: Screenshot of our annotation interface (3/4).

3. You will then rate six summaries of the article for their coverage, accuracy, coherence, and overall quality.

article

\$(article)

Have you finished reading the article?

Give a short title to the article to describe what it is about

You may now rate the summaries below.

Note: consult the rubric if you are unsure of a rating.

How *coherent* is the summary on its own?

view rubric for [Coherence](#)

\$(summary_3)	Coherence: <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/>
\$(summary_6)	Coherence: <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/>
\$(summary_1)	Coherence: <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/>
\$(summary_5)	Coherence: <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/>
\$(summary_2)	Coherence: <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/>
\$(summary_4)	Coherence: <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/> <input type="button" value="5"/>

Figure 8: Screenshot of our annotation interface (4/4).

Article: Downing Street backed a report by think tank Policy Exchange which said selling high value homes when they become vacant would raise £4.5bn a year. That would be enough to build 80,000 to 170,000 social homes, the report said. Labour said new homes were urgently needed but "driving out hard-working families on low wages from whole neighbourhoods" was not the answer. In its Ending Expensive Social Tenancies report, Policy Exchange argues the move could create the largest social house building programme since the 1970s - giving the economy a kickstart. Neil O'Brien, the think tank's director, told the BBC that social housing would still exist in very expensive areas under its proposal, but there would just be "less of it". "The truth is I don't believe anybody has the right to live in the most expensive parts of town. "People do have a right to get housed, just not in the very most expensive areas," he said. He also suggested that the overall number of people waiting for social housing, currently around 1.8 million, could be reduced by about 500,000 if the scheme was implemented. The prime minister's official spokesman said: "This is something that councils can choose to do already. "Councils should be looking for ways to use their social housing stock as efficiently as they can. The waiting list for social housing has increased a lot over passing years. "They need to think about how they can use that social housing stock efficiently. "If they can sell high-value housing to invest in more social housing and find more homes for more people, then that is certainly something they should look at." But Labour said the coalition's "failed" policies were "making the housing crisis worse not better". Shadow housing minister Jack Dromey said: "Councils and housing associations should make effective use of their housing stock but the government should not force them to arbitrarily sell off social homes, breaking up mixed communities and driving out hard-working families on low wages from whole neighbourhoods." He said the government should use a bank bonus tax to fund 250,000 affordable homes and "put unemployed builders back to work" and boost the construction industry. 'Lucky family' Expensive social housing - which Policy Exchange defines as housing worth more than the average property in each region - accounts for 21.8% of the total social housing stock in the UK, it says. This equates to 816,000 properties - out of a total of 3.78 million - which the think tank says could raise up to £159bn if sold. It says London alone has more than £70bn of expensive social housing. About 3.5% of the total stock becomes vacant every year owing to people moving out or dying, the think tank said. This meant the government could sell a total of 28,500 properties each year, raising £5.5bn a year. The figure would stand at £4.5bn after paying off the debt held against the stock, the report said. Mr O'Brien argued that many hard-working people might want to live in a nicer area or in a bigger house but could not afford to. "Rather than having one lucky family with a very expensive house, you would have two families perhaps desperately waiting for social housing, now having a roof over their heads. "That seems fairer to me," he added. The think tank also said the move would be "extremely popular" with all sections of society, claiming that 73% of people, including social tenants, think people should not be given council houses worth more than the average property in a local authority. 'Dramatic erosion' Critics say such a move would push the least well-off out of expensive streets, and into new ghettos. The National Housing Federation, which represents housing associations, says many towns would be "cleansed" of "hardworking people who can't afford to pay high prices". Labour MP Karen Buck, who represents Westminster North, is concerned that lower income families, particularly in London, will be forced out of more affluent areas creating segregated communities of rich and poor. Ms Buck also argued that the Labour government's £8bn social and affordable housing building programme was cut by 60% when the coalition came to power. Housing Minister Grant Shapps - who is in favour of a sell-off - said the government had introduced "radical reforms" to "get Britain building" and to reduce social housing waiting lists. They included investing £19.5bn public and private funding into an affordable housing programme "set to exceed expectations and deliver up to 170,000 homes". Councils could now offer fixed-term tenancies to new tenants to make sure "social housing goes to those in greatest need", he added.

BART: Prime Minister David Cameron has said councils should be allowed to sell off expensive social housing to fund more affordable homes.

LOSSTRUNC: The government has said it would be "appropriate" for councils to sell off social housing in very expensive areas.

LOSSTRUNC + MUFASSA: Councils should sell high-value social housing to help build more homes, the prime minister's office says.

LOSSTRUNC + MUFASSA + WORDTRUNC: Councils should be allowed to sell council houses worth more than the average property to fund new homes, the government says.

Figure 9: Output examples on XSum.

Article: Following Raheem Sterling's interview on Wednesday, in which he said he was not ready to sign a new contract at Liverpool, blogger David Tyrer of Live4Liverpool gives the view from Merseyside. While I hate to use social media as a gauge of opinions, Raheem Sterling's interview didn't go down well at all. It was ill-timed and, regardless of what Sterling and his agent hoped, he didn't come across very well. Some of his answers only fuelled the fire really. I'm hoping that that wasn't the whole point, as we've seen these sorts of situations engineered by agents before. The interview has almost certainly changed the way the fans feel about him. There will be a lot of fans that are of the opinion: 'let him go'. Obviously, with the caveat that we get our money's worth! Raheem Sterling returns to Liverpool training after the international break and shakes hands with manager Brendan Rodgers. Sterling risked angering Liverpool fans after he said in an interview he was not ready to sign a contract. Sterling trains ahead of the weekend's game with Arsenal. It's always disappointing when a young player gets his head turned, but there's a sense of ungratefulness about the whole situation, considering how the club has nurtured him and paid him well throughout. Personally, I think he has the potential to be worth so much more than the £100,000-a-week contract he's turned down. But it's only that: potential. At present, he's arguably in the top five best young players in the world but, obviously at his age, he's also prone to bouts of inconsistency and prolonged poor form. He hasn't been great recently and was awful against Man Utd. Sterling has been linked with a move to Arsenal - the team he is preparing to face at the weekend. Raheem Sterling played for England in the 4-0 Euro 2016 qualifier against Lithuania. And while I'd be willing to see the club give him £100k a week - possibly £120k a week - the club shouldn't do everything it can to keep him. Definitely not. Liverpool fans have a popular mantra: no player is bigger than the club. Admittedly, we stretch the rules for truly great players (Suarez, a recent example) but Sterling is nowhere near. If he wants out, I'm sure the club will handle it the way they have before. Frustration over Sterling's situation has been building for a while, and many fans are now of the belief that if he wants to go he's welcome to. Personally, I don't like players holding the club to ransom. He has as much chance of winning trophies here as he does anywhere (other than money-rich clubs such as Chelsea or Man City). Sterling did not have his best game in a Liverpool shirt during the defeat at Manchester United. Liverpool fans have a mantra that no player is bigger than the club. If Sterling was to move to Arsenal then it would not go down well with Liverpool fans. But if he does end up going, he's worth a lot more than many established players. His ability is so raw but he has almost limitless potential. At the moment, we've seen it in fits and starts but even so, he's easily worth £25-30 million. If I was FSG, I'd hold any interested club to ransom, though - double it. But a move to Arsenal wouldn't go down well. Perhaps better than if it was Chelsea or Manchester United, but honestly there's little Arsenal can offer that we can't. Sure, they're more financially stable but what - in terms of trophies - do Arsenal have to show for the last 12 years? Sterling speaks to the BBC and reveals he is not ready to sign a new contract. I think we're somewhat ahead on that front. Much like Alexis Sanchez, any move to Arsenal will be more about location than anything else. People have drawn comparisons with when Suarez wanted to leave for Arsenal, but the situation is different. Suarez says he wanted Champions League football and we weren't offering that at that time. Not only that, but Suarez was an established top-class player at the time, one of the best in the world. Sterling is purely potential and, given all the club have done for him (cliched, as that sounds), it makes little sense. If he's going to be nurtured into the world-class talent he can become, it won't happen at Arsenal. It may not necessarily happen at Anfield either, but a manager like Brendan Rodgers will give him the best shot of improving to that level over the next two to three years. You can read more from Live4Liverpool [HERE](#) and follow the Twitter account [HERE](#).

BART: Raheem Sterling said he was not ready to sign a new contract at Liverpool. The England winger has been linked with a move to Arsenal. Liverpool fans have a mantra that no player is bigger than the club. But a move to Arsenal would not go down well with the Anfield faithful.

LOSSTRUNC: Raheem Sterling said he was not ready to sign a new contract at Liverpool. The England winger has been linked with a move to Arsenal. Liverpool fans have a mantra that no player is bigger than the club. Sterling has the potential to be worth more than the £100,000-a-week contract.

LOSSTRUNC + MUFASSA: Raheem Sterling has said he is not ready to sign a new contract at Liverpool. The England winger has been linked with a move to Arsenal. Liverpool fans have a mantra that no player is bigger than the club. But a move to Arsenal would not go down well with the fans.

LOSSTRUNC + MUFASSA + WORDTRUNC: Raheem Sterling said he was not ready to sign a new contract at Liverpool. Sterling has been linked with a move to Arsenal - the team he is preparing to face at the weekend. Liverpool fans have a mantra that no player is bigger than the club. If Sterling was to move to Arsenal then it would not go down well with Liverpool fans.

Figure 10: Output examples on CNN/DailyMail.

Mining Effective Features Using Quantum Entropy for Humor Recognition

Yang Liu and Yuexian Hou

College of Intelligence and Computing, Tianjin University, Tianjin, China
{luyou, yxhou}@tju.edu.cn

Abstract

Humor recognition has been studied with several different methods in the past years. However, existing studies on humor recognition do not understand the mechanisms that generate humor. In this paper, inspired by the incongruity theory, any joke can be divided into two components (the setup and the punchline). Both components have multiple possible semantics, and there is an incongruous relationship between them. We use density matrices to represent the semantic uncertainty of the setup and the punchline, respectively, and design Quantum Entropy Uncertainty (QE-Uncertainty) and Quantum Entropy Incongruity (QE-Incongruity) with the help of quantum entropy as features for humor recognition. The experimental results on the SemEval2021 Task 7 dataset show that the proposed features are more effective than the baselines for recognizing humorous and non-humorous texts.

1 Introduction

Humor is one of the most distinctive features of human behavior and a sign of mental maturity (Pasquali, 1990). The study of humor has received extensive attention in the fields of linguistics, philosophy, psychology, and sociology (Mihalcea et al., 2010). Computational humor is of particular interest, with the potential to transform computers into creative and motivational tools (Nijholt et al., 2003).

This paper restricts research to humor recognition in computational humor, which aims to recognize whether a piece of text is humorous. As shown in Figure 1, a joke usually includes two components: the setup and the punchline. The reader generates an expectation of the following text (the punchline) based on the content of the setup, and if the following text violates the reader’s expectation, humor is generated, and vice versa.

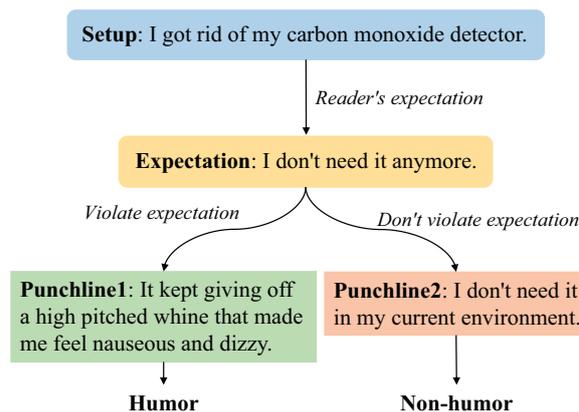


Figure 1: A humor and non-humor example containing the setup and the punchline.

In fact, the incongruity theory of humor can explain the above process of producing humor. The incongruity theory states that humor is generated because a thing (the setup) has multiple underlying concepts, and there is an incongruity between the concept involved in the situation and the real object it represents (the punchline).

Features based on semantic similarity (Mihalcea et al., 2010; Yang et al., 2015) and word association (Liu et al., 2018; Cattle and Ma, 2018) have achieved certain results, but they lack consideration of humorous mechanisms. Xie et al. (2021) calculated the uncertainty and the surprisal values of the joke with the help of the GPT-2. But they did not model the semantic incongruity between the setup and the punchline. While the above approaches are somewhat effective, the incongruity theory requires us to model semantic uncertainty and the incongruity between the setup and the punchline. We take inspiration from quantum theory and use density matrices to represent the uncertainty of text semantics. Specifically, the setup and the punchline are represented as density matrices, respectively. Then, take the quantum entropy of the setup as **Quantum Entropy Uncertainty (QE-Uncertainty)** and the conditional quantum entropy

between the setup and the punchline as **Quantum Entropy Incongruity (QE-Incongruity)**. Experiments conducted on a manually-labeled dataset demonstrate that these two features are better than existing baselines in distinguishing between humorous and non-humorous texts, confirming the necessity of correlating semantic uncertainty with quantum theory.

2 Background

2.1 The Incongruity Theory

The most widely accepted theory for explaining humor is the incongruity theory. The theory suggests that laughter is caused by an incongruity between the understanding of the text and its actual meaning (Mulder and Nijholt, 2002). Immanuel Kant describes humor as “*the sudden transformation of a strained expectation into nothing* (Hickey-Moody and Laurie, 2017).” Schopenhauer (1966) also believed that perceived incongruity exists between a concept and the real object it represents. The incongruity theory has also been developed in the field of linguistics. The Semantic Script-based Theory of Humor (SSTH) proposed by Raskin (1979) is a scripted expression of the incongruity theory. SSTH is our bridge to mathematically model the incongruity theory. SSTH requires humorous texts to meet the following conditions: (1) The text is compatible, fully or in part, with two different (semantic) scripts. (2) The two scripts with which the text is compatible are opposite.

2.2 Density Matrix

The mathematical form of quantum mechanics represents the probability space as a vector space (i.e., the Hilbert space \mathbb{H}^n) (Von Neumann, 2018). Researchers often use Dirac’s notation to represent unit vectors in this space. For example, a unit vector \vec{u} and its transpose \vec{u}^T are represented as $|u\rangle$ and $\langle u|$, respectively. The inner product of two unit vectors $|u\rangle$ and $|v\rangle$ is written as $\langle u|v\rangle$. The projector onto the direction $|u\rangle$ is its own outer product $|u\rangle\langle u|$. The rank of each projector is one and each projector represents a quantum fundamental event, often called a *dyad*. The density matrix (Nielsen and Chuang, 2010) is a generalization of the classical probability distribution. A density matrix ρ can be defined as a mixture of dyads:

$$\rho = \sum_{i=1}^n p_i |\psi_i\rangle\langle\psi_i| \quad (1)$$

where $|\psi_i\rangle$ represents a pure state with probability p_i . The density matrix ρ is symmetric, positive semi-definite, and its trace is one.

2.3 Quantum Entropy

Quantum entropy is a generalization of the quantum case of classical Shannon entropy (Shannon, 1948). If a quantum system is described by a density matrix ρ , its quantum entropy (Von Neumann, 2018) is defined as:

$$S(\rho) = -\text{tr}(\rho \ln \rho) \quad (2)$$

The conditional quantum entropy (Cerf and Adami, 1999) of the density matrix σ given the known density matrix ρ is defined as:

$$\begin{aligned} S(\sigma|\rho) &= S(\sigma\rho) - S(\rho) \\ &= -\text{tr}(\sigma\rho \ln(\sigma\rho)) + \text{tr}(\rho \ln \rho) \end{aligned} \quad (3)$$

unlike classical conditional entropy, conditional quantum entropy can be negative.

3 Methodology

The incongruity theory holds that the prerequisite for humor is that the text has multiple semantic aspects. The reader does not understand one meaning but expects one while the punchline provides another, leading to incongruity. According to the incongruity theory, we should design features to represent the multiple semantic overlaps of the setup, as well as the incongruity of the semantics of the setup and the punchline.

Normalize each word $w_i \in V$ as follows:

$$|w_i\rangle = \frac{\vec{w}_i}{\|\vec{w}_i\|} \quad (4)$$

where $\|\cdot\|$ represents the L_2 -norm. The representation of each word can be viewed as a superposition in Hilbert space.

A sentence of length l is represented by an n -by- n density matrix ρ :

$$\rho = \frac{1}{|l|} \sum_{i=1}^l |w_i\rangle\langle w_i| \quad (5)$$

where the diagonal values of ρ reflect the superposition semantics of sentences, and the non-diagonal values encode the correlation between semantics in a quantum way.

3.1 QE-Uncertainty

We take evidence of humor recognition from the setup, model the setup as a density matrix to represent its uncertainty semantics, and use the quantum entropy of the density matrix to represent the value of uncertainty. Formally, the QE-Uncertainty is calculated as follows:

$$U(\rho) = -\text{tr}(\rho \ln \rho) \quad (6)$$

where ρ represents the density matrix of the setup. The value of QE-Uncertainty reflects the amount of information contained in the text and the uncertainty of semantics. The larger the value, the more information the text contains, and the more likely the text is humorous.

3.2 QE-Incongruity

Another aspect of the incongruity theory is how different the semantics of the punchline is from expectations when the semantics of the setup are known (i.e., how much information we don't know about the punchline). In other words, how much information about the punchline is included in the setup? Specifically, the QE-Incongruity is defined as follows:

$$\begin{aligned} I(\sigma|\rho) &= U(\sigma\rho) - U(\rho) \\ &= -\text{tr}(\sigma\rho \ln(\sigma\rho)) + \text{tr}(\rho \ln \rho) \end{aligned} \quad (7)$$

where ρ and σ represent the density matrices of the setup and the punchline, respectively. The value of QE-Incongruity describes how unknown the semantics of the punchline is when the setup is known. We argue that when the setup contains less semantics in the punchline, there will be incongruity, and there will be humor.

4 Related Work

The existing text humor recognition methods are mainly divided into feature-based methods and deep learning-based methods. [Mihalcea and Strapparava \(2005\)](#) use automatic classification techniques to integrate humor-specific features (alliteration, antonymy, slang) and content-based features into a machine-learning framework for humor classification tasks. [Mihalcea et al. \(2010\)](#) divide the humor text into two components: the setup and the punchline. Humor recognition is performed by calculating the semantic correlation between the setup and the punchline based on the incongruity theory. [Morales and Zhai \(2017\)](#) use a generative language

model combined with background text resources to construct multiple features to identify whether a comment is a humorous text. [Liu et al. \(2018\)](#) combine discourse analysis and sentiment analysis to extract sentiment-related features to address humor recognition. [Xie et al. \(2021\)](#) developed uncertainty and superisal with the help of the prediction results of the pre-trained language model GPT-2. In recent years, with the development of deep learning, some deep learning-based methods have been proposed. [Chen and Lee \(2017\)](#) use convolutional neural networks to identify humor in the TED talks corpus. [Chen and Soo \(2018\)](#) used the highway network architecture to implement deep convolutional neural networks to predict humor on datasets of different types and different languages. [Weller and Seppi \(2019\)](#) used pre-trained BERT for the humor classification task. [Fan et al. \(2020\)](#) combine the Bi-GRU network with phonetic structure and ambiguity for humor recognition.

5 Experiments

5.1 Settings

We build a Support Vector Machine (SVM) classifier for humor classification. Experiments are performed on the SemEval 2021 Task 7¹ dataset modified by [Xie et al. \(2021\)](#). The dataset consists of a total of 3,052 labeled samples, half of which are humor and the other half are non-humor. The text of each sample in the dataset is split into two parts (the setup and the punchline). For each sample in the dataset, the lengths of the setup and the punchline are both below 20, and the percentage of alphabetical letters is greater than 75%, all of which start with alphabetical letters. We use Accuracy(Acc), Precision(P), Recall(R) and F1-Score(F1) as the evaluation metrics. P, R and F1 are macro-averaged. The experiments adopt 10-fold cross-validation, and the result is the average value of repeated experiments.

5.2 Baselines

Semantic similarity and semantic distance are the most commonly used text features, and we choose three such features as our baselines:

- **Path similarity** ([Rada et al., 1989](#)) is a similarity measure based on the shortest path, defined as follows:

$$\text{Sim}_{path} = \frac{1}{1 + D(c_1, c_2)} \quad (8)$$

¹<https://semeval.github.io/SemEval2021/>

where $D(c_1, c_2)$ represents the shortest path in WordNet between concepts c_1 and c_2 .

- **Disconnection** (Yang et al., 2015) is defined as the maximum distance between word pairs in the text.
- **Repetition** (Yang et al., 2015) is defined as the minimum distance between word pairs in the text.

In addition, we consider two GPT-2 based features proposed by Xie et al. (2021) as baselines. They feed the text into GPT-2 model to predict the next token. While predicting the tokens of y , GPT-2 produces a probability distribution v_i over the vocabulary.

- **Uncertainty** is obtained by calculating the average entropy of the probability distribution v_i on the vocabulary, defined as:

$$U(x, y) = -\frac{1}{|y|} \sum_{i=1}^n \sum_{w \in V} v_i^w \log v_i^w \quad (9)$$

where n represents the length of y and V is the vocabulary.

- **Surprisal** describes the degree of surprise when the language model generates the punchline, which is defined as follows:

$$\begin{aligned} S(x, y) &= -\frac{1}{|y|} \log p(y|x) \\ &= -\frac{1}{|y|} \sum_{i=1}^n \log v_i^{y_i} \end{aligned} \quad (10)$$

5.3 Predict Using Individual Features

Table 1 shows the results of individual feature prediction. Compared with the baselines, our proposed features QE-Uncertainty and QE-Incongruity achieve higher scores on all four metrics, with QE-Incongruity achieving the best results. In particular, compared with Uncertainty based on classical Shannon entropy, QE-Uncertainty under our quantum framework is greatly improved. This shows the necessity of quantum generalization for semantic uncertainty problems.

5.4 Boost a Content-Based Classifier

To demonstrate the effectiveness of our proposed features combined with content-based classifiers. We use the 50-dimensional GloVe (Pennington

Table 1: Experimental results of individual features. The results for features with an asterisk are reported by Xie et al. (2021).

Features	P	R	F1	Acc
Random	0.5000	0.5000	0.5000	0.5000
Sim _{path}	0.5123	0.5070	0.4555	0.5062
Disconnection	0.6475	0.5503	0.4610	0.5501
Repetition	0.5592	0.5577	0.5538	0.5567
Uncertainty*	0.5840	0.5738	0.5593	0.5741
Surprisal*	0.5617	0.5565	0.5455	0.5570
QE-Uncertainty	0.6589	0.6318	0.6146	0.6314
QE-Incongruity	0.6690	0.6450	0.6319	0.6451

et al., 2014) embedding as the baseline. We encode the setup and the punchline as the average of their respective word embeddings, resulting in two vectors with dimensions 50. Concatenate these two vectors with our features to form a vector with dimension 101. Finally, put it into an SVM classifier for humor classification. The results are shown in Table 2, our features achieve higher improvements on content-based classifiers compared to baselines.

Table 2: Experimental results of concatenating a content-based classifier. The results for features with an asterisk are reported by Xie et al. (2021).

Features	P	R	F1	Acc
GloVe	0.8233	0.8232	0.8229	0.8234
GloVe+Sim _{path}	0.8246	0.8246	0.8233	0.8237
GloVe+Discon.	0.8262	0.8264	0.8258	0.8263
GloVe+Repeti.	0.8239	0.8241	0.8237	0.8240
GloVe+U*	0.8355	0.8359	0.8353	0.8359
GloVe+S*	0.8331	0.8326	0.8321	0.8326
GloVe+QE-U	0.8361	0.8363	0.8355	0.8359
GloVe+QE-I	0.8363	0.8365	0.8356	0.8360

5.5 Feature Visualization

Figure 2 shows the distribution histograms of the values of QE-Uncertainty and QE-Incongruity for the joke and non-joke samples. From the figure, it can be found that jokes have higher QE-Uncertainty and QE-Incongruity values than non-jokes, which is consistent with what we stated in Section 3.

6 Conclusion

In this paper, we model semantic uncertainty with a quantum framework. Inspired by the incongruity theory, we design two features, QE-Uncertainty and QE-Incongruity. We conduct experiments on the humor dataset, and the experimental results

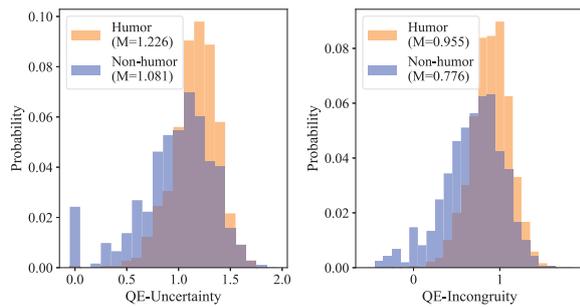


Figure 2: Histograms of our proposed features. The x-axis is the value of the feature, and the y-axis is the proportion of the feature in the total number of samples. **M** is the Median of the current feature.

demonstrate the effectiveness of our proposed features. This suggests that the density matrix is an excellent framework for describing uncertainty and that the quantum entropy of the density matrix is a better feature to distinguish jokes from non-jokes than previously proposed features. We believe that the quantum framework can also be used for semantic uncertainty modeling for other tasks in the future.

Limitations

In this paper, the density matrix representation of text is constructed in an averagely weighted manner, without considering the influence of weights on words. In addition, the density matrix as a text representation does not consider the position information of words. Furthermore, quantum generalization on the problem of multimodal humor recognition is also an interesting topic compared to unimodal humor recognition.

References

Andrew Cattle and Xiaojuan Ma. 2018. [Recognizing humour using word associations and humour anchor extraction](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1849–1858.

N. J. Cerf and C. Adami. 1999. [Quantum extension of conditional probability](#). *Phys. Rev. A*, 60:893–897.

Lei Chen and Chong Min Lee. 2017. [Convolutional neural network for humor recognition](#). *arXiv preprint arXiv:1702.02584v1*.

Peng-Yu Chen and Von-Wun Soo. 2018. [Humor recognition using deep learning](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 113–117.

Xiaochao Fan, Hongfei Lin, Liang Yang, Yufeng Diao, Chen Shen, Yonghe Chu, and Tongxuan Zhang. 2020. [Phonetics and ambiguity comprehension gated attention network for humor recognition](#). *Complexity*, 2020:1–9.

Anna Hickey-Moody and Timothy Laurie. 2017. [Masculinity and ridicule](#). In *Gender: laughter*, pages 215–228. Macmillan Reference USA.

Lizhen Liu, Donghai Zhang, and Wei Song. 2018. [Modeling sentiment association in discourse for humor recognition](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 586–591.

Rada Mihalcea and Carlo Strapparava. 2005. [Making computers laugh: Investigations in automatic humor recognition](#). In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 531–538.

Rada Mihalcea, Carlo Strapparava, and Stephen Pulman. 2010. [Computational models for incongruity detection in humour](#). In *Computational Linguistics and Intelligent Text Processing*, pages 364–374.

Alex Morales and Chengxiang Zhai. 2017. [Identifying humor in reviews using background text sources](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 492–501.

Mauk Mulder and Antinus Nijholt. 2002. [Humour research: State of the art](#). *CTIT Technical Report Series*, (02-34):1–24.

Michael A Nielsen and Isaac L Chuang. 2010. [Quantum Computation and Quantum Information](#). Cambridge University Press.

Anton Nijholt, Oliviero Stock, Alan Dix, and John Morkes. 2003. [Humor modeling in the interface](#). In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, page 1050–1051.

EA Pasquali. 1990. [Learning to laugh: humor as therapy](#). *Journal of Psychosocial Nursing and Mental Health Services*, 28(3):31–35.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Bletner. 1989. [Development and application of a metric on semantic nets](#). *IEEE transactions on systems, man, and cybernetics*, 19(1):17–30.

Victor Raskin. 1979. [Semantic mechanisms of humor](#). In *Annual Meeting of the Berkeley Linguistics Society*, pages 325–335.

- A. Schopenhauer. 1966. *The World as Will and Representation*. Dover books on philosophy. Dover Publications.
- Claude Elwood Shannon. 1948. *A mathematical theory of communication*. *The Bell system technical journal*, 27(3):379–423.
- John Von Neumann. 2018. *Mathematical foundations of quantum mechanics: New edition*, volume 53. Princeton university press.
- Orion Weller and Kevin Seppi. 2019. *Humor detection: A transformer gets the last laugh*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3621–3625.
- Yubo Xie, Junze Li, and Pearl Pu. 2021. *Uncertainty and surprisal jointly deliver the punchline: Exploiting incongruity-based features for humor recognition*. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 33–39.
- Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. *Humor recognition and humor anchor extraction*. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376.

AdapterSoup: Weight Averaging to Improve Generalization of Pretrained Language Models

Alexandra Chronopoulou^{*†∇} Matthew E. Peters[‡] Alexander Fraser[∇] Jesse Dodge^{*‡}

[∇]Center for Information and Language Processing, LMU Munich, Germany

[∇]Munich Center for Machine Learning, Germany

[‡]Allen Institute for Artificial Intelligence, Seattle, WA

Abstract

Pretrained language models (PLMs) are trained on massive corpora, but often need to specialize to specific domains. A parameter-efficient adaptation method suggests training an adapter for each domain on the task of language modeling. This leads to good in-domain scores but can be impractical for domain- or resource-restricted settings. A solution is to use a related-domain adapter for the novel domain at test time. In this paper, we introduce *AdapterSoup*, an approach that performs weight-space averaging of adapters trained on *different* domains. Our approach is embarrassingly parallel: first, we train a set of domain-specific adapters; then, for each novel domain, we determine which adapters should be averaged at test time. We present extensive experiments showing that AdapterSoup consistently improves performance to new domains without extra training. We also explore weight averaging of adapters trained on the *same* domain with different hyper-parameters, and show that it preserves the performance of a PLM on new domains while obtaining strong in-domain results. We explore various approaches for choosing which adapters to combine, such as text clustering and semantic similarity. We find that using clustering leads to the most competitive results on novel domains.

1 Introduction

Large LMs are pre-trained using massive amounts of data in a self-supervised way (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019) and obtain general-domain knowledge. In order to adapt them to a new domain, continuing training using in-domain data has been shown to be helpful (Han and Eisenstein, 2019; Lee et al., 2020; Gururangan et al., 2020). To avoid fine-tuning all parameters, efficient methods such as domain-specific mixtures-of-experts (Gururangan et al., 2022) and

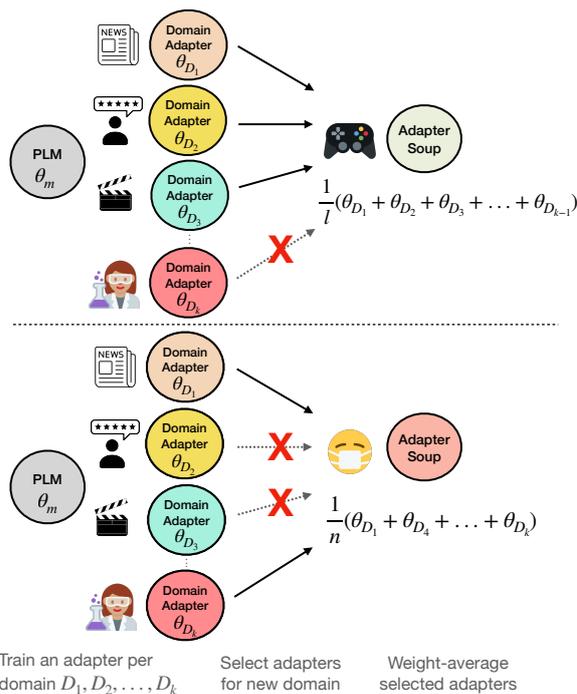


Figure 1: Illustration of AdapterSoup. Starting from the same random seed, an adapter is trained for each domain (*domain adapter*) on top of a PLM. AdapterSoup averages the weights of the adapters that are **most related** to the new domain to improve out-of-domain performance of a PLM at **test time**. The inference cost is independent of the number of adapters (l or n) used.

hierarchical domain adapters (Chronopoulou et al., 2022) have been proposed. Additional in-domain gains can be obtained using weight-space averaging (Wortsman et al., 2022a; Matena and Raffel, 2021). Motivated by this, we propose using weight-space averaging at test time to improve performance on *novel domains without extra training*.

Our approach, AdapterSoup, ensembles adapters in the *weight space* to improve performance on novel domains at test time without parameter updates. To this end, we train adapters on top of a PLM, each in a different domain. We compare several methods for selecting which adapters to

^{*}Correspondence to achron@cis.lmu.de or jessed@allenai.org

[†]Work done during an internship at Allen AI

use for each novel domain at test time and propose weight-space averaging models selected using text clustering. We find that AdapterSoup improves performance on novel domains. We also explore weight averaging adapters trained in the *same* domain, each with a different hyper-parameter configuration, and find that combining models trained with a low learning rate provides competitive in-domain scores, while averaging models trained with high learning rates performs similarly to a general-purpose PLM on novel domains.

Our contributions are the following: **1)** We propose combining domain-adapted PLMs at inference time using adapters. Our approach leads to consistent gains in novel domains. We compare several methods for choosing the models of the AdapterSoup, concluding that text clustering provides the best performance across all domains. **2)** We perform weight-space averaging of PLMs adapted to the same domain with varied hyper-parameters using adapters. We find that we can obtain competitive in-domain scores but also preserve the generalization ability of a PLM.

2 Proposed Approach

Problem Statement. Assuming we have a PLM adapted to k domains D_1, \dots, D_k , we want a model that performs well in a novel domain D_{k+1} without training more parameters. We use the provenance of a piece of text (that is, the *website* from which the text was scraped) as a proxy for *textual domain*. This follows Chronopoulou et al. (2022); Gururangan et al. (2022).

If we assume that we have a PLM fine-tuned on a single domain D_i with different hyper-parameters, we want to combine the fine-tuned models in order to both obtain good in-domain performance and preserve the generalization ability of the PLM to novel domains.

2.1 Cross-Domain AdapterSoup

An illustration of the cross-domain AdapterSoup is provided in Figure 1. Let $f(x, \theta_m)$ be a PLM with input data x and parameters $\theta_m \in \mathbb{R}^d$. We add adapters with a parameter initialization θ_α . While in this work we parameterize θ_α with adapters, our method is general and could be extended to other efficient fine-tuning methods. We only fine-tune the adapters, without updating the parameters θ_m of the PLM, for language modeling using cross-entropy loss. Let us assume that $\theta =$

$\text{FineTune}(\theta_m, \theta_\alpha, \phi, D)$ denote the parameters obtained by fine-tuning a PLM with adapters in a domain D , using hyper-parameters ϕ .

Let ϕ be a fixed hyper-parameter configuration. We vary only the *textual domain*. We first train k different adapters, one for each of the training domains. Then, we combine their weights:

$$\text{AdapterSoup}(x) = f(x, \frac{1}{l} \sum_{i=1}^l \theta_i), \quad (1)$$

i.e., we use the average of the parameters of l fine-tuned models, selected by one of the methods described in §2.3 ($l \leq k$). If $l = k$, this model is a *uniform soup* (Wortsman et al., 2022a).

2.2 Single-Domain AdapterSoup

In this setup, we want to learn an LM that performs well in a single training domain D , while *maintaining* the performance of the initial PLM θ_m in novel domains. To this end, we train adapters on the same domain, varying the *hyper-parameter configuration*. Each of the n models is optimized with different hyper-parameters ϕ_i , with $i \in 1, \dots, n$. We then compute the weight-space average following Equation 1, with $l = 3$. This is similar to logit ensembling, but only adds to the PLM the inference cost of a single adapter, while the added inference cost of logit ensembling scales linearly with the number of adapters.

2.3 Model Selection for AdapterSoup

In this section we describe two methods for selecting the combination of models to create our AdapterSoup (by weight-space averaging) which will be evaluated on a *novel* domain D_{k+1} . Following standard practice (Gururangan et al., 2022; Li et al., 2022) we use a small amount of validation data from the novel domain D_{k+1} for each of the below approaches. We note that we keep the test data unseen and only use it to perform our test-set evaluations.

Sentence similarity. We use pretrained sentence-BERT (Reimers and Gurevych, 2019), an approach that modifies BERT (Devlin et al., 2019) using siamese and triplet networks (Schroff et al., 2015) to obtain sentence embeddings. We compute the embeddings for 100 sentences from each of the training domains D_1, \dots, D_k , plus the novel domain D_{k+1} . Then we compute the average cosine similarity between each of D_1, \dots, D_k and D_{k+1} . We add up to 5 adapters to the AdapterSoup in order of highest cosine similarity (only considering models

Method	10 Evaluation Domains										Avg.
	reuters	techcrunch	fastco	nme	fool	inquisitr	mashable	tripadv	neci	yelp	
GPT-2 (zero-shot)	21.5	27.7	27.9	28.2	23.8	22.4	27.1	40.4	20.7	36.2	27.6
Single Adapter Chosen Using:											
- Sentence similarity	18.9	22.0	22.0	23.1	22.9	18.4	25.3	37.0	18.2	49.4	24.4
- Clustering	17.6	22.4	24.0	21.1	23.3	18.7	23.6	37.7	18.2	44.3	24.0
AdapterSoup (Weight-space average):											
- Uniform	18.2	23.1	22.9	22.2	22.4	18.4	23.1	37.0	19.1	36.2	24.3
- Sentence similarity	17.6	22.0	21.3	20.7	22.2	18.4	22.4	36.2	17.6	35.2	23.4
- Clustering	17.3	21.8	21.3	21.1	22.2	17.8	22.2	34.8	17.6	34.8	23.1
Oracle											
- Best adapter per domain	17.6	22.0	21.5	21.1	22.9	17.8	22.2	37.0	18.2	35.9	23.6
- Clustering + 2 best	17.3	21.8	21.3	20.7	22.0	17.6	22.0	33.4	17.6	33.4	22.7
Hierarchy adapter	16.4	20.1	20.1	20.1	22.2	16.4	22.2	33.1	18.2	34.5	22.3

Table 1: Perplexity (\downarrow) scores on 10 evaluation domains. All single adapter and AdapterSoup experiments have the same inference cost; bold indicates the best perplexity for each novel domain and best average. We find that AdapterSoup using clustering as a selection method on average leads to the best out-of-domain performance.

trained on domains with cosine similarity greater than 0.15 to D_{k+1}). We experimented with several values to define the threshold (3, 5, 10, 15). We did not observe significant improvement when scaling up from 5 to 10 adapters and for that reason, we used up to 5 adapters in each AdapterSoup.

Domain clustering. Our domain clustering approach follows Aharoni and Goldberg (2020). We encode 100 sequences from each of the training domains using a PLM and fit a Gaussian Mixture Model (GMM) with 21 components (equal to the number of training domains), which gives us a domain clustering. We then use 100 sequences from our held-out set (not used for test-set evaluation) and find which clusters they are closest to. We add up to 5 adapters to the AdapterSoup in order of which clusters the most held-out domain text is mapped to. If at least 10% of the sequences of the D_{k+1} is mapped to the cluster of D_i , we add the model trained on D_i to the AdapterSoup.

In-domain. To select the models that perform best in-domain, we exhaustively combine all models trained on a single textual domain (in this case, text found in the website *booking.com*), using combinations of size 3. Each model has been trained with a different hyper-parameter configuration. Specifically, we vary the learning rate and data order. We compare them to the best-performing single model per domain and to a uniform soup.

3 Experimental Setup

Datasets. We assume that text found in a specific website (e.g., *tripadvisor*) can be used as a proxy of a textual domain. We use 21 training domains and 10 evaluation domains (text from 21 and 10 websites accordingly) from the released version

(Dodge et al., 2021) of C4 (Raffel et al., 2020) (details in the Appendix). We hypothesize that the variety of training domains plays an important role in this setting. We randomly sampled domains that belong to the 100 high-resource domains of C4, but further work could consider using M2D2 (Reid et al., 2022), a multi-domain language modeling dataset released concurrently to this work.

Model Architecture. We use GPT-2 (Radford et al., 2019); specifically, we use a publicly available pretrained checkpoint of the small version, i.e., gpt2 from the HuggingFace library (Wolf et al., 2020). We add an adapter to each Transformer (Vaswani et al., 2017) layer after the feed-forward layer. We train only the adapters for language modeling in each training domain. The adapters follow the Bapna and Firat (2019) architecture and have bottleneck size 64. For the cross-domain AdapterSoup, we train all models with an initial learning rate $1e-4$. For the single-domain AdapterSoup, we use different learning rates and data seeds shown in the Appendix.

4 Results

Results are presented in Table 1. For each experiment, we evaluate both perplexity and efficiency.

4.1 Cross-domain

As a first baseline, we use *GPT-2 (zero-shot)*, without further training or additional parameters. This has worse perplexity than all other approaches but is most efficient at inference.

Single Adapters. We then evaluate *Sentence similarity* and *Clustering* in the scenario where only a single adapter is chosen using each approach (this can be thought of as a soup of size 1). This is an

evaluation of how well these two approaches measure similarity between the novel domain D_{k+1} and the training domains; this baseline shows the performance of a single model which can be directly compared to AdapterSoups. Both approaches are significantly better than GPT-2 (zero-shot), and *Clustering* outperforms *Sentence similarity*, suggesting it is better at identifying related domains.

AdapterSoup. We evaluate three types of *AdapterSoup* which differ only in how the models added to the soup are selected. All three are equally as efficient at inference as using a single adapter. *Uniform* is a uniform soup (weight-averaging all trained models). This performs worse than all approaches except GPT-2 (zero-shot); we hypothesize that it performs worse due to negative interference between adapters trained on unrelated domains. Using *Sentence similarity* as described in §2.3 leads to marginally better scores than the single-best adapter per domain, indicating even relatively naively-created soups can outperform the best (oracle) single model. On 8/10 novel domains, the sentence similarity AdapterSoup outperforms the single adapter chosen by Sentence similarity, indicating that the soup leads to better performance. Next, using *Clustering* as described in §2.3 leads to perplexity improvements in 8/10 novel domains compared to sentence similarity, indicating that the method for selecting models for the soup has a large impact. On 9/10 novel domains, the Clustering AdapterSoup outperforms the single adapter chosen by clustering, indicating that our approach leads to better performance.

Oracle Experiments and Larger Models. *Best adapter per domain* shows the performance of the single-best adapter on each novel domain. This is the upper bound for a single adapter, and we see that our *Single Adapter Chosen Using Clustering* matches these scores on 3/10 novel domains, and is close on the rest, suggesting the clustering approach is reasonably good. *Clustering + 2 best* shows the performance of adding the two (oracle) best models to our AdapterSoup made by clustering; our clustering approach is close to these scores, but there is room for future work on better choosing models for the AdapterSoup. *Hierarchy adapter* is taken from Chronopoulou et al. (2022), and is less efficient in terms of both data and parameters.

Selecting Models for the Soup. We qualitatively compare the selection methods for choosing adapters to include in the AdapterSoup for 3 novel

Novel Domain i	Sentence Sim.	Clustering
tripadvisor	booking	booking
	insiderpages	insiderpages lonelyplanet
ncbi	journals	journals
	frontiersin	frontiersin
	springer	springer
reuters	csmonitor	dailymail
	wired	express
	entrepreneur	

Table 2: Domains of models selected for the AdapterSoup using either sentence similarity or clustering. The clustering method seems to more accurately match each novel domain to training domains that are similar to it.

	booking ID	frontiers OOD	journals OOD	yelp OOD
GPT-2 (zero-shot)	29.7	22.2	24.5	36.2
Best single adapter	10.2	27.7	30.3	49.4
AdapterSoup:				
- lr $7e-3$	27.7	23.3	24.8	37.7
- lr $4e-3$	24.5	23.8	25.5	39.6
- lr $1e-3$	11.5	24.0	26.3	42.5
- lr $5e-4$	10.0	26.3	29.1	47.5
- lr $1e-4$	10.4	27.4	30.0	48.9
Best AdapterSoup:				
- in-domain	10.0	26.3	29.1	47.5
- out-of-domain	26.8	22.9	24.5	37.3
Logit ensemble	9.2	25.0	27.7	47.7

Table 3: Perplexity scores in- and out-of-domain (respectively ID and OOD) of models trained on *booking.com*. Low learning rates lead to good in-domain scores, while high learning rates improve the out-of-domain performance.

domains in Table 2. In the case of *tripadvisor*, 2/3 domains *Sentence similarity* and *Clustering* select are identical, while for *ncbi* (science domain) both methods select the same domains. When selecting domains similar to *reuters* (news), clustering seems to find a good match, choosing news domains. However, *Sentence similarity* selects domains that are not quite as related to the novel domain. *Reuters* contains heterogeneous data, so the average cosine similarity on the sentence level is not a suitable metric to find related domains.

4.2 Single-domain

In this section we evaluate how models trained on the same domain can be combined into an AdapterSoup. We train a set of models using adapters on *booking.com* by varying the data order and the learning rate (see Appendix A.3, note our experiments kept the initialization of each adapter fixed), then evaluate all combinations of adapters of size 3, and evaluate the performance of the AdapterSoup both in-domain (*booking.com*) and on 3 held-out domains. We explore this controlled setting to bet-

ter understand the setup described in [Wortsman et al. \(2022a\)](#), who also noted that the learning rate is important; their experiments indicated that smaller learning rates led to better model soups.

Our experiments in Table 3 show a more nuanced result: AdapterSoups made from adapters trained with small learning rates ($5e-4$) performed best in-domain (confirming the result from [Wortsman et al., 2022b](#)), but AdapterSoups made from adapters trained with larger learning rates ($7e-3$, $4e-3$, and $7e-4$) generalize better to novel domains. The number of updates for each adapter is the same, and they all have the same initialization, so we hypothesize that AdapterSoups made from small learning rates act similarly to averaging across steps in gradient descent, leading to a model that is closer to a local optimum. As for why larger learning rates leads to better generalization to novel domains, we hypothesize that each model in the AdapterSoup travels a farther distance from the initialization, leading to learning somewhat more diverse representations. We leave further exploration to future work.

5 Related Work

As training large models from scratch has a severe computational and environmental cost ([Strubell et al., 2019](#); [Dodge et al., 2022](#)), efficient methods such as mixtures-of-experts (MoE) ([Shazeer et al., 2017](#); [Fedus et al., 2021](#); [Artetxe et al., 2022](#)), adapters ([Rebuffi et al., 2017](#); [Houlsby et al., 2019](#); [Pfeiffer et al., 2020](#)), and LoRA layers ([Hu et al., 2022](#)) have recently been proposed. Both adapters and MoEs have shown to work well for domain adaptation ([Cooper Stickland et al., 2021](#); [Gururangan et al., 2022](#); [Chronopoulou et al., 2022](#)). The hierarchy adapter ([Chronopoulou et al., 2022](#)) outperforms our approach but is significantly more expensive. It adds a training cost of $4Ld_{\text{model}}dT$ (following [Kaplan et al., 2020](#)) over the cost of running GPT-2 for a model with L layers, dimension d_{model} , adapter bottleneck size d , average tree depth T ($T = 8$ in the hierarchy adapter paper), while AdapterSoup needs $4Ld_{\text{model}}d$ flops. As a result, training the hierarchy adapter is a factor of T slower than our approach. At inference time, the hierarchy adapter activates 2 paths in the tree and invokes a cost $4Ld_{\text{model}}dT \times 2$, i.e., inference is a factor of $2T$ slower than our approach.

Averaging *weights* of models independently fine-tuned on the same task ([Wortsman et al., 2022a](#)) has shown to improve in-domain performance. [Matena](#)

and [Raffel \(2021\)](#) weight-average fine-tuned PLM models using Fisher merging to avoid intermediate task training and then perform downstream fine-tuning. [Wang et al. \(2022\)](#) fine-tune MoEs using adapters on a downstream task and average their weights at test time. Our paper, however, focuses on improving test-time scores of a model on *novel* domains.

[Wang et al. \(2021\)](#) improve performance in an unseen (target) language by ensembling the source language adapter and language adapters similar to the target language. This approach uses weighted ensembling of the *outputs* of adapters, whereas we ensemble the *weights* of the adapters. AdapterSoup has the inference cost of a single adapter, while [Wang et al. \(2021\)](#) require inference time that scales linearly to the number of adapters.

Contemporaneous work ([Li et al., 2022](#)) also explores performance in novel domains using weight averaging, but uses MoEs instead of adapters.

6 Conclusion

A PLM can be adapted to new domains using adapters. However, this requires training a new set of adapters for each domain. We propose a method based on weight-space averaging of adapters selected using text clustering. Our approach improves performance on novel domains without updating parameters or increasing the inference cost. Future work could explore more sophisticated selection methods to try to match the performance of the oracle experiments.

Limitations

The conclusions we draw in this work about how our approach compares to other approaches (e.g., our baselines) are only supported by evidence on the task of language modeling, with textual domains taken from the C4 dataset. We expect such results to hold more generally, but do not have experimental evidence to support any other scenarios. As with all work on language modeling, the models we have trained could be used to generate language, but we do not have evaluations of generated text (e.g., on fluency, factuality, or other common metrics used to evaluate generated language). Our paper focuses on using adapters; while we expect similar approaches to work for other types of models, we only have evidence to support AdapterSoup working for adapters.

Acknowledgements

We thank Ayyoob Imani for feedback on the final version of the paper and Jonas Pfeiffer for helpful discussions. We also thank Mitchell Wortsman and Ludwig Schmidt for preliminary comments on the first version of this idea.

References

- Roei Aharoni and Yoav Goldberg. 2020. [Unsupervised domain clusters in pretrained language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.
- Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giridharan Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O’Horo, Jeffrey Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Veselin Stoyanov. 2022. [Efficient large scale language modeling with mixtures of experts](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11699–11732, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ankur Bapna and Orhan Firat. 2019. [Simple, scalable adaptation for neural machine translation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.
- Alexandra Chronopoulou, Matthew Peters, and Jesse Dodge. 2022. [Efficient hierarchical domain adaptation for pretrained language models](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1336–1351, Seattle, United States. Association for Computational Linguistics.
- Asa Cooper Stickland, Alexandre Berard, and Vassilina Nikoulina. 2021. [Multilingual domain adaptation for NMT: Decoupling language and domain information with adapters](#). In *Proceedings of the Sixth Conference on Machine Translation*, pages 578–598, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jesse Dodge, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A. Smith, Nicole DeCario, and Will Buchanan. 2022. [Measuring the carbon intensity of ai in cloud instances](#). In *2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’22*, page 1877–1894, New York, NY, USA. Association for Computing Machinery.
- Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. [Documenting large webtext corpora: A case study on the colossal clean crawled corpus](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- William Fedus, Barret Zoph, and Noam Shazeer. 2021. [Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity](#).
- Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022. [DEMIX layers: Disentangling domains for modular language modeling](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States. Association for Computational Linguistics.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don’t stop pretraining: Adapt language models to domains and tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Xiaochuang Han and Jacob Eisenstein. 2019. [Unsupervised domain adaptation of contextualized embeddings for sequence labeling](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China. Association for Computational Linguistics.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for NLP](#). In *Proceedings of the International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 2790–2799.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large](#)

- language models. In *International Conference on Learning Representations*.
- Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv*, abs/2001.08361.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.
- Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.
- Michael Matena and Colin Raffel. 2021. Merging models with fisher-weighted averaging.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.
- Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*.
- Machel Reid, Victor Zhong, Suchin Gururangan, and Luke Zettlemoyer. 2022. M2D2: A massively multi-domain language modeling dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 964–975, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.
- Xinyi Wang, Yulia Tsvetkov, Sebastian Ruder, and Graham Neubig. 2021. Efficient test time adapter ensembling for low-resource language varieties. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 730–737, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yaqing Wang, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022. Adamix: Mixture-of-adapter for parameter-efficient tuning of large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022a. [Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time](#). In *Proceedings of the 39th International Conference on Machine Learning*.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2022b. [Robust fine-tuning of zero-shot models](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7959–7971.

A Appendix

A.1 Training details

We build our code using PyTorch (Paszke et al., 2019) and the HuggingFace library (Wolf et al., 2020). Each model is trained on a single NVIDIA A100 GPU with 40GB of RAM, batch size 64 and gradient accumulation over 5 steps. We train each model for 20 epochs, without using early stopping. We compute semantic similarity using sentence-transformers¹ and a publicly available pretrained model.²

We noticed from preliminary experiments that the choice of random seed is important when averaging weights of domain adapters. We empirically found that averaging domain adapters initialized from different random seeds led to poor performance of AdapterSoup. We suggest initializing the adapters from the same random seed in order to effectively combine adapters trained on various domains.

A.2 Dataset sizes

We use textual corpora from 31 of the 100 most high-resource internet domains of C4. The sizes of the training domains are shown in Table 4, while the sizes of the evaluation domains are shown in Table 5.

A.3 Single-domain AdapterSoup

We present the hyper-parameters we tried in Table 6. In this setup, we computed in- and out-of-domain scores for 455 different combinations (there are 15 models and computed all AdapterSoups of size 3). The trend we observed is that higher learning rates improved results out-of-domain, while lower learning rates provided the best in-domain scores.

A.4 Cross-domain AdapterSoup

We present in Table 7 the evaluation scores of each of the single adapter models. Each adapter has been trained in a different training domain (column 1), and evaluated in 10 novel domains.

¹<https://github.com/UKPLab/sentence-transformers>

²huggingface.co/sentence-transformers/all-mpnet-base-v2

Ind	Training Domain	Train (Eval.) Tokens
1	dailymail.co.uk	25M (3M)
2	wired.com	18M (2M)
3	express.co.uk	16M (2M)
4	npr.org	25M (3M)
5	librarything.com	3M (500K)
6	instructables.com	25M (3M)
7	entrepreneur.com	16M (2M)
8	link.springer.com	28M (4M)
9	insiderpages.com	8M (1M)
10	ign.com	10M (1M)
11	eventbrite.com	11M (1M)
12	forums.macrumors.com	22M (3M)
13	androidheadlines.com	14M (2M)
14	glassdoor.com	4M (500K)
15	pcworld.com	14M (2M)
16	csmonitor.com	23M (3M)
17	lonelyplanet.com	6M (1M)
18	booking.com	30M (4M)
19	journals.plos.org	53M (6M)
20	frontiersin.org	38M (6M)
21	medium	22M (3M)

Table 4: Sizes of training corpora. We fine-tune GPT-2 using adapters on each of these domains. We perform weight-averaging of these 21 domain-adapted LMs.

Ind	Novel Domain	Train (Eval.) Tokens
1	reuters.com	17M (2M)
2	techcrunch.com	13M (2M)
3	fastcompany.com	14M (2M)
4	nme.com	5M (1M)
5	fool.com	34M (4M)
6	inquisitr.com	13M (2M)
7	mashable.com	14M (2M)
8	tripadvisor.com	7M (1M)
9	ncbi.nlm.nih.gov	23M (3M)
10	yelp.com	68M (6M)

Table 5: Sizes of held-out corpora.

Hyper-parameter	Value
learning rates	$7e-3, 4e-3$ $1e-3, 5e-4, 1e-4$
random seed	1, 2, 3

Table 6: Hyper-parameters for single-domain AdapterSoups. We exhaustively compute the AdapterSoup for every combination of 3 models in this set.

Training Domain	Evaluation Domains										
	reuters	techcrunch	fastco	nme	fool	inquisitr	mashable	tripadv.	ncbi	yelp	Avg
dailymail	17.6	23.6	24.0	21.1	23.3	18.4	23.6	39.6	20.5	44.3	25.6
wired	18.0	22.0	21.5	22.0	22.9	18.2	22.2	40.0	19.9	41.3	24.8
express	19.5	25.8	26.0	22.6	25.8	20.1	26.3	42.9	23.3	48.9	28.1
npr	20.1	25.5	25.0	27.7	23.3	20.5	23.6	42.1	21.1	42.9	27.2
librarything	19.5	24.5	24.0	24.8	23.6	19.7	24.8	38.9	21.1	39.3	26.0
instructables	20.5	25.5	25.5	25.5	24.5	20.5	25.5	40.0	21.1	41.7	27.0
entrepreneur	18.2	22.4	22.0	22.6	22.9	18.4	23.1	40.9	21.1	43.4	25.5
springer	19.7	25.0	24.5	24.5	25.3	19.9	26.8	42.9	18.4	43.8	27.1
insiderpages	23.1	28.8	29.1	32.1	25.5	23.1	27.9	37.7	23.3	35.9	28.7
ign	18.9	23.8	23.6	22.6	23.3	18.7	23.6	40.9	21.1	39.6	25.6
eventbrite	19.1	24.3	23.8	23.1	24.3	19.3	25.0	39.6	20.9	41.7	26.1
macrumors	20.3	26.0	26.3	26.3	24.5	20.9	25.5	41.3	22.4	43.4	27.7
androidheadlines	20.7	24.8	25.8	26.0	24.5	20.1	25.3	44.7	22.6	42.9	27.8
glassdoor	20.7	26.0	25.8	27.7	24.8	21.1	26.8	42.5	22.0	42.5	28.0
pcworld	18.7	22.6	22.9	23.6	23.1	18.7	23.1	42.1	21.5	42.9	25.0
csmonitor	18.9	24.0	23.8	24.0	23.6	18.9	23.8	41.3	21.5	43.4	26.3
lonelyplanet	20.7	26.0	25.8	25.0	25.3	20.7	26.6	40.4	22.6	42.9	27.6
booking	27.4	33.4	33.1	35.9	31.5	27.4	35.5	37.0	30.6	49.4	34.1
journals	21.3	26.8	26.0	27.4	26.0	21.5	28.2	46.1	18.2	46.5	28.8
frontiersin	21.1	26.8	25.5	27.7	26.0	27.7	26.0	45.6	19.3	46.5	29.2
medium	17.8	22.2	21.8	21.3	25.0	17.8	25.3	39.3	19.9	43.4	25.4

Table 7: We show the performance of each trained adapter (for the cross-domain setting) on the 10 evaluation domains. Each model has been trained for language modeling with an initial learning rate $1e - 4$ for 20 epochs.

Towards End-to-End Open Conversational Machine Reading

Sizhe Zhou^{1,3}, Siru Ouyang^{2,3}, Zhuosheng Zhang^{2,3} Hai Zhao^{2,3,*}

¹ UM-SJTU Joint Institute, Shanghai Jiao Tong University

² Department of Computer Science and Engineering, Shanghai Jiao Tong University

³ Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University
{sizhezhou, oysr0926, zhangzs}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

In open-retrieval conversational machine reading (OR-CMR) task, machines are required to do multi-turn question answering given dialogue history and a textual knowledge base. Existing works generally utilize two independent modules to approach this problem's two successive sub-tasks: first with a hard-label decision making and second with a question generation aided by various entailment reasoning methods. Such usual cascaded modeling is vulnerable to error propagation and prevents the two sub-tasks from being consistently optimized. In this work, we instead model OR-CMR as a unified text-to-text task in a fully end-to-end style. Experiments on the ShARC and OR-ShARC dataset show the effectiveness of our proposed end-to-end framework on both sub-tasks by a large margin, achieving new state-of-the-art results. Further ablation studies support that our framework can generalize to different backbone models.

1 Introduction

In a multi-turn dialogue comprehension scenario, machines are expected to answer high-level questions through interactions with human beings until enough information is gathered to derive a satisfying answer (Zhu et al., 2018; Zhang et al., 2018; Zaib et al., 2020; Huang et al., 2020; Fan et al., 2020; Gu et al., 2021). As a specific and challenging dialogue comprehension task, conversational machine reading (CMR) (Saeidi et al., 2018) requires machines to understand the given user's initial setting and dialogue history before the machine itself is able to give a final answer or inquire for more clarifications according to rule texts (see Figure 1).

In terms of acquisition of rule texts which are the main reference for tackling the CMR, there

* Corresponding author. This paper was partially supported by Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

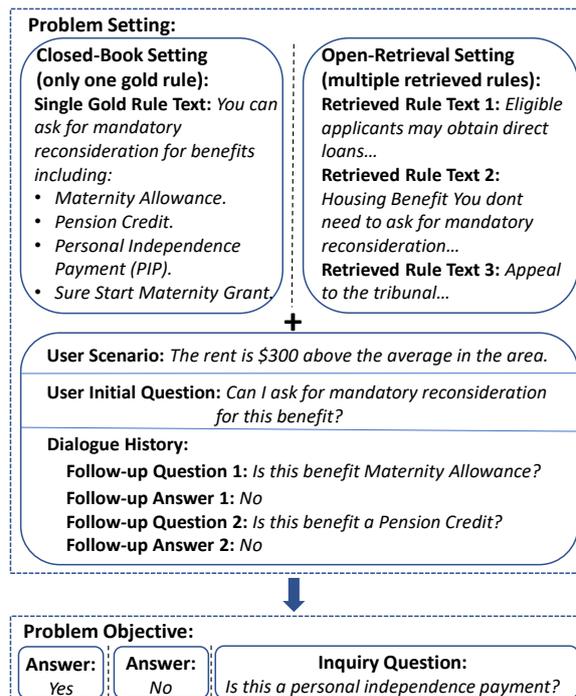


Figure 1: CMR and OR-CMR Task Overview

is closed-book setting where the rule texts are all given and there is correspondingly open-retrieval setting where the rule texts need to be retrieved from a knowledge base (Gao et al., 2021) (see Figure 1). In terms of problem objectives, current approaches in general divide the targets into two categories, one as decision making and one as question generation (Zhong and Zettlemoyer, 2019; Lawrence et al., 2019; Verma et al., 2020; Gao et al., 2020b,c). For decision making sub-task, the machine is required to give decisions to directly answer the user question which concludes the dialogue or generate clarifying questions which continues the dialogue. For question generation sub-task, the machine is required to generate the clarifying questions that are essential to the later final decision making. Following this line of approaching the CMR task, a variety of works

have been proposed mainly based on modeling the matching of elementary conditions (Henaff et al., 2017; Zhong and Zettlemoyer, 2019; Lawrence et al., 2019; Verma et al., 2020; Gao et al., 2020b,c; Ouyang et al., 2021; Zhang et al., 2021) in either a sequential encoding or graph-based manner.

However, by tackling the CMR task with two divided sub-tasks, the corresponding division of the optimization on decision making sub-task and the optimization on the question generation sub-task may result in problems including error propagation, thus hindering further performance advance. Ouyang et al. (2021) has shown that transferring some knowledge between the training of two sub-tasks is beneficial for better performance. However, reducing the gap between two sub-tasks to achieve an end-to-end optimization CMR task still needs further and more comprehensive attempts.

In this work, we propose a completely Unified end-to-end framework for Conversational Machine Reading tasks (UNICMR¹) to tackle the division of optimization challenge by formulating the CMR/OR-CMR task into a single text-to-text task. Our contributions are summarized as follows:

(i) We completely unify two sub-tasks of OR-CMR into a single task in terms of optimization, achieving a fully end-to-end optimization paradigm.

(ii) Experimental results on the OR-ShARC dataset and ShARC dev set show the effectiveness of our method, especially on the question generation sub-task with a relatively small amount of parameters. Furthermore, our method achieves the new state-of-the-art results on all sub-tasks.

(iii) By further ablation studies, we have shown that our proposed framework largely advances the decision making performance, and reduces error propagation thus boosting the question generation performance. We have also shown that our proposed framework can generalize to different backbone models. Qualitative analysis including case study has further verified the effectiveness of our framework.

2 Related Work

2.1 Conversational Machine Reading

The mainstream of research on the conversation-based reading comprehension task focuses on either the decision making (Choi et al., 2018;

Reddy et al., 2019; Sun et al., 2019; Tao et al., 2019; Cui et al., 2020; Yang et al., 2020) or the follow-up utterance generation (Wu et al., 2019; Bi et al., 2019; Ren et al., 2019; Gao et al., 2020a). However, the decision making centered approaches leave out cultivating the machine’s capability to reduce the information gap by clarifying interactions. While the question generation centered approaches neglect exploring the machine’s capability to concentrate on target-oriented information and make vital decisions. In contrast, our work focuses on a more challenging conversation-based reading comprehension task called conversational machine reading (CMR) task (Saeidi et al., 2018; Gao et al., 2021), which requires machines to make decisions and generate clarifying questions in a dialogue given rule texts and user scenarios.

2.2 Open-Retrieval CMR

Most of the current studies on CMR concentrate on the closed-book setting of CMR where the essential reference for the final decision, a piece of rule text corresponding to each dialogue, is given (Zhong and Zettlemoyer, 2019; Verma et al., 2020; Gao et al., 2020b,c). One typical example benchmark is called ShARC (Saeidi et al., 2018). However, in a more realistic and also more challenging setting, the machine is required to retrieve rule texts based on different scenarios. Similar to the open domain question answering setting where the supporting texts are retrieved from external documents to answer factoid questions (Moldovan et al., 2000; Voorhees and Tice, 2000), open-retrieval conversational machine reading (OR-CMR) task is established by requiring the machine to retrieve useful information from a given knowledge base composed of rule texts. In contrast to most of the previous works on CMR, we focus on OR-CMR in pursuit of a more realistic and more challenging setting.

2.3 Joint Optimization of CMR

Existing studies generally approach conversational machine reading task by separating it into two sub-tasks (Zhong and Zettlemoyer, 2019; Verma et al., 2020; Gao et al., 2020b,c), decision making and question generation. Therefore, existing approaches generally focus on different methods to extract the fulfillment of rule-related conditions and conduct explicit entailment reasoning on tracking the conditions in the dialogues. This includes applying attention mechanisms on the sequentially

¹Our source codes are available at <https://github.com/KevinSRR/UniCMR>.

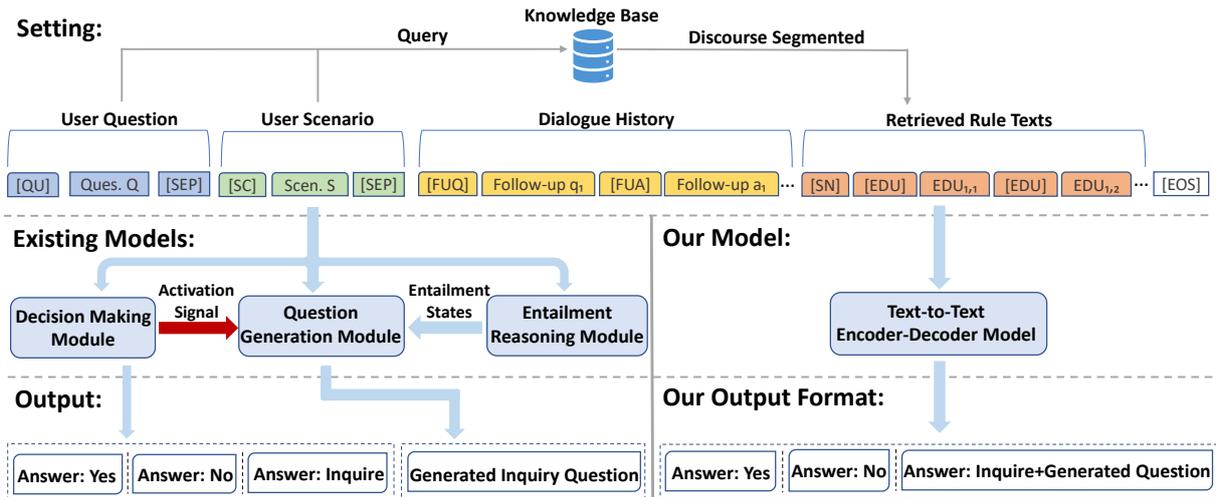


Figure 2: The overall framework for our proposed model (bottom right part) compared with the existing ones (bottom left part). Note that the ways of preprocessing the problem setting input vary from model to model, but they are generally similar. And the setting part only shows our preprocessing overview. Also note that [QU], [SEP], [SC], [SEP], [FUQ], [FUA], [SN], [EDU] are added special tokens while [EOS] is the end-of-sequence token for encoder-decoder model.

encoded user setups and the dialogue (Zhong and Zettlemoyer, 2019; Lawrence et al., 2019; Verma et al., 2020; Gao et al., 2020b,c) and extract discourse structures for better fulfillment matching (Ouyang et al., 2021).

However, one of the major challenges emerges as the division of the optimization of decision making sub-task and the optimization of the question generation sub-task. Zhang et al. (2021) have taken the initial attempt to mitigate the division between two sub-tasks by considering the encoded hidden states from decision making in question generation module. However, it still lacks synergy of optimization and relies on separate feature extractions including the entailment reasoning. In contrast, our work approaches the conversational machine reading task by unifying the two sub-tasks into one, enabling an end-to-end joint model optimization on both the decision making target and question generation target.

3 Problem Formulation

As shown by Figure 1, in traditional CMR task, the machine will be given: user scenario S , user initial question Q , a gold rule text R , and dialogue history $D := \{(q_1, a_1), (q_2, a_2), \dots, (q_n, a_n)\}$ which consists of n follow-up question-answer pairs. The machine is required to do the two sub-tasks:

- **Decision Making.** The machine makes a decision to either answer the user initial question

with *Yes* or *No*, or give *Inquire*² which activates the second sub-task to generate the inquiry question for more clarification.

- **Question Generation.** The machine generates an inquiry question aimed at essential clarifications to answer the user’s initial question.

Beyond CMR, open-retrieval conversational machine reading (OR-CMR) (Gao et al., 2021) further mimics the more challenging second scenario, which is the focus of this work. As shown by Figure 1, the difference between the CMR and OR-CMR lies in the rule text part R . In CMR, the machine is provided with a gold rule text in a closed-book style. While in OR-CMR, the machine needs to retrieve rule texts from a knowledge base in an open-retrieval style alternatively. The machine is given a knowledge base B containing rule texts. Therefore, under the OR-CMR setting, the machine needs to first retrieve m rule texts R_1, R_2, \dots, R_m to complete the input for the same downstream decision making and question generation sub-tasks.

4 Framework

Our model is composed of two main modules: a retriever and a text-to-text encoder-decoder model.

²For the completeness of the conversational machine reading task, there is an additional decision making answer *Irrelevant* which states that the user question is unanswerable. This is the case for CMR task. However, in our work, we mainly follow the setting of OR-CMR and assume that no such answer will be encountered.

The retriever is applied to retrieve rule texts R_1, R_2, \dots, R_m from a given knowledge base B . The text-to-text encoder-decoder model will take in the preprocessed textual input and generate the textual answer directly as a whole. Subsequent extraction methods will be applied for decision making and question generation sub-tasks to obtain the predictions for each sub-task respectively.

4.1 Retriever

To obtain the rule texts, the user scenario S and user initial question Q are concatenated as the input query to the retriever. Our retriever employs the MUDERN TF-IDF-based method (Gao et al., 2021), which takes account of bigram features and scores the similarity between rule texts and queries in the form of bag-of-words vectors weighted by the TF-IDF model. Top-scored m rule texts R_1, R_2, \dots, R_m will then be chosen for the following text-to-text encoder-decoder model.

4.2 Text-to-Text Encoder-Decoder

One of the major challenges of the CMR or OR-CMR task is the division of sub-task optimizations. Motivated by T5 (Raffel et al., 2020) which formulate several traditional NLP tasks into a unified text-to-text generation task, we unify the two sub-tasks by formulating the input and output to our encoder-decoder model as follows.

4.2.1 Input Formulation

Discourse Segmentation. We employ the discourse segmentation approach (Shi and Huang, 2019) to parse the retrieved rule texts into explicit conditions for the model. After discourse segmentation, each retrieved R_i is parsed into N_i elementary discourse units (EDUs) $EDU_{i,1}, EDU_{i,2}, \dots, EDU_{i,N_i}$. Formulation of the final input I is shown by the setting part in Figure 2.

4.2.2 Output Formulation

The output of the text-to-text encoder-decoder will be a sequence of textual tokens $O := \{o_1, o_2, \dots, o_k\}$ where the length k is determined by the model itself but within the maximum generation length hyperparameter. To extract the prediction of the decision making sub-task and the question generation sub-task respectively, we assume the first output token o_1 is model’s prediction, and the following tokens $\{o_2, \dots, o_k\}$ are the generated

follow-up question, which is only meaningful when o_1 represents the *Inquire* decision.

4.2.3 Training Objective

In training stage, the labels $Y := \{y_1, y_2, \dots, y_k\}$ are formulated as: {Yes Token, [EOS]}, {No Token, [EOS]}, and {*Inquire* Token, Follow-up Question Tokens, [EOS]}.³ The training objective is defined as:

$$\mathcal{L} = - \sum_{j=1}^k \log P(y_j | y_{<j}, I; \theta), \quad (1)$$

where I is the input to our encoder-decoder model and θ is all the parameters of our model.

5 Experiments

5.1 Experiment Setups

Datasets. Our training and evaluation is based on the OR-ShARC dataset (Gao et al., 2021). Original dataset ShARC (Saeidi et al., 2018) contains 948 dialogues trees which is then flattened into 32,436 examples with entries composed of rule documents, user setups, dialogue history, evidence, and decision. Derived from ShARC, OR-ShARC modifies the *initial question* to be self-contained and to be independent of gold rule texts. Then the gold rule texts are removed to form the knowledge base B of 651 rules. The train and dev set of ShARC are further split into train, dev, and test set, with sizes 17,936, 1,105, and 2,373, respectively.

The dev and test set each satisfies that around 50% of examples ask questions based on the rule texts used in training (seen) and the remaining asks questions based on the unseen rule texts in training. This feature of the datasets aims to mimic more realistic scenario where user may asks questions on information that the machine has encountered or has never encountered (Gao et al., 2021).

Evaluation Metrics. For decision making sub-task, the evaluation is Micro- and Macro- Accuracy of the decisions. For question generation sub-task, we adopt the $F1_{BLEU}$ (Gao et al., 2021) which calculates the F1 score with precision of BLEU (Papineni et al., 2002) when the predicted decision is *Inquire* and recall of BLEU when the ground truth decision is *Inquire*.

³To make sure Yes Token, No Token and *Inquire* Token have the same length after tokenization, we set the valid tokens of “1”, “2” and “3” to serve as Yes Token, No Token and *Inquire* Token respectively without loss of generality.

Model	Dev Set				Test Set			
	Decision Making		Question Generation		Decision Making		Question Generation	
	Micro	Macro	F1 _{BLEU1}	F1 _{BLEU4}	Micro	Macro	F1 _{BLEU1}	F1 _{BLEU4}
<i>w/ DPR++</i>								
MUDERN	79.7±1.2	80.1±1.0	50.2±0.7	42.6±0.5	75.6±0.4	75.8±0.3	48.6±1.3	40.7±1.1
OSCAR	80.5±0.5	80.9±0.6	51.3±0.8	43.1±0.8	76.5±0.5	76.4±0.4	49.1±1.1	41.9±1.8
<i>w/ TF-IDF</i>								
E ³	61.8±0.9	62.3±1.0	29.0±1.2	18.1±1.0	61.4±2.2	61.7±1.9	31.7±0.8	22.2±1.1
EMT	65.6±1.6	66.5±1.5	36.8±1.1	32.9±1.1	64.3±0.5	64.8±0.4	38.5±0.5	30.6±0.4
DISCERN	66.0±1.6	66.7±1.8	36.3±1.9	28.4±2.1	66.7±1.1	67.1±1.2	36.7±1.4	28.6±1.2
DP-RoBERTa	73.0±1.7	73.1±1.6	45.9±1.1	40.0±0.9	70.4±1.5	70.1±1.4	40.1±1.6	34.3±1.5
MUDERN	78.4±0.5	78.8±0.6	49.9±0.8	42.7±0.8	75.2±1.0	75.3±0.9	47.1±1.7	40.4±1.8
UNICMR _{base}	75.6±0.4	76.5±0.6	53.7±0.5	46.5±0.2	71.7±1.2	72.2±1.1	48.4±1.5	41.5±1.7
UNICMR _{large}	77.7±0.5	78.0±0.6	59.3±1.2 (†8.0)	52.8±0.9 (†9.7)	76.7±1.2 (†0.2)	76.7±1.1 (†0.3)	54.2±1.4 (†5.1)	47.9±1.6 (†6.0)

Table 1: Results on the validation and test set of OR-ShARC. Numerical values in the parentheses show how much our proposed model outperforms the current SOTA model. The first block presents the results of public models with the DPR++ retrieval method, and the second block reports the results of TF-IDF retrieval-based public models and our SOTA model. Our average results with a standard deviation on 3 random seeds are reported. The numbers in brackets (†) indicate the improved accuracy over the previous state-of-the-art model.

Model	Dev Set			
	Decision Making		Question Gen.	
	Micro	Macro	BLEU1	BLEU4
OSCAR	70.1	75.6	63.3	48.1
UNICMR	72.6	78.0	66.3	53.9

Table 2: Results on the validation set of ShARC (with large models). Note that the test set of ShARC is not public hence only the evaluation on dev set is conducted.

Implementation Details. Following the MUDERN model, we employ T5 as our text-to-text encoder-decoder model and initialize the model with the pretrained T5-base and T5-large weights⁴. For the main model either base or large, we set the max generation length as 30, number of beams in generation as 5, and use the first 8 top scored retrieved rule texts in preparing input. The training process utilizes AdamW (Loshchilov and Hutter, 2017) optimizer for 16 epochs with a learning rate of 3e-5. Max gradient norm of 1 is used to conduct gradient clipping. The batch size is 4 with a gradient accumulation step as 8. Random seeds 19, 27, and 95 are applied. Experiments are conducted in two RTX TITAN GPU’s with 24G memory⁵. In training stage, the model with best F1_{BLEU4} score on dev set is kept.

⁴<https://huggingface.co/t5-base>, and <https://huggingface.co/t5-large>, respectively.

⁵Average training run time for UNICMR_{large} is approximately 12 hours with one GPU. Average inference run time for UNICMR_{large} is approximately 10 minutes on dev set and 21 minutes on test set with one GPU.

5.2 Quantitative Results

The effectiveness of our proposed method is verified on both the OR-ShARC and the original ShARC datasets. In addition, we compare the number of parameters with related studies. Tables 1-3 present our main experimental results. We will discuss our findings in the following part.

5.3 Decision Making and Question Generation performance on OR-ShARC.

Referring to our results reported in Table 1, our large unified model has achieved new SOTA question generation performance in both dev and test sets by a large margin. In terms of decision making results, our large model lags behind in the dev set but prevails in the test set performance by maintaining a stable and consistent performance when transferring from dev set to test set.

5.4 Performance on ShARC.

As a reference, the performance of the UNICMR_{large} together with the current SOTA model OSCAR on the dev set of ShARC is reported on Table 2. Note that, in contrast with OR-ShARC (Gao et al., 2021), ShARC benchmark (Saeidi et al., 2018) is in the closed-book setting with the evaluation metric of the question generation sub-task as BLEU. Based on the results in Table 2, it can be seen that UNICMR_{large} maintains a new SOTA performance on dev set by a large margin for both the decision making and the question generation sub-tasks. This shows our unified method is effective for the model’s performance

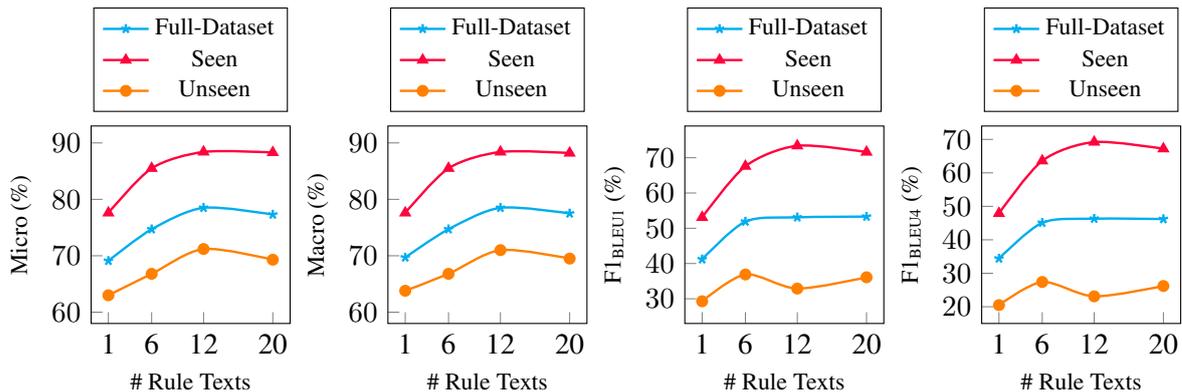


Figure 3: Evaluation performance of our model under different number of retrieved rule texts on test set.

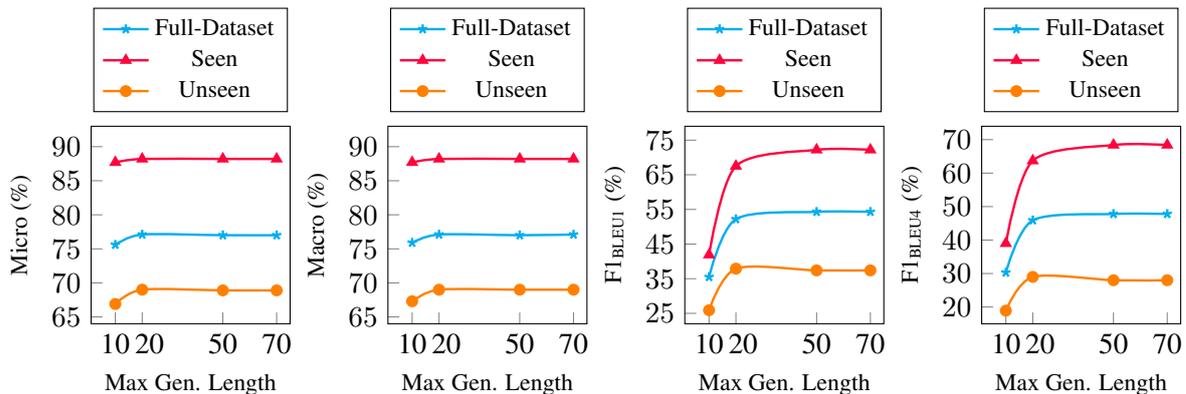


Figure 4: Evaluation performance of our model under different max generation length on test set.

beyond OR-ShARC.

	DISCERN	OSCAR	UNICMR (base/large)
#Param.	330M	1100M	220M/770M

Table 3: The comparison of approximate number of parameters of some current models.

5.5 Comparison of Model Parameter Numbers.

We have approximated total parameters of current high performance models. The information is shown in Table 3. By comparison of the parameter numbers used in current high performance models in Table 3, our UNICMR_{large} (based on T5-large) uses around 770M parameters which generally prevails the current SOTA model OSCAR using around 1100M parameters. Our UNICMR_{base} (based on T5-base) uses 220M parameters but prevails models like DISCERN which uses around 330M parameters. UNICMR_{base} also achieves a close performance to OSCAR in terms of question generation. The above observations verify that our method of unifying optimizing the two sub-tasks

is effective, which enables each sub-task to benefit from the optimization of the other task.

6 Analysis

6.1 Number of Retrieved Rule Texts

The model performance under different choices of the number of retrieved rule texts is shown in Table 7 in Appendix B whose visualization is shown by Figure 3. We see that generally, when the number of rule texts increases, there will be more information which improves our model while also introducing more noise which harms our model. In terms of decision making, our model is quite stable in seen test dataset when the number of rule texts varies. That means our model well captures the useful and trash conditions in rule texts and fulfillment states in dialogue history in the training stage. Besides, The unusual boost of question generation performance in the unseen test set might suggest that using more than the necessary number of rule texts possibly pushes the model to gain more power of generalization in the training stage.

Model	Dev Set				Test Set			
	Micro	Macro	F1 _{BLEU1}	F1 _{BLEU4}	Micro	Macro	F1 _{BLEU1}	F1 _{BLEU4}
UNICMR _{large}	77.7	78.0	59.3	52.8	76.7	76.7	54.2	47.9
Closed-Book	82.1	82.1	67.8	62.8	79.4	79.5	60.5	54.8
w/ DPR++	76.8	77.4	56.8	50.4	75.2	75.2	54.8	48.8
w/o Retriever	71.0	70.9	42.1	35.2	65.8	65.7	35.2	28.7

Table 4: Results of our UNICMR_{large} and UNICMR_{large} with different retriever module setting on the dev and test sets of OR-ShARC benchmark. For Closed-Book setting, the OR-ShARC is turned into a closed-book setting by given the rule texts. For w/ DPR++ setting, the TF-IDF retriever is replaced with DPR++ retriever. For w/o Retriever setting, the OR-ShARC is approached without rule texts.

6.2 Maximum Generation Length

The model performance under the different choices of the maximum generation length is shown in Table 8⁶ in Appendix B whose visualization is shown by Figure 4.

In terms of decision making and question generation, redundant max generation length will not affect the performance of the model but insufficient max generation length will limit the model performance. This means the model well learns the difference between different forms of answers and is able to generate answers of suitable length accordingly. This verifies the feasibility of our end-to-end framework design.

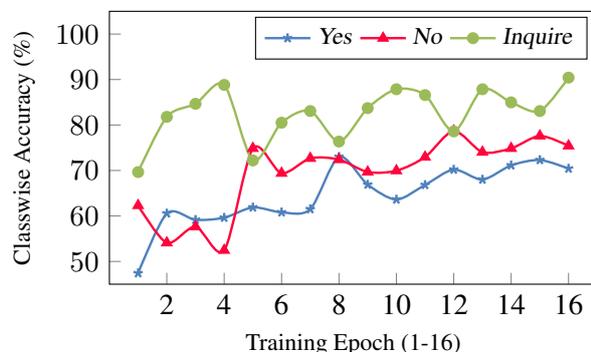


Figure 5: Classwise accuracy on dev set of each epoch.

6.3 Generation Quality Gain Across Training

The classwise accuracy evaluated in the training of the decision making sub-task is shown by Figure 5. By the initial gap between the accuracy for

⁶In Table 7, the hyperparameter m (number of retrieved rule texts) is varied to compare our model performance on the OR-ShARC test set, test set seen and test set unseen divisions respectively. In Table 8, the hyperparameter maximum generation length of the backbone encoder-decoder model is varied to compare our model performance on the same datasets. The corresponding performance of the above two experiments on dev set is shown by Table 9 and Table 10 in Appendix B for reference. Note in these experiments, all the hyperparameters remain the same unless explicitly stated.

“Inquire” and the accuracy for other decisions, our model tends to predict the decision as *Inquire* and generate question when not well fine-tuned. This is due to a gap between the length for the answer *Yes/No* and the length for the answer “*Inquire+Generated Question*”. And also the innate property of pre-trained T5 generation model before well fine-tuned at the beginning which is hence biased towards the longer answer. As the training continues, the accuracy for *Yes* and *No* gradually catches up with *Inquire* even though is slightly lower. This observation shows the existence of the bias of our backbone model and also the effectiveness of our training which large reduces such bias. This also suggests future improvements on more targeted training to eliminate the bias and lessening the discontinuity between the length of output for *Yes/No* and the length of output for “*Inquire+Generated Question*”.

6.4 Contribution of the Retriever Module

To quantify the contribution of the retriever module, we conducted an additional experiment where OR-ShARC is turned into a closed-book setting (see Closed-Book in Table 4). Also, we replaced the TF-IDF retriever with the DPR++ retriever introduced in UNICMR_{large} for reference (see w/ DPR++ in Table 4). Performance of UNICMR_{large} without retriever is also shown (see w/o Retriever in Table 4). The results verify that using the retrieval is beneficial, which reduces the gap between the challenging open-retrieval task and the closed-book task with gold rule texts.

6.5 Discussions of Performance Improvement

To further investigate the source of performance improvement of our method, more comprehensive experimental results are shown here following the deduced conclusions.

First, UNICMR’s unified training format advances the performance of training T5 separately

Model	Dev Set				Test Set			
	BLEU1	BLEU4	F1 _{BLEU1}	F1 _{BLEU4}	BLEU1	BLEU4	F1 _{BLEU1}	F1 _{BLEU4}
<i>w/ T5-large</i>								
UNICMR	67.5	59.1	59.3	52.8	55.8	48.3	54.2	47.9
QG-only whole-evaluation	53.3	47.5	49.5	43.1	45.2	40.1	47.0	39.7
QG-only partial-evaluation	71.1	61.0	47.9	40.8	69.4	59.5	45.8	38.9
<i>w/ BART-base</i>								
UNICMR	58.4	50.2	52.3	45.1	47.3	40.2	46.9	39.8
QG-only whole-evaluation	62.6	51.4	44.1	37.3	60.4	48.9	40.3	33.5
QG-only partial-evaluation	69.2	57.7	43.3	35.3	66.8	56.7	39.9	33.3

Table 5: Question generation performance of UNICMR compared with models trained only on question generation sub-task on OR-ShARC. For QG-only whole-evaluation setting, we use all samples by assigning empty generated question to samples with Yes/No decisions. For QG-only partial-evaluation setting, we use samples only with inquiry questions. The results are generally divided into two parts, one using T5-large as backbone model and one using BART-base as backbone model.

Model	Dev Set		Test Set	
	Micro	Macro	Micro	Macro
<i>w/ T5-large</i>				
UNICMR	77.7	78.0	76.7	76.7
DM-only	73.9	73.7	72.9	72.3
<i>w/ BART-base</i>				
UNICMR	74.8	75.7	71.5	71.8
DM-only	72.5	72.4	68.6	68.3

Table 6: Decision making performance of UNICMR compared with models trained only on decision making sub-task on OR-ShARC. For DM-only setting, we use all samples to train our model only on decision making sub-task. The results are generally divided into two parts, one using T5-large as backbone model and one using BART-base as backbone model.

on decision making. See the performance of T5-large trained for decision making separately (DM-only in Table 6) compared with the original UNICMR_{large} (UNICMR in Table 6) performance. The comparison indicates that UNICMR’s stronger form of unified training improves the model’s decision making ability.

Second, UNICMR’s unified training format advances the performance of training T5 separately on question generation in F1_{BLEU}. Ablation studies here include the T5-large trained with all examples (assign empty to examples with Yes and No decisions) for question generation only (QG-only whole-evaluation in Table 5), T5-large trained with examples with gold inquiry questions for questions generation only (QG-only partial-evaluation in Table 5), and T5-large-based UNICMR (UNICMR in Table 5). The results indicate that:

(i) In terms of F1_{BLEU}, UNICMR has dominantly higher performance than other separately trained models.

(ii) In terms of BLEU⁷, UNICMR is not the best, which shows its source of F1_{BLEU} dominance includes reduction of error propagation.

(iii) For T5-large backbone, UNICMR is higher in BLEU than QG-only partial-evaluation, which means UNICMR’s integration of decision making labels in training is effective.

6.6 Generalizability on Different Backbone Models

Replacing the T5-large backbone with BART-base, and repeating the same experiments (see the same settings but with BART-base as backbone models in Table 6 and Table 5), leads to same general conclusions. This shows the effectiveness of UNICMR’s unified format can well generalize to different end-to-end architectures.

6.7 Error Analysis and Case Study

To reveal more insights into UNICMR, we randomly collect test samples and conduct error analysis (see Figure 6) and case study (see Figure 7 in Appendix A). The ground truth answers are indicated in red, the TF-IDF scores are indicated in green, and the predictions of UNICMR_{large} are indicated in blue. The retrieved rule texts are in descending order in terms of TF-IDF scores.

Error Analysis. The observed test errors are summarized into four aspects: (1) *Noisy Retrieved Rule Texts* which is caused by the innate deficiencies of TF-IDF retriever with bigram

⁷Note that BLEU is measured on samples with *Inquire* as gold labels only while F1_{BLEU} is measured on all samples considering both the BLEU when prediction is *Inquire* and the BLEU when gold label is *Inquire*. For F1_{BLEU} calculation of all QG-only settings, decision making predictions of model trained only on decision making sub-task are used.

Error Type	Dialogue Setups	UniCMR Output
Noisy Retrieved Rule Texts	<p>Scen.: It was a donation of stuff I wasn't using that I gave to Gift Aid and they got me 25% more than anyplace else would.</p> <p>Ques.: Will I have to pay more tax than I've paid?</p> <p>His.: (empty)</p> <p>Gold Rule: Charity donations: tax relief. If the charity or CASC gets back more tax than you've paid, HMRC may ask you to pay more tax to cover the difference.</p> <p>Gold Answer: No</p>	<p>Retrieved Rules:</p> <p>(1) [78.43] Donations through Gift Aid: Charities and community amateur sports clubs (CASCs) can register with HM Revenue and Customs (HMRC) to be part of the Gift Aid scheme. When they're registered, they can claim back the tax you've already paid on your donation.</p> <p>(2) [39.25] Charity donations: tax relief. Donations to charity from individuals are tax free. You can get tax relief if you donate: * through Gift Aid * straight from your wages or pension, through Payroll Giving</p> <p>.....</p> <p>Prediction: Can a charity or community amateur sports club (CASC) register with HM Revenue and customs (HMRC)?</p>
Losing Track of Some Conditions	<p>Scen.: I married my husband Bob when he was 50 in the year 2014. Unfortunately he died of a heart attack in 2015 and never reached the age where he was eligible for a state pension.</p> <p>Ques.: Could I inherit part of my deceased partner's Additional State Pension?</p> <p>His.: Q: Did you partner reach state pension age before April 6 2016? A: No</p> <p>Gold Rule: (same as the first retrieved rule)</p> <p>Gold Answer: Would your partner have reached state pension age on or after 6 April 2016?</p>	<p>Retrieved Rules:</p> <p>(1) [149.46] Inheriting Additional State Pension. You might inherit part of your deceased partner's Additional State Pension if your marriage or civil partnership with them began before 6 April 2016 and one of the following applies: * your partner reached State Pension age before 6 April 2016 * they died before 6 April 2016 but would have reached State Pension age on or after that date</p> <p>(2) [99.16] Inheriting or increasing State Pension from a spouse or civil partner. You may inherit part of or all of your partner's extra State Pension or lump sum if: * they died while they were deferring their State Pension (before claiming) or they had started claiming it after deferring * they reached State Pension age before 6 April 2016 * you were married or in the civil partnership when they died</p> <p>.....</p> <p>Prediction: Did they die before April 6 2016?</p>
Different Condition Ordering	<p>Scen.: I require assistance in turning a local agricultural area into a hunting and gaming area.</p> <p>Ques.: Do I get the Additional State Pension automatically?</p> <p>His.: (empty)</p> <p>Gold Rule: (same as the first retrieved rule)</p> <p>Gold Answer: Have you contracted out of the state?</p>	<p>Retrieved Rules:</p> <p>(1) [49.66] Overview. You get the Additional State Pension automatically if you're eligible for it, unless you've contracted out of it.</p> <p>(2) [41.62] Inheriting Additional State Pension. You might inherit part of your deceased partner's Additional State Pension if your marriage or civil partnership with them began before 6 April 2016 and one of the following applies: * your partner reached State Pension age before 6 April 2016 * they died before 6 April 2016 but would have reached State Pension age on or after that date</p> <p>.....</p> <p>Prediction: Are you eligible for it?</p>
BLEU's Inability on Phrase Variants	<p>Scen.: (empty)</p> <p>Ques.: Can I get payment in lieu?</p> <p>His.: (empty)</p> <p>Gold Rule: (same as the first retrieved rule)</p> <p>Gold Answer: Are you leaving your job?</p>	<p>Retrieved Rules:</p> <p>(1) [21.83] Getting paid instead of taking holidays. The only time someone can get paid in place of taking statutory leave (known as 'payment in lieu') is when they leave their job. Employers must pay for untaken statutory leave (even if the worker is dismissed for gross misconduct).</p> <p>.....</p> <p>Prediction: Did you leave your job?</p>

Figure 6: Error analysis of UNICMR_{large} by comparison with ground truth answers.

features. (2) *Losing Track of Some Conditions* which shows in rare cases UNICMR_{large} might miss some condition fulfillment as UNICMR_{large} does not explicitly model condition fulfillment. (3) *Different Condition Ordering* which is caused by multiple unsatisfied conditions and the flexibility to inquire any of them. (4) *BLEU's Inability on Phrase Variants* which means predictions are penalized by BLEU even if they only differ in unimportant and semantically harmless words.

Case Study. Qualitative improvements of generated inquiries of UNICMR_{large} are summarized into two aspects: (1) *Exactness* which means the capability of capturing the self-contained yet elementary condition units that need to be clarified. (2) *Robustness to Noisy Retrieved Rules* which means the model can filter noisy retrieved rule texts to extract unsatisfied conditions. From the results in Figure 7, it can be seen that UNICMR_{large} generate more suitable inquiries in terms of *Exactness* and achieves excellent performance in terms of *Robustness to Noisy Retrieved Rules*. This suggests that our fully end-to-end framework enables the accurate focus on target conditions and the implicit feature engineering of UNICMR is powerful to filter noisy retrievals regardless of the retriever quality.

7 Conclusion

In this paper, we study open-retrieval setting of the conversation machine reading task and promote a novel framework to first unify the optimizations of the two sub-tasks to achieve optimization synergy. With a retriever module and a parameter-efficient text-to-text encoder-decoder module, we have achieved new SOTA results in both the CMR and the OR-CMR benchmarks. Further experiments shows that our unified training form with an end-to-end optimization method largely contributes to the advanced performance in decision making and reduces the error propagation to boost question generation performance. It's also shown that our framework well generalize to other backbone models. Further qualitative analysis also verifies our framework's effectiveness.

Limitations

Under the challenging open-retrieval setting, a retrieval is required to find the related rules texts. However, the performance of our model may be hindered by the noise introduced by the irrelevant rule texts from the retrieval. To conquer this deficiency, it is beneficial to develop additional filtering methods to alleviate the influence of irrelevant rule texts.

References

- Wei Bi, Jun Gao, Xiaojiang Liu, and Shuming Shi. 2019. [Fine-grained sentence functions for short-text conversation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3984–3993, Florence, Italy. Association for Computational Linguistics.
- Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. [QuAC: Question answering in context](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.
- Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. [MuTual: A dataset for multi-turn dialogue reasoning](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1406–1416, Online. Association for Computational Linguistics.
- Yifan Fan, Xudong Luo, and Pingping Lin. 2020. A survey of response generation of dialogue systems. *International Journal of Computer and Information Engineering*, 14(12):461–472.
- Yifan Gao, Jingjing Li, Michael R Lyu, and Irwin King. 2021. Open-retrieval conversational machine reading. *arXiv preprint arXiv:2102.08633*.
- Yifan Gao, Piji Li, Wei Bi, Xiaojiang Liu, Michael R. Lyu, and Irwin King. 2020a. [Dialogue generation on infrequent sentence functions via structured meta-learning](#).
- Yifan Gao, Chien-Sheng Wu, Shafiq Joty, Caiming Xiong, Richard Socher, Irwin King, Michael Lyu, and Steven C.H. Hoi. 2020b. [Explicit memory tracker with coarse-to-fine reasoning for conversational machine reading](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 935–945, Online. Association for Computational Linguistics.
- Yifan Gao, Chien-Sheng Wu, Jingjing Li, Shafiq Joty, Steven C.H. Hoi, Caiming Xiong, Irwin King, and Michael Lyu. 2020c. [Discern: Discourse-aware entailment reasoning network for conversational machine reading](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2439–2449, Online. Association for Computational Linguistics.
- Jia-Chen Gu, Chongyang Tao, Zhenhua Ling, Can Xu, Xiubo Geng, and Daxin Jiang. 2021. [MPC-BERT: A pre-trained language model for multi-party conversation understanding](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3682–3692, Online. Association for Computational Linguistics.
- Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2017. [Tracking the world state with recurrent entity networks](#). In *International Conference on Learning Representations*.
- Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in building intelligent open-domain dialog systems. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–32.
- Carolin Lawrence, Bhushan Kotnis, and Mathias Niepert. 2019. [Attending to future tokens for bidirectional sequence generation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1–10, Hong Kong, China. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. 2000. [The structure and performance of an open-domain question answering system](#). In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 563–570, Hong Kong. Association for Computational Linguistics.
- Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. Dialogue graph modeling for conversational machine reading. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. [CoQA: A conversational question answering challenge](#). *Transactions of the Association for Computational Linguistics*, 7:249–266.
- Da Ren, Yi Cai, Xue Lei, Jingyun Xu, Qing Li, and Ho fung Leung. 2019. [A multi-encoder neural conversation model](#). *Neurocomputing*, 358:344–354.
- Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. [Interpretation](#)

- of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium. Association for Computational Linguistics.
- Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7007–7014. AAAI Press.
- Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.
- Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM '19*, page 267–275, New York, NY, USA. Association for Computing Machinery.
- Nikhil Verma, Abhishek Sharma, Dhiraj Madan, Danish Contractor, Harshit Kumar, and Sachindra Joshi. 2020. Neural conversational QA: Learning to reason vs exploiting patterns. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7263–7269, Online. Association for Computational Linguistics.
- Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).
- Bowen Wu, Mengyuan Li, Zongsheng Wang, Yifu Chen, Derek F. Wong, Qihang Feng, Junhong Huang, and Baoxun Wang. 2019. Guiding variational response generator to exploit persona. In *Annual Meeting of the Association for Computational Linguistics*.
- Liu Yang, Minghui Qiu, Chen Qu, Cen Chen, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, and Haiqing Chen. 2020. Iart: Intent-aware response ranking with transformers in information-seeking conversation systems. In *Proceedings of The Web Conference 2020, WWW '20*, page 2592–2598, New York, NY, USA. Association for Computing Machinery.
- Munazza Zaib, Quan Z Sheng, and Wei Emma Zhang. 2020. A short survey of pre-trained language models for conversational ai-a new age in nlp. In *Proceedings of the Australasian Computer Science Week Multiconference*, pages 1–4.
- Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3740–3752, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Zhuosheng Zhang, Siru Ouyang, Hai Zhao, Masao Utiyama, and Eiichiro Sumita. 2021. Smoothing dialogue states for open conversational machine reading. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3685–3696, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Victor Zhong and Luke Zettlemoyer. 2019. E3: Entailment-driven extracting and editing for conversational machine reading. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2310–2320, Florence, Italy. Association for Computational Linguistics.
- Pengfei Zhu, Zhuosheng Zhang, Jiangtong Li, Yafang Huang, and Hai Zhao. 2018. Lingke: a fine-grained multi-turn chatbot for customer service. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 108–112, Santa Fe, New Mexico. Association for Computational Linguistics.

A Case Study

To reveal more insights into our framework, we randomly collect test samples and conduct the case study (see Figure 7). The ground truth answers are indicated in red, the TF-IDF scores are indicated in green, and the predictions of UNICMR_{large} are indicated in blue. The retrieved rule texts are in descending order in terms of TF-IDF scores. For the analysis on cases, please refer to Section 6.7.

B Hyperparameter-Related Experiments

In this section, additional experiments related to the hyperparameter m (number of retrieved rule texts) and the hyperparameter maximum generation length are conducted with their results shown in Table 7-10.

In Table 7, the hyperparameter m (number of retrieved rule texts) is varied to compare our model performance on the OR-ShARC test set, test set seen and test set unseen divisions respectively. In Table 8, the hyperparameter maximum generation length of the backbone encoder-decoder model is varied to compare our model performance on the same datasets. The corresponding performance of the above two experiments on dev set is shown by Table 9 and Table 10 respectively. Note in these experiments, all the hyperparameters remain the same unless explicitly stated.

Case Type	Dialogue Setups	UniCMR Output
Exactness	Scen.: (empty) Ques.: Could I authorize the use of marijuana for this patient? His.: (empty) Gold Rule: (same as the first retrieved rule) Gold Answer: If it medically appropriate?	Retrieved Rules: (1) [50.51] Patient Information. Healthcare practitioners may authorize the use of marijuana for any patient as long as it's medically appropriate under the law and the profession's standard of care. The Department of Health doesn't keep a list of those specific practitioners that are authorizing, however there is a list of healthcare practitioners that are allowed to authorize marijuana for medical marijuana. Prediction: Is it medically appropriate under the law and the profession's standard of care?
	Robust to Noisy Retrieved Rules	Scen.: (empty) Ques.: Is this item eligible for the zero rate? His.: Q: Is it rescue equipment? A: No Gold Rule: Items that qualify for the zero rate. The eligible items include: * rescue equipment * resuscitation training dummies Gold Answer: Is it a resuscitation training dummy? Retrieved Rules: (1) [40.81] What you can't claim using your Medicare online account. Although you may have a claim for an eligible item, you can't claim online if: * the item is for a service provided to someone who isn't on your Medicare card * the item is for a service provided more than 2 years ago * the item is for a service provided by a hospital or approved day facility inpatient * you've been bulk billed for the item or had the claim submitted at your doctor's surgery on your behalf (2) [38.50] Items that qualify for the zero rate. You may be able to apply zero VAT when you sell the following to an eligible charity: * equipment for making 'talking' books and newspapers * lifeboats and associated equipment, including fuel * medicine or ingredients for medicine * resuscitation training models (3) [32.56] Items that qualify for the zero rate. The eligible items include: * medical, veterinary and scientific equipment * ambulances * goods for disabled people * motor vehicles for medical use (4) [32.56] (same as the gold rule) Prediction: Is it resuscitation training dummies?

Figure 7: Case study of UNICMR_{large} by comparison with ground truth answers.

Test Set	Decision Making								Question Generation							
	Micro				Macro				F1 _{BLEU1}				F1 _{BLEU4}			
	1	6	12	20	1	6	12	20	1	6	12	20	1	6	12	20
Full-Dataset	69.1	74.7	78.5	77.3	69.7	74.7	78.5	77.5	41.2	51.9	53.1	53.3	34.4	45.1	46.3	46.2
Seen	77.6	85.5	88.4	88.3	77.6	85.5	88.4	88.2	53.1	67.6	73.4	71.6	47.9	63.6	69.2	67.2
Unseen	63.0	66.8	71.2	69.3	63.8	66.8	71.0	69.5	29.3	36.9	32.9	36.1	20.5	27.4	23.1	26.2

Table 7: Comparison of our model under different number of retrieved rule texts on test set.

Test Set	Decision Making								Question Generation							
	Micro				Macro				F1 _{BLEU1}				F1 _{BLEU4}			
	10	20	50	70	10	20	50	70	10	20	50	70	10	20	50	70
Full-Dataset	75.6	77.1	77.0	77.0	75.9	77.1	77.0	77.1	35.5	52.2	54.3	54.3	30.3	45.9	47.8	47.8
Seen	87.7	88.2	88.2	88.2	87.7	88.2	88.2	88.2	41.9	67.5	72.2	72.2	39.0	63.7	68.4	68.4
Unseen	66.9	69.0	68.9	68.9	67.3	69.0	69.0	69.0	25.9	37.9	37.4	37.4	18.9	29.0	28.0	28.0

Table 8: Comparison of our model under different max generation length limit on test set.

Dev Set	Decision Making								Question Generation							
	Micro				Macro				F1 _{BLEU1}				F1 _{BLEU4}			
	1	6	12	20	1	6	12	20	1	6	12	20	1	6	12	20
Full-Dataset	65.4	76.6	77.8	77.6	66.3	76.7	78.2	78.2	36.6	58.2	58.9	58.8	29.5	51.6	53.3	51.8
Seen	78.4	88.2	88.8	90.6	78.2	88.1	88.8	90.5	52.7	71.8	74.6	72.6	47.5	66.8	70.7	68.4
Unseen	54.7	66.9	68.8	66.8	56.4	66.7	69.1	68.7	19.7	40.0	38.6	43.1	10.3	30.5	30.5	32.5

Table 9: Comparison of our model under different number of retrieved rule texts on dev set.

Dev Set	Decision Making								Question Generation							
	Micro				Macro				F1 _{BLEU1}				F1 _{BLEU4}			
	10	20	50	70	10	20	50	70	10	20	50	70	10	20	50	70
Full-Dataset	77.6	76.9	77.1	77.1	78.5	77.4	77.7	77.7	46.9	57.2	66.1	61.1	40.8	50.9	54.8	54.8
Seen	91.0	89.4	89.8	89.8	90.9	89.3	89.7	89.7	47.7	71.0	78.1	78.1	44.1	66.6	73.9	73.9
Unseen	66.5	66.6	66.6	66.6	68.5	67.7	67.7	67.7	36.3	40.6	40.7	40.1	27.6	32.2	31.6	31.6

Table 10: Comparison of our model under different max generation length limit on dev set.

Generative Knowledge Selection for Knowledge-Grounded Dialogues

Weiwei Sun, Pengjie Ren, Zhaochun Ren*

Shandong University, Qingdao, China

sunweiwei@gmail.com, {renpengjie, zhaochun.ren}@sdu.edu.cn

Abstract

Knowledge selection is the key in knowledge-grounded dialogues (KGD), which aims to select an appropriate knowledge snippet to be used in the utterance based on dialogue history. Previous studies mainly employ the classification approach to classify each candidate snippet as “relevant” or “irrelevant” independently. However, such approaches neglect the interactions between snippets, leading to difficulties in inferring the meaning of snippets. Moreover, they lack modeling of the discourse structure of dialogue-knowledge interactions. We propose a simple yet effective generative approach for knowledge selection, called GENKS. GENKS learns to select snippets by generating their identifiers with a sequence-to-sequence model. GENKS therefore captures intra-knowledge interaction inherently through attention mechanisms. Meanwhile, we devise a *hyperlink* mechanism to model the dialogue-knowledge interactions explicitly. We conduct experiments on three benchmark datasets, and verify GENKS achieves the best results on both knowledge selection and response generation.

1 Introduction

To improve the informativeness in open-domain dialogue agents (Freitas et al., 2020), knowledge-grounded dialogues (KGD) are proposed to leverage external structured (Liu et al., 2019) and unstructured (Dinan et al., 2019) knowledge to dialogue responses. In KGD, it is pivotal to embed factual and conversationally appropriate knowledge in responses. Two classes of approaches are considered to embed knowledge: *end-to-end* and *pipeline*. End-to-end models, such as FiD (Izacard and Grave, 2021), process the document and generate the response in one shot. However, they tend to misuse knowledge (Adolphs et al., 2021). Pipeline models address this problem by explicitly identifying a specific knowledge snippet to

*Corresponding author.

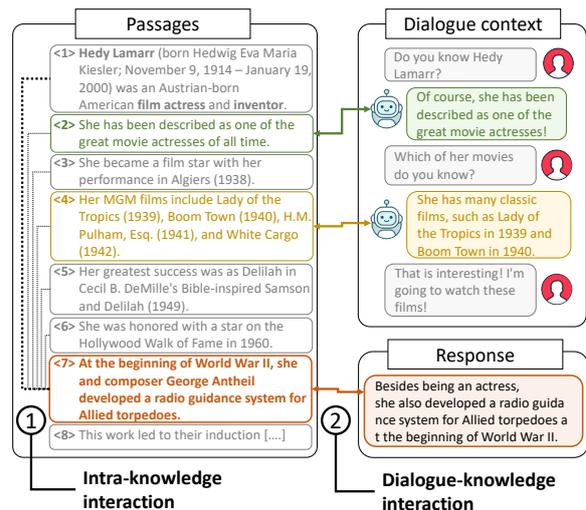


Figure 1: An example of knowledge-grounded dialogues. The dialogue agent selects a knowledge snippet (i.e., <7>) from passages and generates a response based on it. Intra-knowledge interactions and dialogue-knowledge interactions are denoted by ① and ②, respectively.

be used in the response (Adolphs et al., 2021). Typically, pipeline KGD approaches have two sub-steps, i.e., knowledge selection and response generation (Dinan et al., 2019; Kim et al., 2020): The former aims to select knowledge snippets from passages, and the latter generates responses based on them. Knowledge selection plays a vital role in KGD as it directly determines the content of the response (Lian et al., 2019; Meng et al., 2020). In this paper, we focus on selecting knowledge snippets for dialogue to enhance pipeline KGD models.

The *Classification* paradigm dominates knowledge selection studies. In this paradigm, each snippet is independently classified as “relevant” or “irrelevant” (Dinan et al., 2019; Zhao et al., 2020b). However, these approaches ignore *knowledge interactions*, which refer to flows of information within the knowledge or between knowledge and dialogues. As shown in Figure 1, we identify two

types of knowledge interactions in KGD:

Intra-knowledge interaction Intra-knowledge interaction refers to the interactions between snippets. It is worth noting that the meaning of a knowledge snippet is context-dependent and can be ambiguous when taken individually. For example, the <8> snippet in Figure 1 “*This work led to their*” has a referential element *their*, and is difficult to identify its meaning without knowing the remaining context of the sentence. However, with the existence of the remaining context, we can quickly infer that it refers to *Lamarr and George Antheil. This problem challenges existing methods when selecting knowledge on new topics.*

Dialogue-Knowledge interaction Previous works also neglect interactions between dialogue and knowledge. There is a discourse structure and smooth transition of involved knowledge in multi-turn dialogue. For example, *Lamarr’s* profession mentioned in the dialogue in Figure 1 is demonstrated in a parallel and multi-perspective manner, while some other cases follow a shallow-to-deep structure in dialogue.

Some recent efforts attempt to fix these problems within the classification paradigm; for example, [Li et al. \(2022\)](#) build a semantic graph for passages to capture intra-knowledge interaction, [Kim et al. \(2020\)](#) propose sequential knowledge selection to model the dialogue-knowledge interaction as latent variables. However, they are complicated, lack deep semantic interactions, and are challenging to model the two types of knowledge interaction simultaneously.

In this work, we propose **GENKS (Generative Knowledge Selection)**, a simple yet effective *generative* model that addresses these challenges. GENKS first assigns an identifier to each snippet, feeds all the snippets into the model simultaneously, and then selects snippets by generating their identifiers with a sequence-to-sequence Transformer model (e.g., BART ([Lewis et al., 2020a](#))). Compared with KGD methods with the *classification* paradigm, GENKS captures interactions between knowledge snippets through the *self-attention* mechanism in Transformer ([Vaswani et al., 2017](#)). Therefore, GENKS can obviate the ambiguity in snippets with the existence of the rest context and improve the understanding of knowledge. Moreover, we propose a *hyperlink* method to

capture the dialogue-knowledge interactions explicitly and effectively. Finally, we propose to joint knowledge selection and response generation within one generative model.

We evaluate our proposed method on three public KGD datasets: Wizard of Wikipedia ([Dinan et al., 2019](#)), Holl-E ([Moghe et al., 2018](#)), and CMU_DoG ([Zhou et al., 2018](#)). The experimental results show that GENKS significantly improves the accuracy of knowledge selection as well as the quality of response generation, by establishing new state-of-the-art on KGD benchmarks. Improvements are particularly significant on unseen topics, outperforming the BART classification model by up to 8.1% absolute. GENKS also achieves the best results as the number of dialogue turns increased, with an average of 10% improvements over the BART classification model in the last three turns. We also compare our model with recent SOTA end-to-end methods ([Shuster et al., 2021](#)), and find our model can generate responses with fewer hallucinations while having better controllability and interpretability. The effectiveness of the proposed method is also validated through human evaluation and ablative experiments.

Our contributions are summarized as follows: (1) We propose GENKS, which is the first attempt at generative knowledge selection in KGD. (2) GENKS captures intra-knowledge and dialogue-knowledge interactions simultaneously. (3) We propose a hyperlink method to enhance the interactions between dialogue and knowledge. (4) Experiments verify that GENKS establishes a new state-of-the-art on KGD¹.

2 Related work

Knowledge-grounded dialogues With the advances in large-scale language models, dialogue agents can now generate high-quality responses using parametric knowledge ([Thoppilan et al., 2022](#); [Freitas et al., 2020](#); [Bao et al., 2021](#)). However, hallucination remains a challenge, which means that the language model tends to generate plausible-looking statements that are factually incorrect ([Shuster et al., 2021](#)). To address this problem, knowledge-augmented approaches are applied in dialogue generation ([Lewis et al., 2020b](#)). In knowledge-grounded dialogues (KGD), the dialogue models first select a knowledge snippet from

¹The code is available at: <https://github.com/sunweiwei/GenKS>

passages and then generate the responses (Liu et al., 2018; Dinan et al., 2019).

Knowledge selection As the critical step in KGD, knowledge selection has received many studies. The existing methods mainly employ *classification* model with dual-encoder (Dinan et al., 2019; Kim et al., 2020) or cross-encoder (Zhao et al., 2020b) architecture. However, the classification paradigm is unable to capture the knowledge interaction in KGD (Kim et al., 2020; Li et al., 2022). To address this problem, Li et al. (2022) propose a graph-based method to capture the relationship between candidate snippets, Zhan et al. (2021a) and Wu et al. (2021) employ machine reading comprehension model to extract span from long document. Sequential knowledge selection has also been proposed to capture the topic transition in conversations (Kim et al., 2020; Zhan et al., 2021b; Zheng et al., 2020; Meng et al., 2020; Yang et al., 2022). Despite their effectiveness, the existing methods have two drawbacks: (1) they use compact vectors to represent dialogue and knowledge and thus lack deep semantic interactions; (2) they are complicated and challenging to capture intra-knowledge and dialogue-knowledge interactions simultaneously. We address these drawbacks by shifting the modeling paradigm of knowledge selection to identifier generation (Sun et al., 2022), and propose GENKS to capture the two types of interaction simultaneously using Transformer (Vaswani et al., 2017).

Generative knowledge selection A generative paradigm for knowledge selection is not foreign to the NLP community; for example, sequence-to-sequence models have been applied on entity retrieval (Cao et al., 2021), document ranking (Nogueira et al., 2020; Tay et al., 2022), multi-evidence retrieval (Min et al., 2021; Yavuz et al., 2022), and etc. Our proposed model GENKS differs from existing methods in the following ways: (1) we are the first to explore generative knowledge selection in KGD; (2) we consider the effectiveness of intra-knowledge interaction; (3) we design hyperlinks to capture the interaction between knowledge and dialogue.

3 GENKS

We provide an overview of GENKS in Figure 2. As shown in Figure 2, the dialogue data is first serialized into a sequence. Then a sequence-to-sequence model (i.e., BART) is employed to select knowl-

edge and get the response by generating the target sequence autoregressively. In this section, we first formulate the task in Section 3.1. Then, we detail the serialization (Section 3.2) and optimization (Section 3.3) methods.

3.1 Problem formulation

Suppose that we have a case of knowledge-grounded dialogues (C, \mathcal{K}, r) , where $C = (c_1, \dots, c_{|C|})$ is a dialogue context that contains $|C|$ utterances, r is the response to C , $\mathcal{K} = (K_1, \dots, K_{|\mathcal{K}|})$ denotes $|\mathcal{K}|$ passages that are relevant to C ; for each i , $K_i = (k_{i,1}, \dots, k_{i,|K_i|})$ denotes a passage that contains $|K_i|$ snippets. We define $m = \sum_{i=1}^{|\mathcal{K}|} |K_i|$ as the total number of snippets in \mathcal{K} . A knowledge-grounded dialogue agent is decoupled into two modules: a knowledge selection module $P(k|C, \mathcal{K})$ that selects a snippet from \mathcal{K} ; a response generation module $P(r|C, \mathcal{K}, k_s)$ where k_s is the selected snippet from knowledge selection module.

3.2 Serialization

We formulate the knowledge selection task as a procedure of sequence generation. As shown in Figure 2, the dialogue context C and knowledge candidates \mathcal{K} are mapped into a sequence and then fed into a sequence-to-sequence model. The model’s output is converted back to the selected knowledge k or the response r .

Specifically, we first assign an identifier to each snippet in \mathcal{K} , sequentially starting from $\langle k1 \rangle$ to $\langle km \rangle$. Then we convert passages \mathcal{K} into a sequence using a template that packages snippets with the corresponding identifiers and concatenates them in order; see the green block in Figure 2. Similarly, the dialogue context C is serialized by adding task prompts, i.e., task description and speaker name, as shown in the blue block in Figure 2.

In multi-turn dialogues, the knowledge appearing in the dialogue history prompts the discourse structure of knowledge transition and knowledge expression. Hence we propose a *hyperlink* method to capture the dialogue-knowledge interaction explicitly. We provide an example of the hyperlink method in Figure 2. We see that the first utterance of User1 refers to a snippet (whose identifier is $\langle k2 \rangle$) in the passage “Skateboarding”. We thus add a hyperlink to the utterance. The hyperlink includes the identifier and the title of the snippet, i.e., annotating [Skateboarding] $\langle k2 \rangle$ at the beginning of

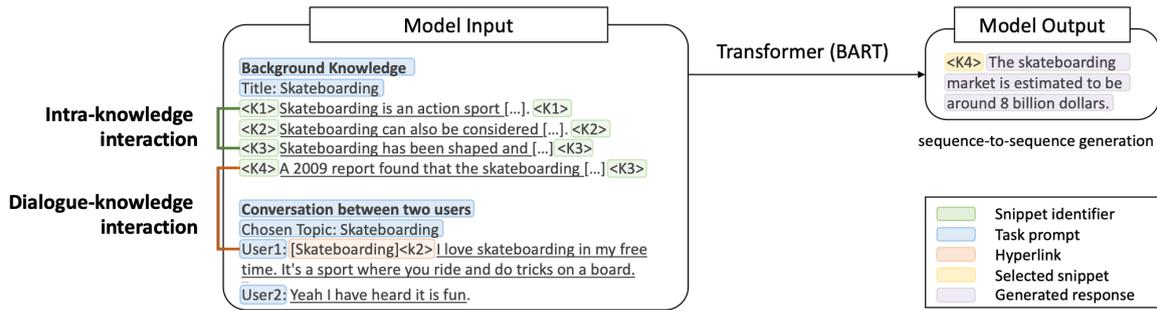


Figure 2: Overview of GENKS. The dialogue context and the knowledge are serialized and fed into a seq-to-seq model, BART. The outputs are the identifier of the selected snippet (i.e., <k5>) and the response.

this utterance (as shown in the red block in Figure 2). Finally, we splice the passages and dialogue context sequences as input for a Transformer model (i.e., BART). Therefore, the model can capture the intra-knowledge and dialogue-knowledge interactions through a *self-attention* mechanism (Vaswani et al., 2017).

3.3 Optimization

The knowledge selection model is optimized by the cross-entropy loss: $\mathcal{L} = -\log P(k_{true}|C, \mathcal{K})$, where k_{true} denotes the label knowledge. Since k_{true} needs to be labeled manually and is not available in some scenarios (Zhou et al., 2018), we construct pseudo-labels for model training following Zhao et al. (2020b) in cases the knowledge label is absent. In particular, we calculate the F1 score (Dinan et al., 2019) between each knowledge snippet and the response. We use the snippet with the highest score as the pseudo label. Such a method is based on the intuition that human responses provide hints regarding the relevance of the snippets (Zhao et al., 2020b; Li et al., 2020).

Since both knowledge selection and response generation are modeled with the *generative* paradigm, we unify the two modules with one joint generative model. In this joint model, the knowledge selection and the response generation are optimized jointly, with shared parameters. To this end, we splice the knowledge identifier k_{true} and response r into one sequence (as shown in Figure 2). Then, we optimize the sequence-to-sequence model using cross-entropy loss on all the tokens of the target sequence. In inference, the model generates knowledge identifier k_s and responses r in an autoregressive fashion. We note that the end-to-end model allows the two tasks to be mutually enhanced and improves the model’s efficiency.

4 Experimental setup

4.1 Datasets

We conduct experiments on Wizard of Wikipedia (WoW) (Dinan et al., 2019), Holl-E (Moghe et al., 2018), and CMU_DoG (Zhou et al., 2018). The statistical details on these three datasets are shown in Table 7 in the appendix.

- **WoW** is an open-domain KGD dataset using Wikipedia passage as background knowledge. The test set of Wizard is split into seen and unseen versions, where the unseen test set contains 58 new topics not discussed in the training data.
- **Holl-E** focuses on the movie domain. The background knowledge consists of plots, comments, and movie reviews collected from different websites. Holl-E has two versions of the test set: single test and multi-reference test. In the multi-reference test, there are multiple human-annotated ground-truth knowledge and corresponding responses for each instance.
- **CMU_DoG** focuses on the domain of movies. The workers discuss a movie in depth given the background knowledge (e.g., introduction, plots, and key scenes).

4.2 Baselines

We compare GENKS with baselines of two categories: (i) *End-to-end methods* that generate response directly without explicit knowledge selection, and (ii) *Pipeline methods* that explicitly select knowledge snippet to be used in response.

The end-to-end methods we consider are:

- **BART** (Lewis et al., 2020a) that generates responses without access to the external passage and uses knowledge inside model parameters instead.
- **BART FID** (Izacard and Grave, 2021) concatenates and encodes each candidate knowledge

with dialogue separately and fuses all the encoded representation in the decoder to generate the response.

- **BART RAG-DPR** is a baseline adopted by Adolphs et al. (2021), which uses DPR-retrieved passages and produces response using RAG.
- **BART FiD-RAG DPR-Poly** (Shuster et al., 2021) uses DPR-Poly to retrieve passage and uses FiD-RAG to generate the response.

Regarding the pipeline baselines, according to their knowledge selection modeling paradigm, we sub-categorize pipeline baselines into four groups:

(1) The *Classification methods*, includes:

- **SKT** (Kim et al., 2020) proposes sequential knowledge selection.
- **DiffKS** (Zheng et al., 2020) captures the knowledge differences between adjacent turns.
- **DukeNet** (Meng et al., 2020) models the knowledge shift and tracking processes with a dual learning scheme.
- **KnowledGPT** (Zhao et al., 2020b) exploits pre-trained language models in KGD.
- **MIKe** (Meng et al., 2021) distinguish user-initiative and system-initiative.
- **K-Mine** (Lotfi et al., 2021) proposes a score-and-aggregate module.
- **TAKE** (Yang et al., 2022) propose a topic-shift aware network.

(2) The *MRC methods*, includes:

- **CoLV** (Zhan et al., 2021a) proposes a collaborative latent variable model.
- **DIALKI** (Wu et al., 2021) proposes a MRC-based model to extract span from passage.

(3) The *Graph-based methods*, includes:

- **Graph** (Li et al., 2022) builds a semantic graph upon candidate documents and employs a GNN model.

(4) And the *Knowledge generation methods*, includes:

- **K2R** (Adolphs et al., 2021) uses the RAG-based model to generate knowledge text and then generates dialogue response based on it.

4.3 Evaluation metrics

In WoW, we choose perplexity (**PPL**) of the ground-truth responses, unigram **F1**² (Dinan et al., 2019), **Knowledge-F1** (Shuster et al., 2021), and **BLEU-4** (Papineni et al., 2002) score as metrics. In Holl-E, we additionally use **ROUGE-1**, and **ROUGE-2** following Meng et al. (2020). In

²<https://github.com/facebookresearch/ParLAI>

CMU_DoG, we additionally use embedding-based metrics includes **Average**, **Extreme**, and **Greedy** following Zhao et al. (2020b).

In addition, we randomly sample 100 examples from the WoW test seen and WoW test unseen, respectively, and recruit three experts for human evaluation. The annotators are asked to judge the model-generated response in four ways:

- **Fluency**, which measures whether the response is fluency in expression;
- **Coherence**, which measures whether the response is coherence to the dialogue context;
- **Relevance**, which measures whether the knowledge used in the response is relevant to the dialogue; and
- **Factuality** measures whether the response’s content is factual. In Factuality evaluation, the experts check the content using Google.

The annotators are asked to assign a score in {0, 1} (representing “nonfactual” and “factual”) for factuality, and a score in {0, 1, 2} (representing “bad”, “fair”, and “good”) for the others.

4.4 Implementation details

We implement the GENKS using BART large (with 400M parameters) (Lewis et al., 2020a) in HuggingFace’s Transformers library. We truncate the dialogue context to 256 tokens, then truncate the knowledge so that the total length is less than 1024 tokens. During inference, the responses are decoded using a greedy search. See Appendix A for more details.

Typically, the number of passages in \mathcal{K} is large, so that the input sequence exceeds the maximum input length of BART (i.e., 1024 tokens). To address this problem, we take advantage of a lightweight passage selector based on DistilBERT (with 66M parameters) (Sanh et al., 2019), which aims to rank the passages in \mathcal{K} . Specifically, we concatenate each passage with dialogue context and encode the sequence using DistilBERT. Finally, the representation of [CLS] token is used to estimate the relevance score of the passage through a learnable MLP classifier. The passage selector is optimized via contrastive learning objective (Nogueira and Cho, 2019), in which the model learns to assign a higher score to positive passages than negative passages. During inference, we keep only the top-1 passage ranked by the passage selector. The passage selector gets Recall@1 of 75.5%, 76.5%, and 68.0% for the WoW test seen, WoW test unseen,

Methods	WoW		Holl-E	
	Seen	Unseen	Single	Multi
<i>Classification methods</i>				
SKT (Kim et al., 2020)	26.8	18.3	29.2	39.2
DukeNet (Meng et al., 2020)	26.4	19.6	30.0	40.3
DiffKS (Zheng et al., 2020)	25.5	19.7	33.0	-
KnowledGPT (Zhao et al., 2020b)	28.0	25.4	-	-
MIKe (Meng et al., 2021)	28.4	21.5	31.9	41.8
K-Mine (Lotfi et al., 2021)	29.7	28.3	31.7	-
TAKE (Yang et al., 2022)	28.8	25.8	-	-
<i>Other methods</i>				
CoLV (Zhan et al., 2021a)	30.1	18.9	32.7	-
DIALKI (Wu et al., 2021)	<u>32.9</u>	<u>35.5</u>	-	-
Graph (Li et al., 2022)	29.4	30.8	<u>37.7</u>	<u>46.1</u>
GenKS	34.2	36.6	37.9	46.8
<i>Variants for comparison</i>				
- BART classification	29.8	29.7	34.0	44.0
- BART classification w/ position	30.1	31.2	34.0	44.0
- Hierarchical classification	30.0	31.4	33.8	43.7
- Without passage selector	31.4	32.0	34.5	44.4
- Unorder knowledge snippets	31.8	33.3	36.5	45.8
- Without hyperlink	33.4	35.4	36.9	45.4

Table 1: Knowledge selection accuracy on WoW (seen and unseen test set) and Holl-E (single reference and multi-reference test set). **Bold** denote the best results with significant improvements over the previous SOTA (t-test, $p < 0.05$). Underline denote second best results.

and Holl-E, respectively.

5 Experimental results

5.1 Performance on knowledge selection

We evaluate the knowledge selection effectiveness of GENKS on WoW and Holl-E, respectively³. In Table 1, we compare the knowledge selection accuracy of GENKS with previous pipeline methods. Results show that GENKS achieves the best knowledge selection accuracy on both datasets and consistently outperforms baselines.

We find that GENKS particularly excels at topics that do not appear during training (see WoW unseen test split). For example, the classification models both have noticeable accuracy drops on the unseen topic. In contrast, models that model the intra-knowledge interaction (e.g., GENKS, Graph, DIALKI) can better understand the knowledge of unseen topics⁴.

To evaluate the Performance of GENKS as dialogue goes deeper, we compare GENKS with

³We cannot evaluate the knowledge selection accuracy on CMU_DoG because the knowledge snippets used in responses are not manually labeled.

⁴The higher results on unseen than seen might be due to the smaller number of topics in the unseen test set.

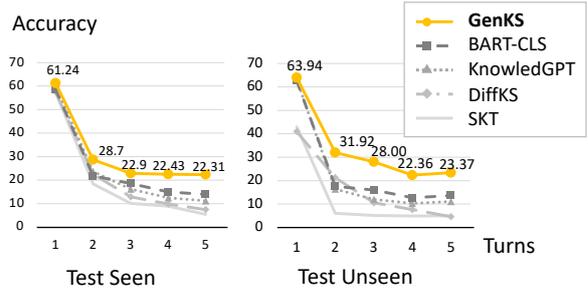


Figure 3: Knowledge selection accuracy over different dialogue turns. BART-CLS represents a text-matching model with cross-encoder architecture.

four classification baselines (SKT, DiffKS, KnowledGPT, and BART-CLS) overturns. Figure 3 shows the results. Both methods achieve good accuracy in the first few turns. However, as the conversation dives deeply into a topic, a significant performance decline can be seen in baseline methods. In contrast, GENKS that explicitly captures the multi-turn dialogue-knowledge interaction, achieves a relatively high accuracy (around 22%-23%).

5.2 Quality of generated responses

We report response generation evaluation results on WoW in Table 2. The results on Holl-E and CMU_DoG are available in Table 8 and Table 9 in the appendix. The results of baselines are cited from original papers or re-evaluated using officially released checkpoints.

Compared with previous pipeline models, GENKS achieve the best Performance on almost all metrics. For example, GENKS surpasses KnowledGPT by 0.7% and 2.4% in terms of F1 on WoW seen and WoW unseen, respectively. Note that the improvements on the unseen test set are more notable than on the seen test set, which agrees with the experimental results regarding knowledge selection. GENKS also achieve competitive results compared to SOTA end-to-end models. For example, GENKS performs comparably to BART FiD-RAG DPR-Poly on WoW seen and outperformed on WoW unseen.

5.3 Ablation study about knowledge selection

To analyze the effect of each component in GENKS, we designed several variants and conducted an ablation study about knowledge selection. Results are listed in Table 1, “Variants for comparison”. The details of compared variants and the findings are as follows:

BART classification We use BART to classify each

Methods	WoW Seen				WoW Unseen			
	PPL	F1	KF1	B4	PPL	F1	KF1	B4
<i>End-to-end models</i>								
BART (Lewis et al., 2020a)	14.7	20.9	17.4	1.7	18.9	18.8	15.1	0.9
BART FiD (Izacard and Grave, 2021)	17.0	21.5	20.0	3.6	18.4	20.6	19.2	3.2
BART RAG-DPR (Adolphs et al., 2021)	11.5	22.6	26.1	3.7	13.1	21.5	22.7	3.0
BART FiD-RAG DPR-Poly (Shuster et al., 2021)	11.4	22.1	29.7	4.1	13.1	21.1	27.1	3.8
<i>Pipeline models</i>								
DukeNet (Meng et al., 2020)	48.3	19.3	18.5	2.4	69.4	17.1	16.5	1.7
CoLV (Zhan et al., 2021a)	39.5	20.3	18.2	2.8	54.3	18.5	17.5	2.1
KnowledGPT (Zhao et al., 2020b)	19.2	22.0	23.8	3.7	22.3	20.5	22.1	3.0
K-Mine (Lotfi et al., 2021)	13.2	21.8	-	-	16.4	21.1	-	-
K2R RAG-DPR (Adolphs et al., 2021)	18.3	22.0	27.3	3.7	22.3	19.9	23.2	2.8
K2R BART RAG-DPR (Adolphs et al., 2021)	17.9	21.3	29.2	3.5	21.1	19.9	24.3	2.5
GenKS	13.1	22.9*	29.5	4.5*	13.2	22.7*	28.1*	4.6*
<i>Ablative variants</i>								
- With BART classification knowledge	14.7	22.0	25.9	3.5	16.2	21.1	24.4	3.1
- Without identifiers generation	13.8	21.7	23.2	3.7	14.1	21.8	23.3	3.9
- Without hyperlink	14.2	22.1	27.2	3.9	15.5	22.3	26.9	4.2
- With oracle knowledge	8.9	38.8	74.2	13.1	10.5	38.9	74.5	12.8

Table 2: Evaluation results on WoW seen and unseen test set in terms of response quality. We compare against the ground-truth dialogue response in terms of perplexity (PPL), F1, Knowledge F1 (KF1), and BLEU-4 (B4). The four groups lists previous end-to-end models, previous pipeline models, GenKS, and ablative variants. The best results are highlighted with **bold**, and the second-best results are highlighted with underline. * indicates significant improvements over all baselines with p-value < 0.05.

candidate snippet into two classes: “relevant” or “irrelevant”. The results show that BART in the classification paradigm performs worse than GENKS by a large margin.

BART classification w/ position To understand the influence of position bias, we splice the snippet’s position into the classification model’s input. We find that the results are improved to a certain extent (about 1% improvement), but there is still a clear gap compared with GENKS.

Hierarchical classification This variant first uses the passage selector model of GENKS to rank the passages and then selects the snippets in the top-ranked passage using BART classification w/ position. The results show that the passage selector does not affect the classification model’s Performance.

Without passage selector When the passage selector model of GENKS is removed, the model has more probability of truncating the label knowledge, resulting in an evident decline in Performance.

Unorder knowledge snippets To disable the intra-knowledge interaction, we unorder the snippets so that order of the snippets is inconsistent with the original passages. This variant shows a decline in selection accuracy, especially on unseen topics, indicating that keeping the order of the snippets in

the passage is necessary.

Without hyperlinks We remove the hyperlinks in the dialogue context. About a 1% accuracy drop is seen, indicating the effectiveness of hyperlinks.

5.4 Ablation study about response generation

As shown in Table 2, we also conduct an ablation study about response generation. The details of compared variants and the findings are as follows:

With BART classification knowledge When replacing the generated identifier with the knowledge selected by BART classification, a performance decline is witnessed—the F1 value drops by 0.7% and 1.8% on Wizard seen and unseen, sustaining the effectiveness of the knowledge selection of GENKS.

Without identifier generation This variant removes the identifier generation by directly generating the response. We see notable performance drops, especially in the KF1 metric. The results indicate that explicit training and inference about knowledge selection enable to use of more appropriate knowledge in response generation.

Without hyperlinks This variant removes hyperlinks from GENKS. It performs worse than GENKS, probably due to its lower accuracy of knowledge selection than GENKS.

Use the oracle knowledge We replace the model-predicted snippet identifier with the oracle one

Methods	WoW Seen				WoW Unseen			
	Flu.	Coh.	Rel.	Fact.	Flu.	Coh.	Rel.	Fact.
BART	1.82	1.51	1.45	0.82	1.76	1.50	1.47	0.76
BART FiD	1.88	1.70	1.55	0.84	1.85	1.67	1.53	0.82
TMN	1.59	1.41	1.08	0.62	1.42	1.30	0.98	0.59
DukeNet	1.69	1.56	1.22	0.71	1.66	1.47	1.10	0.72
KnowledGPT	1.89	1.67	1.58	0.87	1.87	1.68	1.51	0.83
GenKS	1.90	1.72	1.69	0.89	1.91	1.71	1.67	0.91

Table 3: Human evaluation results. Flu, Coh, Rel, and Fact denote Fluency, Coherence, Relevance, and Factuality, respectively.

	KS	RG	ACC	F1
BART RAG (Adolphs et al., 2021)	0	11.5	-	21.5
BART FiD* (Shuster et al., 2021)	0	25.4	-	21.1
KnowGPT (Zhao et al., 2020b)	12.5	7.4	25.4	20.5
K2R* (Adolphs et al., 2021)	11.8	8.2	-	19.9
DIALI* (Li et al., 2022)	8.2	-	35.5	-
Graph* (Li et al., 2022)	15.4	-	30.8	-
GenKS	2.9	8.8	36.6	22.9

Table 4: Inference time (minutes) on one GPU on WoW unseen test set. The values of model with * are estimated based on the model size and input/output length. KG and RG denote inference time for knowledge selection and response generation stage, respectively.

(knowledge used by ground-truth response). The results (e.g., KF1=74) suggest that GENKS can effectively locate and incorporate the corresponding knowledge into the responses following the guidance of the identifier.

5.5 Human evaluation

Table 3 shows the human-evaluating results. Results show that GENKS consistently outperforms baselines on all datasets. The Fleiss’ kappa value is above 0.60, indicating substantial agreement among the annotators. GENKS outperforms KnowledGPT by about 0.02 and DukeNet by about 0.20 in terms of response generation evaluation metrics (i.e., *Fluency* and *Context Coherence*). Moreover, for the *Knowledge Relevance*, the annotators agree that GENKS is capable of selecting knowledge that is more relevant to the dialogue and generating more informative responses than baselines. The *Factuality* results show that by explicitly identifying the knowledge snippet used in response, GENKS can reduce the hallucination of response generation.

Methods	WoW Seen			WoW Unseen		
	F1	KF1	B4	F1	KF1	B4
GenKS	22.9	29.5	4.5	22.7	28.1	4.6
GenKS-2	22.4	29.3	4.2	22.2	27.6	4.2
GenKS (5 Snippets)	22.3	27.6	4.2	21.8	25.5	4.1
GenKS (3 Snippets)	21.1	29.3	3.2	20.0	20.9	2.9
GenKS (128 Tokens)	21.5	25.6	3.5	20.7	22.9	3.4
GenKS (64 Tokens)	20.7	23.3	3.0	20.1	20.6	2.9

Table 5: Analytical experiment results on WoW. The first group compares GENKS and its variant GENKS, which selects two snippets instead of one. The second group includes the results of GENKS with different maximum number of snippets inputs or maximum input tokens.

5.6 Efficiency evaluation

To evaluate the efficiency of GENKS, we compare the model with previous end-to-end models and pipeline models. The results listed in Table 4 show that GENKS is more efficient than previous pipeline models. We infer that this phenomenon is because GENKS jointly models knowledge selection and response generation, avoiding repeated encoding of dialogue history and knowledge. As a pipeline method, we also find that GENKS achieves comparable efficiency compared to end-to-end models like RAG, but benefits from explicit knowledge selection.

5.7 Analytical experiment

Multi-snippets selection GENKS select a single snippet following the experimental setup outlined in the baselines (Dinan et al., 2019), but it can also select multiple snippets by generating multiple identifiers. We test a variant of our GENKS model, GENKS-2, which selects two snippets by generating two identifiers consecutively. We compared its performance with the original GENKS on the WoW dataset. The results are listed in Table 5 group 1. GENKS-2 performs slightly worse than the original GenKS, likely because the WoW dataset only uses one snippet in response annotation and therefore does not benefit from using multiple snippets (Dinan et al., 2019). Nevertheless, the results suggest that the proposed generative knowledge selection approach has the ability to select multiple knowledge.

Hyper-parameter analysis We also conduct ablation experiments on the number of input snippets to the model and maximum input tokens. The results are listed in Table 5 group 2. We find that

Topic	Budweiser
User	I think Budweiser taste terrible. Have you ever had it?
Know 1	Produced in various breweries around the world, Budweiser is a filtered beer available in draft and packaged forms.
Res 1	Yes, I have. It is produced in various breweries around the world!
Know 2	Budweiser is an American-style pale lager produced by Anheuser-Busch, currently part of the transnational corporation Anheuser-Busch InBev.
Res 2	Yes, I have. I know that it is an American-style pale lager produced by Anheuser-Busch.

Table 6: Examples of GENKS outputs on the WoW.

reducing the number or length of knowledge reduces model effectiveness.

5.8 Case study

To better understand end-to-end baselines and our model, we provide an example in Table 6, which shows that GENKS appropriately changes its response prediction when providing different knowledge snippets⁵. Therefore, GENKS is more controllable and interpretable than end-to-end models, where the end-to-end system is a black box. We provide more case studies in Appendix B.

6 Conclusion

In this paper, we have proposed GENKS, a simple yet effective knowledge-grounded dialogue model. GENKS is a generative model, which learns to select knowledge snippets by generating their identifiers. Benefiting from the modeling of intra-knowledge interaction and dialogue-knowledge interaction, GENKS effectively addresses the challenges of *ambiguity* and *discourse structure*. Our experiments have shown that GENKS establishes a new state-of-the-art on three knowledge-grounded dialogue benchmarks. Notably, GENKS particularly excels at new topics and as the dialogue goes deeper. GENKS also outperforms SOTA end-to-end models. Hence, we believe GENKS reveals a new paradigm for knowledge selection in open-domain dialogue.

⁵Note that this example only aims to show the output of the model. In fact, according to <https://en.wikipedia.org/wiki/Budweiser>, Budweiser is also a famous lager from the Czech Republic, and the American Budweiser being sold and known as Bud through most of the European Union.

Limitations

The limitations of this work include the modular modeling of passage reranks, which reduces the efficiency of the approach. Besides, we only conduct human evaluation on one popular dataset, i.e., Wizard of Wikipedia. Furthermore, the effectiveness of GENKS is only verified in the English dataset. Research on other languages establishes a new challenge, especially for languages with limited knowledge and annotated data. In future work, we would like to explore more efficient passage rerank techniques on knowledge-grounded dialogues. We will also conduct human evaluation for more datasets. Besides, generative knowledge selection can be extended to future studies about conversational recommendation.

Ethics statement

The paper proposes a knowledge-grounded dialogue system to generate a response using external knowledge. The intended use of this system is to perform chit-chat with the user on topics such as books and movies. The system is developed using large pre-trained language models (i.e., BART), who are trained on large-scale web data known to contain biased or discriminatory content. The datasets (i.e., WoW, Holl-E, CMU_DoG) that we train on also include subjective knowledge (comments on movies) that may express the bias of the writers. Although the system is able to reduce the hallucination of response compared to end-to-end models, the outputs from our system may still contain non-factual information and should not be considered as advice for any critical decision-making.

Acknowledgements

This work was supported by the National Key R&D Program of China with grant No. 2020YFB1406704, the Natural Science Foundation of China (62272274, 62202271, 61902219, 61972234, 62072279, 62102234), the Natural Science Foundation of Shandong Province (ZR2021QF129), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

- Leonard Adolphs, Kurt Shuster, Jack Urbanek, Arthur D. Szlam, and Jason Weston. 2021. [Reason first, then respond: Modular generation for knowledge-infused dialogue](#). *ArXiv*, abs/2111.05204.
- Siqi Bao, H. He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhihua Wu, Zhen Guo, Hua Lu, Xinxian Huang, Xin Tian, Xinchao Xu, Yingzhan Lin, and Zhengyu Niu. 2021. [Plato-xl: Exploring the large-scale pre-training of dialogue generation](#). *ArXiv*, abs/2109.09519.
- Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. [Autoregressive entity retrieval](#). In *ICLR 2021*.
- Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. [Wizard of wikipedia: Knowledge-powered conversational agents](#). In *ICLR 2019*.
- Daniel De Freitas, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. [Towards a human-like open-domain chatbot](#). *ArXiv*, abs/2001.09977.
- Gautier Izacard and Edouard Grave. 2021. [Leveraging passage retrieval with generative models for open domain question answering](#). In *EACL 2021*.
- Byeongchang Kim, Jae Hyun Ahn, and Gunhee Kim. 2020. [Sequential latent knowledge selection for knowledge-grounded dialogue](#). In *ICLR 2020*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. [Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *ACL 2020*, pages 7871–7880.
- Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *NIPS 2020*, volume 33, pages 9459–9474.
- Lin-Xiao Li, Can Xu, Wei Wu, Yufan Zhao, Xueliang Zhao, and Chongyang Tao. 2020. [Zero-resource knowledge-grounded dialogue generation](#). In *NIPS 2020*, pages 8475–8485.
- Sha Li, Madhi Namazifar, Di Jin, Mohit Bansal, Heng Ji, Yang Liu, and Dilek Z. Hakkani-Tür. 2022. [Enhancing knowledge selection for grounded dialogues via document semantic graphs](#). In *NAACL 2022*.
- Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Q. Li, and Jie Zhou. 2019. [Incremental transformer with deliberation decoder for document grounded conversations](#). In *ACL 2019*, pages 12–21.
- Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. [Learning to select knowledge for response generation in dialog systems](#). In *IJCAI 2019*, pages 5081–5087.
- Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. 2018. [Knowledge diffusion for neural dialogue generation](#). In *ACL 2018*, pages 1489–1498.
- Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. 2019. [Knowledge aware conversation generation with explainable reasoning over augmented graphs](#). In *EMNLP 2019*, pages 1782–1792.
- Ehsan Lotfi, Maxime De Bruyn, Jeska Buhmann, and Walter Daelemans. 2021. [Teach me what to say and i will learn what to pick: Unsupervised knowledge selection through response generation with pretrained generative models](#). In *EMNLP 2021 | NLP4ConvAI*, pages 254–262.
- Chuan Meng, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tengxiao Xi, and M. de Rijke. 2021. [Initiative-aware self-supervised learning for knowledge-grounded conversations](#). In *SIGIR 2021*, page 522–532.
- Chuan Meng, Pengjie Ren, Zhumin Chen, Weiwei Sun, Zhaochun Ren, Zhaopeng Tu, and M. de Rijke. 2020. [Duketnet: A dual knowledge interaction network for knowledge-grounded conversation](#). In *SIGIR 2020*, page 1151–1160.
- Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021. [Joint passage ranking for diverse multi-answer retrieval](#). In *EMNLP 2021*, pages 6997–7008.
- Nikita Moghe, Siddharth Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. [Towards exploiting background knowledge for building conversation systems](#). In *EMNLP 2018*, pages 2322–2332.
- Rodrigo Nogueira and Kyunghyun Cho. 2019. [Passage re-ranking with bert](#). *ArXiv*, abs/1901.04085.
- Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy J. Lin. 2020. [Document ranking with a pre-trained sequence-to-sequence model](#). In *Findings of EMNLP 2020*, pages 708–718.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *ACL 2002*, pages 311–318.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter](#). *ArXiv*, abs/1910.01108.
- Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. [Retrieval augmentation reduces hallucination in conversation](#). In *Findings of EMNLP 2021*, pages 3784–3803.

- Tianxiang Sun, Xiangyang Liu, Xipeng Qiu, and Xuanjing Huang. 2022. [Paradigm shift in natural language processing](#). *Int. J. Autom. Comput.*, 19:169–183.
- Yi Tay, Vinh Quang Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. [Transformer memory as a differentiable search index](#). *ArXiv*, abs/2202.06991.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam M. Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, Yaguang Li, Hongrae Lee, Huaixiu Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, I. A. Krivokon, Willard James Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Hartz Søraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravindran Rajakumar, Alena Butryna, Matthew Lamm, V. O. Kuzmina, Joseph Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. [Lamda: Language models for dialog applications](#). *ArXiv*, abs/2201.08239.
- Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *NIPS 2017*, volume 30.
- Zequi Wu, Bo-Ru Lu, Hannaneh Hajishirzi, and Mari Ostendorf. 2021. [Dialki: Knowledge identification in conversational systems through dialogue-document contextualization](#). In *EMNLP 2021*.
- Chenxu Yang, Zheng Lin, JiangNan Li, Fandong Meng, Weiping Wang, Lan Wang, and Jie Zhou. 2022. [Take: Topic-shift aware knowledge selection for dialogue generation](#). In *COLING 2022*.
- Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, Nishit Shirish Keskar, and Caiming Xiong. 2022. [Modeling multi-hop question answering as single sequence prediction](#). In *ACL 2022*, pages 974–990.
- Haolan Zhan, Lei Shen, Hongshen Chen, and Hainan Zhang. 2021a. [Colv: A collaborative latent variable model for knowledge-grounded dialogue generation](#). In *EMNLP 2021*, pages 2250–2261.
- Haolan Zhan, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Yongjun Bao, and Yanyan Lan. 2021b. [Augmenting knowledge-grounded conversations with sequential knowledge transition](#). In *NAACL 2021*, pages 5621–5630.
- Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2020a. [Low-resource knowledge-grounded dialogue generation](#). In *ICLR 2020*.
- Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020b. [Knowledge-grounded dialogue generation with pre-trained language models](#). In *EMNLP 2020*, pages 3377–3390.
- Chujie Zheng, Yunbo Cao, Daxin Jiang, and Minlie Huang. 2020. [Difference-aware knowledge selection for knowledge-grounded conversation generation](#). In *Findings of EMNLP 2020*, pages 115–125.
- Kangyan Zhou, Shrimai Prabhumoye, and Alan W. Black. 2018. [A dataset for document grounded conversations](#). In *EMNLP 2018*, pages 708–713.

	WoW	Holl-E	CUM_DoG
Training size	18,430	7,228	3,373
Validation size	1,948	930	229
Test size	965 / 968	913	619
Number of topics	1,365	858	30
Avg. Turn per dialogue	9.0	10.1	22.2
Avg. Num of snippets	62.5	57.3	36.3
Avg. Num of passages	11.6	5.9	4.0

Table 7: Statistics of two experimental datasets, Wizard of Wikipedia (WoW), Holl-E, and CMU_DoG. The two numbers in WoW indicate the size of seen and unseen test set, respectively.

	F1	B4	KF1	RG1	RG2
BART (Lewis et al., 2020a)	34.7	22.3	29.1	38.0	27.9
CoLV (Zhan et al., 2021a)	-	20.3	-	32.0	25.8
MIKe (Meng et al., 2021)	32.1	21.1	-	38.0	25.2
Graph (Li et al., 2022)	-	-	-	42.5	34.4
GenKS	36.7	24.3	31.3	42.3	35.2

Table 8: Results on Holl-E in term of response quality. RG1 and RG2 denote ROUGE-1 and ROUGE-2 respectively. Best results are heighten with **bold**.

A Implementation details

We use gradient clipping with a maximum gradient norm of 0.1. We optimize the model for up to 5 epochs with a batch size of 16 on 4 3090 GPUs with 24G memory. We choose the model checkpoints by evaluating the metrics on the validation set for each epoch. During inference, the responses are decoded using a greedy search. We have tried some advanced decoding algorithms (e.g., nucleus sampling) and found no improvement. The training of the model can be completed within 5h, and the latency of the model inference for one example is within 0.1s. The passage rerank model gets Recall@1 of 75.5%, 76.5%, 61.0% for WoW test seen, WoW test unseen, and Holl-E, respectively.

	PPL	F1	Avg.	Ext.	Greedy
ITDD (Li et al., 2019)	26.0	10.4	0.748	0.390	0.587
DRD (Zhao et al., 2020a)	46.1	10.8	0.791	0.406	0.613
TMN (Dinan et al., 2019)	75.2	9.9	0.789	0.399	0.615
KGPT (Zhao et al., 2020b)	20.6	13.5	0.837	0.437	0.654
GenKS	16.1	14.1	0.847	0.442	0.668

Table 9: Results on CMU_DoG in term of response quality. The best results are highlighted with **bold**.

Topic	Nickelback
User	Do you like Nickelback?
...	
User	Yes, what more can you tell me about Nickelback?
System	Chad Kroeger is the leading vocalist of the group.
User	Amazing. What about the other group members
Passage	The band is composed of guitarist and lead vocalist <i>Chad Kroeger</i> , guitarist, keyboardist and backing vocalist <i>Ryan Peake</i> , bassist <i>Mike Kroeger</i> , and drummer <i>Daniel Adair</i> .
TMN	i think the song is a very popular song
DukeNet	chad kroeger is a great band
KGPT	the lead vocalist is chad kroeger. he is also the drummer.
GenKS	Chad Kroeger, Ryan Peake, Mike Kroeger and Daniel Adair.
Human	Ryan Peake is the keyboardist an Mike Kroeger is the bassist.

Table 10: Case study on the Wizard Test Unseen dataset. This table shows an example where both GENKS and baselines select the proper knowledge.

Topic	List of national parks of the United States
User	I love national parks dont you
...	
User	I live in Arkansas and love the Hot Springs National Park and have been there many times, really it is beautiful
DukeNet	i have heard the city is located in the ouachita mountains among the us
KGPT	i've been to the ouachita mountains, too! i've been to the ouachita mountains in the ouachita mountains.
GenKS	I've never been to Hot Springs, but I've always wanted to go to there.
Human	I've never been to that one! I bet its beautiful!

Table 11: Case study on the Wizard Test Unseen dataset. This table shows an example where both models select the wrong knowledge.

B Case study

To better understand baselines and our model, we present two examples in Table 10 and Table 11. Table 10 show example where both GENKS and baselines select out the proper knowledge (i.e., the knowledge snippet shown in green). We see that the response generated by GENKS is more appropriate to the dialogue context than baselines, while KnowledGPT's response does not answer User2's question and is also factually incorrect. In Table 11, we observed that although neither GENKS nor the baselines selected the label knowledge, the response generated by GENKS is still more natural and coherence. We also find that KnowledGPT is more colloquial than GENKS but has problems with hallucinations.

Evaluating the Tradeoff Between Abstractiveness and Factuality in Abstractive Summarization

Markus Dreyer¹ Mengwen Liu¹ Feng Nan¹ Sandeep Atluri¹ Sujith Ravi^{2*}
Amazon¹ SliceX² AI
{mddreyer, mengwliu, nanfen, satluri}@amazon.com
ravi.sujith@gmail.com

Abstract

Neural models for abstractive summarization tend to generate output that is fluent and well-formed but lacks semantic faithfulness, or factuality, with respect to the input documents. In this paper, we analyze the tradeoff between abstractiveness and factuality of generated summaries across multiple datasets and models, using extensive human evaluations of factuality. In our analysis, we visualize the rates of change in factuality as we gradually increase abstractiveness using a decoding constraint, and we observe that, while increased abstractiveness generally leads to a drop in factuality, the rate of factuality decay depends on factors such as the data that the system was trained on. We introduce two datasets with human factuality judgements; one containing 10.2k generated summaries with systematically varied degrees of abstractiveness; the other containing 4.2k summaries from five different summarization models. We propose new factuality metrics that adjust for the degree of abstractiveness, and we use them to compare the abstractiveness-adjusted factuality of previous summarization works, providing baselines for future work.¹

1 Introduction

Summarization is the task of generating a semantically faithful, well-formed and concise text representation of the input. Automatically generated summaries have traditionally been *extractive* (Luhn, 1958; Edmundson, 1969; Neto et al., 2002; Erkan and Radev, 2004; Wong et al., 2008), leading to issues with readability and coherence, as different extracted fragments may not fit well when taken out of their original contexts (Poibeau and Sag-gion, 2012). Researchers have also invested in methods for *abstractive* summarization, aiming to paraphrase the input documents' main points

Input: The National Zoo's giant panda cub made his debut Wednesday in a five-minute explosion of cuteness confined to a live stream because of the coronavirus pandemic. (...) The zoo is closed because of the pandemic and has not said when it will reopen. (...)

Summary 1: The National Zoo's giant panda cub made his debut Wednesday in a five-minute video live-streamed from the zoo's live stream because it's closed due to the pandemic. The zoo has not said when it will reopen.

Summary 2: The National Zoo's giant panda cub made its debut Wednesday in a five-minute video live-streamed from the zoo's live stream because it's closed due to the pandemic. It's not clear when the zoo will reopen.

Summary 3: The National Zoo is still closed due to the pandemic, but the National Zoo's giant panda cub has made its debut—and it was a pretty cute moment. The cub was born Wednesday, and the live-streamed birth lasted just five minutes.

More abstractive
↓

Figure 1: Three successively more abstractive summaries generated from the same input article, with MINT abstractiveness scores (Section 2.1) of 46.1%, 67.2%, 79.5%. Fragments extracted from the input are marked from red (longer fragments) to yellow (shorter fragments). The bottom summary has factual errors.

without borrowing their exact lexical expressions (Radev and McKeown, 1998; Saggion and Lapalme, 2002; Ganesan et al., 2010; Genest and Lapalme, 2012; Radford et al., 2019; Gehrmann et al., 2019; Lewis et al., 2019; Zhang et al., 2020). Abstractive summaries generated by today's neural models tend to be fluent and well-formed, but lack semantic faithfulness (Cao et al., 2017; Kryscinski et al., 2019). Observed rates of factual errors in abstractive summaries have ranged from 30% to over 75% (Cao et al., 2017; Maynez et al., 2020). The research community is developing automatic factuality metrics (Wang et al., 2020; Kryscinski et al., 2020; Goodrich et al., 2019; Goyal and Durrett, 2020; Ribeiro et al., 2022) and methods that attempt to increase factuality (Fan et al., 2018; Scialom et al., 2019; Zhang et al., 2019; Falke et al., 2020; Cao and Wang, 2021). However, the factuality problem of abstractive summaries cannot be well understood without considering the *degree* of abstractiveness of a given summary: Any summary is on a spectrum between *extractive* and

*Work conducted during his position at Amazon.

¹Code and data are available at <https://github.com/amazon-science/abstractive-factual-tradeoff>.

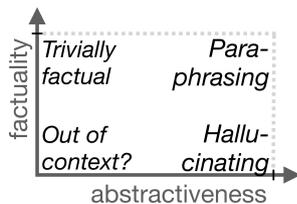


Figure 2: Four extremes at the abstractiveness-factuality spectrum.

paraphrasing, which results in summaries that are more *abstractive*, is harder and prone to semantic errors. As an example, Figure 1 shows part of a Washington Post article and three summaries with increasing abstractiveness, which we have generated using our abstractiveness constraints (Section 2.2). The first two summaries are correct, but the third, most abstractive, summary has factual errors, misinterpreting the input.

Few authors have discussed this connection explicitly. Lebanoff et al. (2019) observe that abstractive summaries consisting of concatenated extracted fragments tend to be more factual than those created by more complex fusion. Durmus et al. (2020) observe that models trained on the more *extractive* CNN/DM dataset (Hermann et al., 2015) create more factual summaries than models trained on the more *abstractive* XSum dataset (Narayan et al., 2018). We show that such models differ in factuality even when we bias them to generate summaries that have similar levels of abstractiveness. Our analysis (Section 4) situates summarization models on the spectrum outlined in Figure 2, where factual summaries range from “trivially factual” (extractive) to truly “paraphrasing” (abstractive). We make the following contributions:

1. We systematically explore the relationship of abstractiveness and factuality and show how factuality decays with increasing abstractiveness. We argue that factuality rates of different systems cannot be compared without taking their degrees of abstractiveness into account.
2. We introduce new factuality metrics that take abstractiveness into account and evaluate the abstractiveness-factuality tradeoff across various datasets and summarization models. We establish baselines that will allow others to demonstrate progress on mitigating the abstractiveness-factuality tradeoff.
3. We introduce a new dataset containing 10.2k summaries with systematically varied degrees

abstractive (See et al., 2017). Summaries that are extractive to a larger extent tend to be more factual since copying text from the input into the summary rarely introduces factual errors while the task of

of abstractiveness along with human factuality judgements, and a second dataset containing 4.2k summaries from five summarization models with their human factuality judgements.

2 Abstractiveness

2.1 Measuring Abstractiveness

In this paper, we wish to analyze the relationship of abstractiveness and factuality of generated summaries. We start by proposing a comprehensive abstractiveness metric. Abstractiveness measures the amount of rephrasing, i.e., the degree to which the words, phrases and sequences of the generated text have *not* been extracted from the corresponding input; a fully abstractive summary method expresses the main points of the input in its own words. To measure abstractiveness, most authors list the proportions of summary n -grams of varying lengths that are novel, i.e., do not occur in the corresponding inputs (See et al., 2017; Narayan et al., 2018; Gao et al., 2019). Grusky et al. (2018) proposed a new metric also based on contiguous overlapping text spans, *density*, measuring the average length of extracted fragments in a summary. Others have proposed metrics that take common *non-contiguous* subsequences into account, e.g., *perfect fusion_k* (Durmus et al., 2020) measures the percentage of summary sentences that assemble substrings from k source sentences in their original order.

Based on these previous works, we define a comprehensive abstractiveness metric that combines measures of contiguous and non-contiguous extractive summary fragments, making it sensitive to different kinds of abstractiveness and therefore suitable as a general abstractiveness metric. We define this metric as a ratio, in order to facilitate combining it with a factuality metric of the same $[0,1]$ range (Section 4). Let $\chi(\mathbf{x}, \mathbf{y}) = \text{hmean}(p_1, p_2, p_3, p_4, \text{lcsr})$ be a measure of *extractive* overlap between input \mathbf{x} and summary \mathbf{y} , using the harmonic mean of multiple component measures. Each p_n , short for $p_n(\mathbf{x}, \mathbf{y})$, is the n -gram precision of the n -grams in \mathbf{y} with respect to \mathbf{x} , i.e., the percentage of n -grams in \mathbf{y} that are extracted from \mathbf{x} .² Following common practice (Papineni et al., 2002), we use n -grams up to length four. We do not include density in $\chi(\mathbf{x}, \mathbf{y})$ as its range is unbounded. The measure lcsr (longest common sub-

²We smooth all n -gram counts (Chen and Cherry, 2014) to avoid undefined or zero harmonic mean values in highly abstractive summaries. See Appendix A for details.

x the supreme court reserved its verdict on a batch
of pleas which have raised questions
y the supreme court reserved its decision on a batch
of pleas that have raised questions

Figure 3: Example of input and highly extractive generated output. The color coding is the same as in Fig. 1.

sequence ratio), short for $\text{lcsr}(\mathbf{x}, \mathbf{y})$, is the length of the longest common subsequence (LCS) between \mathbf{x} and \mathbf{y} divided by the length of \mathbf{y} . lcsr , inspired by ROUGE-L (Lin, 2004), generalizes perfect fusion_k to consider *all* instances of non-contiguous overlaps between input and summary. Adding a measure of non-contiguous overlap is important as it detects overlaps that are long but broken up by minor changes, such as synonyms, as in the example in Figure 3. Finally, the MINT (Metric for lexical independence of generated text) abstractiveness measure is defined as $\text{MINT}(\mathbf{x}, \mathbf{y}) = 1 - \chi(\mathbf{x}, \mathbf{y})$. For a set of inputs and their summaries, we report the average MINT score. See Figure 1 for the MINT scores of three increasingly abstractive example summaries. In Section 5, we show that MINT scores correlate highly with density scores.

The described MINT score capitalizes on prior work to provide a comprehensive and unified metric for abstractiveness of conditionally generated text, combining measures of contiguous and non-contiguous overlap into a single percentage score. The implementation of MINT we provide will facilitate standardized comparisons of abstractiveness across different works.

2.2 Nonlinear Abstractiveness Constraints

We now introduce nonlinear abstractiveness constraints (NAC), which enable us to control the degree of abstractiveness at decoding time; it will allow us to use a trained summarization model to decode input multiple times while applying constraints to control the abstractiveness of the generated text output (e.g., see Figure 1). We will use this technique to analyze the impact of abstractiveness on factuality (Section 4).

Let $\mathcal{F}(\mathbf{x}, \mathbf{y})$ be the set of the longest extractive fragments in the decoding output \mathbf{y} with respect to the input \mathbf{x} . In Figure 1, such fragments are marked in color for each summary. We define a function $\lambda_h(|\mathbf{f}|)$ that assigns a discount probability to any extractive fragment $\mathbf{f} \in \mathcal{F}(\mathbf{x}, \mathbf{y})$:

$$\lambda_h(|\mathbf{f}|) = 2^{-|\mathbf{f}|^2/h^2} \quad (1)$$

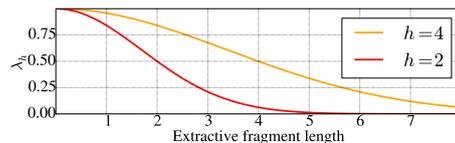


Figure 4: λ_h defines discounts for extractive fragments based on their lengths. Smaller h values lead to more abstractive summaries.

We configure this function³ with h , interpreted as the length of an extracted fragment for which $\lambda_h = 0.5$. Decreasing h results in a λ_h that discounts shorter extractive fragments more strongly, leading to increased abstractiveness (see Figure 4). Our discount penalty grows nonlinearly, affecting longer extractive fragments more strongly than multiple shorter ones with the same combined length. To see why we choose a **nonlinear penalty**, consider for example that extracting a 10-gram makes a summary more extractive than using ten words from the article separately, since an extracted 10-gram will be highly recognizable as stemming from the input. This nonlinearity is in contrast to Weber et al. (2018), which used a linear penalty to control the amount of copying in a pointer network.

In decoding, we search for the summary $\hat{\mathbf{y}}$ that maximizes the product of the summarization model probability, $p_M(\mathbf{y} | \mathbf{x})$, and the discount probabilities of the extractive fragments $\mathcal{F}(\mathbf{x}, \mathbf{y})$:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} p_M(\mathbf{y} | \mathbf{x}) \times \prod_{\mathbf{f} \in \mathcal{F}(\mathbf{x}, \mathbf{y})} \lambda_h(|\mathbf{f}|) \quad (2)$$

Beam Decoding. The model probability $p_M(\mathbf{x}, \mathbf{y})$ in neural text generation models (Section 5.1.1) decomposes for token-by-token decoding as $\prod_{i=1}^{|\mathbf{y}|} p_M(y_i | \mathbf{x}, y_1, \dots, y_{i-1})$. Similarly, we decompose the application of the λ_h function for any partial or completed extractive fragment \mathbf{f} :

$$\lambda_h(|\mathbf{f}|) = \prod_{l=1}^{|\mathbf{f}|} \frac{\lambda_h(l)}{\lambda_h(l-1)} \quad (3)$$

Therefore, to successively apply λ_h at each output position i in beam decoding, each candidate for token y_i is evaluated to check whether choosing it would extend an extractive fragment to length l . If so, its model probability $p_M(y_i | \dots)$ is multiplied with $\lambda_h(l)$ and the $\lambda_h(l-1)$ that was applied to the previous token y_{i-1} is divided out. We are not

³Additionally, the exponent used in $|\mathbf{f}|^2$ and h^2 could be configured, but we keep it at 2 in our experiments. A larger exponent would result in a steeper descent around h .



Figure 5: Screenshot (part) of a Mechanical Turk task (HIT) to judge the factuality of a summary sentence (in blue) with respect to news articles. Darker green article sentences are more similar to the blue summary sentence. The full task showed sentences from two more articles in the same cluster; from the Multi-News test set.

aiming to control the length of the generated output; instead we penalize the model in proportion to the length of any phrases it would extract from the input and encourage it to use novel phrases instead. **Extraction Rewards.** We can choose to apply an extraction *reward*, rather than a penalty, by using the inverse $1/\lambda_h$; smaller values of h then result in summaries that are more *extractive*.

3 Factuality

We now describe metrics for factuality, before we can describe the relationship between abstractive-ness and factuality (Section 4). By factuality of a summary y , we mean factual consistency with the input x , rather than objective factuality or universal truth. Measuring factuality automatically is an active area of research (Gabriel et al., 2020). Factuality is most naturally measured by human annotators; we describe our setup for human factuality annotation first, then move to automatic metrics.

3.1 Human-annotated Factuality

We use Amazon’s Mechanical Turk (AMT) to measure the factuality of automatically generated summaries with human annotators. These annotators are untrained, so we use multiple mitigation strategies to obtain high-quality judgements. We simplify the task: To avoid overwhelming annotators with long text, we select a single sentence per summary and ask the annotators if it is factually consistent with the shown article(s). The other sentences of the summary are given as well for context, shown in gray (see Figure 5). The article(s) are shortened to show a total of 9 sentences that were determined to be semantically most similar to the selected summary sentence;⁴ the remaining article parts are replaced by “...”. The summary sentence is selected at random in proportion to its length.

⁴We measure cosine similarity of sentence encodings computed by the Universal Sentence Encoder (Cer et al., 2018).

For each summary, we get judgements only for the randomly selected sentence. Aggregated over a set of summaries, we measure the average chance of any randomly selected summary sentence to be factual. We have verified high correlation of these factuality rates with the factuality rates obtained through professional annotators who judged complete summaries with respect to the full articles (see Appendix C).

We provide detailed task instructions, including examples for intrinsic and extrinsic factual errors (Maynez et al., 2020). We require that potential annotators pass a custom qualification test of finding factuality errors. Only workers with at least 100 completed tasks on AMT with an acceptance rate of 95%+ may take the test; 15% of those pass, enabling them to work on our tasks. We use three annotators per task and use MACE (Hovy et al., 2013) to aggregate annotations and recover the most likely binary factuality judgement per summary. We add summaries for which we know the correct factuality annotation and repeatedly check the annotators’ accuracy on those summaries while they are annotating; all answers from annotators who fall below a threshold are replaced by answers from additional annotators. Appendix C describes more details on our setup and fair compensation.

For any set of generated summaries, we create the AMT tasks, get an aggregate binary judgement per summary based on the multiple answers as described, and report the mean of all human binary summary factuality judgements; we call this score **FACTH** (Table 1). We collect human factuality judgements for 10.2k BART summaries with varying degrees of abstractive-ness, and for 4.2k summaries from five different summarization models.

Released Datasets. We release these human judgements as datasets called **CONSTRAINTSFACT** (Section 5.1) and **MODELSFACT** (Section 5.2). Previous datasets with human factuality judgements

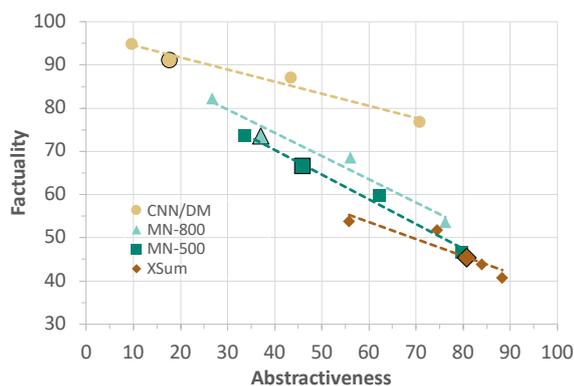


Figure 6: Human factuality judgements (FACTH) for different degrees of abtractiveness (MINT). Each color represents a BART model trained on a particular dataset, decoded with varying decoding constraints (Sec. 2.2); large outlined symbols mean no constraints.

(Wang et al., 2020; Kryscinski et al., 2020; Maynez et al., 2020; Pagnoni et al., 2021) are substantially smaller, with under 5k summaries each, and our CONSTRAINTSFACT dataset is the first that evaluates the factuality of summaries with systematically varied degrees of abtractiveness.

3.2 Automatically Measured Factuality

Measuring factuality *automatically* is an active research area; Pagnoni et al. (2021) gives an overview over recent metrics and compares their correlations to human judgements, where DAE (Goyal and Durrett, 2020, 2021) and FactCC (Kryscinski et al., 2020) perform well. DAE is an entailment model that classifies the factuality of the dependency arcs in the summary, resulting in fine-grained judgements at the subsentence level. FactCC is a BERT-based binary classifier trained on pairs of input and output sentences, where the output sentence is annotated as either factual or non-factual.

4 Abtractiveness-Factuality Tradeoff

The metrics for factuality and abtractiveness along with the abtractiveness constraints allow us to systematically explore the relationship between abtractiveness and factuality. We can control abtractiveness and observe the effect on factuality, i.e., we can vary the amount of lexical overlap between input and generated summary and observe the extent to which the summary preserves the input semantics.

Factuality Trend Lines. To explore this relationship, we train summarization models on different

datasets. For any trained summarization model, we decode the test set multiple times with different h values for λ_h (Equation 1), resulting in sets of summaries with varying degrees abtractiveness. For each of these test set decodings, we measure abtractiveness using MINT and the corresponding factuality using human annotations, unless otherwise noted. This results in a series of (abtractiveness, factuality) points for any trained summarization model, which can be plotted, along with a linear trend line. Figure 6 shows such a plot; Section 5.1.2 discusses its details.

F@50 Score. Given each trend line, we can read off the factuality at 50% abtractiveness, an intuitively interpretable metric, which we call F@50; it provides a comparison of the factuality of different models with a fixed degree of abtractiveness.

MINT-adjusted Factuality Scores. We characterize the tradeoff on any single decoding output using a weighted average between factuality and abtractiveness, $(\phi F + A)/(\phi + 1)$. To measure abtractiveness A , we use MINT; to measure factuality F , we use human-measured factuality or an automatic metric with $[0,1]$ range like DAE or FactCC, resulting in abtractiveness-adjusted factuality metrics $\mu\mathbf{FactH}$, $\mu\mathbf{DAE}$, $\mu\mathbf{FactCC}$, etc.

We give factuality a higher weight, since factual semantic representation of the input is a fundamental requirement for summarization and low factuality can have negative societal impact (Zellers et al., 2019), while abtractiveness is a desirable stylistic property. When two measures are combined into one comprehensive evaluation metric there is no *a priori* correct mixture weight; we follow common practice to give the more important measure twice the weight (Kohonen et al., 2010; Li et al., 2020; Preuß et al., 2021; Opitz and Frank, 2021) and set ϕ to 2. By this definition, a system whose factuality decreases by x units, as compared to another system, must make up for the lost factuality by $2x$ units in abtractiveness to get the same score. When two systems have the same factuality, the score prefers the one with higher abtractiveness.

4.1 Discussion

The abtractiveness-adjusted factuality metrics address the issue that in the past, factuality rates of different systems have been compared without taking abtractiveness into account. However, if one system has a higher factuality rate than another, it may

	λ	MINT	FACTH	μ FACTH	F@50
CNN/DM	$1/\lambda_2$	9.7	94.8	66.5	84.4
	none	17.6	91.2	66.7	
	λ_4	43.5	87.0	72.5	
	λ_2	70.8	76.7	74.7	
MN-800	$1/\lambda_2$	26.8	82.2	63.7	68.9
	none	37.0	73.5	61.3	
	λ_4	56.1	68.5	64.4	
	λ_2	76.2	53.5	61.1	
MN-500	$1/\lambda_2$	33.6	73.5	60.2	64.4
	none	45.9	66.5	59.6	
	λ_4	62.3	59.7	60.6	
	λ_2	79.7	46.5	57.6	
XSum	$1/\lambda_1$	55.8	53.7	54.4	56.7
	$1/\lambda_2$	74.5	51.7	59.3	
	none	80.8	45.3	57.2	
	λ_4	84.0	43.7	57.1	
	λ_2	88.3	40.7	56.5	

Table 1: Abstractiveness and factuality on 600 test samples per setting. The 17 MINT and FACTH numbers are as shown in Figure 6; we add μ FACTH and F@50.

have achieved this by copying phrases from the input into the summary with minimal rephrasing, i.e., by having a low degree of abstractiveness. Such a system may produce high-quality summaries, but their factuality rate cannot directly be compared to the factuality numbers of more abstractive summarization systems. Summarization methods that are highly factual and abstractive are able to rephrase the input with few factual errors; when we compare the factuality of abstractive summarizers we must control for the amount of such rephrasing. The abstractiveness-adjusted factuality metrics we propose enable us to compare the factuality of abstractive summarization models even when they perform different amounts of rephrasings.

As an analogy, consider precision and recall. High precision can be trivially achieved with low recall, just as high factuality can be achieved with low abstractiveness. Therefore when comparing the precision of different retrieval systems, their recall numbers are taken into account by using the F-score.⁵ Similarly, we argue that factuality comparisons must take abstractiveness into account.

Dataset	Train	Valid	Test
CNN/DM	287,227	13,368	11,490
XSum	204,045	11,332	11,334
Multi-News	44,972	5,622	5,622

Table 2: Train/valid/test split on public datasets.

5 Experiments

5.1 Comparison Across Datasets Using NAC

Datasets. We use CNN/DM (Hermann et al., 2015), XSum (Narayan et al., 2018), and Multi-News (Fabbri et al., 2019), all of which contain English-only text. CNN/DM contains news articles from CNN and DailyMail paired with bullet point summaries. XSum contains articles from BBC News, using each article’s first sentence as summary.⁶ In Multi-News, each summary is written by a professional editor and paired with a cluster of news articles. For all three public datasets, we use the provided training/validation/test split. The sizes of the three datasets are listed in Table 2. From each of the three datasets, we use 600 samples to compare human and automatic factuality judgements.⁷

5.1.1 Setup

We use the BART (Lewis et al., 2020) sequence-to-sequence model, which was pretrained on 160GB of text and gives competitive results on CNN/DM and XSum. Our models use the provided model checkpoints for the CNN/DM and the XSum datasets as well as the recommended decoding settings. For Multi-News (MN), we train a model on the training set, starting from the `bart.large` pretrained model.⁸ For Multi-News, we truncate the input documents per cluster so that their combined length does not exceed N words, following Fabbri et al. (2019). We train models with $N = 800$ and $N = 500$, called MN-800 and MN-500, respectively. We measure the MINT scores for the reference summaries in these datasets; these can be compared to the MINT scores obtained in

⁵In our case, we use a weighted arithmetic mean instead because an F score would steeply decline to zero as abstractiveness goes to zero, which is undesirable for output whose factuality is high.

⁶Following Wang et al. (2020), we reinsert the first sentences whenever we measure factuality of XSum summaries on AMT or with automatic metrics.

⁷For Multi-News and XSum, we take the first 600 samples per test set. For CNN/DM, we take the first 300 and the last 300 test samples, from CNN and Daily Mail, respectively.

⁸We train for five epochs (learning rate: $2e-5$) and limit output to 50 to 300 tokens.

decoding (Section 5.1.2). The test set references for MN-500 have a MINT score of 78.2%, compared to 72.8% for MN-800. MINT is higher for MN-500 since the shorter truncation removes article content that could otherwise overlap with the summaries. The MINT scores for the CNN/DM and XSum references are 59.6% and 87.8%, respectively; XSum is the most abstractive dataset.

5.1.2 Results

We use each of the four BART models to decode its respective test set multiple times, with varying abstractiveness constraints, resulting in 17 outputs. For each one, we obtain human factuality judgements on the corresponding 600 samples, resulting in 17 x 600 human factuality judgements – our CONSTRAINTSFACT dataset –, which we aggregate into 17 mean FACTH scores; we also compute the corresponding 17 MINT scores. Figure 6 plots the resulting abstractiveness and human-measured factuality for each of the four models, thereby providing a visual representation of the abstractiveness-factuality tradeoff for these models. Table 1 shows the same 17 MINT and FACTH values, along with μ FACTH and F@50 scores.

The lower right of Figure 6 shows five lozenges (◆). The larger one represents the decoding with our **XSum**-trained model using default settings; the other four red points represent decodings under the same model, but with different abstractiveness constraints that result in more *extractive* ($1/\lambda_h$) or more *abstractive* (λ_h) summaries (Section 2.2). The five red points are associated with a dashed linear trend line. Compared to the other points in the figure, abstractiveness is high and factuality low – the model tends to paraphrase its input, often incorrectly. It took a strong extractive reward ($1/\lambda_1$), which we did not use for the models trained on other datasets, to bias this model toward lower abstractiveness and higher factuality.

For the **Multi-News** models, four decodings using MN-500 are shown as squares (■), decodings under MN-800 as triangles (▲). The MN-800 model is more factual across the abstractiveness spectrum. This can be explained by the fact that for MN-500, larger parts of the input are truncated (Section 5.1.1) that the untruncated reference summary in training may still refer to; the MN-500 model learns to hallucinate more.

The four decodings for **CNN/DM** are shown as bullets (●). Its model output without abstractiveness constraint (large bullet) is the most extractive;

the extraction reward to its left (using $1/\lambda_2$) cannot make it much more extractive; however, there is room to the right, and the abstraction rewards (λ_4 and λ_2) move its abstractiveness far into the abstractiveness level of Multi-News and XSum.

F@50 Scores. One of the main takeaways of this study is that different systems can have different factuality rates at the same level of abstractiveness. Previous authors have observed that XSum summaries are highly abstractive and less factual, and that CNN/DM summaries are at the opposite side of that spectrum. We confirm this; however, we add that we can bias the XSum model to create less abstractive summaries and the CNN/DM model to create more abstractive models, so that **their abstractiveness becomes comparable**, and the factuality rates still differ considerably: Based on the trend line, the F@50 score of the XSum model is 56.7%, while the CNN/DM model’s F@50 is 84.4%. MN-800 and MN-500 lie in the middle.

μ FACTH Scores. The μ FACTH scores adjust FACTH for abstractiveness. They penalize the CNN/DM model for its low abstractiveness and reward the XSum model for its high abstractiveness, bringing them closer together, compared to their more divergent FACTH scores. The μ FACTH scores for MN-800 and MN-500 are also close (59.6% versus 61.3% for λ =none), as MN-800 is more factual but also less abstractive.

Summary Quality and Abstractiveness. Table 3 lists ROUGE-L scores for the different decodings, along with abstractiveness metrics, measured on the *full* test sets. ROUGE scores aim to measure summary quality by comparing the generated summaries with the reference summaries, while abstractiveness metrics measure overlap between the generated summaries and the input. Decodings without abstractiveness constraints replicate previous works’ ROUGE scores (Lewis et al., 2020; Fabri et al., 2019) (Appendix H). The λ_4 constraint can **dramatically increase abstractiveness while leaving ROUGE scores virtually unchanged**. We also conduct a human evaluation of informativeness and coherence, comparing unconstrained summaries with summaries generated with the λ_4 decoding constraint; the unconstrained decoding is preferred for XSum but the constrained decoding is preferred for CNN/DM, and results are mixed for Multi-News, see Appendix D. The density scores (Grusky et al., 2018) in the table have high correla-

	λ	RL	MINT	p3	p4	lcsr	density
CNN/DM	$1/\lambda_2$	37.9	9.0	89.0	84.7	93.1	28.9
	none	41.0	16.8	79.5	72.1	89.4	15.4
	λ_4	41.5	43.7	50.0	35.1	77.8	4.6
	λ_2	39.3	70.3	26.4	12.6	67.4	2.2
MN-800	$1/\lambda_2$	44.8	26.6	71.1	64.1	69.5	20.7
	none	45.8	37.1	58.9	50.1	63.3	13.4
	λ_4	45.8	56.3	38.7	27.0	51.9	4.3
	λ_2	44.0	76.4	20.7	10.4	41.6	2.0
MN-500	$1/\lambda_2$	44.6	34.1	63.7	56.4	61.0	17.6
	none	45.5	45.9	50.2	41.4	54.2	10.6
	λ_4	45.1	62.2	33.4	22.7	44.8	3.6
	λ_2	43.3	79.8	17.8	8.8	35.9	1.8
XSum	$1/\lambda_1$	30.8	53.8	41.7	32.3	66.9	5.8
	$1/\lambda_2$	36.0	73.9	23.0	14.1	57.7	3.0
	none	36.8	80.2	17.6	9.2	54.5	2.4
	λ_4	36.8	83.6	14.6	6.6	52.8	2.2
	λ_2	36.3	88.1	10.8	4.1	49.8	1.9

Table 3: Impact of λ on ROUGE-L F_1 (RL) and abstractiveness metrics on the full test sets. p3, p4, lcsr are component scores in MINT (Sec. 2.1), density is average length of extracted fragments (Grusky et al., 2018). ROUGE measures overlap with reference summaries, abstractiveness metrics measure input overlap.

tion with the MINT scores.

5.2 Comparison Across Different Models

We also compare the abstractiveness-factuality tradeoffs of summarization models from the literature. We obtain outputs of four summarization models other than BART: BERTSUM (Liu and Lapata, 2019) is a transformer model in which only the encoder is pretrained; PGCONV (See et al., 2017) is a pointer-generator network; BOTTOMUP (Gehrmann et al., 2018) and ABSRL (Chen and Bansal, 2018) select source fragments to constrain an abstractive generation model. We obtain human factuality judgements of the five model outputs on 600 samples of CNN/DM and XSum, respectively, and release this as our MODELSFACT dataset; we apply automatic metrics (e.g., DAE) as well as our abstractiveness-adjusted variants (e.g., μ DAE) to the *full* test sets. Table 4 shows the results. For CNN/DM, we find that the highly extractive model PGCONV receives the highest automatic and human factuality scores, while the abstractiveness-adjusted variants favor BART or ABSRL, whose outputs represent better tradeoffs between abstractiveness and factuality. On XSum, BART’s output is considerably more factual than BERTSUM’s across all factuality metrics, while BART has only slightly lower abstractiveness; as a result, BART is

	Model	MINT	μ FACTH	μ DAE	μ FactCC			
CNN/DM	BART	16.8	66.4	91.2	67.4	92.6	56.2	75.9
	BERTSUM	14.1	64.7	90.0	57.8	79.6	57.0	78.5
	PGCONV	5.5	63.5	92.5	64.0	93.3	62.3	90.7
	BOTTOMUP	17.2	50.6	67.3	55.0	73.9	54.3	72.9
	ABSRL	18.9	60.6	81.5	62.3	84.0	64.1	86.8
XSum	BART	80.2	56.9	45.3	67.3	60.8	53.9	40.8
	BERTSUM	82.8	52.1	36.8	61.5	50.8	50.8	34.8

Table 4: Abstractiveness (MINT) and factuality of different models. For each factuality metric, we first list its MINT-adjusted variant in green. Example: BART’s μ FACTH is 66.4, while the unadjusted FACTH is 91.2. All numbers are percentage scores $\in [0,100]$.

also favored by all MINT-adjusted factuality metrics. Detailed results including additional factuality metrics are described in Appendix G.

The MINT-adjusted variants of factuality metrics put factuality rates into perspective. We encourage authors who compare factuality rates across summarization models to also compare MINT-adjusted variants (e.g., μ DAE), to account for differing levels of abstractiveness.

6 Related Work

Abstractiveness-Factuality Tradeoff: Durmus et al. (2020) observe that abstractiveness at test time depends on the abstractiveness of the training data and that highly abstractive summaries tend to be less factual. We control for abstractiveness and see that factuality rates between different systems can vary widely at the *same* abstractiveness levels. Recently, Ladhak et al. (2022) present an alternative framework to evaluate the faithfulness-extractiveness tradeoff, requiring training multiple models on subsets of the training data to measure the tradeoff, while we use constraints to analyze tradeoffs that a single model makes. **Increasing Abstractiveness:** Kryściński et al. (2018) use policy gradient with a novelty reward to encourage abstraction in a pointer-generator (PG) (Gulcehre et al., 2016; See et al., 2017). Weber et al. (2018) penalize copying tokens during PG decoding. Our constraints apply to general sequence-to-sequence models and include nonlinear penalties. Song et al. (2020) control copying in training abstractive summarization models by masking the summary tokens with different probabilities, depending on whether they are seen in the input document or not. In contrast, our technique does not require retraining to

obtain varying degrees of abstractiveness.

7 Conclusions

We presented new metrics and datasets for evaluating the relationship of abstractiveness and factuality. As part of our analysis, we presented abstractiveness constraints, which can bias a summarization model to increase or decrease the level of abstractiveness while generating summaries, using nonlinear penalties or rewards based on the length of summary fragments extracted from the source. Through automatic and human factuality evaluations, including 10.2k human factuality judgements of summaries with systematically varied abstractiveness, we shed light on how abstractiveness interacts with factuality, across multiple datasets and models. We proposed new metrics to measure the tradeoff, including F@50 and MINT-adjusted factuality rates, such as μ DAE and μ FactCC, and we established baselines for future research.

Limitations

The abstractiveness constraints we have presented can be used to increase or decrease the abstractiveness of the generated text. Dedicated code is needed to integrate such constraints into a decoder. The constraints are needed to obtain trend lines as in Figure 6, as well as the F@50 score. However, the MINT-adjusted factuality scores, such as μ FactH, μ DAE or μ FactCC can be computed for any summarization system, without the need for implementing abstractiveness constraints, as we have done in Section 5.2.

Ethical Considerations

We have analyzed the factuality of generated text in relation to the abstractiveness of the source texts; we have also proposed new metrics that let researchers compare the factuality of different generative models. As such, we consider our work a contribution toward text generation methods that make fewer factual mistakes and become therefore more reliable and responsible. However, any advance in text generation methods can be used by bad actors to cheaply generate misleading or harmful texts.

We hired annotators on the Mechanical Turk platform to judge machine-generated summaries. Our first ethical consideration with respect to this data collection is fair and prompt pay for the work of the annotators. We describe in Appendix C that

we paid all human subjects a fair average pay of \$12.50 USD per hour, based on observed median time spent per HIT. As described (Section 3.1), we automatically approved the annotators' work promptly and paid bonuses as appropriate. The annotators' privacy and confidentiality were respected at all times.

References

- Shuyang Cao and Lu Wang. 2021. [CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. [Faithful to the original: Fact aware neural abstractive summarization](#). *CoRR*, abs/1711.04434.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. [Universal Sentence Encoder](#). *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*, pages 169–174.
- Boxing Chen and Colin Cherry. 2014. [A systematic comparison of smoothing techniques for sentence-level BLEU](#). In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Yen-Chun Chen and Mohit Bansal. 2018. [Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting](#). In *Proc. of ACL*.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. [Supervised learning of universal sentence representations from natural language inference data](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.
- Esin Durmus, He He, and Mona Diab. 2020. [Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization](#). In *Association for Computational Linguistics (ACL)*.
- H. P. Edmundson. 1969. New methods in automatic extracting. *J. ACM*, 16:264–285.
- Günes Erkan and Dragomir R. Radev. 2004. [Lexrank: Graph-based lexical centrality as salience in text summarization](#). *J. Artif. Int. Res.*, 22(1):457–479.

- Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. [Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.
- Tobias Falke, Leonardo F.R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2020. [Ranking generated summaries by correctness: An interesting but challenging application for natural language inference](#). In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 2214–2220. Association for Computational Linguistics (ACL).
- Lisa Fan, Dong Yu, and Lu Wang. 2018. [Robust Neural Abstractive Summarization Systems and Evaluation against Adversarial Information](#). In *NIPS Interpretability and Robustness for Audio, Speech and Language Workshop*.
- Saadia Gabriel, Asli Celikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2020. [Go Figure! A Meta Evaluation of Factuality in Summarization](#). Technical report.
- Kavita A. Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. [Opinosis: A graph based approach to abstractive summarization of highly redundant opinions](#). In *International Conference on Computational Linguistics*.
- Shen Gao, Xiuying Chen, Piji Li, Zhangming Chan, Dongyan Zhao, and Rui Yan. 2019. [How to write summaries with patterns? learning towards abstractive summarization through prototype editing](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3741–3751, Hong Kong, China. Association for Computational Linguistics.
- Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. [Bottom-Up Abstractive Summarization](#). In *Proc. of EMNLP*.
- Sebastian Gehrmann, Zachary Ziegler, and Alexander Rush. 2019. [Generating abstractive summaries with finetuned language models](#). In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 516–522, Tokyo, Japan. Association for Computational Linguistics.
- Pierre-Etienne Genest and Guy Lapalme. 2012. Fully abstractive approach to guided summarization. In *Annual Meeting of the Association for Computational Linguistics*.
- Ben Goodrich, Vinay Rao, Peter J Liu Mohammad Saleh, Google Brain, Peter J Liu, and Mohammad Saleh. 2019. [Assessing The Factual Accuracy of Generated Text](#). In *International Conference on Knowledge Discovery and Data Mining (KDD)*.
- Tanya Goyal and Greg Durrett. 2020. [Evaluating factuality in generation with dependency-level entailment](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.
- Tanya Goyal and Greg Durrett. 2021. [Annotating and modeling fine-grained factuality in summarization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.
- Max Grusky, Mor Naaman, and Yoav Artzi. 2018. [Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.
- Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. [Pointing the unknown words](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149.
- Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. [Teaching machines to read and comprehend](#). In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1693–1701.
- Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. [Learning whom to trust with MACE](#). In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1120–1130, Atlanta, Georgia. Association for Computational Linguistics.
- Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. [Semi-supervised learning of concatenative morphology](#). In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86, Uppsala, Sweden. Association for Computational Linguistics.
- Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. [Neural text summarization: A critical evaluation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

- Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. [Evaluating the factual consistency of abstractive text summarization](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346.
- Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. 2018. [Improving abstraction in text summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817, Brussels, Belgium. Association for Computational Linguistics.
- Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2022. [Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization](#). In *Proc. of ACL*.
- Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. [Analyzing sentence fusion in abstractive summarization](#). In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 104–110, Hong Kong, China. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Xiangci Li, Hairong Liu, and Liang Huang. 2020. [Context-aware stand-alone neural spelling correction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 407–414, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yang Liu and Mirella Lapata. 2019. [Text Summarization with Pretrained Encoders](#). In *Proc. of EMNLP*.
- Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2:159–165.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.
- Joel Larocca Neto, Alex Alves Freitas, and Celso A. A. Kaestner. 2002. Automatic text summarization using a machine learning approach. In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence, SBIA ’02*, page 205–215, Berlin, Heidelberg. Springer-Verlag.
- Juri Opitz and Anette Frank. 2021. [Towards a decomposable metric for explainable evaluation of text generation from AMR](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1504–1518, Online. Association for Computational Linguistics.
- Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. [Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Thierry Poibeau and Horacio Saggion. 2012. Automatic Text Summarization: Past, Present and Future. In *Multi-source, Multilingual Information Extraction and Summarization*, pages 3–13.
- Svenja Preuß, Luna Pia Bley, Tabea Bayha, Vivien Dehne, Alessa Jordan, Sophie Reimann, Fina Roberto, Josephine Romy Zahm, Hanna Siewerts, Dirk Labudde, and Michael Spranger. 2021. [Automatically identifying online grooming chats using CNN-based feature extraction](#). In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 137–146, Düsseldorf, Germany. KONVENS 2021 Organizers.
- Dragomir R. Radev and Kathleen R. McKeown. 1998. [Generating natural language summaries from multiple on-line sources](#). *Computational Linguistics*, 24(3):469–500.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

- Leonardo F. R. Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. 2022. [FactGraph: Evaluating factuality in summarization with semantic graph representations](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3238–3253, Seattle, United States. Association for Computational Linguistics.
- Horacio Saggion and Guy Lapalme. 2002. Generating indicative-informative summaries with sumum. *Computational Linguistics*, 28:497–526.
- Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. [Answers unite! unsupervised metrics for reinforced summarization models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3246–3256, Hong Kong, China. Association for Computational Linguistics.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.
- Kaiqiang Song, Bingqing Wang, Zhe Feng, Ren Liu, and Fei Liu. 2020. Controlling the amount of verbatim copying in abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(05), pages 8902–8909.
- Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020.
- Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Kyunghyun Cho. 2018. Controlling decoding for more abstractive summaries with copy-based networks. *arXiv preprint arXiv:1803.07038*.
- Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. [Extractive summarization using supervised and semi-supervised learning](#). In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 985–992, Manchester, UK. Coling 2008 Organizing Committee.
- Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, F. Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *NeurIPS*.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. [PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.
- Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D. Manning, and Curtis P. Langlotz. 2019. [Optimizing the Factual Correctness of a Summary: A Study of Summarizing Radiology Reports](#).

A Measuring Abtractiveness with MINT

***N*-gram Overlap.** Each p_n , short for $p_n(x, y)$, is the n -gram precision of the n -grams in y with respect to x , i.e., the percentage of n -grams in y that are extracted from x .⁹ For highly abstractive outputs, higher-order n -gram precision can be zero, leading to an undefined or zero harmonic mean value. We prevent this by smoothing the n -gram counts from which n -gram precisions are calculated, such that each n -gram count is the average of itself and the smoothed $(n - 1)$ -gram count and the unsmoothed $(n + 1)$ -gram count. The smoothed 0-gram count is defined as the 1-gram count plus one. We chose this method for its simplicity and effectiveness; it is described as method 5 in [Chen and Cherry \(2014\)](#).

Harmonic Mean. We use the harmonic mean, in analogy to the definition of the F_1 score, as it is a mean function designed to aggregate ratios with different denominators.

For a completely extractive summary that extracts sentences in the original order, the MINT score is 0. The score increases as the order of the extractive fragments is changed with respect to the input, their lengths are decreased and new words and fragments are introduced that are not part of the input x . The use of the length-normalized LCS score (lcsr) is inspired by ROUGE-L; it is a useful addition to the n -gram precisions as it can detect the extraction of longer n -grams broken up by minor edits. As an example, consider the (x, y) pair shown in Figure 3. Only 4 of the 12 summary four-grams match the input, i.e., $p_4=33.3\%$, although very high overlap is apparent due to the fact that a 15-word fragment from the input was extracted with only the words “verdict” and “which” minimally changed by synonym substitution. The lcsr score reflects this and measures $12/15=80.0\%$ overlap. On the other hand, the n -gram precisions used in the MINT score are valuable in detecting textual overlaps that are not part of the longest common subsequence.

⁹MINT has elements of ROUGE ([Lin, 2004](#)) and BLEU ([Papineni et al., 2002](#)). We do not use the *modified* n -gram precisions, like BLEU does, because n -grams extracted multiple times from x should count as such every time.

B Details on the Abtractiveness Constraints

Log Space. We have described the abtractiveness constraints in probability space. In practice, we equivalently search for \hat{y} in log space using log probabilities and the log of λ_h defined in Equation 1. It can be shown that $\log \lambda_h(|f|) = \frac{-|f|^2}{(1.20112 \times h)^2}$.

C Details on Our Mechanical Turk Setup

We provide additional details on the strategies we use to obtain high-quality judgements on Amazon Mechanical Turk. We give detailed instructions to the annotators, with definitions and examples of different factual errors (see Figure 7). We also add a request to write a short explanation when a sentence is judged as not factual.

Tasks with Known Answers. We add a number of tasks with known answers, enabling us to estimate the accuracy of workers who work on multiple of these.

Automatic Quality Checks. Workers who complete the tasks too quickly, write no or very short explanation texts or have low accuracy on the tasks with known answers are automatically removed from our worker pool. Their answers are replaced with new answers.

Bonus. We use a bonus incentive structure. Every worker who passes the automatic quality checks receives a bonus at the end.

Check Against Professional Annotators. We have seven sets of 150 automatically generated summaries each, which we had previously sent to professional news editors to annotate factuality. Those annotators rated the complete summaries with respect to the complete inputs – no sentences were preselected to simplify the task. We re-annotated these summary-article pairs using our Mechanical Turk setup, and the resulting per-set factuality rates correlated highly ($r=.88$) with those previously obtained from the professional annotators ($p < .05$).

As a further quality check, we sent one set of 600 summaries to Mechanical Turk twice, several weeks apart. The two factuality rates obtained for that same set were close – 91.2% and 92.0%.

Instructions (Click to collapse)

Please evaluate whether the **blue sentence** from the summary is consistent with the information in the articles. Select **no** if the blue sentence is not consistent, i.e., its facts are not supported by the articles.

Select **no** in cases like these:

- The blue sentence **contradicts** information in the articles. The blue sentence might say "A fire broke out in Seattle", but an article says it broke out in Portland. Or the blue sentence might say "the Republicans won the election", but the articles indicate that the Democrats won instead.
- The blue sentence **adds** a fact that is not mentioned anywhere in the articles. For example, the blue sentence might say that "A fire broke out at 2am", but the articles don't mention the time when the fire broke out.

Meaning of the colors:

- Summary:** The gray sentences in the summary are displayed to give context only. Please evaluate the **blue sentence** only.
- Articles:** The sentences in the articles have green background color to help you find information more quickly. Article sentences with **darker green** background color are more related to the blue sentence. The least related sentence have been removed, indicated by three dots (...).

Please evaluate the **blue sentence** in the summary. (See instructions above.)

Summary:
 A North Carolina couple is suing the producers of Hgtv's love it or list it because they say the show turned their dream home into a Shoddily constructed one, the Raleigh news& observer reports. **Deena Murphy and Timothy Sullivan say they agreed to take part in the show under the guise of moving into a rental property with their teenage foster children, but the reality show's principals -- designer Hilary Farr, real estate agent David Visentin, and contractor Eric Eremita -- are "actors or television personalities playing a role for the camera," not people who "played more than a casual role in the actual renovation process," according to the lawsuit filed against big coat TV and contractor Aaron Fitz construction.** The lawsuit claims the couple were "victims of shoddy work and unfair trade practices" that left their floors, windows, and other parts of their home damaged. The couple says they gave \$140,000 to big coat for renovations, but were told the rest of the money was used to pay Fitz and other Subcontractors. " One of the things they're doing in this lawsuit is kind of blowing the secrecy off of reality TV," today legal's hosts say. big coat denies the couple's claims. " We believe that this claim is in no way supported by any of the facts of the case, and we will be defending ourselves vigorously in this matter," the company says in a statement, per the Huffington Post.

Article 1
 ...
 Deena Murphy and Timothy Sullivan are suing the production company for HGTV's "Love It or List It," claiming the hit show turned their dream home into a nightmare. The lawsuit against Big Coat TV and one of its contractors alleges the couple were "victims of shoddy work and unfair trade practices" that left their floors, windows and other parts of their home damaged.
 ...
 TODAY The program's hosts, David Visentin and Hilary Farr "One of the things they're doing in this lawsuit is kind of blowing the secrecy off of reality TV," said TODAY legal analyst Lisa Bloom.
 ...
 TODAY North Carolina couple Deena Murphy and Timothy Sullivan are suing the show's production company.
 ...

Article 3
 A North Carolina couple is suing the producers of Love It Or List It, saying the show left them with a house that was shoddily constructed. The Raleigh News & Observer says that Deena Murphy and Timothy Sullivan agreed to participate in the hit HGTV series under the guise that they were considering a move to a rental property with their teenage foster children. The problem, according to the suit against Big Coat TV and Aaron Fitz Construction, was that the show's principals--designer Hilary Farr, real estate agent David Visentin,

Figure 7: Instructions for the factuality annotation task on Amazon Mechanical Turk, as well as the summary and part of the article text shown to the worker.

Qualification Test. For all our evaluations on Mechanical Turk (see Section 3.1), we first set up a short qualification test that can be taken by any worker from a country whose main language is English, who has completed 100 or more HITs so far with an acceptance rate of 95% or higher. The qualification test consists of just three questions from our factual consistency setup; two of which must be answered correctly, along with an explanation text (5 words or more) to explain when “not factually consistent” was chosen. 53% of workers who start the test provide answers to all three questions, and 27.6% of these answer at least two correctly and provide a reasonable explanation text, i.e., only 14.6% of the test takers are granted the qualification.

The qualification enables workers to work on our factual consistency HITs as well as our HITs judging informativeness and coherence.

Fair Compensation. The factual consistency task pays \$0.15 per HIT with a bonus of \$0.05. It can be done quickly, given the fact that a single summary sentence is evaluated and the related sentences in the article are highlighted. The task of evaluating informativeness and coherence (see Appendix D) pays \$0.50 per HIT with a bonus of \$0.25, as more text is displayed, compared to the factuality task. These amount to an average pay of \$12.50 per hour, including the bonus, based on median time spent per HIT. The bonus is paid to workers who spend at least 10 seconds per HIT, give short explanation texts for their decisions and maintain high accuracy on HITs with known answers.

	CNN/DM		MN-800		XSum	
	inf.	coh.	inf.	coh.	inf.	coh.
prefer off	36.5	36.7	39.8	35.8	18.8	18.7
prefer λ_4	46.5	39.2	34.7	39.8	16.5	16.3
both equal	17.0	24.2	25.5	24.3	64.7	65.0

Table 5: Human quality evaluation of summaries generated with no abstractiveness constraint (“off”) versus λ_4 . We asked which summary is more informative or coherent, respectively. MN-800 stands for Multi-News with the input documents truncated to 800 words total (Section 5.1.1).

D Human Evaluation of Informativeness and Coherence

We conduct a human evaluation to determine the informativeness and coherence of the summaries generated with the λ_4 decoding constraint (Equation 1), which increases abstractiveness, as compared to not using any abstractiveness constraint. We use the same setup as for the factuality task, including a qualification test, three annotators per task and aggregation using MACE.

We use the following definitions of *informativeness* and *coherence* for the human evaluation:

- *Informativeness*: The more informative summary is better at expressing the main points of the news story. It contains information that is more relevant and important. It has fewer unimportant details. Its content is more similar to the human-written summary.
- *Coherence*: The more coherent summary has better structure and flow, is easier to follow. The facts are presented in a more logical order.

The results are shown in Table 5. For the **CNN/DM** model, the output without decoding constraints is the most extractive, and the raters preferred the more abstractive version generated with the decoding constraint, both for informativeness and coherence. For the **XSum** model, where the output with the decoding constraint disabled is already highly abstractive, the result is reversed. For **Multi-News**, the result is mixed: Raters found the output with no decoding constraints more informative, but less coherent.

Data	Size	DAE	FactCC	FEQA	QAGS
All	4.2k	.44	.35	.27	.44
CNN/DM	3.0k	.35	.24	.05	.27
XSum	1.2k	.39	.17	†.01	.25

Table 6: Pearson correlations to human factuality judgements on the MODELSFACT dataset. The result with the † symbol is not significant.

E More On Automatic Factuality Metrics

When we apply FactCC to a summary, we apply it separately to each summary sentence and use the mean score per summary. For each sentence that we score with FactCC, we shorten the input document by selecting ten sentences with the highest cosine embedding similarity (Conneau et al., 2017), in order to fit the input to the length limits.

In the following two appendix sections, we use not only DAE and FactCC, as described in the main text, but also two metrics based on question answering: FEQA (Durmus et al., 2020) and QAGS (Wang et al., 2020). **FEQA** generates questions from masked summary sentences whose masked entities are used as “gold” answers; these are compared to the answers obtained from a QA model on the input. In **QAGS**, a question generation model generates questions from the summary, a QA model answers these questions from both summary and input, and the similarity of the answer pairs is evaluated.

F Correlating Human and Automatic Factuality Judgements

Table 6 shows correlations of the human judgements with different automatic metrics on the MODELSFACT dataset, complementing earlier studies (Gabriel et al., 2020; Pagnoni et al., 2021). We compute correlations at the level of individual summaries. To make meaningful comparisons between the human and the automatic scores, we apply the automatic metrics here to the *single* randomly selected sentence per summary that the human annotators judged. Overall, we observe here that DAE has the highest correlations with human judgements.

Data	Model	MINT	μ FACTH	μ DAE	μ FactCC	μ FEQA	μ QAGS
CNN/DM	BART	16.8	66.4 91.2	67.4 92.6	56.2 75.9	47.2 62.4	61.7 84.2
	BERTSUM	14.1	64.7 90.0	57.8 79.6	57.0 78.5	47.6 64.4	60.8 84.2
	PGCONV	5.5	63.5 92.5	64.0 93.3	62.3 90.7	45.2 65.0	58.1 84.4
	BOTTOMUP	17.2	50.6 67.3	55.0 73.9	54.3 72.9	47.3 62.3	58.2 78.7
	ABSRL	18.9	60.6 81.5	62.3 84.0	64.1 86.8	49.6 65.0	61.3 82.5
XSum	BART	80.2	56.9 45.3	67.3 60.8	53.9 40.8	50.9 36.2	53.4 40.1
	BERTSUM	82.8	52.1 36.8	61.5 50.8	50.8 34.8	46.6 28.4	46.0 27.6

Table 7: Abtractiveness (MINT) and factuality of different summarization models. For each factuality metric, we first list its MINT-adjusted variant in green. Example: BART’s μ FACTH is 66.4, while the unadjusted FACTH is 91.2. All numbers are percentage scores $\in [0,100]$.

G Comparison Across Different Models

Here we offer an extended description of our comparison of the abtractiveness-factuality tradeoffs of summarization models from the literature, including the use of additional automatic factuality metrics (see Appendix E).

Table 7 shows human and automatic factuality scores, as well as MINT-adjusted versions of these scores. We observe that all factuality metrics favor the output of the PGCONV model on **CNN/DM**; however, its low abtractiveness indicates that its output falls into the “trivially factual” quadrant (Figure 2). The MINT-adjusted variants (shown in green) penalize such low abtractiveness, favoring the BART or ABSRL models instead, whose outputs represent better tradeoffs between abtractiveness and factuality. Human factuality raters (FACTH) rank ABSRL in fourth place, while FactCC, FEQA and QAGS rank it highly; we hypothesize that ABSRL makes factual errors that these measures cannot detect well. On **XSum**, BART’s output is considerably more factual than BERTSUM’s across all factuality metrics, while BART has only slightly lower abtractiveness; as a result, BART is also favored by all MINT-adjusted factuality metrics. BART’s pretraining of both encoder and decoder may be contributing to its factuality, in accordance with [Maynez et al. \(2020\)](#). Note that for DAE, we apply the Ent-C model on CNN/DM output and the XSUM-HUMAN model on XSum output. Appendix H.2 shows **ROUGE** scores.

H ROUGE Scores

H.1 BART Models

The aim of this paper is not to improve ROUGE scores, but to gain insights about the tradeoff between abtractiveness and factuality. We do, however, stress that the BART models we use in our analysis are competitive with the start of the art. We list our ROUGE-1, ROUGE-2 and ROUGE-L F_1 scores, as well as their averages; see the RL scores in Table 3 as well:

- For CNN/DM, our λ =none decoding has 44.1/21.2/41.0 with an average of 35.4, same as the average of 35.4 in [Lewis et al. \(2020\)](#).
- For XSum, our λ =none decoding has 45.3/21.9/36.8 with an average of 34.7, compared to an average of 34.9 in [Lewis et al. \(2020\)](#).
- For Multi-News, our MN-800 λ =none decoding has 50.2/20.5/45.8 with an average of 38.8, compared to improved ROUGE F_1 results of 44.5/16.0/40.3 with an average of 33.6 by [Fabbri \(personal communication\)](#) for [Fabbri et al. \(2019\)](#).

H.2 Comparing Summarization Models

To complement our comparison of different models in Section 5.2, we list the ROUGE-L F_1 scores of the five models in Table 8.

I Additional Experimental Details

We used AWS p3.8x and p3.16x EC2 machines for all our experiments, except we ran FEQA on the Multi-News summaries on a p3dn.24xlarge machine, as it required more memory.

	Model	RL
CNN/DM	BART	41.0
	BERTSUM	39.2
	PGCONV	36.4
	BOTTOMUP	38.3
	ABSRL	37.3
XSum	BART	36.8
	BERTSUM	31.3

Table 8: ROUGE-L F_1 scores for the models compared in Section 5.2.

The BART model has 406,290,432 parameters. Fine-tuning BART on the Multi-News training set took about 2.5 hours on 4 GPUs; we fine-tuned for 5 epochs following instructions on the fairseq BART webpage, without further hyperparameter search. For CNN/DM and XSum we used the provided checkpoints.¹⁰ The minimum and maximum length for Multi-News decoding was determined by the lengths of the training reference summaries.

¹⁰See <https://github.com/pytorch/fairseq/tree/master/examples/bart>.

Fairness in Language Models Beyond English: Gaps and Challenges

Krithika Ramesh, Sunayana Sitaram, Monojit Choudhury

Microsoft Corporation

{t-kriramesh, sunayana.sitaram, monojitc}@microsoft.com

Abstract

With language models becoming increasingly ubiquitous, it has become essential to address their inequitable treatment of diverse demographic groups and factors. Most research on evaluating and mitigating fairness harms has been concentrated on English, while multilingual models and non-English languages have received comparatively little attention. In this paper, we survey different aspects of fairness in languages beyond English and multilingual contexts. This paper presents a survey of fairness in multilingual and non-English contexts, highlighting the shortcomings of current research and the difficulties faced by methods designed for English. We contend that the multitude of diverse cultures and languages across the world makes it infeasible to achieve comprehensive coverage in terms of constructing fairness datasets. Thus, the measurement and mitigation of biases must evolve beyond the current dataset-driven practices that are narrowly focused on specific dimensions and types of biases and, therefore, impossible to scale across languages and cultures.

1 Introduction

Language models are known to be susceptible to developing spurious correlations and encoding biases that have potentially harmful consequences in downstream tasks. Whilst prior work has documented these harms (Dev et al., 2021) (Bender et al., 2021) (Kumar et al.), there remains much to be studied and criticism for the existing research (or lack thereof) that remains to be addressed.

In the context of language models, fairness can manifest in two forms; *representational* and *allocational* harms. **Representational** harms generally refer to cases where demographic groups end up being misrepresented. This includes stereotypes and negative associations with these groups and even a lack of acknowledgment of certain groups that are underrepresented in the data. **Allocational** harms,

on the other hand, refer to the inequitable distribution of resources and opportunities to groups with different demographic attributes associated with them. The nature of allocational harms can vary based on the **sociocultural, economic, and legal** settings where the system has been deployed. However, it can also take shape in terms of the model’s functionality across languages with fewer resources (Choudhury and Deshpande, 2021; Liu et al., 2021). While current literature adopts a Euro-American-centric view of fairness, work such as Sambasivan et al. (2021) pushes to recognize algorithmic fairness from a more inclusive lens.

Bias crops up in multiple steps of the pipeline (Hovy and Prabhumoye, 2021) (Sap et al., 2022), including the annotation process, the training data, the input representations, model architecture, and the structure of the research design. Thus, measures to mitigate bias in one of these components alone will likely not suffice as a corrective measure, necessitating human intervention at different stages of the pipeline.

Most work that addresses fairness in NLP addresses it from an Anglo-centric context, with comparatively significantly less work done in grammatically-gendered and low-resource languages. Their inability to capture social and cultural nuances and demographic variations is well-documented (Talat et al., 2022). Despite this, they are ubiquitous, with applications ranging diverse fields, from legal contexts to healthcare. That said, there is insufficient documentation of the harms that could stem from unfair models trained for downstream tasks involving natural language generation, despite Arnold et al. (2018); Bhat et al. (2021); Buschek et al. (2021) indicating the influence of these systems on users. Apart from this, these NLP systems also reinforce and reproduce the social and racial hierarchies observed in society and fail to recognize underrepresented communities that are already marginalized (Dev et al., 2021;

Lauscher et al., 2022b). The ramifications of neglecting these issues are diverse and far-reaching, from minor inconveniences for users in less harmful contexts to compromising their privacy as well as depriving them of opportunities and resources (Cirillo et al., 2020; Köchling and Wehner, 2020).

Finally, while the interplay and tradeoff between privacy, efficiency, and fairness in tabular data has received extensive examination (Hooker et al., 2020; Lyu et al., 2020) comparatively fewer studies have been conducted in NLP (Tal et al., 2022; Ahn et al., 2022; Hessenthaler et al., 2022).

The contributions of this work center around drawing attention to the current state of research on fairness in the context of linguistic and cultural issues in non-English languages and in the context of multilingual models. While thorough survey studies such as Sun et al. (2019); Stanczak and Augenstein (2021); Bhatt et al. (2022) yield valuable insights into some of these aspects, none address the current state of the work in multilingual fairness. Our paper provides insights into the following:

- This work surveys and presents challenges and unanswered questions with respect to fairness in both monolingual and multilingual NLP.
- We analyze bias from both a linguistic and cultural lens for non-English languages and present a comprehensive overview of the literature in bias pertaining to grammatically gendered languages and multilinguality.
- We bring to the forefront challenges in multilingual fairness and begin a dialogue for creating more equitable systems for multilingual NLP.

2 Bias in Monolingual Setups for English

2.1 Metrics for Measurement

Prior to delving into the complexities of fairness in multilingual systems, it is essential to first examine the prevalent biases and challenges in monolingual systems. By prefacing the discussion on bias in multilingual systems with an overview of the current state of fairness evaluation and identifying areas for improvement, we aim to shed light on the potential for similar issues to arise in multilingual systems, as many of the biases present in monolingual systems are likely to persist in multilingual contexts. Some of the initial work on analyzing biases in NLP models (Bolukbasi et al., 2016) propose quantitative measures of evaluating bias in

word embeddings. Broadly speaking, bias measures are subcategorized into *i) intrinsic* and *ii) extrinsic* measures. Intrinsic metrics quantify bias in the model’s pre-trained representations, whereas extrinsic metrics deal with bias observed in the outputs of the downstream task the model is trained for.

Caliskan et al. (2017); May et al. (2019); Nadeem et al. (2021); Nangia et al. (2020) are commonly used in papers evaluating language models for fairness. Caliskan et al. (2017) proposes the Word Embedding Association Test (WEAT). A fundamental criticism of WEAT is that it can be exploited to overestimate the bias in a model (Ethayarajh et al., 2019). The Sentence Encoder Association Test (SEAT) metric (May et al., 2019) was proposed to address WEAT’s limitation of measuring bias only over static word embeddings. SEAT is an adaptation of WEAT that allows us to measure bias over contextualized embeddings.

StereoSet (Nadeem et al., 2021), and CrowS-Pair (Nangia et al., 2020) are crowdsourced datasets specifically geared toward measuring the model’s stereotypical proclivity over multiple dimensions, which are inclusive of gender, race, and religion, among others. Blodgett et al. (2021) points out the flaws in the data quality, such as invalid stereotype/anti-stereotype pairs, reliance on indirect group identifiers as a proxy for demographic identification, and logical incongruities in the sentence pairs.

Several other intrinsic measures and adaptations of the aforementioned ones have also been proposed (Kurita et al., 2019; Webster et al., 2020; Kaneko and Bollegala, 2021; Lauscher et al., 2021). Recent studies (Delobelle et al., 2022; Meade et al., 2022) that perform comparative evaluations across these measures provide valuable insights into how and where the metrics can be used, along with their potential drawbacks.

2.2 Intrinsic vs Extrinsic Evaluation

While intrinsic measures are valuable in that they indicate the existence of representational bias in systems, the current literature on fairness evaluation largely concentrates on intrinsic metrics alone. Considerably less work has been done on addressing bias in extrinsic evaluation, with several downstream tasks needing concrete metrics to evaluate bias in their outputs. This is a pressing issue due to the lack of correlation between intrinsic and extrin-

sis measures (Goldfarb-Tarrant et al., 2020; Cao et al., 2022; Delobelle et al., 2022). As emphasized in Orgad and Belinkov (2022), incorporating extrinsic evaluation measures is crucial for several reasons, including the greater relevance of these metrics to bias mitigation objectives. Aside from this, evaluating fairness on the downstream task’s outputs allows us to gauge more precisely how a particular demographic may be affected by the biases in the system.

Although work done in fairness evaluation in NLP primarily concentrates on monolingual studies, there remain several unanswered questions and inconclusive results. For instance, although May et al. (2019) claims to use semantically bleached templates, experiments in Delobelle et al. (2022) suggest that they retain some degree of semantic significance. While several bias evaluation methods use template-based data, recent findings (Alnegheimish et al., 2022) suggest that this approach may be unreliable and advocate the use of natural sentence prompts.

2.3 Fairness From the Lens of Multiple Social Dimensions

The focus of much of the existing body of literature is on gender bias, with little that covers other dimensions like race and religion. Evaluation metrics should be able to evaluate harms in language models over the intersectionality of multiple identities, akin to what would realistically be expected in real-world data. While previous research (Talat et al., 2022; Kirk et al., 2021) has emphasized the importance of fairness evaluation and mitigation over intersectional identities, there is relatively sparse work that attempts to address the same (Tan and Celis, 2019; Subramanian et al., 2021; Hassan et al., 2021; Lalor et al., 2022; Câmara et al., 2022). It is also crucial to gauge if reducing bias across one dimension could affect biases in the other dimensions. Most fairness measures do not account for the intersectionality of identities and standards of justice outside the predominantly Western sphere of distributive justice (Sambasivan et al., 2021; Lundgard, 2020).

Whilst there has been an increase in proposing novel methods to mitigate bias in language models, there needs to be more work in benchmarking these debiasing techniques to assess their relative effectiveness. Meade et al. (2022) represents a step forward in this direction. Despite criticism (Etha-

yarajh et al., 2019; Blodgett et al., 2021) of some evaluation metrics, they are still consistently used (and not always in conjunction with other metrics) in bias evaluation studies.

3 Linguistic Aspects

The linguistic variations between languages pose additional problems in the realm of multilingual NLP. Take, for example, the concept of gender, which has multiple definitions in linguistic terms (namely, **grammatical, referential, lexical and bio-social gender**) (Stanczak and Augenstein, 2021). Section 3.1 delves into how the grammatically gendered nature of languages can affect bias in multilingual and monolingual spaces alike. **Referential** gender, on the other hand, deals with terms that referentially address a person’s gender, such as pronouns. Terms that non-referentially describe gender fall under the umbrella of **lexical** gender, and the **bio-social** definition of gender involves a mixture of phenotypic traits, gender expression, and identity as well as societal and cultural aspects that influence them (Ackerman, 2019).

Although initial forays into this field investigate bias caused by grammatical gender, problems in these systems can also crop up due to the other definitions of gender. Referential gender terms are not always aligned when used in conjunction with lexically gendered terms, particularly with respect to pronoun-based anaphors for queer-identifying individuals. Several default assumptions regarding the individual’s gender identity are made as a consequence (Cao and Daumé III, 2021).

There are multiple varying forms of pronoun complexity (Lindström, 2008; Ballard, 1978). Apart from this, there are instances of substantial variations in their linguistic forms even among languages within a specific region, as highlighted in Nair (2013). Linguistics also involves the presence of constructs like deictic pronouns and honorific pronouns (Goddard, 2005), which in some cases can lead to the pronouns used to reference someone changing based on their social dynamic within the community (Lauscher et al., 2022c). These linguistic aspects represent another line of work that must be addressed for lower-resourced communities that communicate using languages that utilize these.

Lexical gender, while non-referential, finds its own challenges due to the variation of these terms across languages. For example, while certain relationships with individuals in a family may have an

exact mapping in other languages, more often than not (particularly with Southeast Asian languages), there is no precise mapping, and the system ends up making an approximation or ignoring the term altogether. Such issues may also be likely to perforate to other axes such as race, religion, caste, and so forth. In particular, considering that one method of training multilingual embeddings relies on alignment-based approaches, it is imperative that we keep in mind how these design choices could affect the representations of these terms.

Whilst utilizing linguistic features in methods to evaluate and mitigate gender bias is a relatively new field of study, previous work has demonstrated that additional linguistic context can result in performance gains (Volkova et al., 2013; Wallace et al., 2014), thus in alignment with the claim from Hovy and Yang (2021) that LMs must utilize social context to be able to reach human-level performance on tasks. Sun et al. (2021) utilizes linguistic features to capture cross-cultural similarities, and thus, to select languages that are optimal for cross-lingual transfer. However, it is essential to acknowledge that languages are susceptible to cultural and linguistic shifts that occur at both global and local levels over time, as noted in Hamilton et al. (2016). Pretrained models also have the capability to embed sociodemographic information, as evinced by Lauscher et al. (2022a).

It has also been noted that other linguistic forms of gender do not translate well to sociological gender (Cao and Daumé III, 2021). Furthermore, the scarcity of non-binary gender options in different languages can lead to the misgendering of non-binary individuals in these languages, as they may be constricted to fit into a binarized definition of sociological gender.

3.1 Grammatically Gendered Languages

Linguistics recognizes multiple forms of gender (Cao and Daumé III, 2020), as observed in grammatically gendered languages where most or all nouns, including those referring to inanimate objects, possess a syntactic concept of gender. These languages can have anywhere between 2 to 20 forms of grammatical gender divisions. There has been an almost exclusive focus on English for evaluating gender bias, even in the setting of monolingual models and systems. English, however, is not a grammatically-gendered language. This may limit the transferability of techniques used for bias

evaluation and mitigation to other languages that are grammatically gendered.

Zhou et al. (2019) examines bias from the view of grammatically gendered languages by decomposing the gendered information of words in the embedding space into two components; i) semantic and ii) syntactic. For instance, the Spanish word for "man" (*hombre*) is both semantically and syntactically gendered. However, the Spanish word for "water" (*agua*) is not semantically gendered but is considered a feminine noun. The proximity of female occupation words to the feminine side and male occupation words to the masculine side of the semantic gender direction suggests the presence of bias in these Spanish embeddings. Zhou et al. (2019) also demonstrates via experiments on bilingual embeddings that, post-alignment, masculine-gendered words are closer to the English equivalent of the occupation words than feminine-gendered ones. The paper also proposes bias mitigation methods and demonstrates that the quality of the embeddings is preserved via word-translation experiments. Nevertheless, the validity of these mitigation measures would need to be verified by testing them on downstream tasks. Gonen et al. (2019) show that grammatical gender affects the word representations in Italian and German and that inanimate nouns end up being closer to words of the same gender. They propose to address this through the precise use of a language-specific morphological tool and a careful approach to removing all the gender signals from a given text.

The grammatical properties of a language might show some interesting properties to be taken into account when dealing with the fairness of large language models, particularly for gender bias. Studies directed toward them could yield insights into observable trends across language families, with Gonen et al. (2019) demonstrating how the alignment of languages in the embedding space is negatively affected by grammatical gender. They could also prove helpful when analyzing bias in multilingual models, where both grammatically gendered and non-gendered languages are aligned to the same embedding space. The research and datasets available for extrinsic evaluation over other languages remain an area with scope for improvement.

Apart from these grammatical properties that affect the results we observe, the translation of existing bias evaluation datasets into other languages to create parallel corpora does not suffice

when dealing with languages apart from English. This is partly because most languages are inherently rooted in cultural context. Any data curated for these languages must incorporate socio-cultural and linguistic aspects unique to the language/region. Depriving NLP systems of cultural context could consequently lead to entire axes over which social biases are measured being ignored. The cultural significance of words and phrases in various languages can vary significantly, as demonstrated in [Mohamed et al. \(2022\)](#), as well as in characteristics such as metaphorical tendencies ([Gutiérrez et al., 2016](#)) and communication styles ([Miehle et al., 2016](#); [Suszczyńska, 1999](#)). [Hovy and Yang \(2021\)](#) includes an overview and critique of this in the current state of NLP literature, which they claim adopts an oversimplified view and focuses on the information content alone while ignoring the social context of this content. [Milios and BehnamGhader \(2022\)](#); [España-Bonet and Barrón-Cedeño \(2022\)](#) illustrate the inefficiency of direct translation methods, and [España-Bonet and Barrón-Cedeño \(2022\)](#) advocates for the creation of culturally-sensitive datasets for fairness assessment. However, [Kaneko et al. \(2022\)](#) proposes a way to generate parallel corpora for other languages that bears high correlation with human bias annotations.

4 Multilingual Models

Multilingual spaces allow the embeddings of multiple languages to be aligned so that the mappings of every word to its equivalent in other languages are close to each in these embedding spaces. There are numerous ways of training multilingual language models ([Hedderich et al., 2021](#)) using monolingual and unlabeled data. Multilingual language models can improve cross-lingual performance on low-resource languages leveraging the data available to higher-resourced languages up to a certain number of languages. Beyond a point, however, the performance across these languages on cross-lingual and monolingual tasks begins to dip as the number of languages increases ([Conneau et al., 2020](#)). However, few studies explore the impact of multilingual training on biases. [Hovy and Yang \(2021\)](#) illustrate how language and culture share a strong association, and [Khani et al. \(2021\)](#); [Sun et al. \(2021\)](#) reveal that geographical and cultural proximity among languages could enhance the performance of models.

Languages provide much insight into a society's

cultural norms, ideologies, and belief systems ([Hershovich et al., 2022](#); [Wilson et al., 2016](#)). Often, the properties unique to a language are not clearly mapped to other languages or even other dialects within a language, with no direct translations available for several phrases and terminology. Whether or not language models can retain this cultural information and context while utilizing information from higher-resourced languages still requires investigation.

4.1 An Outline of Fairness Evaluation in the Context of Multilinguality

Several datasets have been put forward for the purpose of multilingual evaluation, and [Table 1](#) describes these datasets along with details regarding their utility. These include the languages they cover, whether or not they evaluate bias over pre-trained representations or a downstream task, and the downstream tasks and dimensions they cater toward.

[Zhao et al. \(2020\)](#) was among the first papers to quantify biases in multilingual spaces and does so using both extrinsic and intrinsic evaluation techniques. Their findings indicate that some factors that influence bias in multilingual embeddings include the language's linguistic properties, the target language used for the alignment of the embeddings, and transfer learning on these embeddings induces bias. Additionally, there is the possibility that non-Germanic languages do not align well with Germanic ones, and further work would be required to derive conclusions as to how this affects fairness measurements.

[Huang et al. \(2020\)](#) released the first multilingual Twitter corpus for hate speech detection, annotated with the author's demographic attributes (age, country, gender, race/ethnicity), which allows for fairness evaluation across hate speech classifiers. Through experiments, they prove that variations in language, which are highly correlated with demographic attributes ([Preoțiuc-Pietro and Ungar, 2018](#); [Osiapem, 2007](#)), can result in biased classifiers. However, there are some promising results from [Liang et al. \(2020\)](#), which proposes a novel debiasing method using [Dufter and Schütze \(2019\)](#). While the multilingual model is originally debiased over English, results show its effectiveness for zero-shot debiasing over Chinese.

[Câmara et al. \(2022\)](#) measures both unisectional and intersectional social biases over gender, race,

Dataset	Languages	Task	Metric	Dimensions
https://github.com/MSR-LIT/MultilingualBias	English, Spanish, German, French	Text Classification	I, E	Gender
https://github.com/xiaoleihuang/DomainFairness	English, Italian, Portuguese, Spanish	Text Classification	E	Gender
https://github.com/kanekomasahiro/bias_eval_in_multiple_mlm	German, Japanese, Arabic, Spanish, Portuguese, Russian, Indonesian, Chinese	Masked Language Modelling	I	Gender
https://github.com/ascamara/ml-intersectionality	English, Arabic, Spanish	Text Classification	E	Gender, Race/Ethnicity, Intersection
https://github.com/liangsheng02/densray-debiasing/	English, Chinese	Masked Language Modelling	I	Gender
https://github.com/xiaoleihuang/Multilingual_Fairness_LREC	English, Italian, Portuguese, Spanish, Polish	Text Classification	E	Age, Country, Gender, Race/Ethnicity
https://github.com/coastalcph/fairlex	English, German, French, Italian and Chinese	Text Classification	E	Gender, Age, Region, Language, Legal Area

Table 1: Datasets for fairness evaluation beyond English. I = Intrinsic, E = Extrinsic

and ethnicity in multilingual language models. This is particularly relevant, as in a practical setting, treating identities as composites of various demographic attributes is a necessity. Kaneko et al. (2022) measures gender bias in masked language models and proposes a method to use parallel corpora to evaluate bias in languages shown to have high correlations with human bias annotations. In cases where manually annotated data doesn't exist, this could prove helpful.

Although there has been research on fairness in multimodal contexts (Wolfe and Caliskan, 2022; Wolfe et al., 2022), in a first-of-its-kind study, Wang et al. (2022) looks at fairness from a multilingual view in multimodal representations. Whilst they find that multimodal representations may be individually fair, i.e., similar text representations across languages translate to similar images, this concept of fairness does not extend across multiple groups.

Talat et al. (2022) expresses criticism over the primary data source for multilingual large language models being English, which they claim is reflective of cultural imperialism. They also advocate for these models to be used only for languages they have been trained for to retain the cultural context unique to a language. The multilingual datasets commonly used tend to be parallel corpora derived directly from English translations, neglecting the socio-cultural nuances specific to a given language, as evidenced by the CommonCrawl corpora (Dodge et al., 2021).

Moreover, recent literature (Al Kuwatly et al., 2020; Parmar et al., 2022; Sap et al., 2022) presents us with yet another potential issue; lack of demographic variation in the annotation of these dataset results could contribute to bias in the pipeline. As of yet, several languages (Aji et al., 2022; Joshi et al., 2020) (such as Hindi, Arabic, and Indonesian, which have tens to hundreds of million of native speakers) have had little to no fairness benchmark-

ing datasets developed for them, an indicator that much remains to be done to develop more equitable language models.

4.2 An Outline of Fairness Mitigation in the Context of Multilinguality

Due to multilingual spaces being a composite of the embeddings of various languages with different linguistic and semantic properties, it would serve mitigation techniques well to consider these differences. Other methods could use these distinctions to reduce bias in downstream tasks. Zhao et al. (2020), for one, show that balancing the corpus and transferring it to a grammatically gendered language's embedding space could reduce bias, and that using debiased embeddings could also aid with bias mitigation.

Huang (2022) takes inspiration from the FEDA domain adaption technique (Daumé III, 2007) to use it to mitigate bias in multilingual text classification and compares this with other mitigation methods. These debiasing baselines involve adversarial training, masking out tokens associated with demographic groups, and instance weighting to reduce the impact of data instances that could lead to more biased classifiers. While Liang et al. (2020) show that zero-shot debiasing can be beneficial for this purpose, further study would be required to ascertain if this is a feasible possibility.

4.3 Problems in Multilingual Evaluation and Mitigation

A major challenge in multilingual fairness is the lack of datasets (including parallel corpora) and literature for evaluation across tasks. Much of the research conducted in monolingual contexts has yet to be replicated in a multilingual setting, which would enable us to determine whether or not bias trends in monolingual spaces are directly transferable to multilingual contexts. Research and data resources also tend to neglect less-represented

demographics, notably those local to a particular region. Further, datasets require thorough documentation, as variations in annotator information can result in different types of biases infiltrating the pipeline (Mohamed et al., 2022; Joshi et al., 2016; Bracewell and Tomlinson, 2012). These could include attitudes towards other cultures and languages, which must be assessed and reported during data collection. Multilingual users speak multiple languages, and there is no work on evaluating bias in language contact settings such as code-switching. Certain axes along which systems may discriminate may be contained to a given region. Due to the underrepresented nature of marginalized identities (such as immigrant communities), models will likely not learn useful representations of these identities.

5 Culture

Language and culture are intrinsically linked with each other. However, NLP research has historically placed a considerable emphasis on the information content of the data, as opposed to the contextual information surrounding the same data. Hovy and Yang (2021) propose a broad taxonomy of 7 social factors that encompasses various aspects of this contextual information. This could be incorporated into models to improve performance and make them aware from a socio-cultural perspective.

The differences between a pair of languages or even a pair of dialects could reflect across multiple attributes; this could lead to variations in language’s *phonology, tone, text, and lexical forms*. Some of these attributes are controlled by the speaker and receiver involved. Despite evidence of gains in performance by leveraging these features, systems still retain the potential to discriminate against marginalized communities, as evinced in Sap et al. (2019). This necessitates the proposal of evaluation methods to analyze the potential harms that people from different cultural backgrounds might expose themselves to via the use of such systems.

Multilingualism also entails the need to navigate the nuances of language, including the potential for stereotypes and discriminatory language, which may not have precise equivalents in other languages. Cultural taboos and stereotypes can be highly localized. As an example, pregnant or lactating women are discouraged from consuming nutritious food in certain cultures (Meyer-Rochow,

2009). Such contextual information might be underrepresented or nonexistent in the data that the model is exposed to. While some culture-specific behaviors may be prohibited or frowned upon in some parts of the world, there are yet other places that may encourage or remain indifferent to these very same behaviors.

Additionally, the axes we consider require to be treated differently in different cultural and linguistic settings. Take, for instance, gender. While gender has, for the most part, been treated as a binary variable in these studies, this does not echo what is observed in real-world settings, where several individuals have non-binary gender identities (Devinney et al., 2022). Non-binary gender identities encompass a broad spectrum of gender identities, and the term is generally considered an umbrella term for any identity outside the binary. The inability of models to incorporate this additional information on gender has subsequently led to them developing meaningless representations of non-binary genders in text (Dev et al., 2021). This translates to the systematic erasure of their identities. Baumler and Rudinger (2022) show that much remains to be done concerning addressing non-binary identities outside the Western context. For instance, several non-binary identities, such as the Aravanis and the Māhūs (local to India and Hawaii, respectively) are likely to have little to no meaningful coverage in the training data of the models. These identities can also have unanswered nuances in literature; for example, the Acaults of Myanmar do not consider transsexualism, transvestism, and homosexuality to be distinct categories. This is also applicable to languages such as Arabana-Wangkangurru, which make use of deictic pronouns (previously discussed in Section 3) (Lauscher et al., 2022c; Hercus, 1994).

Further, given that models are highly susceptible to the kind of data they are trained on, it is unlikely that our models can recognize that certain forms of prejudice are more frequent in specific socio-cultural environments than others. The targets of this discrimination are also likely to vary from region to region, another nuance that models find difficult to account for. India and Nepal, for instance, are two countries that still suffer from the effects of the hierarchy of a historically caste-based society that (despite sharing similar roots) bear differences in terms of representation of the various castes and how they are referred to (Jodhka et al., 2010; Rao,

2010). It is important to note that the ability of a system to incorporate information from these social factors to mitigate biases is task-dependent. Downstream tasks like machine translation and dialogue/response generation may depend more on cues related to speaker and receiver characteristics from the taxonomy proposed in [Hovy and Yang \(2021\)](#) than other tasks. Extrinsic metrics for machine translation focus primarily on the gender bias of the mappings of nouns and pronouns from one language to another ([Cho et al., 2019](#)). On the other hand, more open-ended, subjective tasks like NLG are prone to encoding underlying biases and stereotypes across multiple axes and reproducing these in their outputs ([Henderson et al., 2018](#)).

It is critical to consider intersectionality in these studies, as every individual is a composite of multiple identities across multiple axes. When conducting inquiries into the biased nature of these systems, we encourage researchers to use metrics that treat fairness as an intersectional concept and keep in line with the recommendations as suggested in [Talat et al. \(2022\)](#); [Blodgett et al. \(2020\)](#) to document the affected demographics. Testing the validity and reliability of bias measurement and debiasing metrics is essential to ensuring the effectiveness of proposed methods ([Blodgett et al., 2020](#)), and it is crucial to report any limitations of the same.

6 Moving Towards Inclusive Systems in All Languages

The issue of fairness in multilingualism presents a number of challenges. Although current practitioners encourage making systems multicultural and developing systems to be used only for specific cultural contexts ([Talat et al., 2022](#)), we posit that this may not be a viable solution due to various practical considerations. The vast diversity of cultures and ethnicities across the world presents significant difficulties in terms of creating equitable multilingual systems. Even within languages such as English, several dialectal variants, both of the regional and social kind ([Nguyen et al., 2016](#)), still need to be accounted for. [Blodgett and O'Connor \(2017\)](#) is an example of how this could further stigmatize oppressed communities. Language and various social aspects related to language are ever-evolving. Modeling aspects such as lexical variants and the syntactical difference between languages, elements like phonology, and speech inflections in spoken language could contribute to the complexity

of these systems.

Several countries have diverse concentrations of people from all regions of the world with unique backgrounds. The intricacies of the social interactions resulting from the population's diverse linguistic backgrounds and issues arising from language contact make the study of the fairness of multilingual systems that would be deployed to cater to these populations essential. It is not possible to make models agnostic to demographic attributes. Even with the omission of certain attributes, models can still exhibit bias based on factors such as linguistic variations in dialect, or the linguistic features employed, as demonstrated by [Hovy and Søgaard \(2015\)](#) who highlight the improved performance of NLP systems on texts written by older individuals. The data that large language models (LLMs) are trained on tends to be biased towards certain demographic strata ([Olteanu et al., 2019](#)). Although curating more diverse datasets and following recommendations to mitigate bias in the data pipeline would be a step forward to mitigating this problem ([B et al., 2021](#)), various resource constraints could hinder this or make it impractical.

Due to all these challenges and the ubiquity of language technologies that are used by large populations of non-English speaking users, addressing fairness and bias, taking into account diverse linguistic, socio-linguistic, and cultural factors, is of utmost importance. Interdisciplinary and multicultural teams are crucial to identifying, measuring, and mitigating harms caused by bias in multilingual models. Better evaluation benchmarks covering diverse linguistic phenomena and cultures will lead to better fairness evaluation.

Regarding data collection, as discussed in [Section 3.1](#), it would be prudent to avoid directly translating datasets for training or evaluation in applications where fairness is critical. As we have shown in this survey, it is not enough to collect datasets in multiple languages for measuring and mitigating bias, although even these are lacking for most languages worldwide. Zero-shot techniques that ignore the cultural nuances of a language should be used with care in fairness-critical applications, as linguistically similar languages may have different cultural values and vice versa. Finally, multilingual models and systems need to incorporate shared value systems that take into account diverse cultures, although some cultural differences may still go unacknowledged.

Limitations

Our work surveys fairness literature in languages other than English, including bias measurement and mitigation strategies. Although we call out the fact that bias in literature is studied from an Anglo-centric point of view, it is conceivable that we miss many diverse perspectives on linguistic and cultural aspects of bias in different languages and cultures of the world due to the relatively heterogeneous background (in terms of nationality, ethnicity and field of study) of the authors. There may also be other relevant work in the social science literature that we may have missed including in this survey.

References

- Lauren Ackerman. 2019. Syntactic and cognitive issues in investigating gendered coreference. *Glossa*, 4.
- Jaimeen Ahn, Hwaran Lee, Jinhwa Kim, and Alice Oh. 2022. [Why knowledge distillation amplifies gender bias and how to mitigate from the perspective of DistilBERT](#). In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 266–272, Seattle, Washington. Association for Computational Linguistics.
- Alham Fikri Aji, Genta Indra Winata, Fajri Koto, Samuel Cahyawijaya, Ade Romadhony, Rahmad Mahendra, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasajo, Timothy Baldwin, Jey Han Lau, and Sebastian Ruder. 2022. [One country, 700+ languages: NLP challenges for underrepresented languages and dialects in Indonesia](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7226–7249, Dublin, Ireland. Association for Computational Linguistics.
- Hala Al Kuwatly, Maximilian Wich, and Georg Groh. 2020. [Identifying and measuring annotator bias based on annotators’ demographic characteristics](#). In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 184–190, Online. Association for Computational Linguistics.
- Sarah Alnegheimish, Alicia Guo, and Yi Sun. 2022. [Using natural sentence prompts for understanding biases in language models](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2824–2830, Seattle, United States. Association for Computational Linguistics.
- Kenneth C. Arnold, Krysta Chauncey, and Krzysztof Z Gajos. 2018. Sentiment bias in predictive text recommendations results in biased writing. *Proceedings of the 44th Graphics Interface Conference*.
- Senthil Kumar B, Aravindan Chandrabose, and Bharathi Raja Chakravarthi. 2021. [An overview of fairness in data – illuminating the bias in data pipeline](#). In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 34–45, Kyiv. Association for Computational Linguistics.
- William L. Ballard. 1978. [More on yuchi pronouns](#). *International Journal of American Linguistics*, 44(2):103–112.
- Connor Baumler and Rachel Rudinger. 2022. [Recognition of they/them as singular personal pronouns in coreference resolution](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3426–3432, Seattle, United States. Association for Computational Linguistics.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the dangers of stochastic parrots: Can language models be too big?](#) . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’21*, page 610–623, New York, NY, USA. Association for Computing Machinery.
- Advait Bhat, Saaket Agashe, and Anirudha Joshi. 2021. [How do people interact with biased text prediction models while writing?](#) In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 116–121, Online. Association for Computational Linguistics.
- Shaily Bhatt, Sunipa Dev, Partha Talukdar, Shachi Dave, and Vinodkumar Prabhakaran. 2022. [Re-contextualizing fairness in NLP: The case of India](#). In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 727–740, Online only. Association for Computational Linguistics.
- Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. [Language \(technology\) is power: A critical survey of “bias” in NLP](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.
- Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. [Stereotyping Norwegian salmon: An inventory of pitfalls in fairness benchmark datasets](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1004–1015, Online. Association for Computational Linguistics.

- Su Lin Blodgett and Brendan O’Connor. 2017. [Racial disparity in natural language processing: A case study of social media african-american english](#).
- Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Kalai. 2016. [Man is to computer programmer as woman is to homemaker? debiasing word embeddings](#). *CoRR*, abs/1607.06520.
- David Bracewell and Marc Tomlinson. 2012. [The language of power and its cultural influence](#). In *Proceedings of COLING 2012: Posters*, pages 155–164, Mumbai, India. The COLING 2012 Organizing Committee.
- Daniel Buschek, Martin Zurn, and Malin Eiband. 2021. [The impact of multiple parallel phrase suggestions on email input and composition behaviour of native and non-native english writers](#). *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.
- Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. [Semantics derived automatically from language corpora contain human-like biases](#). *Science*, 356(6334):183–186.
- António Câmara, Nina Taneja, Tamjeed Azad, Emily Allaway, and Richard Zemel. 2022. [Mapping the multilingual margins: Intersectional biases of sentiment analysis systems in English, Spanish, and Arabic](#). In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 90–106, Dublin, Ireland. Association for Computational Linguistics.
- Yang Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. [On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 561–570, Dublin, Ireland. Association for Computational Linguistics.
- Yang Trista Cao and Hal Daumé III. 2020. [Toward gender-inclusive coreference resolution](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4568–4595, Online. Association for Computational Linguistics.
- Yang Trista Cao and Hal Daumé III. 2021. [Toward gender-inclusive coreference resolution: An analysis of gender and bias throughout the machine learning lifecycle*](#). *Computational Linguistics*, 47(3):615–661.
- Won Ik Cho, Ji Won Kim, Seok Min Kim, and Nam Soo Kim. 2019. [On measuring gender bias in translation of gender-neutral pronouns](#). In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 173–181, Florence, Italy. Association for Computational Linguistics.
- Monojit Choudhury and Amit Deshpande. 2021. [How linguistically fair are multilingual pre-trained language models?](#) *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12710–12718.
- Davide Cirillo, Silvina Catuara-Solarz, Czuee Morey, Emre Guney, Laia Subirats, Simona Mellino, Analisa Gigante, Alfonso Valencia, María José Rementería, Antonella Santucci Chadha, and Nikolaos Mavridis. 2020. [Sex and gender differences and biases in artificial intelligence for biomedicine and healthcare](#). *npj Digital Medicine*, 3(1):81.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Hal Daumé III. 2007. [Frustratingly easy domain adaptation](#). In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.
- Pieter Delobelle, Ewoenam Tokpo, Toon Calders, and Bettina Berendt. 2022. [Measuring fairness with biased rulers: A comparative study on bias metrics for pre-trained language models](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1693–1706, Seattle, United States. Association for Computational Linguistics.
- Sunipa Dev, Masoud Monajatipoor, Anaelia Ovalle, Arjun Subramonian, Jeff Phillips, and Kai-Wei Chang. 2021. [Harms of gender exclusivity and challenges in non-binary representation in language technologies](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1994, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hannah Devinney, Jenny Björklund, and Henrik Björklund. 2022. [Theories of “gender” in nlp bias research](#). In *2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’22*, page 2083–2102, New York, NY, USA. Association for Computing Machinery.
- Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. [Documenting large webtext corpora: A case study on the colossal clean crawled corpus](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Philipp Dufter and Hinrich Schütze. 2019. [Analytical methods for interpretable ultradense word embeddings](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1185–1191, Hong Kong, China. Association for Computational Linguistics.
- Cristina España-Bonet and Alberto Barrón-Cedeño. 2022. [The \(undesired\) attenuation of human biases by multilinguality](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages –, Online and Abu Dhabi, UAE. Association for Computational Linguistics.
- Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019. [Understanding undesirable word embedding associations](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1696–1705, Florence, Italy. Association for Computational Linguistics.
- Cliff Goddard. 2005. [The languages of east and south-east asia: An introduction](#).
- Seraphina Goldfarb-Tarrant, Rebecca Marchant, Ricardo Muñoz Sanchez, Mugdha Pandya, and Adam Lopez. 2020. [Intrinsic bias metrics do not correlate with application bias](#).
- Hila Gonen, Yova Kementchedjhieva, and Yoav Goldberg. 2019. [How does grammatical gender affect noun representations in gender-marking languages?](#) In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 463–471, Hong Kong, China. Association for Computational Linguistics.
- E.D. Gutiérrez, Ekaterina Shutova, Patricia Lichtenstein, Gerard de Melo, and Luca Gilardi. 2016. [Detecting cross-cultural differences using a multilingual topic model](#). *Transactions of the Association for Computational Linguistics*, 4:47–60.
- William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. [Cultural shift or linguistic drift? comparing two computational measures of semantic change](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2116–2121, Austin, Texas. Association for Computational Linguistics.
- Saad Hassan, Matt Huenerfauth, and Cecilia Ovesdotter Alm. 2021. [Unpacking the interdependent systems of discrimination: Ableist bias in nlp systems through an intersectional lens](#).
- Michael A. Hedderich, Lukas Lange, Heike Adel, Janik Strötgen, and Dietrich Klakow. 2021. [A survey on recent approaches for natural language processing in low-resource scenarios](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.
- Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. 2018. [Ethical challenges in data-driven dialogue systems](#). In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 123–129, New York, NY, USA. Association for Computing Machinery.
- Luise Hercus. 1994. [A grammar of the arabawangkurru language : Lake eyre basin, south australia](#).
- Daniel Hershcovich, Stella Frank, Heather Lent, Miryam de Lhoneux, Mostafa Abdou, Stephanie Brandl, Emanuele Bugliarelli, Laura Cabello Piqueras, Ilias Chalkidis, Ruixiang Cui, Constanza Fierro, Katerina Margatina, Phillip Rust, and Anders Søgaard. 2022. [Challenges and strategies in cross-cultural NLP](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6997–7013, Dublin, Ireland. Association for Computational Linguistics.
- Marius Hessenthaler, Emma Strubell, Dirk Hovy, and Anne Lauscher. 2022. [Bridging fairness and environmental sustainability in natural language processing](#).
- Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. 2020. [Characterising bias in compressed models](#).
- Dirk Hovy and Shrimai Prabhumoye. 2021. [Five sources of bias in natural language processing](#). *Language and Linguistics Compass*, 15(8):e12432.
- Dirk Hovy and Anders Søgaard. 2015. [Tagging performance correlates with author age](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 483–488, Beijing, China. Association for Computational Linguistics.
- Dirk Hovy and Diyi Yang. 2021. [The importance of modeling social factors of language: Theory and practice](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 588–602, Online. Association for Computational Linguistics.
- Xiaolei Huang. 2022. [Easy adaptation to mitigate gender bias in multilingual text classification](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 717–723, Seattle, United States. Association for Computational Linguistics.
- Xiaolei Huang, Linzi Xing, Franck Dernoncourt, and Michael J. Paul. 2020. [Multilingual Twitter corpus and baselines for evaluating demographic bias in hate speech recognition](#). In *Proceedings of the*

- Twelfth Language Resources and Evaluation Conference*, pages 1440–1448, Marseille, France. European Language Resources Association.
- Surinder S. Jodhka, Ghanshyam Shah, Tudor Kalinga, P Silva, Paramsothy Sivapragasam, Thanges, Sri Lanka, Uddin Iftekhhar, Chowdhury, Bangladesh Zulfi, Ali Shah, Krishna Bhattachan, Tej Sunar, Yasso Bhattachan, Nepal Senapati, Sobin George, Martin Macwan, S Thorat, Vincent Manoharan, and Gowhar Yakooob. 2010. Comparative contexts of discrimination: Caste and untouchability in south asia. *Economic and Political Weekly*, 45.
- Aditya Joshi, Pushpak Bhattacharyya, Mark Carman, Jaya Saraswati, and Rajita Shukla. 2016. [How do cultural differences impact the quality of sarcasm annotation?: A case study of Indian annotators and American text](#). In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 95–99, Berlin, Germany. Association for Computational Linguistics.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. [The state and fate of linguistic diversity and inclusion in the NLP world](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.
- Masahiro Kaneko and Danushka Bollegala. 2021. [Unmasking the mask – evaluating social biases in masked language models](#).
- Masahiro Kaneko, Aizhan Imankulova, Danushka Bollegala, and Naoaki Okazaki. 2022. [Gender bias in masked language models for multiple languages](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2740–2750, Seattle, United States. Association for Computational Linguistics.
- Nikzad Khani, Isidora Tourni, Mohammad Sadegh Rasooli, Chris Callison-Burch, and Derry Tanti Wijaya. 2021. [Cultural and geographical influences on image translatability of words across languages](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 198–209, Online. Association for Computational Linguistics.
- Hannah Kirk, Yennie Jun, Haider Iqbal, Elias Benussi, Filippo Volpin, Frédéric A. Dreyer, Aleksandar Shtedritski, and Yuki Markus Asano. 2021. [How true is gpt-2? an empirical analysis of intersectional occupational biases](#). *CoRR*, abs/2102.04130.
- Alina Köchling and Marius Claus Wehner. 2020. [Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of hr recruitment and hr development](#). *Business Research*, 13(3):795–848.
- Sachin Kumar, Vidhisha Balachandran, Lucille Njoo, Antonios Anastasopoulos, and Yulia Tsvetkov. [Language generation models can cause harm: So what can we do about it? an actionable survey](#).
- Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. [Measuring bias in contextualized word representations](#). In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, Florence, Italy. Association for Computational Linguistics.
- John Lalor, Yi Yang, Kendall Smith, Nicole Forsgren, and Ahmed Abbasi. 2022. [Benchmarking intersectional biases in NLP](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3598–3609, Seattle, United States. Association for Computational Linguistics.
- Anne Lauscher, Federico Bianchi, Samuel R. Bowman, and Dirk Hovy. 2022a. [SocioProbe: What, when, and where language models learn about sociodemographics](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7901–7918, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Anne Lauscher, Archie Crowley, and Dirk Hovy. 2022b. [Welcome to the modern world of pronouns: Identity-inclusive natural language processing beyond gender](#).
- Anne Lauscher, Archie Crowley, and Dirk Hovy. 2022c. [Welcome to the modern world of pronouns: Identity-inclusive natural language processing beyond gender](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1221–1232, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Anne Lauscher, Tobias Lükken, and Goran Glavas. 2021. [Sustainable modular debiasing of language models](#). *CoRR*, abs/2109.03646.
- Sheng Liang, Philipp Dufter, and Hinrich Schütze. 2020. [Monolingual and multilingual reduction of gender bias in contextualized representations](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5082–5093, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Eva Lindström. 2008. [Language complexity and inter-linguistic difficulty](#), page 217–242.
- Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. 2021. [Visually grounded reasoning across languages and cultures](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10467–10485, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Alan Lundgard. 2020. [Measuring justice in machine learning](#). In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. ACM.
- Lingjuan Lyu, Xuanli He, and Yitong Li. 2020. [Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness](#).
- Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. [On measuring social biases in sentence encoders](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, Minneapolis, Minnesota. Association for Computational Linguistics.
- Nicholas Meade, Elinor Poole-Dayana, and Siva Reddy. 2022. [An empirical survey of the effectiveness of debiasing techniques for pre-trained language models](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1878–1898, Dublin, Ireland. Association for Computational Linguistics.
- Victor Benno Meyer-Rochow. 2009. [Food taboos: their origins and purposes](#). *Journal of Ethnobiology and Ethnomedicine*, 5(1):18.
- Juliana Miehle, Koichiro Yoshino, Louisa Pragst, Stefan Ultes, Satoshi Nakamura, and Wolfgang Minker. 2016. [Cultural communication idiosyncrasies in human-computer interaction](#). In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 74–79, Los Angeles. Association for Computational Linguistics.
- Aristides Milios and Parishad BehnamGhader. 2022. [An analysis of social biases present in bert variants across multiple languages](#). *ArXiv*, abs/2211.14402.
- Youssef Mohamed, Mohamed Abdelfattah, Shyma Alhuwaider, Feifan Li, Xiangliang Zhang, Kenneth Ward Church, and Mohamed Elhoseiny. 2022. [Artelingo: A million emotion annotations of wikiart with emphasis on diversity over language and culture](#).
- Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. [StereoSet: Measuring stereotypical bias in pretrained language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.
- Ravi Sankar S Nair. 2013. Tribal languages of kerala.
- Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. [Crows-pairs: A challenge dataset for measuring social biases in masked language models](#).
- Dong Nguyen, A. Seza Doğruöz, Carolyn P. Rosé, and Franciska de Jong. 2016. [Computational Sociolinguistics: A Survey](#). *Computational Linguistics*, 42(3):537–593.
- Alexandra Olteanu, Carlos Castillo, Fernando D. Diaz, and Emre Kıcıman. 2019. [Social data: Biases, methodological pitfalls, and ethical boundaries](#). *Frontiers in Big Data*, 2.
- Hadas Orgad and Yonatan Belinkov. 2022. [Choose your lenses: Flaws in gender bias evaluation](#). In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 151–167, Seattle, Washington. Association for Computational Linguistics.
- Iyabo Osiapem. 2007. [Florian coulmas, sociolinguistics: The study of speakers’ choices -](#). *Language in Society - LANG SOC*, 36.
- Mihir Parmar, Swaroop Mishra, Mor Geva, and Chitta Baral. 2022. [Don’t blame the annotator: Bias already starts in the annotation instructions](#).
- Daniel Preotiuc-Pietro and Lyle Ungar. 2018. [User-level race and ethnicity predictors from Twitter text](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1534–1545, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Jasmine Rao. 2010. [The caste system: Effects on poverty in india, nepal and sri lanka](#). *Glob. Majority E-J.*, 1.
- Nithya Sambasivan, Erin Arnesen, Ben Hutchinson, Tulsee Doshi, and Vinodkumar Prabhakaran. 2021. [Re-imagining algorithmic fairness in india and beyond](#). *CoRR*, abs/2101.09995.
- Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. [The risk of racial bias in hate speech detection](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.
- Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. [Annotators with attitudes: How annotator beliefs and identities bias toxic language detection](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5906, Seattle, United States. Association for Computational Linguistics.
- Karolina Stanczak and Isabelle Augenstein. 2021. [A survey on gender bias in natural language processing](#). *CoRR*, abs/2112.14168.
- Shivashankar Subramanian, Xudong Han, Timothy Baldwin, Trevor Cohn, and Lea Frermann. 2021. [Evaluating debiasing techniques for intersectional biases](#).
- Jimin Sun, Hwijeen Ahn, Chan Young Park, Yulia Tsvetkov, and David R. Mortensen. 2021. [Cross-cultural similarity features for cross-lingual transfer](#)

- learning of pragmatically motivated tasks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2403–2414, Online. Association for Computational Linguistics.
- Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy. Association for Computational Linguistics.
- Małgorzata Suszczyńska. 1999. Apologizing in english, polish and hungarian: Different languages, different strategies. *Journal of Pragmatics*, 31(8):1053–1065.
- Yarden Tal, Inbal Magar, and Roy Schwartz. 2022. Fewer errors, but more stereotypes? the effect of model size on gender bias. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 112–120, Seattle, Washington. Association for Computational Linguistics.
- Zerak Talat, Aurélie Névéol, Stella Biderman, Miruna Clinciu, Manan Dey, Shayne Longpre, Sasha Lucicioni, Maraim Masoud, Margaret Mitchell, Dragomir Radev, Shanya Sharma, Arjun Subramonian, Jaesung Tae, Samson Tan, Deepak Tunuguntla, and Oskar Van Der Wal. 2022. You reap what you sow: On the challenges of bias evaluation under multilingual settings. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 26–41, virtual+Dublin. Association for Computational Linguistics.
- Yi Chern Tan and L. Elisa Celis. 2019. Assessing social and intersectional biases in contextualized word representations. *CoRR*, abs/1911.01485.
- Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1815–1827, Seattle, Washington, USA. Association for Computational Linguistics.
- Byron C. Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. 2014. Humans require context to infer ironic intent (so computers probably do, too). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 512–516, Baltimore, Maryland. Association for Computational Linguistics.
- Jialu Wang, Yang Liu, and Xin Wang. 2022. Assessing multilingual fairness in pre-trained multimodal representations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2681–2695, Dublin, Ireland. Association for Computational Linguistics.
- Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *CoRR*, abs/2010.06032.
- Steven Wilson, Rada Mihalcea, Ryan Boyd, and James Pennebaker. 2016. Disentangling topic models: A cross-cultural analysis of personal values through words. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 143–152, Austin, Texas. Association for Computational Linguistics.
- Robert Wolfe and Aylin Caliskan. 2022. American == white in multimodal language-and-image ai. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '22, page 800–812, New York, NY, USA. Association for Computing Machinery.
- Robert Wolfe, Yiwei Yang, Bill Howe, and Aylin Caliskan. 2022. Contrastive language-vision ai models pretrained on web-scraped multimodal data exhibit sexual objectification bias.
- Jieyu Zhao, Subhabrata Mukherjee, Saghar Hosseini, Kai-Wei Chang, and Ahmed Hassan Awadallah. 2020. Gender bias in multilingual embeddings and cross-lingual transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2896–2907, Online. Association for Computational Linguistics.
- Pei Zhou, Weijia Shi, Jieyu Zhao, Kuan-Hao Huang, Muhao Chen, Ryan Cotterell, and Kai-Wei Chang. 2019. Examining gender bias in languages with grammatical gender. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5276–5284, Hong Kong, China. Association for Computational Linguistics.

Global-Local Modeling with Prompt-Based Knowledge Enhancement for Emotion Inference in Conversation

Renxi Wang and Shi Feng

School of Computer Science and Engineering

Northeastern University

Shenyang, China

realreasonwang@gmail.com, fengshi@cse.neu.edu.cn

Abstract

The ability to recognize emotions in conversations is necessary and important for the online chatbot to do tasks such as empathetic response generation and emotional support. Present researches mainly focus on recognizing emotions through a speaker’s utterance, while research on emotion inference predicts emotions of addressees through previous utterances. Because of the lack of the addressee’s utterance, emotion inference is more challenging than emotion recognition. In this paper, we propose a global-local modeling method based on recurrent neural networks (RNN) and pre-trained language models (PLM) to do emotion inference, which utilizes the sequence modeling ability of RNNs and abundant knowledge from PLMs. Moreover, we take the whole dialogue history as input of PLM to generate knowledge by in-context learning. Experimental results show that our model with knowledge enhancement achieves state-of-the-art performance on all three datasets.¹

1 Introduction

The task of emotion recognition in conversation (ERC) (Poria et al., 2019b) aims to identify emotion labels of an utterance, where the whole dialogue history along with the current utterance is given. However, in emotion inference in conversation (EIC) the current utterance is lacking but the dialogue history and the current addressee are known (Li et al., 2021a). For example, Figure 1 shows a conversation between A and B. In the third turn, ERC detects A’s emotion using all available information while EIC predicts A’s emotion using all the information except the last utterance. ERC is a popular task that has been explored widely and deeply, while EIC is a new task that measures the emotion understanding ability of models from a different perspective.

¹The code is available at <https://github.com/Reason-Wang/DialogueGLP>.

Person A

Person B

[neutral]: Hi, what are you doing here?

[happy]: Thank you! It’s so nice of you to invite me.

[happy]: I am celebrating my child’s birthday. Would you like to join us?

Figure 1: A conversation between A and B. The text in orange might be helpful for ERC. The text in green might be helpful for EIC.

In ERC, some previous works utilize sequence-based neural networks to model the context and speaking parties (Majumder et al., 2019; Hu et al., 2021a; Li et al., 2021a). These approaches first finetune a model on utterances to classify emotions. Then this model is used to extract features of utterances. As shown in Figure 1, B makes A happy because he/she is inviting A to join in a party. However, B feels happy because he/she is celebrating his/her child’s birthday. The finetuning tends to keep the semantics that is helpful to classify the current emotion and the feature extraction compresses the utterance’s information, which may cause information loss that is valuable to infer A’s emotion. Some works model dialogues at utterance-level with graph-based models (Ghosal et al., 2019, Shen et al., 2021). They have the same problem as sequence-based models. Also, it becomes difficult for them to distinguish similar emotions as their layers deepen (Li et al., 2022). Pre-trained language models do not need to do the feature extraction process and they contain knowledge suitable for EIC. However, these models can not naturally process sequential utterances from different parties. Motivated by this, we propose global-local modeling method to combine different abilities from these models. Specifically, we use a sequence-based model to get the representation

of the dialogue history and a pre-trained model to process utterances that are close to the addressee’s turn in which his/her emotion is to be inferred. In our framework, the global representation and local utterances can attend to each other, which we believe is helpful for EIC.

Some researchers introduce external knowledge to improve the performance of emotion detection (Ghosal et al., 2020, Li et al., 2021b). They generate commonsense knowledge using COMET (Bosselut et al., 2019) which is trained on ATOMIC (Sap et al., 2019). However, this knowledge is limited to certain event types. Also, it is generated based on a single utterance instead of the whole dialogue, which further limits the quality of the knowledge. Recent advancements of in-context learning (Liu et al., 2022) show that it is possible to generate high-quality knowledge when language models are provided with appropriate examples. Based on the above analysis, we propose a knowledge generation method specially designed for EIC task based on prompt learning. Specifically, we use templates to obtain two kinds of knowledge: I. We let GPT fill the dialogue, thus we get pseudo utterances that may be spoken by the addressee and take them as knowledge. II. We ask GPT how the addressee feels and take generated texts as knowledge. Our knowledge is more precise since we take the whole dialogue history as input. Also, it is more diverse because GPT is trained on a large number of texts in different fields.

2 Methods

2.1 Problem Definition

Given a dialogue $D = [(U_1, p_1), (U_2, p_2), \dots, (U_m, p_m), p_{m+1}]$, where U_i is the utterance in i -th turn and p_i is the participant in i -th turn. For $i = m + 1$, p_i is the addressee, otherwise p_i is the speaker. The task is to predict the addressee’s emotion e using D .

2.2 Global Model

We use DialogueInfer (Li et al., 2021a) as our global model. We first finetune a RoBERTa-Large (Liu et al., 2019) model to predict the emotion label of utterances as ERC task. Then we use the finetuned model to extract features of utterances and get a 1024-dimensional vector u_i for each utterance U_i . These representations of utterances are then put into DialogueInfer to get the representation of the dialogue. DialogueInfer is a model designed for

the EIC task. It adopts addressee-aware modules to capture the persistence and contagiousness of utterances. Formally, the output of the global model can be defined as:

$$\begin{aligned} h_t, c_t &= \mathbb{1}\{p_t = p_{m+1}\}LSTM_a(u_t, (h_{t-1}, c_{t-1})) \\ &\quad + \mathbb{1}\{p_t \neq p_{m+1}\}LSTM_o(u_t, (h_{t-1}, c_{t-1})) \\ h_g &= h_{m+1} \end{aligned} \quad (1)$$

where $t = 1, 2, \dots, m$ is the turn step, $\mathbb{1}\{\text{condition}\}$ is the indicator function and returns 1 if the condition is true otherwise 0, $h_t \in \mathbb{R}^{d_1}$ and $c_t \in \mathbb{R}^{d_1}$ are hidden state and cell state respectively, d_1 is the hidden dimension in LSTM unit, h_g is the global representation of the dialogue. The final output $h_g \in \mathbb{R}^{d_1}$ is then fed into the local model.

2.3 Local Model

We employ RoBERTa (Liu et al., 2019) as the local model. RoBERTa shares the same architecture as BERT (Devlin et al., 2019) and is trained with masked language modeling objective function. We concatenate the last k utterances to form the input. To make the local model addressee-aware as global model, we prepend a speaker prefix to indicate whether the utterance comes from the addressee. The final text input is:

$$\begin{aligned} U_t &= \text{prfix}(p_{m-k+1})U_{m-k+1} </s> \text{prfix}(p_{m-k+2}) \\ &\quad U_{m-k+2} </s> \dots \text{prfix}(p_m) U_m \end{aligned} \quad (2)$$

$$\text{prfix}(p_i) = \begin{cases} \text{"I:"}, & p_i = p_{m+1} \\ \text{"Other:"}, & p_i \neq p_{m+1} \end{cases} \quad (3)$$

where $</s>$ is the special token that indicates the separation of utterances.

To fuse the global information, we add the global representation h_g to the first token’s embedding of the text input. The whole process can be formulated as:

$$\hat{h}_g = W^T h_g + b \quad (4)$$

$$H = [\text{Emb}(U_t[0]) + \hat{h}_g; \text{Emb}(U_t[1 :])] \quad (5)$$

$$h_e = \text{RoBERTa-Model}(H) \quad (6)$$

where $W \in \mathbb{R}^{d_1 \times d_2}$ is the matrix to project dimensions, d_2 is the hidden dimension in RoBERTa model, Emb is the embedding layer of RoBERTa.

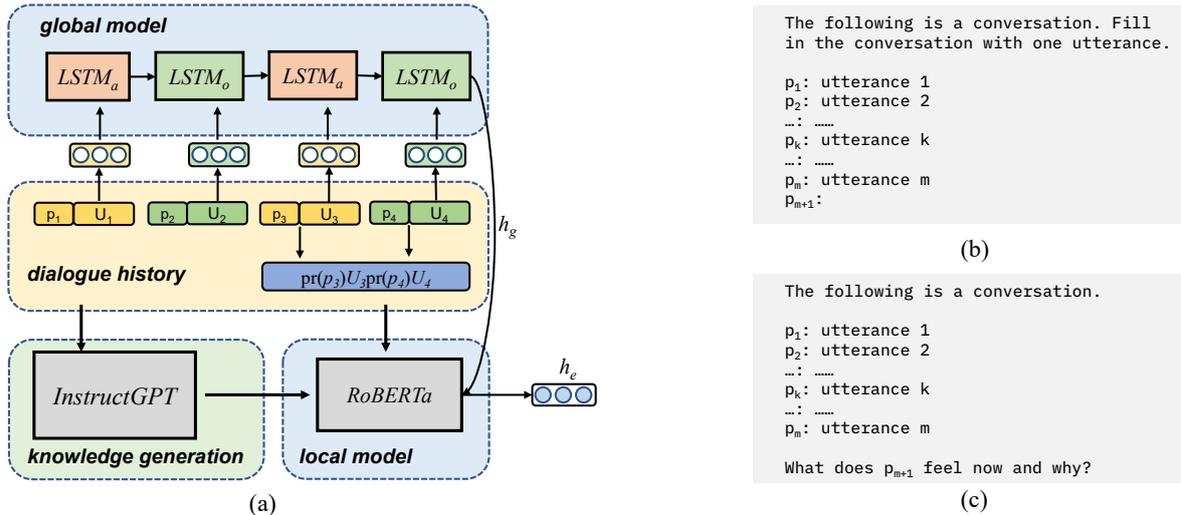


Figure 2: (a) Our framework to infer the emotion. (b) The template to generate pseudo utterances. (c) The template to generate feelings and corresponding reasons.

2.4 Prompt Based Knowledge Generation

GPT-3 (Brown et al., 2020) is a powerful model which generates informative and accurate texts when provided with appropriate examples. The model is further finetuned to align with users so the outputs are more truthful and less toxic (Ouyang et al., 2022). We use the resulting model, called InstructGPT, to generate two kinds of knowledge.

Pseudo Utterances We take the dialogue history as input and let InstructGPT generate the utterance that might be spoken by the addressee. Figure 2 shows the template to generate pseudo utterances. After obtaining these knowledge texts, we first prepend the addressee prefix to them. Then we append them to the text input in the local model.

Feelings and Corresponding Reasons Since InstructGPT is able to do many tasks, we ask InstructGPT directly about the addressee’s emotions and corresponding reasons. The output from the model is taken as knowledge and is used the same way as pseudo utterances. Figure 2 shows the template to generate this kind of knowledge.

2.5 Classifier

We use the first token’s representation h_e as the final output. A softmax layer is employed after a linear projection layer:

$$p_e = \text{softmax}(W^T h_e + b) \quad (7)$$

where $W \in \mathbb{R}^{d_2 \times c}$ is the projection matrix, c is the number of emotions, $p_e \in \mathbb{R}^c$ is the probability distribution over different emotions.

3 Experiments

3.1 Implementation

We train our model on three datasets: DailyDialog (Li et al., 2017), MELD (Poria et al., 2019a) and EmoryNLP (Zahiri and Choi, 2018). We first finetune a RoBERTa-Large model on the training set of each dataset. The batch size is set to 16 and the model with the best performance on the development set is saved. We then use this model to extract features of the datasets. For emotion inference, we set the learning rate to 1e-5. AdamW is used as the optimizer to update parameters. In the first two epochs, we only update the global model and freeze the local model. After that, we finetune the whole model. We find this updating scheme makes training more stable.

For other baselines, we adapted their official codes to make them applicable to EIC task. The parameters we used in training referred to their original paper. We select all the models based on their best performance on the development set. We use cross entropy as the loss function.

3.2 Main Results

Table 1 shows the main results of our experiments. Our base model without knowledge augmentation already performs best on DailyDialog and MELD. In most cases, the generated knowledge improves the performance. However, knowledge U (pseudo utterances) decreases the performance measured by macro F1 on DailyDialog. COMET decreases the performance measured by macro F1 on DailyDia-

Model	DailyDialog		MELD		EmoryNLP	
	macro F1	weighted F1	macro F1	weighted F1	macro F1	weighted F1
DialogueRNN*	36.28	71.67	16.77	33.96	17.62	20.49
DialogueCRN*	33.82	72.23	16.00	35.44	16.92	21.29
DialogueInfer*	34.43	71.00	17.06	35.41	17.64	20.28
DialogueGCN†	37.62	70.89	16.15	34.59	16.86	20.39
DAG†	34.87	71.93	18.27	34.94	17.88	21.86
CoG-BART‡	35.51	72.10	17.15	34.60	17.46	21.35
CoMPM‡	37.67	68.60	17.53	34.67	17.70	21.21
DialogueGLP	40.64	73.55	18.66	37.08	17.35	21.37
DialogueGLP(C)	39.64	73.97	17.46	37.14	16.69	19.97
DialogueGLP(U)	39.30	74.83	19.02	37.39	17.97	21.41
DialogueGLP(F)	40.79	75.10	19.65	37.32	17.70	21.84
DialogueGLP(F+U)	40.93	75.11	20.88	38.42	19.13	22.08

Table 1: Comparison of our models and sequence-based (*), graph-based (†) and transformer-based (‡) models. DialogueInfer and our models are designed for EIC. Others are designed for ERC. (C) denotes knowledge enhancement with COMET, (U) and (F) denote knowledge enhancement with pseudo utterances and feelings respectively. We report the mean score over 5 random seeds.

Model	DailyDialog	MELD	EmoryNLP
DialogueGLP	73.55	37.08	21.37
w/o global model	72.74	36.66	20.60
w/o local model	71.37	35.44	20.93
w/o addressee-aware	73.51	36.58	20.07

Table 2: Ablation studies on three datasets.

log and MELD. Generally, knowledge F (feelings and corresponding reasons) are better than knowledge U. Our prompt-based knowledge generation method is better than COMET. We also concatenate the two generated prompt-based knowledge (U+F). The performance is further improved compared to single knowledge augmentation.

Knowledge F consists of emotions and reasons for those emotions. To explore whether only InstructGPT is enough to predict the emotions, we let it directly infer the emotions of addressees in DailyDialog. The resulting weighted F1 is 34.65, which shows that it is not good at inferring emotions and the main performance boost of DialogueGLP(F) comes from the part of reasons.

Ablation Analysis To explore the effectiveness of different modules in our model, we also do ablation studies on the three datasets. To remove the addressee information, we simply replace the global model with a single LSTM and the addressee prefix with the speaker’s name. The results show that the local model is generally more important than other modules. Since DailyDialog is dyadic, the second to last utterance in our input texts must be from the addressee. Therefore, the addressee information is less important in DailyDialog.

4 Related Work

Emotion recognition in conversation has been a popular area where different models have been proposed. We divide them into three categories: sequence-based, graph-based and transformer-based. DialogueRNN (Majumder et al., 2019) models different parties and global state by different recurrent neural networks. DialogueInfer (Li et al., 2021a) adopts two LSTMs to process utterances by whether they are from addressees. DialogueCRN (Hu et al., 2021b) iteratively does retrieving and reasoning process to extract and integrate emotional clues. DialogueGCN (Ghosal et al., 2019) utilizes graph neural networks to connect utterances with surrounding utterances. DAG (Shen et al., 2021) uses a directed acyclic graph network to gather information over long distances. CoG-BART (Li et al., 2022) adopts supervised contrastive learning and response generation as auxiliary tasks. CoMPM (Lee and Lee, 2022) combines speaker’s memory using a pre-trained model as an extractor.

Some works focus on introducing knowledge to help detect emotions. KET (Zhong et al., 2019) retrieve commonsense knowledge from ConceptNet (Speer et al., 2017) and NRC_VAD (Mohammad, 2018). COSMIC (Ghosal et al., 2020), DialogueInfer (Li et al., 2021b) and ToDKAT (Zhu et al., 2021) incorporates commonsense knowledge generated by COMET (Bosselut et al., 2019). GKP (Liu et al., 2022) generates knowledge from language models with prompt learning to do commonsense reasoning.

5 Conclusions

In this paper we combine the ability of sequence models and pre-trained models and propose global-local modeling method to do emotion inference in conversation. Moreover, we take the whole dialogue as input and generate knowledge with prompt learning. Experiments show that our model has achieved state-of-the-art performance on three datasets. Ablation studies show the effectiveness of different modules in our model.

Limitations

Since in our framework the global model needs to first compute the global representation then the local model outputs the emotion distribution, it takes longer time to train and inference than other models. We utilize a pre-trained model in our framework, which requires large GPU memory.

Acknowledgements

The work was supported by National Natural Science Foundation of China (62272092, 62172086).

References

- Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. [COMET: Commonsense transformers for automatic knowledge graph construction](#). In [Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics](#), pages 4762–4779, Florence, Italy. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. [Advances in neural information processing systems](#), 33:1877–1901.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In [Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 \(Long and Short Papers\)](#), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. [COSMIC: COMmonSense knowledge for eMotion identification in conversations](#). In [Findings of the Association for Computational Linguistics: EMNLP 2020](#), pages 2470–2481, Online. Association for Computational Linguistics.
- Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. [DialogueGCN: A graph convolutional neural network for emotion recognition in conversation](#). In [Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing \(EMNLP-IJCNLP\)](#), pages 154–164, Hong Kong, China. Association for Computational Linguistics.
- Dou Hu, Lingwei Wei, and Xiaoyong Huai. 2021a. [DialogueCRN: Contextual reasoning networks for emotion recognition in conversations](#). In [Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing \(Volume 1: Long Papers\)](#), pages 7042–7052, Online. Association for Computational Linguistics.
- Zhe Hu, Zuohui Fu, Yu Yin, and Gerard de Melo. 2021b. [Context-aware interaction network for question matching](#). In [Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing](#), pages 3846–3853, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Joosung Lee and Woojin Lee. 2022. [CoMPM: Context modeling with speaker’s pre-trained memory tracking for emotion recognition in conversation](#). In [Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies](#), pages 5669–5679, Seattle, United States. Association for Computational Linguistics.
- Dayu Li, Xiaodan Zhu, Yang Li, Suge Wang, Deyu Li, Jian Liao, and Jianxing Zheng. 2021a. [Emotion inference in multi-turn conversations with addressee-aware module and ensemble strategy](#). In [Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing](#), pages 3935–3941, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Dayu Li, Xiaodan Zhu, Yang Li, Suge Wang, Deyu Li, Jian Liao, and Jianxing Zheng. 2021b. [Enhancing emotion inference in conversations with commonsense knowledge](#). [Knowledge-Based Systems](#), 232:107449.
- Shimin Li, Hang Yan, and Xipeng Qiu. 2022. [Contrast and generation make bart a good dialogue emotion recognizer](#). In [Proceedings of the AAAI Conference on Artificial Intelligence](#), volume 36, pages 11002–11010.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. [DailyDialog: A manually labelled multi-turn dialogue dataset](#). In [Proceedings of the Eighth International Joint Conference on Natural Language Processing \(Volume 1: Long Papers\)](#), pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

- Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. [Generated knowledge prompting for commonsense reasoning](#). In [Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics \(Volume 1: Long Papers\)](#), pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). [CoRR](#), abs/1907.11692.
- Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. [Dialoguernn: An attentive rnn for emotion detection in conversations](#). In [Proceedings of the AAAI conference on artificial intelligence](#), volume 33, pages 6818–6825.
- Saif Mohammad. 2018. [Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words](#). In [Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics \(Volume 1: Long Papers\)](#), pages 174–184, Melbourne, Australia. Association for Computational Linguistics.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. [Training language models to follow instructions with human feedback](#). [arXiv preprint arXiv:2203.02155](#).
- Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019a. [MELD: A multimodal multi-party dataset for emotion recognition in conversations](#). In [Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics](#), pages 527–536, Florence, Italy. Association for Computational Linguistics.
- Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019b. [Emotion recognition in conversation: Research challenges, datasets, and recent advances](#). [IEEE Access](#), 7:100943–100953.
- Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. [Atomic: An atlas of machine commonsense for if-then reasoning](#). In [Proceedings of the AAAI conference on artificial intelligence](#), volume 33, pages 3027–3035.
- Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. 2021. [Directed acyclic graph network for conversational emotion recognition](#). In [Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing \(Volume 1: Long Papers\)](#), pages 1551–1560, Online. Association for Computational Linguistics.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. [Conceptnet 5.5: An open multilingual graph of general knowledge](#). In [Thirty-first AAAI conference on artificial intelligence](#).
- Sayed M Zahiri and Jinho D Choi. 2018. [Emotion detection on tv show transcripts with sequence-based convolutional neural networks](#). In [Workshops at the thirty-second aai conference on artificial intelligence](#).
- Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. [Knowledge-enriched transformer for emotion detection in textual conversations](#). In [Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing \(EMNLP-IJCNLP\)](#), pages 165–176, Hong Kong, China. Association for Computational Linguistics.
- Lixing Zhu, Gabriele Pergola, Lin Gui, Deyu Zhou, and Yulan He. 2021. [Topic-driven and knowledge-aware transformer for dialogue emotion detection](#). In [Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing \(Volume 1: Long Papers\)](#), pages 1571–1582, Online. Association for Computational Linguistics.

A Case Study

Figure 3 shows a dialogue example between A and B from DailyDialog (Li et al., 2017) dataset and generated knowledge by COMET (Bosselut et al., 2019) and prompt learning. The task is to infer A’s emotion. For COMET, we take each utterance as input and generate three types of knowledge as Li et al. (2021b), which are oReact, oWant and oEffect. For prompt-based knowledge generation, we formulate the input by the template and dialogue history and input it into InstructGPT. As a result, we get a pseudo utterance that may be spoken by A and feelings of A.

As Figure 3 shows, our knowledge summarizes the dialogue well and is much more human-readable. This property makes our knowledge more suitable to concatenate with text inputs. COMET trained on ATOMIC (Sap et al., 2019) generates knowledge based on events. Therefore only one utterance can be taken as input instead of longer contexts. If an utterance contains multiple events, the generated knowledge may not be accurate. Also, the knowledge generated by COMET often repeats.

B Datasets Preprocessing

The datasets can not be directly used. We take each dialogue as an example and the emotion of the last utterance as the label. In training, we do not use the last utterance. DailyDialog is a dyadic dialogue dataset for emotion recognition. It contains more than 10,000 dialogues. We take each dialogue as a training example and take the last speaker of the dialogue as the addressee and the corresponding emotion as the label. MELD (Poria et al., 2019a) is a multimodal multiparty dialogue dataset designed for emotion recognition. However, it contains less than 2,000 dialogues. To get more training examples, we cut one dialogue into more dialogues. We keep a dialogue at least three utterances and cut it wherever the next speaker is different from the current speaker. Figure 3 shows how we process a dialogue with eight utterances. EmoryNLP (Zahiri and Choi, 2018) is a ERC dataset collected from *Friends*. We preprocess it the same way as MELD. Table 3 shows the statics after we preprocess the three datasets. We get each split of datasets from their original splits.

C Adapting Codes to EIC

Since EIC is a new task, there are not many baselines for EIC. We adapt models that are original

Dataset	train	dev	test
DailyDialog	11118	1000	1000
MELD	6125	685	1540
EmoryNLP	8345	1124	1140

Table 3: Statics of processed datasets.

designed for ERC to do EIC. In our baselines, only DialogueInfer is designed for EIC. For other models, we mainly modify the code of their inputs and outputs.

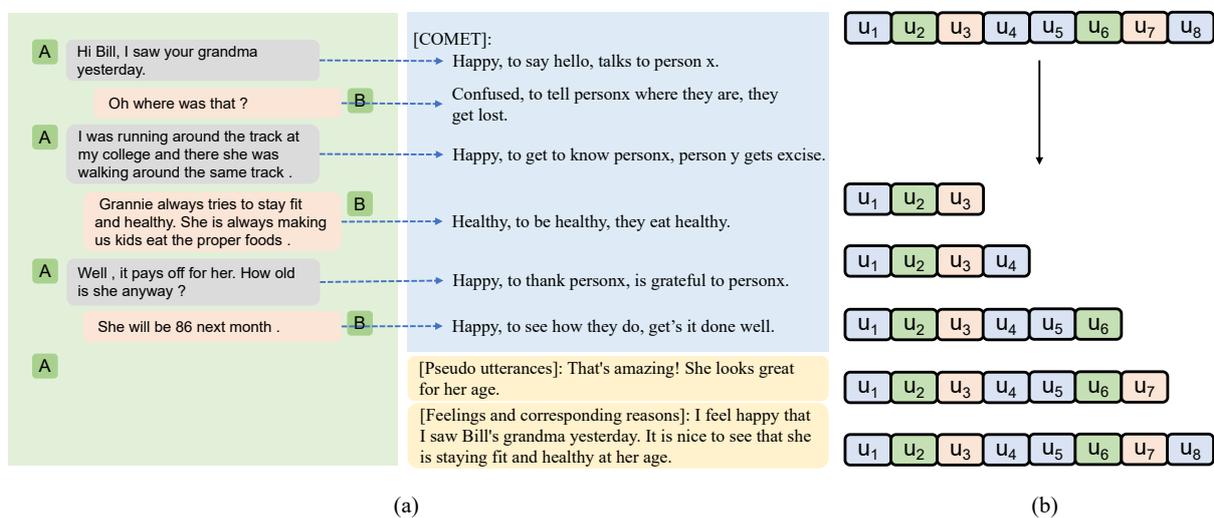


Figure 3: (a) A dialogue example from DailyDialog and generated knowledge from COMET and our method. (b) An example of how we cut dialogues. This dialogue contains eight utterances. Colors denote speakers. We cut it into five dialogues.

Headline Token-based Discriminative Learning for Subheading Generation in News Article

Joonwon Jang

Sejong University / Republic of Korea
joonwon.lainshower@gmail.com

Misuk Kim*

Sejong University / Republic of Korea
mskim.sju@gmail.com

Abstract

The news subheading summarizes an article’s contents in several sentences to support the headline limited to solely conveying the main contents. So, it is necessary to generate compelling news subheadings in consideration of the structural characteristics of the news. In this paper, we propose a subheading generation model using topical headline information. We introduce a discriminative learning method that utilizes the prediction result of masked headline tokens. Experiments show that the proposed model is effective and outperforms the comparative models on three news datasets written in two languages. We also show that our model performs robustly on a small dataset and various masking ratios. Qualitative analysis and human evaluations also show that the overall quality of generated subheadings improved over the comparative models.

1 Introduction

The news headline summarizes the article to grab the attention and interest of the readers (Dor, 2003; Ifantidou, 2009; Ecker et al., 2014). However, the headline is written in a brief form of short sentences with topic-related phrases (Yamada et al., 2021), making it hard for users to grasp the entire content of the news article from the headline alone. To tackle this problem, some news vendors provide a *subheading*, usually located right below the headline, to convey its main content within several sentences. This component can provide a core and informative summary of a news article that cannot be conveyed by the headline alone. Mainly, subheadings are written by professional news writers with concise content that corresponds to the main body of the news.

Recently, Hasan et al. (2021) released XLSum, a multilingual news summary dataset, referring to subheading as a summary of the article. Therefore, generating subheadings can be considered an abstractive summarization problem that needs to cap-

ture the topical knowledge from the body of the article. The main approach is to add auxiliary signals to make the model aware of topical knowledge. Dou et al. (2021) and Aralikkatte et al. (2021) add an external guidance signal by lexical similarity between input text and summary. Yamada et al. (2021) extracts the context word sequences from the reference to reflect some important phrases from the article. Although these external auxiliary sources provide diverse topical signals, they are cost-intensive to heuristically manipulate and have limitations in guiding the overall topical information of the article. Other approaches incorporate contrastive learning into sequence-to-sequence (seq2seq) model, allowing the model to learn topical representation of the input text (Lee et al., 2021; Liu et al., 2021; Wu et al., 2020). They explicitly constructs positive or negative inputs to introduce contrastive loss as an augmentation of MLE training.

In this work, we propose a novel framework for generating compelling news subheadings by discriminating whether each token in the reconstructed headline is the same as the token in the original headline. Unlike previous approaches that use heuristically extracted topical information or positive and negative pairs, we utilize *headline* that fundamentally implies the topic of the entire article. We make full use of this indispensable object as a guide signal through token-based discriminative learning. We conducted comparative experiments on three datasets written in English or Korean to evaluate the performance of our model and verified our model through additional qualitative analysis with human evaluations.

2 Datasets

We used one English news summarization dataset and two Korean news summarization datasets.

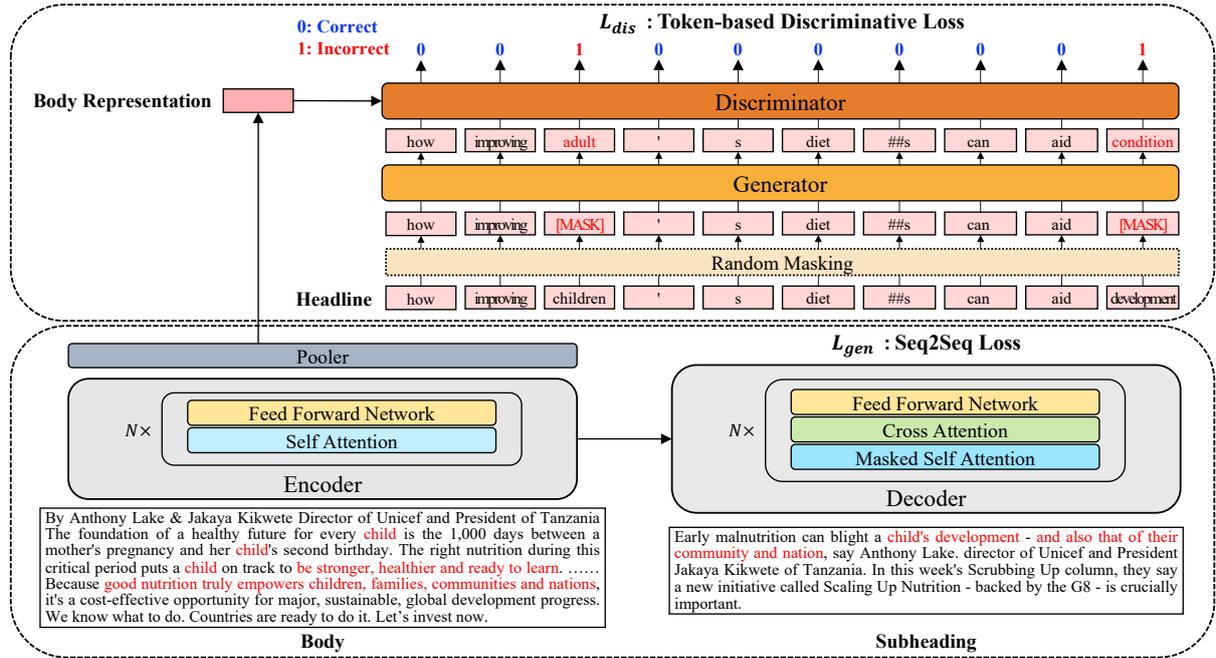


Figure 1: Framework of the proposed model. The lower part of the figure represents subheading generation, and the upper part of the figure represents token-based discriminative learning. Prediction results of discriminator encourage encoder to focus on the topical information.

2.1 XLSum

XLSum (Hasan et al., 2021) is a highly abstract multilingual news summarization dataset containing online articles crawled from the British Broadcasting Corporation (BBC). They regard a bold paragraph containing one or two sentences at the beginning of each article as a summary, a subheading. We use the English and Korean versions with train set, valid set, and test set pairs of $\{306522, 11535, 11535\}$ and $\{4407, 550, 550\}$, respectively.

2.2 YonhapNews

However, the size of the training dataset in XLSum’s Korean version is insufficient for fine-tuning. We construct a dataset from YonhapNews, one of the most reliable news outlets in South Korea, to evaluate the performance of the model with a sizable Korean dataset. In YonhapNews, the subheading is located right below the headline to condense the body in abstractly, like the BBC. Train set, valid set, and test sets in YonhapNews are $\{208750, 26094, 26094\}$ and will be released for academic use.¹

¹<https://github.com/Lainshower/Subheading-Gen>

3 Proposed Method

As shown in Figure 1, our proposed model consists of subheading generation (bottom) and token-based discriminative learning (top) parts. The loss occurring in each part is defined as L_{gen} and L_{dis} , and the model is trained by minimizing the following loss:

$$L = L_{gen} + \lambda \cdot L_{dis}, \quad (1)$$

where λ is a weighted hyperparameter for the two losses.

3.1 Subheading Generation

We use BART (Lewis et al., 2020) as our seq2seq model, where the encoder takes a body B as an input, generates an input representation of the body, and passes it to the decoder, which outputs the subheading \hat{S} . The loss of subheading generation is as follows:

$$L_{gen} = - \sum_{i=1}^M \log(p(s_i | s_{1:i-1}, B; \theta)), \quad (2)$$

where a body $B = (b_1, b_2, \dots, b_N)$ and its subheading $S = (s_1, s_2, \dots, s_M)$ consists of N token vectors b_i and M token vectors s_j , respectively, and the seq2seq model is optimized to learn the θ parameters to minimize the negative log-likelihood.

Dataset	Model	Size	Rouge-1	Rouge-2	Rouge-L	BERTScore
XLSum-ENG	BART	139M	36.52	15.33	30.49	77.68
	MT5	582M	37.81	14.59	28.30	74.48
	T5	220M	37.87	15.97	29.09	75.96
	Pegasus	571M	39.57	16.63	32.45	77.93
	T5 w/multi	247M	35.37	14.71	29.84	75.74
	Ours	282M	39.84	18.07	33.77	78.96
XLSum-KOR	BART	124M	25.23	15.91	23.37	74.60
	MT5	582M	29.16	14.03	25.40	69.94
	T5	247M	13.99	4.10	13.48	67.92
	T5 w/multi	247M	17.55	7.03	16.60	71.58
	Ours	269M	27.41	17.56	25.48	75.58
YonhapNews	BART	124M	21.41	11.00	19.21	72.19
	MT5	582M	25.24	11.28	21.13	70.70
	T5	247M	20.10	8.70	17.96	71.67
	T5 w/multi	247M	22.21	9.91	19.91	71.21
	Ours	269M	24.15	13.21	21.93	73.08

Table 1: Subheading generation performance for each dataset. Size represents the number of parameters in each model. Our model outperforms the comparative models except for Rouge-1 score for XLSum-KOR and YonhapNews.

3.2 Token-based Discriminative Learning

Inspiring by [Chuang et al. \(2022\)](#), we inject topical knowledge of the article by discriminating whether tokens in the reconstructed headline are the same as the original. As described in Appendix A, headline has high lexical similarity with subheading compared to other objects in the article. Therefore, headline reconstruction helps model to aware of topical information which is inherent in the headline. The loss of token-based discriminative learning is as follows:

$$L_{dis} = \sum_{i=1}^L \left(-\mathbb{1}(h'_i = h_i) \log D(H'|B, H) - \mathbb{1}(h'_i \neq h_i) \log(1 - D(H'|B, H)) \right), \quad (3)$$

where a headline $H = (h_1, h_2, \dots, h_L)$ consist of L tokens and the reconstructed headline H' is $H' = G(H_{masked})$ where G is the generator and a masked headline H_{masked} is obtained with random mask $M = [m_1, m_2, \dots, m_L], m_t \in [0, 1], H_{masked} = H \cdot M$. Using the compressed body representation of the encoder, the discriminator D predicts whether the tokens in the reconstructed headline H' are the same as the original headline H . As shown in Figure 1, the generator predicts the masked headline into “*how improving adult’s diet can aid condition*”. Using the encoded body representation and partially mispredicted headline, the model trains the incorrectly predicted “*adult*”, and “*condition*” and the correctly predicted rest of the tokens, respectively. Back-propagated gradients of the discriminator D cause the encoder to include the topical information of the article in the body representation by classifying whether the tokens

in the reconstructed headline H' come from the original headline or not.

Model	Generated Subheading
BART	The UK’s oil and gas industry generated negative tax receipts in 2015-16, according to HM Revenue and Customs (HMRC).
MT5	Have led to a fall in tax receipts from UK oil and gas production, according to HM Revenue and Customs (HRMC)
T5	revenues have fallen to their lowest level since records began in the 1960s, according to new figures from HM Revenue and Customs (HMRC)
Pegasus	tax receipts from oil and gas production in the UK have fallen to their lowest level, according to HM Revenue and Customs (HMRC)
Ours	Revenues from the North Sea oil and gas industry have fallen to their lowest level since records began, according to HM Revenue and Customs (HMRC).

Table 2: Example of generated subheading for each model. The original headline is “*North Sea receipts hit record low*” and the reference subheading is “*The UK government has incurred a loss from North Sea oil and gas production for the first time since records began nearly 50 years ago*”. The body of the article can be found in the XLSum-ENG test set with the corresponding id= ‘uk-scotland-scotland-business-36388621’.

4 Experiments

4.1 Experimental Setting

We use pretrained models BART and ELECTRA ([Clark et al., 2020](#)). Unlike [CLS] representation of BERT ([Devlin et al., 2019](#)), BART doesn’t have a special input representation token. As such, we use an average pooler to compress the output of the encoder and freeze the generator to keep generating noise headline for token-based discriminative learning. Optimal parameters

were obtained in the search spaces with learning rate $\{1e-5, 2e-5, 3e-5, 4e-5\}$, masking ratio $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, and lambda $\{0.1, 0.01\}$.

4.2 Comparative Models

BART, T5 (Raffel et al., 2020), and MT5, (Xue et al., 2021) were used as comparative models in all datasets. Also, Pegasus (Zhang et al., 2020) for English were used as comparative models. For a fair comparison, we use the concatenated body with a headline in the input of the comparative models.

4.3 Experimental Results

Table 1 shows the results of subheading generation performance for each dataset. Model performance was evaluated using Rouge (Lin, 2004) and BERTScore (Zhang et al., 2019). In XLSum-ENG, our model outperforms all comparative models. In particular, our model performs better than MT5 or Pegasus, which have more than double the model size. In other words, token-based discriminative learning can improve generation performance more efficiently than simple concatenation. Our model outperforms the comparative models in all other metrics except the Rouge-1 score for the Korean language datasets. MT5 records the highest Rouge-1 score on both Korean datasets. However, because Korean is decomposed into many sub-words due to its morphological richness, it is not suitable to evaluate performance with Rouge-1 score alone. In particular, in terms of BERTScore, our model scored 5.64% and 2.32% higher than MT5 in XLSum-KOR and YonhapNews, respectively. This indicates that our model can generate semantically relevant subheadings. Moreover, good performance on small datasets (i.e., XLSum-KOR) demonstrate the robustness of our model.

Table 2 shows an example of the generated subheadings for each model. We can see that our model utilizes “North Sea” and “record low” from the headline to better condense topical information in the article. Additional qualitative results are described in Appendix C.

	XLSum-ENG	XLSum-KOR	YonhapNews
BART	2.53 (0.64)	2.00 (0.38)	3.53 (0.92)
MT5	2.71 (0.61)	2.07 (0.59)	2.60 (0.51)
T5	2.64 (0.74)	2.20 (0.41)	2.73 (0.46)
Pegasus	3.64 (0.50)	-	-
Ours	4.21 (0.70)	3.67 (0.49)	3.40 (0.74)

Table 3: The average score of human evaluation for XLSum-ENG, XLSum-KOR, and YonhapNews. Numbers in parentheses indicate the standard deviations.

4.4 Human Evaluations

We conduct human evaluations to verify whether the subheadings of the proposed method are more topically relevant than the baselines. Three samples were randomly selected from each test dataset, and subheadings generated along with their corresponding headlines and body were shown to the workers and evaluated on a five-point Likert scale. Table 3 shows that our model generates topic-relevant subheadings better than the baselines on two datasets, and is particularly robust on a small dataset (i.e., XLSum-KOR). In the case of YonhapNews, BART showed the highest score, but the independent t-test showed that the average difference between Ours and BART was insignificant ($p > 0.663$).

4.5 Comparison with Multi-task Learning

To verify whether our model effectively learns topical information from the headlines, we conduct additional experiments with a multi-task learning. Different prefixes were used to know the model what the current training task is. One task maps news body text to subheading, and the other maps news body text to headline. We experiment with T5 because it has less discrepancy with our pre-training objectives. T5 w/multi rows in Table 1 show the multi-task learning results, demonstrating that our method is more effective in learning headline information.

4.6 Ablation Studies

We perform ablation studies in terms of masking ratio to analyze the effectiveness of token-based discriminative learning. Figure 2 shows the results of the Rouge-2 score and BERTScore according to the masking ratio for each dataset. We also plot the performance of the T5 with similar parameter sizes to ours. Our model outperforms T5 in all masking ratio ranges. This indicates that our model is not significantly sensitive to masking ratio. In particular, the large performance difference of XLSum-KOR demonstrates the robustness of our model on the small dataset. The original headline is completely incorrectly reconstructed if the masking ratio exceeds 0.4, limiting ability of the model to learn crucial topical information from the headline considering the token length of the headline. However, for small masking ratios such as 0.1, the generator can completely reconstruct the original headline, but it is limited in maximizing the benefits of token-based discriminative learning. Headline to-

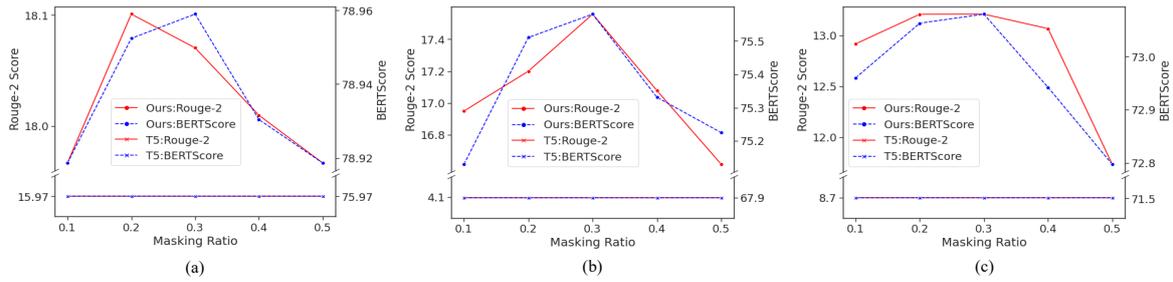


Figure 2: Rouge-2 score and BERTScore according to masking ratio for (a) XLSum-ENG, (b) XLSum-KOR, and (c) YonhapNews. The performance of the T5 in two Korean datasets is plotted with a ‘x’ marker. The masking ratio between 0.2 and 0.3 shows the best performance in all datasets.

ken length distribution is described in Appendix B. Therefore, it is recommended to set the masking ratio between 0.2 and 0.3 in order to utilize the token-based discriminative learning for subheading generation entirely.

5 Conclusions

In this paper, we propose a novel model for generating a subheading for news article. Along with token-based discriminative learning, our model can effectively utilize topical information from a headline that is essential in articles and does not require additional manipulated information. Experiments on three datasets written in two different languages show the effectiveness of the proposed model. Also, qualitative results and human evaluation show that the overall quality of generated subheadings is improved compared to comparative models. We expect that our model will be extended in future research to an abstractive summarization task that include both a headline and a body text, such as legal texts or papers.

Limitations

Our study outperformed all comparative models in generating subheadings through token-based discriminative learning. However, the experiments mainly used limited languages such as English and Korean due to a lack of large-scale multilingual training data and the need for significant GPU resources. We, therefore, encourage further investigations to expand the versatility of the proposed model by utilizing large-scale multilingual language datasets to verify expandable applications in various morphological characteristics.

Ethics Statement

As YonhapNews is one of the most reliable media outlets in South Korea, articles from YonhapNews are published through a rigorous verification process and will be deleted or revised if they contain any form of bias. However, the period of data collected is three years, and there may be past article content that has not been modified by new facts, so we cannot guarantee that all articles in YonhapNews dataset are completely unbiased. Nevertheless, this dataset has sufficient potential to develop into various studies and thus is released for academic uses.

Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) (Project Number: 2021-0-00469) and the Technology Innovation Program (or Industrial Strategic Technology Development Program-Knowledge service industry technology development project - service core technology development) funded by the Ministry of Trade, Industry & Energy(MOTIE, Korea) (Project Number: 20018758).

References

- Rahul Aralikkatte, Shashi Narayan, Joshua Maynez, Sascha Rothe, and Ryan McDonald. 2021. [Focus attention: Promoting faithfulness and diversity in summarization](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6078–6095, Online. Association for Computational Linguistics.
- Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljagic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. [DiffCSE: Difference-based contrastive learning for](#)

- sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Daniel Dor. 2003. On newspaper headlines as relevance optimizers. *Journal of pragmatics*, 35(5):695–721.
- Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.
- Ullrich KH Ecker, Stephan Lewandowsky, Ee Pin Chang, and Rekha Pillai. 2014. The effects of subtle misinformation in news headlines. *Journal of experimental psychology: applied*, 20(4):323.
- Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XLsum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.
- Elly Ifantidou. 2009. Newspaper headlines and relevance: Ad hoc concepts in ad hoc contexts. *Journal of Pragmatics*, 41(4):699–720.
- Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021. Contrastive learning with adversarial perturbations for conditional text generation. In *International Conference on Learning Representations*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Junpeng Liu, Yanyan Zou, Hainan Zhang, Hongshen Chen, Zhuoye Ding, Caixia Yuan, and Xiaojie Wang. 2021. Topic-aware contrastive learning for abstractive dialogue summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1229–1243, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Hanlu Wu, Tengfei Ma, Lingfei Wu, Tariro Manyumwa, and Shouling Ji. 2020. Unsupervised reference-free summary quality evaluation via contrastive learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3612–3621, Online. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Kosuke Yamada, Yuta Hitomi, Hideaki Tamori, Ryohhei Sasano, Naoaki Okazaki, Kentaro Inui, and Koichi Takeda. 2021. Transformer-based lexically constrained headline generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4085–4090, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Similarity between Subheading and Headline

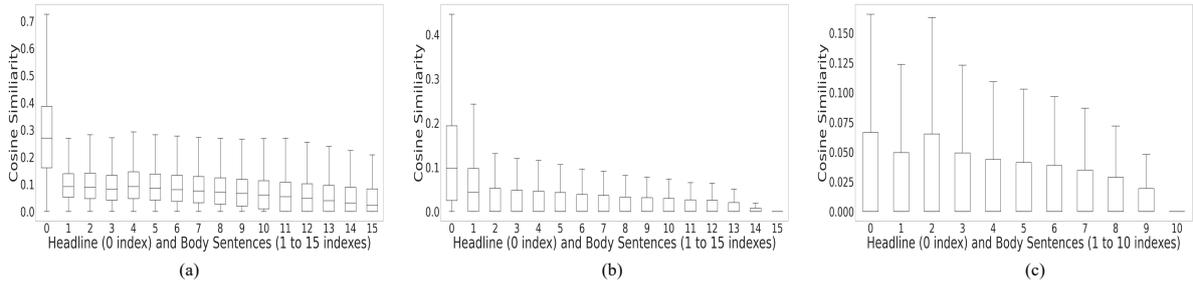


Figure A.1: Cosine similarity between the TF-IDF values of subheading-headline (0 index in X-axis) and between the TF-IDF values of the subheading-body sentences (1 to N indexes in X-axis) of the news article. Each figure (a), (b), and (c) represents for dataset XLSum-ENG, XLSum-KOR, and YonhapNews, respectively.

B Length of Headline

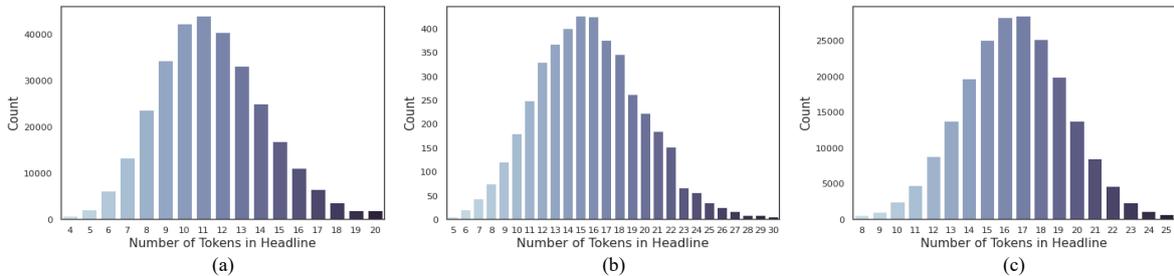


Figure B.1: Distribution of the token length in the headline of (a) XLSum-ENG, (b) XLSum-KOR, and (c) YonhapNews.

C Qualitative Results

Model	Generated Subheading
BART	신종 코로나바이러스 감염증(코로나19) 백신 구매 계약을 맺은 영국 정부가 존슨 총리에게 코백스를 통해 백신을 기부할 것을 촉구했다. (The British government, which signed a contract to purchase a new coronavirus infection (COVID-19) vaccine, urged Prime Minister Johnson to donate the vaccine through COVAX.)
MT5	단체들이 신종 코로나바이러스 감염증(코로나19) 백신을 구매한 후에도 잉여 물량의 대부분을 공유할 것이라고 촉구했다. (Organizations urged that they will share most of the surplus even after purchasing a new coronavirus infection (COVID-19) vaccine.)
T5	신종 코로나바이러스 감염증(코로나19) 백신을 사용할 수 있다고 밝혔다. (It was announced that a new coronavirus infection (COVID-19) vaccine can be used.)
Ours	영국이 신종 코로나바이러스 감염증(코로나19) 백신의 잉여 물량을 저개발국에 기부하겠다고 밝혔다. (The <u>UK</u> has announced that it will <u>donate remaining of COVID-19 vaccine</u> to underdeveloped countries.)

Table C.1: Example of generated subheading for each model in XLSum-KOR. The original headline is “코로나19 백신: 전국민 접종하고도 1억 도즈 남는 영국... ‘남는 백신 기부할 것’ (COVID-19 Vaccine: 100 million doses of UK left after national vaccination... ‘Donate the remaining vaccines.’)” and the reference subheading is “국제 구호 단체들이 보리스 존슨 영국 총리에게 영국이 저개발국가에 기부할 수 있는 코로나19 백신이 얼마나 되는지를 조속히 밝힐 것을 촉구하고 있다. (International aid organizations are urging British Prime Minister Boris Johnson to reveal as soon as possible how many COVID-19 vaccines Britain can donate to underdeveloped countries.)” The body of the article can be found in the test set with corresponding id=‘international-56553770’.

Model	Generated Subheading
BART	11억원 기부...군부대도 방역에 힘 보탤다 (Donating 11 billion won ... The military unit also helped with quarantine)
MT5	잇단 후원...경북도교육청, 장병 130여명 투입해 방역 지원 (Continuous sponsorship ... Gyeongbuk Provincial Office of Education dispatched 130 soldiers to provide quarantine support)
T5	각계서 정금·성금 잇따라 (a series of donations from all walks of life)
Ours	착한 임대인 운동 확산...제201특공여단, 경산역·버스터미널 방역 지원 (Spreading the Good Renters Movement ... The 201st Special Forces Brigade, provide quarantine support to <u>Gyeongsan</u> Station and Bus Terminal)

Table C.2: Example of generated subheading for each model in YonhapNews. The original headline is “‘코로나19 극복 함께해요’...대구·경북에 성금·물품 답지 (‘Let’s overcome COVID-19 together’...Daegu and Gyeongsangbuk-do collect donations to deliver goods)” and the reference subheading is “착한 임대인 운동 공공기관에 확산...군부대는 방역 지원 (The Good Renters Movement Spreads to Public Institutions... Military units provide quarantine support)”. This news article covers the quarantine support from various industry fields, including the rental industry and the military service, in response to the COVID-19 in Daegu and Gyeongsangbuk-do, city and state located in Korea. Subheading generated by our model include “Gyeongsan”, located in Gyeongsang-do, showing that it reflects the locational information of quarantine support that occurs through the headline. The body of the article can be found in ‘<https://www.yna.co.kr/view/AKR2020030914600053>’

Decipherment as Regression: Solving Historical Substitution Ciphers by Learning Symbol Recurrence Relations

Nishant Kambhatla Logan Born Anoop Sarkar

School of Computing Science, Simon Fraser University

8888 University Drive, Burnaby BC, Canada

{nkambhat, loborn, anoop}@sfu.ca

Abstract

Solving substitution ciphers involves mapping sequences of cipher symbols to fluent text in a target language. This has conventionally been formulated as a search problem, to find the decipherment key using a character-level language model to constrain the search space. This work instead frames decipherment as a sequence prediction task, using a Transformer-based causal language model to learn recurrences between characters in a ciphertext. We introduce a novel technique for transcribing arbitrary substitution ciphers into a common *recurrence encoding*. By leveraging this technique, we (i) create a large synthetic dataset of homophonic ciphers using random keys, and (ii) train a decipherment model that predicts the plaintext sequence given a recurrence-encoded ciphertext. Our method achieves strong results on synthetic 1:1 and homophonic ciphers, and cracks several real historic homophonic ciphers. Our analysis shows that the model learns recurrence relations between cipher symbols and recovers decipherment keys in its self-attention.¹

1 Introduction

Text may be considered a special kind of recurrent sequence, where letters repeat at intervals which conform to a language’s character n -gram distribution. The hidden mapping between cipher text and plain text can be viewed as a model that predicts this recurrent sequence. Can we use self-attention to recover the mapping from a sequence of recurrent symbols to crack the cipher?

In this work, we exploit this idea for decipherment by building upon recent successes of Transformer models (Vaswani et al., 2017) in reasoning-based regression tasks such as mathematical reasoning (Saxton et al., 2019; Li et al., 2021) and learning the mathematical function for recurrent

¹https://github.com/protonish/decipher_symbol_recurrence

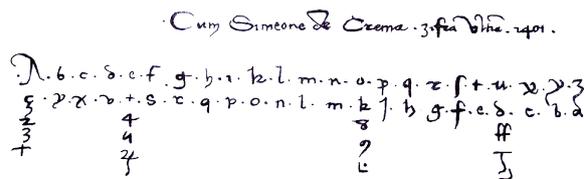


Figure 1: The homophonic substitution key for the *Simeone de Crema* written in Mantua in 1401 AD. The top line maps each character in the alphabet to its reversed-alphabet equivalent; each vowel is substituted by three additional symbols.

sequences (D’Ascoli et al., 2022). We rethink decipherment as a regression task that predicts a natural language plaintext by learning a recurrence relation between integer-coded ciphertext symbols.

There exist large collections of historical ciphers (see de-crypt.org)², in the form of encrypted letters and more informal communications, of which many remain undeciphered. Many of these texts employ complex *homophonic substitution ciphers*, which mask the frequencies of letters by using a larger alphabet than the underlying language. Figure 1 shows the first known homophonic cipher from 1401 AD³. Automated computational decipherment of such texts is challenging (Pettersson and Megyesi, 2019; Megyesi et al., 2020). Prior work has mainly focused on using clever heuristics and/or search algorithms to explore the space of cipher keys and score multiple candidate plaintexts under character language models (LMs) (Knight et al., 2006; Corlett and Penn, 2010; Hauer et al., 2014; Berg-Kirkpatrick and Klein, 2013; Nuhn et al., 2013, 2014; Kambhatla et al., 2018) In contrast Aldarrab and May (2021) train a sequence-to-sequence model to solve simple (one-to-one) substitution ciphers. This approach, however, cannot solve complex homophonic ciphers as it relies on frequency information which such ciphers obscure.

In this paper, in a departure from frequency and

²<https://de-crypt.org/decrypt-web/RecordsList>

³https://en.wikipedia.org/wiki/Francesco_I_Gonzaga

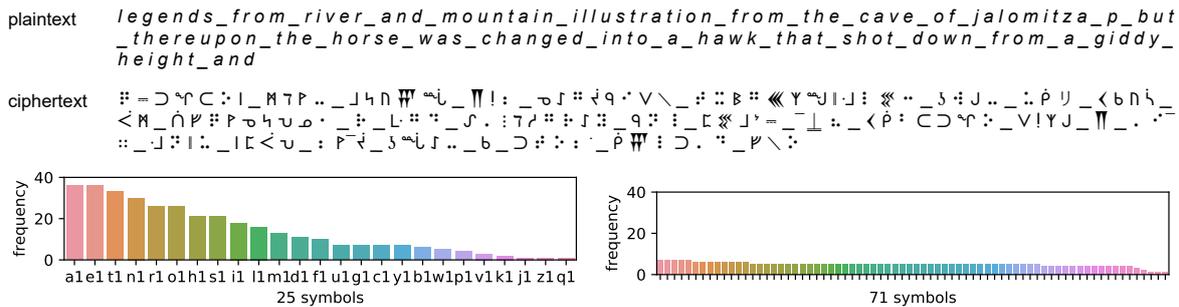


Figure 2: Part of an arbitrary 424 characters long homophonic cipher with 71 symbols and its plaintext. **Bottom** row juxtaposes symbol frequencies in 1:1 vs homophonic encipherments of the plaintext.

heuristic search-based techniques, we make the following **contributions**: ① We create a sequence-to-sequence dataset comprising 2 million unique homophonic ciphers and their plaintext. ② We propose a novel *recurrence encoding* which encodes information about the position and repetition of symbols in a cipher. This can be applied to both 1:1 and homophonic ciphers. ③ This encoding allows us to treat decipherment as a sequence prediction task conditioned on an integer sequence. We introduce a novel approach to solving homophonic ciphers by using a Transformer LM to translate integer-encoded ciphertexts to plaintexts. We also provide exhaustive analysis to show the strengths of our model. ④ We demonstrate near-perfect results on synthetic homophonic ciphers. Additionally, we show fully automated decipherment of TNA_SP106/5 and BnF-f01, two real historical ciphers.

Our analysis shows that reproducing the input ciphertext before generating the plaintext helps our model to learn the relations between cipher symbols. This enables it to implicitly learn the decipherment key with high accuracy, and to determine which symbols are homophones of one another. The decipherment is highly constrained by this implicit key even in the face of disfluent plaintexts, and as a result our model is able to produce decipherments into Latin and late Middle/early modern English, despite being trained on modern English.

2 Decipherment of Substitution Ciphers

A 1:1 or *simple substitution cipher*, the oldest known technique for obscuring written information, defines a 1-1 mapping between plaintext characters and ciphertext symbols. This mapping can easily be broken with frequency analysis (Hauer et al., 2014; Kambhatla et al., 2018; Aldarrab and May,

Method	Search	Train
❄ <i>n</i> -gram LM (2010)	A*	✗
❄ LM + Bay. Inf.(2011)	sampling	✗
❄ LM + HMM (2013)	EM; 1M restarts	✓
❄ <i>n</i> -gram LM (2013; 2014)	beam	✗
❄ lstm LM (2018)	beam	✗
🔥 Generative LM (Ours)	✗	✓

Table 1: Summary of different methods used for solving homophonic ciphers. Prior approaches are predominantly search-based and use a frozen language model to score partial candidate hypotheses.

2021) which leverages the fact that these ciphers preserve the distribution of character frequencies in the underlying language.

Homophonic ciphers are substitution ciphers where one plaintext character may be encoded by more than one ciphertext symbol. In this way, frequent plaintext characters can be mapped to many infrequent ciphertext symbols, resulting in a flattened frequency distribution (Figure 2).

2.1 Background

Traditional approaches to natural language decipherment of homophonic substitution ciphers—the main focus of this work—are entirely search-based (Table 1). Nuhn et al. (2013) perform a beam search using an offline, frozen character language model to score candidate decipherments, and Nuhn et al. (2014) improve the rest-cost estimation for this technique. Kambhatla et al. (2018) further improve the rest-cost heuristic by using a frozen neural LM to score hypotheses. Corlett and Penn (2010), on the other hand, use A* search. Berg-Kirkpatrick and Klein 2013 uses 1 million random restarts to learn HMMs for decipherment.

The ability of such inference-only methods to generalize can be limited, as it depends on the underlying language model that is used to score the

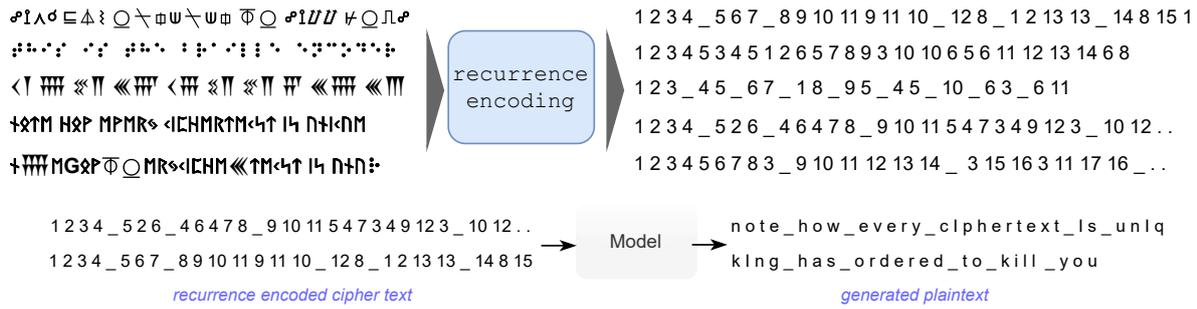


Figure 3: **Top-left:** Before *recurrence encoding*. Every ciphertext is unique with a different key, with distinct keys and plaintexts, and varying lengths. There is no relation between identical symbols in different ciphers (e.g. Φ in cipher 1 and 5). These dissimilarities make it nearly impossible to train a model to generalize to new, unseen homophonic ciphers. **Top-right:** After *recurrence encoding*. Arbitrary ciphertexts are converted to recurrent integer sequences. The encoding is applied to each cipher independently: the same symbol may receive different encodings in different ciphertexts depending on where it first occurs, and two ciphers may receive the same encoding despite using different alphabets. **Bottom:** Recurrence-encoded symbols decipher to different values depending on context.

constructed hypotheses. Even an efficient search can take a long time to find the whole cipher key (Nuhn et al., 2013; Kambhatla et al., 2018).

3 Our Generative Decipherment Model

We frame decipherment as a novel sequence generation task—we train a generative language model that learns the relations between recurring symbols in a homophonic cipher and generates the corresponding deciphered plaintext message.

3.1 Converting Arbitrary Ciphers into Recurrent Integer Sequences

In a substitution cipher, any character may be substituted for any other, limiting what one can generalize between different ciphertexts (Figure 3). However, unrelated ciphers may still exhibit similar patterns of letter distribution and repetition and display latent characteristics of the plaintext language. For example, the characters at the beginning of two unrelated ciphers are likely drawn from the same distribution of word-initial letters in the underlying plaintext language. So we can generalize better by treating ciphers as recurrent sequences.

We propose a novel *recurrence encoding* to highlight where cipher symbols first occur and how they are repeated within a ciphertext. This encoding replaces the n th unique symbol in a ciphertext with the number n wherever that symbol occurs (Figure 3). This converts arbitrary ciphertexts into integer sequences, and thus provides a coherent connection between ciphers with distinct keys or disjoint alphabets.

3.2 Modelling Symbol Recurrence Relations

In this section, *ciphertext* specifically refers to a recurrence-encoded integer sequence.

Following Wang et al. (2021) and Zhang et al. (2022) in MT, we use a causal language model (LM) as a replacement for a Transformer-based encoder-decoder model (Vaswani et al., 2017). For a source sequence X and the target Y ,

$$[X^l, Y^l] = \text{FFN} \circ \text{SelfAttn} \left([X^{l-1}, Y^{l-1}], \text{Mask} \right) \quad (1)$$

where l is the layer index, FFN is a feed-forward network, and *Mask* denotes the attention mask. Our **CausalLM** model is a unidirectional LM, with causal masking over both the source and the target. This optimizes the joint distribution of cipher (src) and plaintext (tgt) sequences:

$$\begin{aligned} \mathcal{L}^{\text{CLM}}(X, Y) &= \mathcal{L}^{\text{SRC}} + \mathcal{L}^{\text{TGT}} \\ &= -\log P(X) - \log P(Y|X) \end{aligned} \quad (2)$$

CausalLM is therefore forced to sequentially predict the ciphertext just as it predicts the plaintext. This formulation encourages the model to learn a coherent relationship between the ciphertext and plaintext characters within each training sample.

Baseline Models. To understand the importance of causal attention masking, we also consider the following models which do not generate the ciphertext:

Seq2Seq Following (Aldarrab and May, 2021), this is a character level Transformer architecture that is only optimized on the target-side (plaintext) loss: $\mathcal{L}^{\text{Seq2Seq}}(X, Y) = \mathcal{L}^{\text{TGT}} = -\log P(Y|X)$

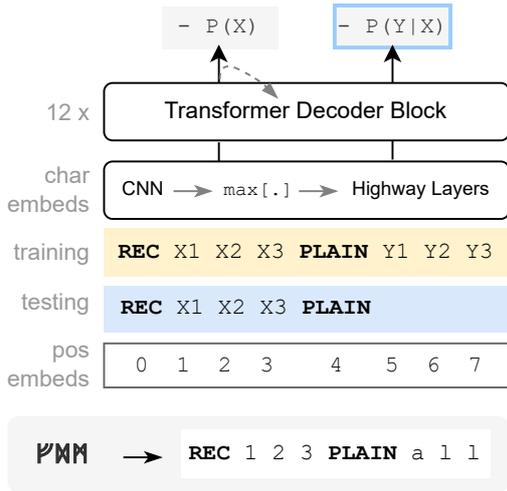


Figure 4: Schematic depiction of our Transformer LM model for an example ciphertext. X denotes the recurrence encoded ciphertext and Y is the plaintext, prepended by REC and PLAIN tags respectively.

Target-Only CausalLM is only optimized on the target-side loss \mathcal{L}^{TGT} , and incurs no loss when generating the source text.

PrefixLM combines SelfAttn and masked-SelfAttn: the ciphertext is attended at all times, while the plaintext uses a causal mask:

$$Mask(i, j) = 1, \text{ if } i \geq j \text{ or } j \leq |X|; \text{ else } 0 \quad (3)$$

where $1 \leq i, j \leq (|X| + |Y|)$. This setting mimics an encoder-decoder by modelling the conditional distribution of the plaintext target given the ciphertext source with target-only objective $\mathcal{L}^{PLM}(X, Y) = \mathcal{L}^{TGT}$.

All models build character embeddings with a convolutional neural network and highway networks over character inputs (Kim et al., 2016).

4 Experimental Setup

4.1 Data

Length	Keys	#Train	#Valid	#Test
300-400	30-45	460,467	25,582	25,581
	45-60	503,695	25,582	25,581
	30-85	542,611	25,582	25,581
300-700	30-45	460,467	25,582	25,581
	45-60	542,611	25,582	25,581
	30-85	1,046,306	25,582	25,581

Table 2: Summary of the synthetic homophonic ciphers used in our experiments. All ciphers are unique.

We extract 1000 English books from Project

Algorithm 1 Allocate Homophonic Symbols

Plaintext sample y of length n
 Plaintext chars, $y_{freq} = \text{Counter}(y).\text{most_common}()$
 Approx. cipher symbols, $\#\text{sym}$

```

procedure HOMOPHONIC( $y_{freq}, n, \#\text{sym}$ )
  sym_count = 0  $\triangleright$  final num. of cipher symbols
  sym_per_char = dict()  $\triangleright$  num. symbols per plain char
  for char, freq in  $y_{freq}$  do
    char_weight = int(freq / n)
     $w_{sym} = \text{int}(\text{char\_weight} * \#\text{sym})$ 

    num_sym =  $\begin{cases} 1, & w_{sym} == 0 \\ w_{sym}, & \text{otherwise} \end{cases}$ 

    sym_count += num_sym
    sym_per_char[char] = num_sym
  return sym_count, sym_per_char

```

Gutenberg⁴ to create training, validation and test sets. We also use $\sim 200k$ English sentences from news-commentary v9 from WMT14 En-De. Combining these, we generate homophonic ciphers with lengths and keys summarized in Table 2.

Synthetic Homophonic Ciphers. To train a model that can generalize to unseen ciphers, we first generate synthetic homophonic ciphers using Algorithm 1 to flatten the frequency distribution of a text. This technique allocates multiple less-frequent ciphertext symbols to common plaintext letters to yield strong homophonic ciphers.

For simple substitution ciphers, we use the same English data as above to create 1.2M synthetic substitution ciphers with lengths up to 256. Following previous work on 1:1 ciphers (Nuhn et al., 2013; Kambhatla et al., 2018; Aldarrab and May, 2021), we evaluate on 50 test ciphers of lengths up to 128 (16,32,64) and beyond 128 (128,256) from the Wikipedia page on History⁵. All our experimental settings include data with word boundaries denoted by the space symbol ($_$). We train our multilingual model on length 256 1:1 ciphers from the 13 language data in Aldarrab and May (2021)⁶ which includes training, validation and test splits.

4.2 Model Details

Our main model uses a Transformer decoder-based auto-regressive language model. Our model comprises a 12 layer decoder with 12 attention heads and a feed-forward dim. of 1536, totalling 23M trainable parameters. We use character filters of

⁴<https://github.com/pgcorpus/gutenberg>

⁵<https://en.wikipedia.org/wiki/History>

⁶<https://github.com/NadaAldarrab/s2s-decipherment>

[(1, 64), (2, 128), (3, 192), (4, 256)] with a character dim of 4 and 2 highway layers. For the seq2seq model, we implement a 6 layer encoder-decoder Transformer with the same settings as above. All models are implemented using the fairseq toolkit (Ott et al., 2019).

All models train for about 30 iterations over the data at ~ 100 minutes per epoch on 4xA6000 GPUs. Inference uses a beam size of 200 unless otherwise stated. Inference speed is about 400 chars/second on a single Titan RTX.

Evaluation Following prior work (Kambhatla et al., 2018; Aldarrab and May, 2021), we evaluate on Symbol Error Rate (SER), the proportion of ciphertext symbols which are wrongly recovered.

5 Homophonic Substitution Ciphers

We experiment on solving our own synthetic homophonic ciphers, and on automatically deciphering two real world homophonic ciphers that have previously only been cracked manually.

5.1 Results on Synthetic Ciphers

Table 3 reports results on synthetic homophonic data using recurrence encoding. On 400- and 700-character long ciphers, our best model with causal attention and up to 65 keys achieves near perfect decipherment. As expected, we observe the best results on long ciphers, which provide more context. Even on challenging ciphers with up to 85 keys over 400 characters (4.5 chars/symbol), our model attains an average error rate of 2.25%, averaging only 1 wrong character each. As will be shown in Section 8 (Table 7), our model also implicitly recovers decipherment keys with $> 98\%$ accuracy.

#keys	Model	Max Len.	
		400	700
30-45	Seq-to-Seq	72.30	fail
	PrefixLM	54.73	69.50
	CausalLM (tgt)	29.99	37.20
	CausalLM	0.40	0.21
40-65	PrefixLM	69.50	54.73
	CausalLM (tgt)	29.99	37.20
	CausalLM	0.83	0.80
30-85	PrefixLM	70.52	71.82
	CausalLM (tgt)	42.05	42.69
	CausalLM	2.25	2.19

Table 3: SER% on synthetic, homophonic, recurrence-encoded ciphers. All plaintexts use 26 unique letters.

These results show the strength of recurrence encoding together with a causal LM objective.

Generating cipher + plaintext vs. plaintext only:

The best results by a significant margin are obtained through language modeling with a causal attention mask—the only model that is trained to generate the ciphertext before the intended plaintext. All other approaches fail to give adequate results in any setting. A sequence-to-sequence model obtains poor results on ciphers of 400 characters and fails to converge on 700 char long ciphers.⁷ Prefix attention is consistently worse than causal attention, and its performance varies unpredictably across input length and number of keys. Causal attention with target-only loss is worse than causal attention, suggesting that reproducing the source text may play an important role in solving this task. We consider this idea further in our analysis of the model’s attention in Section 8.

5.2 Results on Zodiac 408 Cipher

We compare our method on the famous Zodiac 408 cipher that has 408 characters written with 54 different symbols (~ 7.5 characters per symbol). For this particular cipher, all models including ours are trained on the English Gigaword corpus for fair comparison with other models in Table 4. From Table 4, our generative LM is both better, and faster by orders of magnitude.

Method	Search	SER (%)	Speed
LM+EM (2013)	1M restarts	11.0	–
n -gram LM (2013)	beam 100K	94.6	4000
	beam 1M	2.7	35000
LSTM LM (2018)	beam 100K	2.4	5600
	beam 1M	1.9	50000
Ours (greedy)	beam 1	1.9	1 sec
Ours (best)	beam 200	1.9	2 sec

Table 4: Zodiac 408. Methods for simple (1:1) substitution ciphers can not be used on Zodiac408. The last column shows inference speed in seconds. Our method is much faster because it auto-regressively generates the decipherment whereas previous methods perform an exhaustive search to find the mapping for each symbol.

5.3 Solving historical substitution ciphers

Most historical ciphers are centuries old and can be challenging to solve—the encipherment scheme may be peculiar to the author; the language may be

⁷NMT models are known to suffer from long sentences (Neishi and Yoshinaga, 2019; Varis and Bojar, 2021)

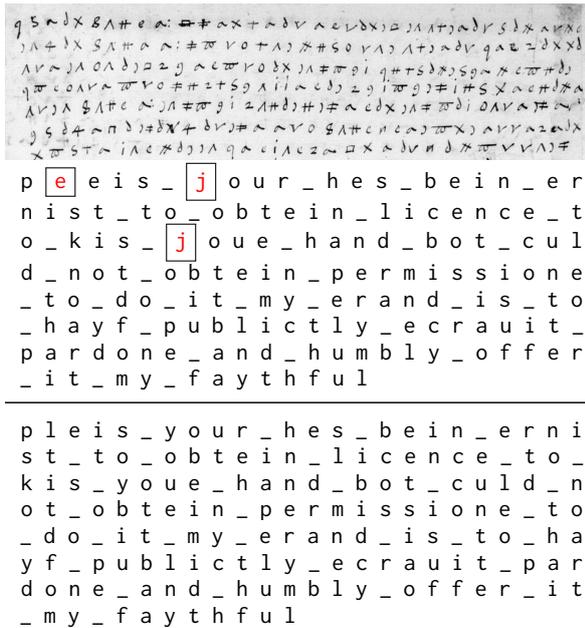


Figure 5: Predicted (middle) vs true (bottom) decipherment of the first few lines of BnF-f01 cipher (top). Errors in red and boxed.

archaic or unstandardised, and thus out-of-domain for models trained on modern data; or there may be human errors in the transcription.

Though our model is trained only on synthetic ciphers with no further finetuning, we hypothesize that it may nonetheless see success on real medieval ciphers. This section demonstrates those successes. **1. TNA_SP106/5** Also called CharlesI_(0096),⁸ this is a strong homophonic cipher that was written in 1624 in the United Kingdom, during the reign of King Charles I of England. It is a very difficult cipher which is 171 characters long, using 47 unique symbols to encipher 27 plaintext letters (each cipher symbol appears 3.6 times on average).

Using our homophonic 40-65 key model with beam size 1000, we attain a readable decipherment with a symbol error rate (SER) 18%.

2. BnF_fr2988_f01 BnF-f01⁹ (Fig. 5) is a 2 page long enciphered letter from between 1524–1549 in Italy, believed to be addressed to King Henry VIII. This homophonic cipher uses 35 symbols, with archaic spellings in the underlying plaintext that make it a challenging target due to the different character distribution from our training set. Examples include pleis→(please), faythful→(faithful),

⁸de-crypt.org/decrypt-web/RecordsView/420

⁹de-crypt.org/decrypt-web/RecordsView/2323

obtein→(obtain) and gretast→(greatest). Our best homophonic model trained with ciphers between [30-45] keys is able to crack it with SER 1.13%. The first few lines of model output and corresponding plaintext are shown in Figure 5. Although our model was never explicitly trained for robustness, the recurrence encoding helps it to overcome the unexpected plaintext distribution and maintain a consistent key to recover the message.

6 Does our technique generalize to 1:1 substitution ciphers?

To exploit the well-known weakness of simple substitution ciphers, Aldarrab and May (2021) proposed *frequency ranking* whereby cipher symbols are replaced by their frequency ranks across all 1-1 substitution ciphers. We use recurrence encoding and frequency ranking with our best performing causal LM architecture¹⁰ and compare with several baselines, including Aldarrab and May (2021).

cipher length →	<128	>128
Beam + 6-gram (Nuhn et al., 2013)	22.00	0.00
Beam + LM ((Kambhatla et al., 2018))	10.89	0.00
Beam + LM + Freq. Match (ibid.)	11.32	0.00
Seq2Seq + Freq. (Aldarrab and May)	7.68	0.00
Causal LM + Freq.	10.56	0.00
Causal LM + Rec.	11.30	0.02

Table 5: On simple substitution ciphers of length >128, our performance equals or exceeds all baselines. Freq. and Rec. denote frequency and recurrence encodings.

Our model performs well on simple substitutions (Table 6) using frequency ranking. While the scores on very short ciphers (16, 32, 64) only match the performance of beam-search based methods, on ciphers longer than 128, our model achieves close to 0 SER. Recurrence encoding is less effective in shorter sequences (< 128) and requires more context to be effective compared to frequency ranks which more directly indicate the plaintext character distribution. However, recurrence encoding is not required in this context as the character distributions are not flattened.

7 Unknown Plaintext Language

As Megyesi et al. (2020) reports, several historical ciphers in libraries and archives have no information on the plaintext language. We evaluate on the

¹⁰Recall that recurrence encoding doesn’t work well with an encoder-decoder model (Table 3).

Latin Borg Cipher¹¹, a ca. 17th century manuscript, to learn if our model can decipher without explicit knowledge of the underlying language.

7.1 Multilingual model with no language ID

Following Aldarrab and May (2021), we train our decipherment model on ciphers from 13 different languages without language ID and apply it to the Borg cipher (page 0011v). Both models are trained on frequency based encoding. Compared to 5.47% SER in the baseline, our model achieves a better SER of 4.1%.

	SER (%)
Multilingual Seq2Seq (2021)	5.47
Multilingual Causal LM (ours)	4.10

Table 6: Using our multilingual model trained on 13 languages to solve the 1:1 Borg cipher.

7.2 Zero-shot decipherment in an unseen language

Though the Borg MS uses a simple substitution cipher, the plaintext language (Latin) is out-of-domain for our main model which was trained on English only (Sec. 5). Zero-shot inference on the first 400 characters of page 0011v results using our recurrence-encoding based model in an SER of 45.14%. A mere 3 manual interventions—fixing two words (*aperitione*, *emorrhoidarum*) and correcting one (*cumo* to *cum*)—however, are sufficient to solve the entire cipher with **SER 3.89%**. See Appendix B for details on human-in-the-loop decipherment.

This shows that our model learns to consistently produce the same output for a given symbol regardless of the plaintext distribution, which is crucial for cracking ciphers and separates this from a conventional text generation model.

8 Analysis

8.1 On the significance of learning the distribution of ciphertext characters

Section 5 demonstrated that CausalLM significantly outperforms other models on synthetic homophonic data. This is the only approach which must model the distribution of ciphertext characters: Seq2Seq and PrefixLM allow the model to freely attend to the full ciphertext, while target-only loss gives no penalty for mistakes in the ciphertext. These settings remove the incentive to learn

¹¹<https://cl.lingfil.uu.se/~bea/borg/>

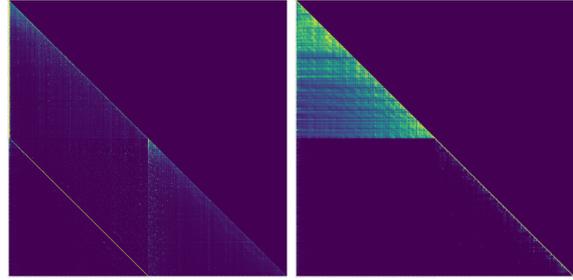


Figure 6: **Left:** self-attention map from our causal LM. **Right:** self-attention map over the same sentence from our causal LM with target-only loss.

the distribution of ciphertext characters, whereas CausalLM must sequentially predict the ciphertext just as it predicts the plaintext.

Figure 6 illustrates the impact of target-only loss by showing the final layer of self-attention scores for an example input. CausalLM exhibits a strong diagonal pattern in the lower-left of the attention matrix, showing that it attends monotonically to cipher symbols when producing corresponding plaintext symbols. With target-only loss, attention is roughly uniform over all ciphertext symbols when reproducing the input, as there is no penalty for mistakes in this section and consequently no need to attend to relevant context cues. This model does not strongly attend to the ciphertext at any point when generating the plaintext.

Key Recovery The model’s self-attention implicitly recovers decipherment keys. We construct a $n_{\text{plaintext_symbols}} \times n_{\text{cipher_symbols}}$ matrix where cell (p, c) sums the mean attention over all layers paid to c when producing p . More common symbols receive more attention, so we divide each column by the frequency of the corresponding symbol. Figure 7 depicts such a matrix, normalized so that the largest value in each row is 1: the largest value in most columns clearly corresponds to the decipherment key. Table 7 reports key error rate (KER) using this technique, as well as variants without adjusting for frequency or normal-

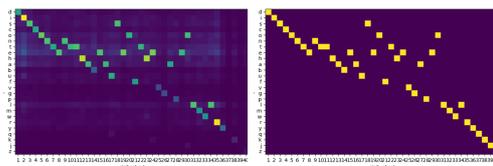


Figure 7: **Left:** Avg. attention paid to ciphertext symbols when generating plaintext symbols in one test case. **Right:** True key, where a dark cell indicates which character the symbol in each column deciphers to.

	KER (%)
Row-norm.	3.40
Col-norm.	3.63
Row-norm. + Frq. adj.	1.54
Col-norm. + Frq. adj.	3.63

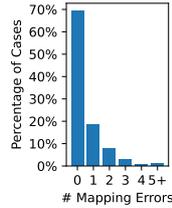


Table 7: (l) Key error rate (%) averaged over 2000 homophonic test cases. Values multiplied by 100 for readability. (r) Distribution of mapping errors.

izing by column rather than by row; the distribution of errors is shown beside the table. With our best technique, the median and mode number of erroneous mappings is 0, i.e. we perfectly recover the key from the self-attention in most cases.

Our model’s ability to accurately produce Latin highlights that it learns to obey the inferred key even in the face of conflicting signals from an out-of-domain plaintext.

8.2 Recovering character recurrence relations

Homophone Recovery Attention from ciphertext symbols to other ciphertext symbols reveals which characters are homophonic. Figure 8 (left) shows, for a sample input, the average self-attention from a ciphertext symbol towards other ciphertext symbols; Figure 8 (right) shows which of these symbols are homophones. The largest self-attention scores roughly correlate with cells representing homophonic symbol pairs. Comparing to Figure 7 (left) we see that the homophone pairs which are *not* recovered involve those symbols for which the model lacks a confident plaintext mapping.

This behaviour arises as the model reproduces the input ciphertext. This can be observed by averaging self-attention over chunks of successive time-

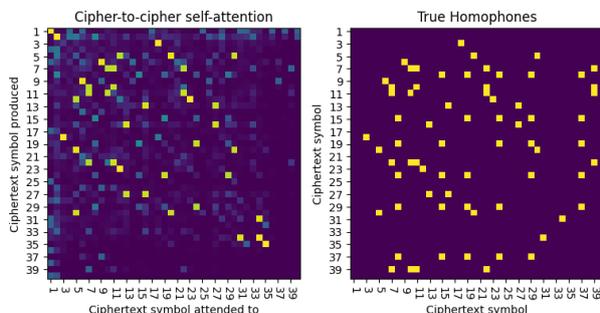


Figure 8: **Left:** Avg. attention from ciphertext symbols to other ciphertext symbols when reproducing one test case. **Right:** Dark cells indicate homophonous symbols.

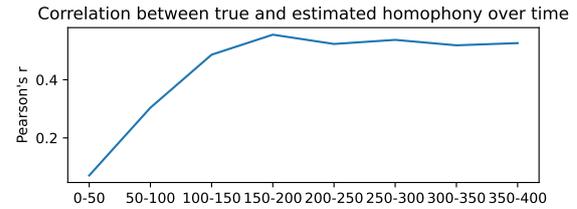


Figure 9: Pearson’s r between self-attention scores at different time-steps and reference matrices encoding homophones (cf. Figure 8). Averaged over 50 ciphertexts.

steps (rather than the entire input) and measuring Pearson’s r between the resulting matrices and a reference matrix as in Figure 8 (right). As seen in Figure 9, the correlation between homophony and self-attention grows steadily as the model reproduces the input, reaching a plateau after ~ 200 tokens. (See also Figure 13 in Appendix C.) This demonstrates the importance of CausalLM, as it shows that the model makes crucial inferences about its input as it reproduces that input.

We emphasize that our model learns homophony relations and recovers decipherment keys *implicitly*, in just a single pass over the input ciphertext. Prior search-based techniques required a search over many candidate plaintexts in order to recover these same relations explicitly.

9 Other Related Work

Computational decipherment based techniques have seen a wide range of applications ranging such as identifying unknown languages and scripts (Hauer and Kondrak, 2016), writing systems (Born et al., 2019, 2021, 2022) and lost languages (Snyder et al., 2010; Luo et al., 2019), offensive language detection (Wu et al., 2018; Qian et al., 2019), and, more recently, towards improving neural machine translation (Kambhatla et al., 2022). While decipherment has strong connections to cryptography research, we limit the scope of this work to natural language based decipherment. Knight et al. (2006) proposed an unsupervised noisy channel based technique for decipherment. Hauer et al. 2014 solved short ciphers with Monte-Carlo tree search. Greydanus (2017) train a seq-to-seq LSTM to solve polyalphabetic substitution ciphers including Enigma, but only explore supervised known-plaintext attacks. CipherGAN (Gomez et al., 2018) exploits learned letter embedding distributions, but requires a large volume of ciphertext and only handles 1:1 substitution and Vigenère ciphers. Luo et al. (2021) and Aldarrab and May (2022) pro-

pose techniques to decipher undersegmented ciphers. Aldarrab and May (2021) train a sequence-to-sequence neural translation model to decipher from character frequencies. In contrast, we introduce a novel encoding which is suitable for homophonic inputs, and demonstrate that a causal LM is more effective than a seq-to-seq model in a homophonic setting.

Cross-attention from encoder-decoder architectures has been shown to have limited explanatory power in translation settings (Moradi et al., 2019, 2021). In spite of this, Born et al. 2022 show that encoder self-attention implicitly captures information which replicates expert intuitions about document structure in an undeciphered script. In a similar vein, our key recovery experiments offer evidence of yet another way self-attention may be fruitfully exploited in the decoder in a decipherment setting.

10 Conclusion and Future Work

We introduce a novel recurrence encoding to represent distributional information which is invariant across plaintexts and ciphertexts, even under homophonic ciphers. This allows us to train a Transformer LM for decipherment using synthetic ciphertext-plaintext pairs. Our model achieves strong results on unseen homophonic substitution ciphers, and achieves the first fully-automatic decipherment of several historical ciphers. We show that language models vastly outperform sequence-to-sequence models on this task, and that causal attention masking (which forces our model to reproduce the ciphertext before deciphering it) is crucial to solve homophonic inputs. Our analysis shows that our model implicitly learns homophony relations and the decipherment key while reproducing the input. In a zero-shot setting, our model accurately decipheres into Latin despite being trained on English; This work marks a successful departure from search-based solutions to homophonic substitution ciphers, and introduces language models as a viable tool for future decipherment work.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. The research was partially supported by the Natural Sciences and Engineering Research Council of Canada grants NSERC RGPIN-2018-06437 and RGPAS-2018-522574 and a Department of National Defence

(DND) and NSERC grant DGDND-2018-00025 to the third author, and by an NSERC award CGSD3-547773-2020 to the second author.

Limitations

A key limitation of our model is the combined sequence length of the ciphertext and the plaintext. As the standard self-attention mechanism of the Transformer uses $O(n^2)$ time and space with respect to sequence length, modelling longer ciphers (eg. 1500 chars) is extremely compute inefficient. This also restricts our model’s ability to handle ciphers such as the Beale Pt. 2 cipher. A possible avenue for an extension of this work is to address this issue by leveraging more sophisticated self attention mechanisms like the linformer (Wang et al., 2020).

Ethics Statement

This work is concerned with decoding encrypted correspondences, and therefore the techniques in the paper are designed to reveal information that has been purposefully obscured and might violate the privacy of the authors. However, an encryption system such as the homophonic substitution cipher is primarily seen in centuries old historical ciphers, and is both relatively weak and obsolete. The methods might have little impact beyond any applications intended towards decipherment of ancient ciphers or machine translation. Further, the more standard encryption techniques such as the AES/RSA are very sophisticated and cannot be attacked with the model discussed in the paper.

We note that our proposal of this method is not as a replacement to expert code-breakers, but as a new tool at their disposal. Our model cannot “cheat” except by disobeying the key, and we’ve shown that it does consistently follow the key (Section 8.2). Thus our model output is no more or less trustworthy than the equivalent produced by a human. Since there is no guarantee that the model will always produce the right decipherment, it is imperative that domain experts assess the text produced by this model in the same way they would assess proposals from an amateur decipherer with little/no domain knowledge.

References

Nada Aldarrab and Jonathan May. 2021. [Can sequence-to-sequence models crack substitution ciphers?](#) In

- Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7226–7235, Online. Association for Computational Linguistics.
- Nada Aldarrab and Jonathan May. 2022. [Segmenting numerical substitution ciphers](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 706–714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Taylor Berg-Kirkpatrick and Dan Klein. 2013. [Decipherment with a million random restarts](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 874–878, Seattle, Washington, USA. Association for Computational Linguistics.
- Logan Born, Kate Kelley, Nishant Kambhatla, Carolyn Chen, and Anoop Sarkar. 2019. [Sign clustering and topic extraction in Proto-Elamite](#). In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 122–132, Minneapolis, USA. Association for Computational Linguistics.
- Logan Born, Kathryn Kelley, M. Willis Monroe, and Anoop Sarkar. 2021. [Compositionality of complex graphemes in the undeciphered Proto-Elamite script using image and text embedding models](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4136–4146, Online. Association for Computational Linguistics.
- Logan Born, M. Monroe, Kathryn Kelley, and Anoop Sarkar. 2022. [Sequence models for document structure identification in an undeciphered script](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9111–9121, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Eric Corlett and Gerald Penn. 2010. [An exact A* method for deciphering letter-substitution ciphers](#). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1040–1047, Uppsala, Sweden. Association for Computational Linguistics.
- Stéphane D’Ascoli, Pierre-Alexandre Kamienny, Guillaume Lample, and Francois Charton. 2022. [Deep symbolic regression for recurrence prediction](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4520–4536. PMLR.
- William F. Friedman. 1922. The index of coincidence and its applications in cryptography. In *Department of Ciphers. Publ 22.*, Geneva, Illinois. Riverbank Laboratories.
- Aidan N. Gomez, Sicong Huang, Ivan Zhang, Bryan M. Li, Muhammad Osama, and Lukasz Kaiser. 2018. [Unsupervised cipher cracking using discrete gans](#).
- Sam Greycanus. 2017. [Learning the enigma with recurrent neural networks](#).
- Bradley Hauer, Ryan Hayward, and Grzegorz Kondrak. 2014. [Solving substitution ciphers with combined language models](#). In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2314–2325, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.
- Bradley Hauer and Grzegorz Kondrak. 2016. [Decoding anagrammed texts written in an unknown language and script](#). *Transactions of the Association for Computational Linguistics*, 4:75–86.
- Nishant Kambhatla, Logan Born, and Anoop Sarkar. 2022. [CipherDAug: Ciphertext based data augmentation for neural machine translation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 201–218, Dublin, Ireland. Association for Computational Linguistics.
- Nishant Kambhatla, Anahita Mansouri Bigvand, and Anoop Sarkar. 2018. [Decipherment of substitution ciphers with neural language models](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 869–874, Brussels, Belgium. Association for Computational Linguistics.
- Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. 2016. [Character-aware neural language models](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).
- Kevin Knight, Beáta Megyesi, and Christiane Schaefer. 2011. [The copiale cipher](#). In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 2–9, Portland, Oregon. Association for Computational Linguistics.
- Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. [Unsupervised analysis for decipherment problems](#). In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 499–506, Sydney, Australia. Association for Computational Linguistics.
- Wenda Li, Lei Yu, Yuhuai Wu, and Lawrence Charles Paulson. 2021. [Isarstep: a benchmark for high-level mathematical reasoning](#). In *ICLR*.
- Jiaming Luo, Yuan Cao, and Regina Barzilay. 2019. [Neural decipherment via minimum-cost flow: From Ugaritic to Linear B](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3146–3155, Florence, Italy. Association for Computational Linguistics.

- Jiaming Luo, Frederik Hartmann, Enrico Santus, Regina Barzilay, and Yuan Cao. 2021. [Deciphering Undersegmented Ancient Scripts Using Phonetic Prior](#). *Transactions of the Association for Computational Linguistics*, 9:69–81.
- Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldspühl. 2020. [Decryption of historical manuscripts: the decrypt project](#). *Cryptologia*, 44(6):545–559.
- Pooya Moradi, Nishant Kambhatla, and Anoop Sarkar. 2019. [Interrogating the explanatory power of attention in neural machine translation](#). In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 221–230, Hong Kong. Association for Computational Linguistics.
- Pooya Moradi, Nishant Kambhatla, and Anoop Sarkar. 2021. [Measuring and improving faithfulness of attention in neural machine translation](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2791–2802, Online. Association for Computational Linguistics.
- Masato Neishi and Naoki Yoshinaga. 2019. [On the relation between position information and sentence length in neural machine translation](#). In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 328–338, Hong Kong, China. Association for Computational Linguistics.
- Malte Nuhn, Julian Schamper, and Hermann Ney. 2013. [Beam search for solving substitution ciphers](#). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1576, Sofia, Bulgaria. Association for Computational Linguistics.
- Malte Nuhn, Julian Schamper, and Hermann Ney. 2014. [Improved decipherment of homophonic ciphers](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1764–1768, Doha, Qatar. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [fairseq: A fast, extensible toolkit for sequence modeling](#). In *Proceedings of NAACL-HLT 2019: Demonstrations*.
- Eva Pettersson and Beata Megyesi. 2019. [Matching keys and encrypted manuscripts](#). In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 253–261, Turku, Finland. Linköping University Electronic Press.
- Jing Qian, Mai ElSherief, Elizabeth Belding, and William Yang Wang. 2019. [Learning to decipher hate symbols](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3006–3015, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sujith Ravi and Kevin Knight. 2011. [Bayesian inference for zodiac and other homophonic ciphers](#). In *ACL*, pages 239–247.
- David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. [Analysing mathematical reasoning abilities of neural models](#). In *International Conference on Learning Representations*.
- Benjamin Snyder, Regina Barzilay, and Kevin Knight. 2010. [A statistical model for lost language decipherment](#). In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1048–1057, Uppsala, Sweden. Association for Computational Linguistics.
- Dusan Varis and Ondřej Bojar. 2021. [Sequence length is a domain: Length-based overfitting in transformer models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8246–8257, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Shuo Wang, Zhaopeng Tu, Zhixing Tan, Wenxuan Wang, Maosong Sun, and Yang Liu. 2021. [Language models are good translators](#).
- Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. [Linformer: Self-attention with linear complexity](#). *arXiv preprint arXiv:2006.04768*.
- Zhelun Wu, Nishant Kambhatla, and Anoop Sarkar. 2018. [Decipherment for adversarial offensive language detection](#). In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 149–159, Brussels, Belgium. Association for Computational Linguistics.
- Biao Zhang, Behrooz Ghorbani, Ankur Bapna, Yong Cheng, Xavier Garcia, Jonathan Shen, and Orhan Firat. 2022. [Examining scaling and transfer of language model architectures for machine translation](#).

A Hyperparameters and Settings

Preprocessing Following the previous work (Kambhatla et al., 2018; Aldarrab and May, 2021), we preprocessed the Project Gutenberg data by stripping the text of all non text elements, then lower-casing all characters, and removing all non-alphabetic and non-space characters. Our final plaintexts consists of the 26 letters of English alphabet and the `_` symbol to denote space only.

Multilingual Data. The 13 language multilingual data¹² released by Aldarrab and May (2021) consists of 2.2M ciphers in Catalan, Danish, Dutch, Finnish, French, German, Hungarian, Italian, Latin, Norwegian, Portuguese, Spanish, and Swedish languages.

Layers	12
Attn Heads	12
FF Dim.	1536
Char Embed	4
Highway Layers	2
Dropout	0.1
Attn. Dropout	0.1
Batch Size	32000
Peak lr	0.0005
Early Stopping	No
Max Epoch	20

Table 8: The hyperparameters for our model and training.

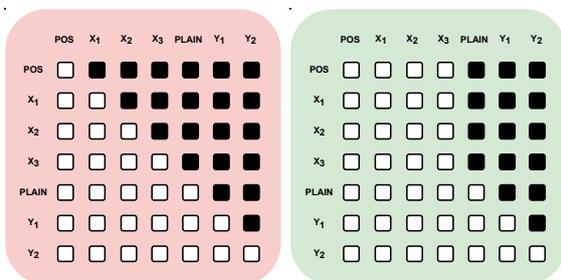


Figure 10: A high level depiction of the causal (left) and the prefix (right) attention masks. X denotes the ciphertext and Y is the plaintext, both prepended by POS and PLAIN tags respectively.

B Towards Decipherment with Human Intervention

In a realistic decipherment setting, the nature of the cipher under attack will not be known. It is possible that some symbols will be polyphonic, that the plaintext language will be out-of-domain, or that the text will be short or corrupted (intentionally, to prevent decipherment, or as a result of damage in the case of historical documents). In such cases, the outputs from an automated decipherment may require manual emendation; real computer-assisted decipherments have previously relied on post-editing by domain experts, as in the decipherment of the Copiale cipher Knight et al. 2011. We demonstrate two examples of how our model can be used for human-in-the-loop decipherment in this more realistic setting.

Zodiac 408 A famous cipher from the Zodiac killer of the 1970s, this text contains 408 characters written with 54 different symbols (~7.5 characters per symbol). There are six polyphonic¹³ symbols, making the text out-of-domain for our model which was only trained on homophonic ciphers. Table 9 shows an example of assisted decipherment based on corrected words. When a correction is identified, it can be appended to the model input, for example:

```
orig. input: REC <cipher> PLAIN
user prompt: REC <cipher> PLAIN i _ l i k e
```

Since our model is a left-to-right language model on both cipher and plaintext, we can interrupt it at any point during plaintext generation to introduce a correction, which is then used as a new constraint on decoding the plaintext. Correcting only 5 polyphonous characters gives **SER 0.4%**, establishing a new state of the art on this cipher.

Borg Cipher The Borg Cipher¹⁴ is a ca. 17th century manuscript written in enciphered Latin. Though the text uses a simple substitution cipher, the plaintext language is out-of-domain for our model which was trained on English.

Since our model was never trained on Latin, zero-shot inference on the first 400 characters results in an SER of 45.14%. But fixing the words `aperitione _ emorrhoidarum` and correcting

¹²<https://github.com/NadaAldarrab/s2s-decipherment>

¹³Different from a homophonic symbol, a *polyphonic* cipher symbol encodes more than one plaintext character.

¹⁴<https://cl.lingfil.uu.se/~bea/borg/>

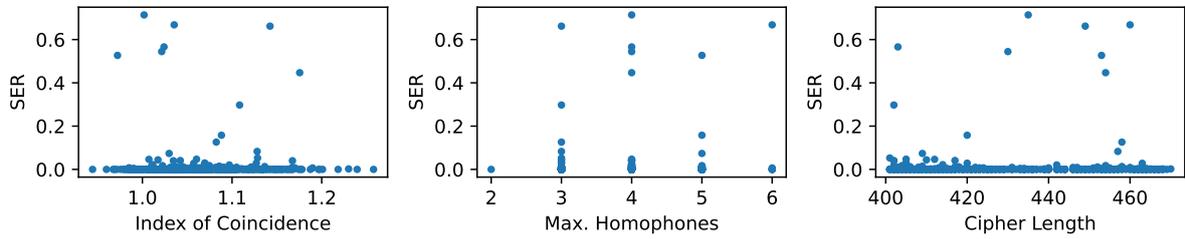


Figure 12: SER vs. index of coincidence, number of homophones of the most homophonic ciphertext symbol, and cipher length. SER is not significantly correlated with any of these metrics.

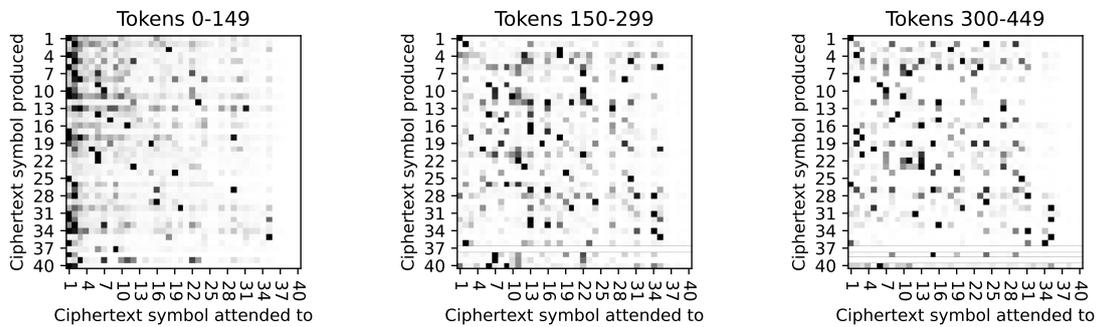


Figure 13: Mean self-attention from each ciphertext symbol to every other ciphertext symbol, averaged across different time-steps. This figure uses the same input as Figure 8. Note how the left subfigure, representing the earliest time-steps, does not meaningfully resemble the reference matrix from Figure 8, implying that the model has not learned which tokens are homophones at this early stage.

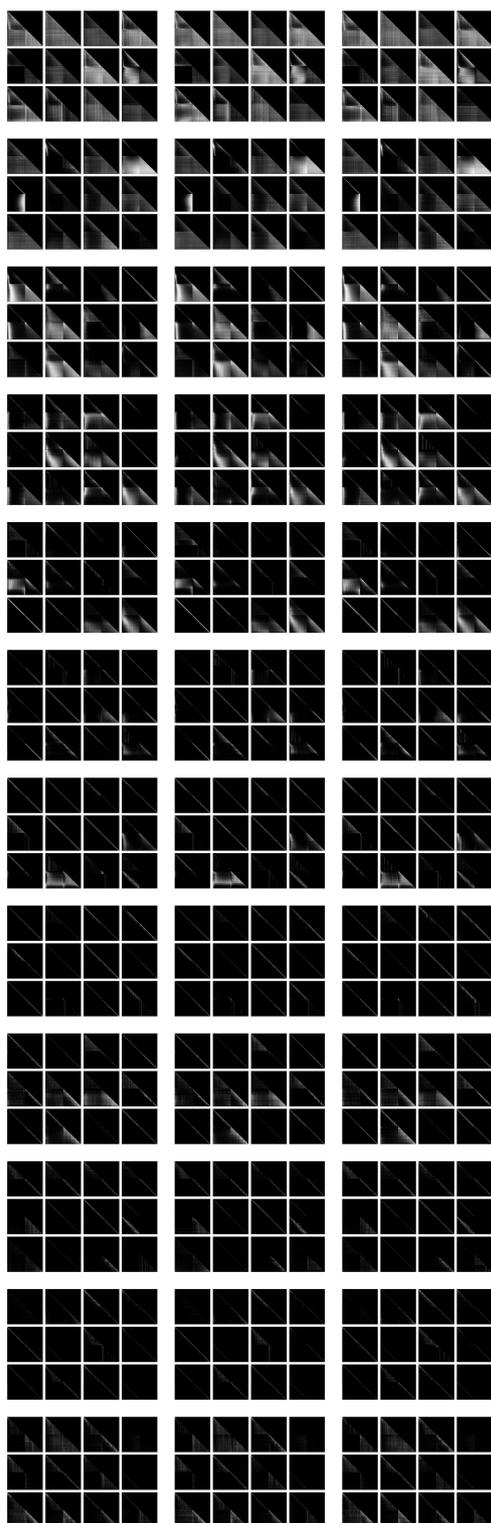


Figure 14: Attention maps for each head in each layer on three sample inputs. (Full-size figure available in supplemental material.) Each column represents a distinct ciphertext, and each row a Transformer layer, with the output layer on the bottom. Each cell is divided into 12 sub-figures, showing the self-attention maps from each of the 12 heads in the corresponding layer on the corresponding input.



Figure 15: The original full version of the **Zodiac-408** cipher published by the San Francisco Examiner on August 3rd 1969.

95 λδχ βλθε α: # # αχ + αδν α εδχ) δ λ λ τ α δ ν δ χ λ ν κ
λ λ τ δ χ β λ θ ε α λ: # π ν ο τ λ) # # σ ο ν λ) λ τ α δ ν γ α ε 2 δ χ δ
λ ν α λ ο λ δ) δ 2 γ α ε π ρ ο δ χ λ # π ρ ι γ # τ σ δ κ) σ γ α # ε π # δ)
γ π ε ο λ ν α π ρ ο # # 2 τ σ γ λ ι α ε δ) 2 γ ι π γ) # ι # σ χ α ε # δ χ α
λ ν) λ β λ θ ε α λ # π γ ι 2 λ # δ) #) # α ε δ χ) λ # π δ ι ο λ ν α # ε
γ σ δ 4 α π δ) # δ χ + δ ν) # α α ν ο β λ θ ε η ε α) π χ) λ ν γ α 2 α δ χ
χ π σ τ α ι λ ε # δ) λ γ α ε ι λ ε 2 α π χ α δ ν η δ χ π ν λ) #
π γ ι π χ χ α χ 2 γ χ α σ ι δ χ π ν λ) ι λ ε τ α δ ε) λ σ π)
β λ θ ε α: λ ν ο α ε χ) π ρ ο 2 γ λ γ δ ν ω λ ν α δ ν χ # 2 2 π
λ α ε δ χ) # π) π γ γ α ε δ χ) λ π ο # ε ν # α β λ θ ε α: χ
λ ε ο δ χ α π π ν α γ π ε ο λ ι β λ θ ε α χ γ α ε π ν χ α 2 λ χ)
ν λ # # λ ν χ δ χ) δ ν β λ θ ε χ λ ν α: τ # # λ 2 α β λ θ α
λ ν ν α 2 α δ χ χ α 4 δ χ) λ ε α) α δ ν δ ν χ λ ν) δ ν # σ
ε λ # τ σ α π ν ο ρ ο π δ ν η α δ ε λ ι σ γ ι α τ α γ ε λ χ # ε
δ ν η λ ι π ν α ι π χ) δ λ ν α π ν ο γ π ε γ α π η ε ν δ χ
γ 2 δ ν β λ θ ε α π σ 2 λ ν ο α ε χ λ σ σ λ # ε λ ι ε α
σ δ η δ λ ν) λ ε α ο # χ λ # γ 2 δ ν) # π) χ) π) λ) # π) # α
χ π σ γ α # δ ε τ α π τ σ α) λ # α σ γ α β λ θ ε α: ν λ ε
γ ε α χ α ε # α # γ 2 χ α σ ι ι ε λ 2 α ρ ι ε δ 2 ο π δ ν η α
δ ε λ ι # δ χ σ γ ι α π ν ο χ) π) α π) # α λ γ σ γ ε α 2 α
δ ο α) λ # α σ γ) # δ χ 2 π) α ε π γ γ α ε δ χ 2 π γ τ α
γ α ε ι λ ε 2 δ) δ ν) # δ χ 2 π ν α ε π) # π) π σ) # π)
π ε ε σ ε α ο γ λ + π ν ν δ χ # # δ) π ν ο π σ) # π) π
ε) λ τ α + π ν ν γ χ # # δ) λ ε) # π) λ ν ο α ε χ # χ γ δ
δ λ ν α) λ τ α # λ ε ε # γ δ) τ α χ #) σ α 2 α π ν
δ χ λ ι β λ θ ε α ν ν α 2 γ λ 2 π γ τ α) # α) ε π #
α σ λ ι χ # 2 λ ν α 2 π ν τ α π ω) # α ε δ) λ η α
) # δ ε δ ν) ε π ν # α τ # # π ε β λ θ ε ι ε δ ν ο δ χ
2 π γ τ α β ε) # π γ 2 η π ο # λ ν) α ν χ α π ν ο 2 π + γ ε λ 2
λ χ) # π) π γ λ ν α) # π γ ε # # 2 τ σ α χ #) α) # π γ # γ σ
ε λ τ # α δ χ) β λ θ ε α: λ 2 π +) # π γ ε π γ γ λ γ ν α
2 λ ν) π: π + α π ν ο γ α ε ι λ ε 2 δ) τ α β λ θ ε 2 α π ν δ χ
γ σ γ ε λ ο # χ α) # λ η ε δ α) η π ο α ι ι α χ) δ χ) # α
λ ν α δ) # π σ ο ε α 2 λ # α) # α χ ε π) δ α γ λ σ δ χ
α γ λ ι β λ θ ε α ν ν α 2 α δ χ) # π)) ε π # α σ δ χ
τ α π σ 2 λ γ α ν γ λ χ χ δ + σ α) λ χ π # χ β λ θ ε ι:
π) λ λ ν) α ε δ ν ι λ ε ο π ε τ σ λ # ο λ π # γ χ π

Figure 16: Page 1 of the BnF-f01 cipher from 1500s.

Handwritten text in a ciphery script, likely a form of the BnF-f01 cipher. The text is arranged in approximately 20 lines, written in a dense, slanted hand. The characters are a mix of letters and symbols, including Greek letters like alpha, beta, gamma, delta, epsilon, zeta, eta, theta, iota, kappa, lambda, mu, nu, xi, omicron, pi, rho, sigma, tau, upsilon, phi, chi, psi, omega, and various numbers and symbols. The text is difficult to decipher due to the complexity of the cipher and the handwriting.

Figure 17: Page 2 of the BnF-f01 cipher from 1500s.

A Survey on Recent Advances in Keyphrase Extraction from Pre-trained Language Models

Mingyang Song, Yi Feng and Liping Jing*

Beijing Key Lab of Traffic Data Analysis and Mining

Beijing Jiaotong University

Beijing, China

{mingyang.song, 21112027, lpjing}@bjtu.edu.cn

Abstract

Keyphrase Extraction (KE) is a critical component in Natural Language Processing (NLP) systems for selecting a set of phrases from the document that could summarize the important information discussed in the document. Typically, a keyphrase extraction system can significantly accelerate the speed of information retrieval and help people get first-hand information from a long document quickly and accurately. Specifically, keyphrases are capable of providing semantic metadata characterizing documents and producing an overview of the content of a document. In this paper, we introduce keyphrase extraction, present a review of the recent studies based on pre-trained language models, offer interesting insights on the different approaches, highlight open issues, and give a comparative experimental study of popular supervised as well as unsupervised techniques on several datasets. To encourage more instantiations, we release the related files mentioned in this paper¹.

1 Introduction

Keyphrase extraction is a fundamental task in NLP for identifying and extracting a set of keyphrases from the document that could summarize the important information discussed in the source document (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019). Keyphrases have enabled accurate and fast searching for the document from a large text corpus and have exhibited their potential in improving many NLP tasks, such as text summarization (Zhang et al., 2004). Various information filtering and extracting techniques are becoming critical with the ever-increasing amount of text data. Owing to its potential importance, keyphrase extraction has received more and more attention from

NLP researchers. However, the keyphrase extraction task is far from being solved: state-of-the-art performance on keyphrase extraction is still lower than other core NLP tasks. Our goal in this paper is to investigate the state-of-the-art models in keyphrase extraction, examine the primary sources of errors made by existing systems, and discuss the challenges ahead.

The first keyphrase extraction task was organized by Turney (1999), which defines the keyphrase extraction task as “the automatic selection of important and topical phrases from the body of a document”. Since then, there have been numerous keyphrase extraction models (Witten et al., 1999; Turney, 2000; Tomokiyo and Hurst, 2003; Hulth, 2004; Wan and Xiao, 2008a; Jiang et al., 2009; Liu et al., 2009; Grineva et al., 2009; Nguyen and Phan, 2009; Bougouin et al., 2013; Caragea et al., 2014; Danesh et al., 2015; Bougouin et al., 2016; Florescu and Caragea, 2017a; Campos et al., 2018a; Alzaidy et al., 2019). In the past two decades, keyphrase extraction methods have experienced the development from traditional approaches to deep learning methods (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019). With the recent development of Pre-trained Language Models (PLMs) (Devlin et al., 2019; Liu et al., 2019), many NLP tasks have significantly changed, that is, how to adopt and leverage pre-trained language models in the specific task. Therefore, many keyphrase extraction models (Sun et al., 2020a; Song et al., 2021) adopt PLMs as the embedding layer.

We present a comprehensive survey of recent advances in neural keyphrase extraction. We describe the neural keyphrase extraction systems based on pre-trained language models, which depend on different paradigms (e.g., one-stage (Wang et al., 2020) and two-stage (Sun et al., 2020a)), various tasks (e.g., classification and ranking (Mu et al., 2020; Sun et al., 2020a)), different learning strategies (e.g., supervised (Song et al., 2021) and un-

* Corresponding author.

¹<https://github.com/MySong7NLP/KeyphraseExtractionSurvey>

supervised (Ding and Luo, 2021)), and variants of pre-trained language models (e.g., BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)).

Furthermore, we re-implement and collect the results of the mentioned models on several benchmark keyphrase extraction datasets. We illustrate the results in Table 3 and Table 2 and discuss in Section 6 how neural keyphrase extraction systems have improved performance over past works, including supervised and unsupervised models. Furthermore, we provide resources, including links to share the current neural keyphrase extraction systems and links to share the code for each category of the neural keyphrase extraction approaches. To the best of our knowledge, this is the first survey focusing on the keyphrase extraction task based on recent pre-trained language models.

Overall, this paper first discusses previous surveys on keyphrase extraction in Section 2.1 and give a briefly introduction about pre-trained language models in Section 2.2. Then we highlight standard, past, and recent benchmark keyphrase extraction datasets (from shared tasks and other research) in Section 3 and evaluation metrics in Section 4. We then describe neural keyphrase extraction systems in Section 5. Next, we give the analysis and discussion in Section 6. Finally, we summarize the conclusions and future directions of neural keyphrase extraction in Section 7. The limitations of our work is presented in Section 8.

2 Preliminary

In this section, we claim the differences between the current survey and the existing surveys. Next, we present the background of pre-trained language models and their importance in NLP.

2.1 Previous Surveys

The first comprehensive keyphrase extraction survey was Hasan and Ng (2014), which covered a variety of unsupervised and supervised keyphrase extraction models, highlighted common features used by existing models during that time, and explained evaluation metrics that are still in use today. Papagiannopoulou and Tsoumakas (2019) present a more recent keyphrase extraction survey that mainly included many unsupervised and supervised models based on deep learning. Furthermore, Papagiannopoulou and Tsoumakas (2019) also provides a list of popular keyphrase extraction datasets and a thorough empirical study.

The existing keyphrase extraction surveys primarily cover early feature-engineered and neural-based keyphrase extraction models (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019). There is not yet, to our knowledge, a comprehensive survey of keyphrase extraction based on pre-trained language models.

2.2 Pre-trained Language Models

Recently, pre-trained language models have advanced the state-of-the-art in many NLP tasks ranging from textual similarity to text summarization (Zhang et al., 2019; Liu and Lapata, 2019; Zhong et al., 2020) and named entity recognition (Zhou et al., 2021). State-of-the-art pre-trained models include LSTM-based language models (e.g., ELMo (Peters et al., 2018)) and Transformer-based language models (e.g., BERT² (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)). Specifically, the transformer-based models learn bidirectional representations for words based on a masked language model and sentence adjacency training objective (Devlin et al., 2019). Simply using contextualized embeddings obtained from the transformer-based pre-trained language models in place of traditional embeddings has resulted in state-of-the-art performance on a range of NLP tasks. Therefore, pre-trained language models have been employed as encoders for obtaining word-, sentence-, and document-level representations to assist the downstream tasks.

3 Keyphrase Extraction Dataset

Since the first shared task on KE (Turney, 1999), many shared tasks and benchmark datasets for KE have been created. Specifically, OpenKP (Xiong et al., 2019), Inspec (Hulth, 2003), NUS (Nguyen and Kan, 2007), Krapivin (Krapivin and Marchese, 2009), SemEval2010 (Kim et al., 2010), SemEval2017 (Augenstein et al., 2017), and KP20k (Meng et al., 2017) were created from scientific articles in English.

Compared with other datasets, KP20k contains a large amount of annotation data, so it is often used as the dataset to train the neural-based KE models recently. Meanwhile, in recent papers (Sun et al., 2020a; Song et al., 2021), Inspec (Hulth, 2003), NUS (Nguyen and Kan, 2007), Krapivin (Krapivin and Marchese, 2009), SemEval2010 (Kim et al., 2010), and SemEval2017 (Augenstein et al., 2017)

²<https://huggingface.co/bert-base-cased>

Dataset	Type	Long	# Doc.	Avg. # Words	Present KPs (%)
KP20k (Meng et al., 2017)	Scientific Paper Abstract	✗	568.00k	188.47	57.40
Inspec (Hulth, 2003)	Scientific Paper Abstract	✗	2.00k	130.57	55.69
SemEval2017 (Augenstein et al., 2017)	Scientific Paper Abstract	✗	0.50k	176.13	42.01
NUS (Nguyen and Kan, 2007)	Full Scientific Paper	✓	0.21k	7644.43	67.75
Krapivin (Krapivin and Marchese, 2009)	Full Scientific Paper	✓	2.30k	8420.76	44.74
SemEval2010 (Kim et al., 2010)	Full Scientific Paper	✓	0.24k	7434.52	88.70
DUC2001 (Wan and Xiao, 2008b)	News Document	✗	0.31k	724.63	97.82
OpenKP (Xiong et al., 2019)	Open Domain Web Content	✗	147.20k	900.40	100.00

Table 1: This table shows the statistics of different recent popular datasets. **Long** indicates whether the dataset belongs to a long document. **# Doc.** is the number of documents in the dataset. **Avg. # Words** is the average number of words for documents in the indicated dataset. **Present KPs (%)** indicates the percentage of keyphrases, which are presented in the documents.

datasets are often used as the zero-shot test sets to verify the robustness of the KE models trained by the KP20k dataset. Furthermore, KE tasks have also been organized on newswire articles in English, e.g., DUC2001 (Wan and Xiao, 2008b). Table 1 summarizes the statistics of several commonly used benchmark datasets.

4 Keyphrase Extraction Evaluation

This section describes evaluation metrics for measuring recent state-of-the-art keyphrase extraction baselines on commonly-used datasets. Designing a suitable evaluation metric for the keyphrase extraction task is by no means an easy study (Hasan and Ng, 2014). To score the output of a keyphrase extraction model, the traditional approach, which is also adopted by the SemEval-2010 (Kim et al., 2010) shared task on keyphrase extraction, is (1) to create a mapping between the keyphrases in the ground-truth keyphrases and those in the model output adopting exact and partial matching (Papagiannopoulou and Tsoumakas, 2019), and then (2) score the output using evaluation metrics such as precision (P), recall (R), and F1-score (F1).

As mentioned earlier, such evaluation usually operates based on exact matches between the predicted and ground-truth keyphrases. However, such a strategy cannot account for partial matches or semantic similarity. For example, if the prediction is "keyphrase extraction model" and the ground truth is "keyphrase extraction system", despite both semantic similarity and partial matching, the score will be 0. These minor deviations are ubiquitous in

keyphrase extraction, yet they are harshly penalized by the "exact match" evaluation metrics.

5 Neural Keyphrase Extraction Models with Pre-trained Language Models

There are two popular pipelines in the keyphrase extraction task, including one-stage and two-stage frameworks, as illustrated in Figure 1. The former mainly refers to using the task reformulation to address the keyphrase extraction task, which often treats the keyphrase extraction task as a sequence labeling task. The latter represents a more general framework, which usually operates in two procedures: (1) extracting a set of words/phrases that serve as candidate phrases using some heuristics and (2) determining which candidate phrases are keyphrases using supervised or unsupervised methods (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019).

Typically, supervised methods perform better on specific domain tasks. However, this kind of method takes a lot of labor to annotate the corpus, and the model after training may overfit and not work well on other KE datasets. On the contrary, unsupervised methods do not need to annotate the corpus and usually have better data generalization in different domains. Still, the performance is often insufficient due to the lack of annotated data. Overall, we defined the above two procedures as the candidate keyphrase extraction and keyphrase importance estimation. In this paper, we distinguish the existing methods into three categories depending on the recent state-of-the-art baselines (with

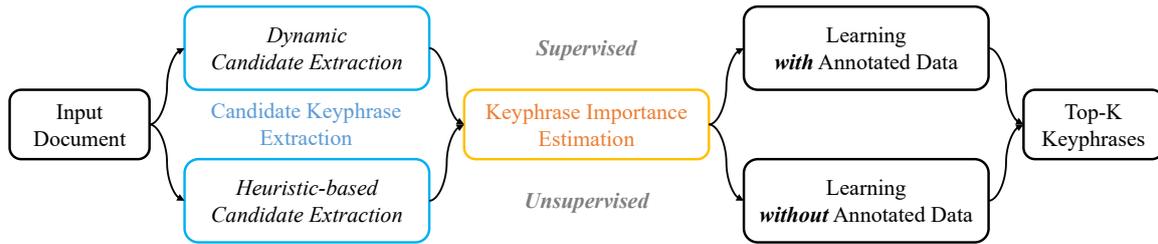


Figure 1: The overall architecture of the two-stage supervised and unsupervised keyphrase extraction framework.

pre-trained language models as the backbone), including two-stage unsupervised, two-stage supervised, and one-stage supervised models.

5.1 Two-Stage Unsupervised Keyphrase Extraction Models

As noted before, unsupervised keyphrase extraction systems generally extract a set of phrases from the source document as candidates by using heuristic rules. These rules are designed to avoid spurious instances and keep the number of candidates to a minimum (Hasan and Ng, 2014). The main steps of the commonly used candidate keyphrases extraction methods for the recent unsupervised keyphrase extraction models are as follows, (1) tokenizing the document and tagging the document with part-of-speech (POS) tags via the StanfordCoreNLP Tools³; (2) extracting candidate phrases based on part-of-speech tags by the regular expression via the python package NLTK⁴. Furthermore, different pruning heuristics have been designed for pruning candidates that are unlikely to be keyphrases to obtain a better candidate set (Huang et al., 2006; Kumar and Srinathan, 2008; El-Beltagy and Rafea, 2009; Newman et al., 2012; You et al., 2009). After obtaining candidates, keyphrases are determined by estimating the importance of each candidate through various strategies. Here, to facilitate the introduction, we divide the methods of importance estimation into two categories, namely, traditional methods and embedding-based methods.

Traditional unsupervised keyphrase extraction systems can be mainly divided into statistics-based (Jones, 2004; Campos et al., 2018b), topic-based (Liu et al., 2009; Jardine and Teufel, 2014), and graph-based (Mihalcea and Tarau, 2004; Wan and Xiao, 2008b; Bougouin et al., 2013; Florescu and Caragea, 2017b) methods. Generally, these

models primarily use different features of documents (e.g., word frequency, position, linguistic properties, topic, length, the relationship between words, external knowledge-based information, etc.) to estimate the importance of each candidate phrase and discriminate whether a candidate phrase is a keyphrase (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019).

However, these traditional unsupervised models estimate the importance scores of candidate phrases based on the surface-level features, ignoring the high-level features (e.g., syntactic and semantic information) of natural languages, which leads to extract wrong keyphrases. Therefore, recent studies focus on embedding-based models (Wang et al., 2015; Mahata et al., 2018a; Papagiannopoulou and Tsoumakas, 2018; Sahrawat et al., 2020; Kulkarni et al., 2022; Song et al., 2022b), which leverage pre-trained embeddings (containing high-level features) to obtain phrase and document embeddings and calculate the importance scores of candidate phrases for extracting keyphrases. Wang et al. (2015) is the first work to explore utilizing word embedding and frequency to generate weighted edges between words, then using the weighted PageRank algorithm to compute and rank candidate scores. Key2vec (Mahata et al., 2018a) proposes an effective way of processing text documents for training multi-word phrase embeddings that are used for topic representations of scientific articles and ranking of keyphrases extracted from them using the topic-weighted PageRank algorithm. Mahata et al. (2018b) uses a combination of theme-weighted personalized PageRank algorithm and neural phrase embeddings for extracting and ranking keyphrases. EmbedRank (Bennani-Smires et al., 2018) ranks candidate phrases by measuring the semantic similarity between each candidate phrase and document embeddings.

With the development of pre-trained language

³<https://stanfordnlp.github.io/CoreNLP>

⁴<https://github.com/nltk>

models (e.g., ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019)), SIFRank⁵ (Sun et al., 2020b) improves candidate phrase and document embeddings from EmbedRank with the pre-trained language model ELMo (Peters et al., 2018) and achieves better performance. JointGL⁶ (Liang et al., 2021) integrates boundary-aware phrase centrality (the semantic similarities are calculated between all candidate phrases for identifying which candidate is better) and phrase-document relevance (the semantic similarities are calculated between candidate phrases and their corresponding document) from both local and global views, then used both jointly to determine the importance of each candidate. AttentionRank⁷ (Ding and Luo, 2021) adopts a pre-trained language model to calculate the self-attention of a candidate within the context of a sentence, and the cross-attention between a candidate and sentences within the source document to evaluate the local and global importance of each candidate. MDERank⁸ (Zhang et al., 2021) proposes to rank candidates using the similarity between the BERT embeddings of the source document and the masked document. Totally, these models achieve state-of-the-art performance in the unsupervised keyphrase extraction task, benefiting from the development of representation learning.

5.2 Two-Stage Supervised Keyphrase Extraction Models

Different from two-stage unsupervised approaches, supervised approaches generally combine candidate keyphrase extraction and keyphrase importance estimation via *an end-to-end learning framework*, guide the whole model to rank and extract keyphrases through annotated data and optimize the two stages simultaneously. Therefore, to obtain sufficient candidates, the recent supervised models (Xiong et al., 2019; Sun et al., 2020a; Song et al., 2021, 2022a) directly extract n-grams from the document as candidates. Then propose, various approaches to estimate the importance scores of candidates. To estimate the importance of candidate phrases, similar to unsupervised models, supervised models (Xiong et al., 2019; Sun et al., 2020a; Song et al., 2021) also obtain phrase and document representations by adopting pre-trained

language models as the backbone, including ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), etc.

Firstly, BLING-KPE (Xiong et al., 2019) formulates keyphrase extraction as an n-gram level keyphrase chunking task to determine whether a candidate is a keyphrase, which incorporates pre-trained embeddings (i.e., ELMo (Peters et al., 2018)) into a convolutional transformer network to model n-gram representations. BLING-KPE achieves significant improvement over previous models. To leverage external knowledge to assist keyphrase extraction, SMART-KPE⁹ (Wang et al., 2020) also shows that incorporating multimodal information in web pages, such as font, size, and DOM features, can bring further improvement for open-domain web keyphrase extraction. Later, Ainslie et al. (2020) replaces the full self-attention of Transformers with local-global attention, which significantly boosts the keyphrase extraction performance for long documents. SKE-BASE-RANK (Mu et al., 2020) proposes a span-based keyphrase extraction model to model the relationships between candidates and the document in context.

JointKPE¹⁰ (Sun et al., 2020a) proposes an open-domain keyphrase extraction approach built on pre-trained language models (Devlin et al., 2019; Liu et al., 2019), which can capture both local phraseness and global informativeness when extracting keyphrases. JointKPE learns to rank keyphrases by estimating their informativeness in the whole document and is jointly trained on the keyphrase chunking task to guarantee the phraseness of keyphrase candidates. KIEMP¹¹ (Song et al., 2021) proposes estimating the importance score of each candidate from multiple perspectives and introducing a matching module to match the high-level concept between the document and candidates to enhance the relevance of extracted keyphrases. To extract more relevant keyphrases, HyperMatch¹² (Song et al., 2022a) proposes a new matching framework and explores keyphrase extraction in the hyperbolic space. Concretely, HyperMatch first maps phrase and document representations into the same hyperbolic space and explicitly models the relevance between candidate phrases and the document as the phrase-document relevance via the Poincaré distance to extract keyphrases.

⁵<https://github.com/sunylgdx/SIFRank>

⁶https://github.com/xnliang98/uke_ccrank

⁷<https://github.com/hd10-iupui/AttentionRank>

⁸<https://github.com/linhanz/mderank>

⁹<https://github.com/victorywys/SMART-KPE>

¹⁰<https://github.com/thunlp/BERT-KPE>

¹¹<https://github.com/MySong7NLPer/KIEMP>

¹²<https://github.com/MySong7NLPer/HyperMatch>

Model	DUC2001			Inspec			SemEval2010			SemEval2017		
	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15
Traditional Two-Stage Models												
TF-IDF (Jones, 2004)	9.21	10.63	11.06	11.28	13.88	13.83	2.81	3.48	3.91	12.70	16.26	16.73
YAKE (Campos et al., 2018b)	12.27	14.37	14.76	18.08	19.62	20.11	11.76	14.4	15.19	11.84	18.14	20.55
TextRank (Mihalcea and Tarau, 2004)	11.80	18.28	20.22	27.04	25.08	36.65	3.80	5.38	7.65	16.43	25.83	30.50
SingleRank (Wan and Xiao, 2008b)	20.43	25.59	25.70	27.79	34.46	36.05	5.90	9.02	10.58	18.23	27.73	31.73
TopicRank (Bougouin et al., 2013)	21.56	23.12	20.87	25.38	28.46	29.49	12.12	12.90	13.54	17.10	22.62	24.87
PositionRank (Florescu and Caragea, 2017b)	23.35	28.57	28.60	28.12	32.87	33.32	9.84	13.34	14.33	18.23	26.30	30.55
Two-Stage Embedding-based Unsupervised Keyphrase Extraction Models with Static Embeddings												
EmbedRank2v (Bennani-Smires et al., 2018)	24.02	28.12	28.82	31.51	37.94	37.96	3.02	5.08	7.23	20.21	29.59	33.94
KeyGames (Saxena et al., 2020)	24.42	28.28	29.77	32.12	40.48	40.94	11.93	14.35	14.62	-	-	-
Two-Stage Embedding-based Unsupervised Keyphrase Extraction Models with PLMs												
SIFRank (Sun et al., 2020b)	24.27	27.43	27.86	29.11	38.80	39.59	-	-	-	22.59	32.85	38.10
JointGL (Liang et al., 2021)	28.62	35.52	36.29	32.61	40.17	41.09	13.02	19.35	21.72	-	-	-
AttentionRank (Ding and Luo, 2021)	-	-	-	24.45	32.15	34.49	11.39	15.12	16.66	23.59	34.37	38.21
MDERank (Zhang et al., 2021)	23.31	26.65	26.42	27.85	34.36	36.40	13.05	18.27	20.35	20.37	31.21	36.63

Table 2: Performance of unsupervised keyphrase extraction models on the DUC2001, Inspec, SemEval2010 and SemEval2017 test sets. F1 scores on the top 5, 10, and 15 keyphrases are reported. The best results are bolded. The results of baseline models are those presented in the original papers or better results published in other papers recently.

5.3 One-Stage Supervised Keyphrase Extraction Models

A major limitation of the above two-stage supervised approaches is classifying the labels of each candidate phrase independently while ignoring the dependencies that could potentially exist between candidates. Therefore, recent studies (Golapalli et al., 2017; Basaldella et al., 2018; Wang et al., 2018; Alzaidy et al., 2019; Sun et al., 2019; Mu et al., 2020; Sahrawat et al., 2020) formulated keyphrase extraction as sequence labeling and showed that using linear-chain Conditional Random Fields improved the performance over baseline models for this task. Then, Mu et al. (2020) proposes SKE-BASE-CLS and -RANK, which directly extracts span-based phrase representations from all the document tokens via pre-trained language models and further learn to capture the interaction between them and their corresponding document to get better ranking results. Furthermore, this kind of model can extract overlapped keyphrases (Mu et al., 2020).

6 Discussion

In this section, we report the results of the recent unsupervised and supervised keyphrase extraction baselines, which all adopt pre-trained language

models as the backbone, as shown in Table 2 and Table 3. Specifically, Table 2 presents the results of the traditional unsupervised methods and the unsupervised embedding-based keyphrase extraction baselines discussed in Section 5.1 on the DUC2001 (Wan and Xiao, 2008b), Inspec (Hulth, 2003), SemEval2010 (Kim et al., 2010), and SemEval2017 (Augenstein et al., 2017) datasets. Embedding-based two-stage models without PLMs indicate that the models do not use pre-trained language models as the backbone to obtain representations. Table 3 shows the results of all the different categories of the supervised keyphrase extraction systems discussed in Section 5.2 and Section 5.3 on the KP20k (Meng et al., 2017) and OpenKP (Xiong et al., 2019) datasets.

Our first finding from the survey is those two-stage embedding-based systems with static embeddings outperform two-stage traditional methods, despite the latter’s access to different valuable features (e.g., word frequency, position, linguistic properties, topic, length, the relationship between words, external knowledge-based information, etc.). This further demonstrates the necessity of studying embedding-based methods.

Our second finding is those embedding-based systems with PLMs outperform embedding-based approaches with static embeddings in most cases.

Model	KP20k		OpenKP		
	F1@5	F1@10	F1@1	F1@3	F1@5
One-Stage Supervised Keyphrase Extraction Models					
SMART-KPE+Full (Wang et al., 2020)	-	-	38.0	40.1	34.4
BERT-TagKPE [†]	38.8	31.7	32.1	36.1	31.4
BERT-SpanKPE [†]	36.8	30.8	31.8	33.2	28.9
RoBERTa-TagKPE [‡]	39.3	32.0	36.1	38.0	33.0
RoBERTa-SpanKPE [‡]	37.3	30.9	34.7	36.1	31.3
Two-Stage Supervised Keyphrase Extraction Models					
BLING-KPE (Xiong et al., 2019)	-	-	26.7	29.2	20.9
SKE-BASE-CLS (Mu et al., 2020)	38.6	32.6	-	-	-
BERT-ChunkKPE [†]	41.2	33.7	34.0	35.6	31.1
RoBERTa-ChunkKPE [†]	40.8	33.7	35.5	37.3	32.4

SKE-BASE-RANK (Mu et al., 2020)	39.2	33.0	-	-	-
BERT-RankKPE [†]	41.3	34.0	34.2	37.4	32.5
RoBERTa-RankKPE [†]	41.7	34.3	36.1	39.0	33.7
HyperMatch (Song et al., 2022a)	41.6	34.3	36.4	39.4	33.8

BERT-JointKPE [†]	41.1	33.8	34.9	37.6	32.5
RoBERTa-JointKPE [†]	41.9	34.4	36.4	39.1	33.8
KIEMP (Song et al., 2021)	42.1	34.5	36.9	39.2	34.0

Table 3: Results of different categories of supervised keyphrase extraction models on two benchmark keyphrase datasets. F1 scores on the top 1, 3, 5, and 10 keyphrases are reported. [†] indicates the results are reported by their corresponding paper (Sun et al., 2020a), and [‡] denotes that these results are re-evaluated by ourselves via the code which is provided by its corresponding paper (Sun et al., 2020a). The best results are highlighted in bold. The results of baseline models are those presented in the original papers or better results published in other papers recently.

However, not all embedding-based systems with PLMs are superior to embedding-based systems with static embeddings. The former generally outperforms the latter when adopting the same importance estimation strategy, but the estimation strategy can significantly affect the results of keyphrase extraction. To sum up, effectively using pre-trained embeddings to estimate the importance score of each candidate is a critical part of improving the performance of keyphrase extraction. Furthermore, there is still interesting progress to be made by leveraging a self-supervised learning strategy to optimize embedding-based systems. MDERank uses a simple yet effective contrastive learning strategy to optimize embedding-based systems, achieving better performance.

Our third finding is that the embedding-based methods have slight improvement on long document datasets (e.g., SemEval2010), and all unsupervised methods have poor effects on long document datasets. This demonstrates that keyphrase extraction from long documents is still a challeng-

ing problem.

Our final finding is that two-stage supervised keyphrase extraction methods are superior to one-stage supervised keyphrase extraction methods, as illustrated in Table 3. In addition, the two-stage method has higher scalability and adaptability than the one-stage method, such as handling long and extremely long documents.

7 Conclusion and Future Directions

We summarize the recent neural keyphrase extraction models based on pre-trained language models. Our survey of models for keyphrase extraction, covering both unsupervised and supervised models, has yielded several important insights. The analysis revealed that there are at least six major challenges ahead.

7.1 Improving the Quality of Generated Candidate Keyphrases

Many heuristic rules have proven effective with a high recall to cover most of the gold keyphrases

of source documents, which determines the upper bound of the performance of keyphrase extraction (Hasan and Ng, 2014). Intuitively, better candidate keyphrase extraction strategies are required to generate a set of candidate keyphrases with a higher recall from the source document to improve the upper-bound performance of keyphrase extraction. Recent work (Jawahar et al., 2019) demonstrates that the intermediate layers of BERT encode a rich hierarchy of linguistic information, with surface features at the bottom, syntactic features in the middle, and semantic features at the top, as mentioned in Section 2.2. They also observe that BERT mostly captures phrase-level information in the lower layers and gradually dilutes this information in higher layers. In addition, the number of candidate keyphrases will increase as the document length increases. Therefore, how constructing candidate keyphrases using the potential knowledge of pre-trained language models is a valuable research direction.

7.2 Improving Evaluation Metric

As mentioned in Section 4, the existing evaluation metrics occur when a keyphrase extraction system extracts a keyphrase from candidates that is semantically equivalent to a ground-truth keyphrase but is considered erroneous by a scoring function because it fails to recognize that the predicted keyphrase and the corresponding gold keyphrase are semantically equivalent.

In other words, an evaluation error is not made by a keyphrase extraction system, but a mistake due to an unformed scoring function (Hasan and Ng, 2014). Therefore, a more suitable evaluation metric is required to evaluate the predicted keyphrases by adopting the semantic-based matching metric instead of the exact matching evaluation metric. In the future, using pre-trained language models (e.g., BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)) to construct a new semantic-aware evaluation metric similar to BERTScore (Zhang et al., 2020) may be an interesting and valuable research direction.

7.3 Reducing Over-Generation Error

Over-generation errors occur when a keyphrase extraction system correctly predicts a candidate as a keyphrase because it contains a word that frequently appears in the associated document but at the same time erroneously outputs other candidates

as keyphrases because they have the same word in the document.

As mentioned before, for example, if the prediction is "keyphrase extraction challenge" and the ground truth is "keyphrase extraction system", despite both semantic similarity and partial matching, the score will be 0. These minor deviations are ubiquitous in keyphrase extraction, yet they are harshly penalized by the "exact match" evaluation metrics. There are often some non-keyphrases in the candidates. Half of the content of such phrases is very relevant to the core information of the document, but the other half is meaningless. These candidate keyphrases are usually hard to extract and treated as hard samples, which is one of the main reasons for reducing keyphrase extraction performance. The above issues can be solved by modifying the traditional evaluation metrics with semantic weighting.

7.4 Handling Long Document

Generally, two main challenges exist in keyphrase extraction systems equipped with pre-trained language models (e.g., BERT (Devlin et al., 2019)) as the backbone when extracting keyphrases from a long document, especially for an extremely long document.

The first challenge is that pre-trained language models can not directly model the complete context information when facing long documents due to the length limitation of pre-trained language models.

The second challenge is that as the length of the document increases, the difficulty of estimating the importance scores of candidate phrases also increases (specifically for the number of candidates), resulting in the reduction of keyphrase extraction accuracy.

7.5 Improving Domain Generalization

For news or scientific documents, the authors usually annotate a set of keyphrases for their articles (Meng et al., 2017; Augenstein et al., 2017). However, there is typically a lack of keyphrases as the label information for their corresponding documents in other specific domains.

Most existing keyphrase extraction datasets and studies are based on news or scientific documents and lack datasets and research related to other domains. Therefore, the task worthy of investigation is to transfer the keyphrase extraction model from the scientific domain to other domains to build a

domain-specific keyphrase extraction model with various domain generalization strategies.

7.6 Probing Pre-trained Language Model for Keyphrase Extraction

In addition to using transformer-based pre-trained language models (e.g., BERT) in NLP tasks and end applications, research has also been done on BERT, especially to reveal what linguistic information is available in different parts of the model (Jawahar et al., 2019; de Vries et al., 2020; Chen et al., 2021). It has been noted that BERT progressively acquires linguistic information roughly in the same order as the classic language processing pipeline (Tenney et al., 2019a,b): surface features are expressed in lower layers, syntactic features more in middle layers, and semantic ones in higher layers (Jawahar et al., 2019). Making full use of the above hierarchy information may effectively improve the performance of keyphrase extraction.

8 Limitations

The main goal of this paper is to provide a survey of the existing models. Since we do not propose new models, there are no potential social risks to the best of our knowledge. Our work may benefit the research community by providing more introspection into the current state-of-the-art neural keyphrase extraction approaches with pre-trained language models.

9 Acknowledgments

We thank the three anonymous reviewers for their helpful comments. This work was partly supported by the Fundamental Research Funds for the Central Universities (2019JBZ110); the National Natural Science Foundation of China under Grant 62176020; the National Key Research and Development Program (2020AAA0106800); the Beijing Natural Science Foundation under Grant L211016; CAAI-Huawei MindSpore Open Fund; and Chinese Academy of Sciences (OEIP-O-202004).

References

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Václav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. [ETC: encoding long and structured inputs in transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*,

pages 268–284. Association for Computational Linguistics.

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. [Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents](#). In *WWW*, pages 2551–2557. ACM.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. [Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications](#). In *SemEval@ACL*, pages 546–555. Association for Computational Linguistics.

Marco Basaldella, Elisa Antolli, Giuseppe Serra, and Carlo Tasso. 2018. [Bidirectional LSTM recurrent neural network for keyphrase extraction](#). In *Digital Libraries and Multimedia Archives - 14th Italian Research Conference on Digital Libraries, IRCDL 2018, Udine, Italy, January 25-26, 2018, Proceedings*, volume 806 of *Communications in Computer and Information Science*, pages 180–187. Springer.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. [Simple unsupervised keyphrase extraction using sentence embeddings](#). In *CoNLL*, pages 221–229. Association for Computational Linguistics.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. [Topicrank: Graph-based topic ranking for keyphrase extraction](#). In *IJCNLP*, pages 543–551. Asian Federation of Natural Language Processing / ACL.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2016. [Keyphrase annotation with graph co-ranking](#). In *COLING*, pages 2945–2955. ACL.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018a. [A text feature based automatic keyword extraction method for single documents](#). In *ECIR*, volume 10772 of *Lecture Notes in Computer Science*, pages 684–691. Springer.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018b. [Yake! collection-independent automatic keyword extractor](#). In *ECIR*, volume 10772 of *Lecture Notes in Computer Science*, pages 806–810. Springer.

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. [Citation-enhanced keyphrase extraction from research papers: A supervised approach](#). In *EMNLP*, pages 1435–1446. ACL.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Moshu Chen, and Liping Jing. 2021. [Probing bert in hyperbolic spaces](#). In *International Conference on Learning Representations*.

- Soheil Danesh, Tamara Sumner, and James H. Martin. 2015. [Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction](#). In **SEM@NAACL-HLT*, pages 117–126. The *SEM 2015 Organizing Committee.
- Wietse de Vries, Andreas van Cranenburgh, and Malvina Nissim. 2020. [What’s so special about bert’s layers? a closer look at the nlp pipeline in monolingual and multilingual models](#). In *EMNLP (Findings)*, pages 4339–4350. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.
- Haoran Ding and Xiao Luo. 2021. [Attentionrank: Unsupervised keyphrase extraction using self and cross attentions](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1928.
- Samhaa R. El-Beltagy and Ahmed A. Rafea. 2009. [Kp-miner: A keyphrase extraction system for english and arabic documents](#). *Inf. Syst.*, 34(1):132–144.
- Corina Florescu and Cornelia Caragea. 2017a. [A new scheme for scoring phrases in unsupervised keyphrase extraction](#). In *ECIR*, volume 10193 of *Lecture Notes in Computer Science*, pages 477–483.
- Corina Florescu and Cornelia Caragea. 2017b. [Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents](#). In *ACL (1)*, pages 1105–1115. Association for Computational Linguistics.
- Sujatha Das Gollapalli, Xiaoli Li, and Peng Yang. 2017. [Incorporating expert knowledge into keyphrase extraction](#). In *AAAI*, pages 3180–3187. AAAI Press.
- Maria P. Grineva, Maxim N. Grinev, and Dmitry Lizorkin. 2009. [Extracting key terms from noisy and multitheme documents](#). In *WWW*, pages 661–670. ACM.
- Kazi Saidul Hasan and Vincent Ng. 2014. [Automatic keyphrase extraction: A survey of the state of the art](#). In *ACL (1)*, pages 1262–1273. The Association for Computer Linguistics.
- Chong Huang, Yonghong Tian, Zhi Zhou, Charles X. Ling, and Tiejun Huang. 2006. [Keyphrase extraction using semantic networks structure analysis](#). In *ICDM*, pages 275–284. IEEE Computer Society.
- Anette Hulth. 2003. [Improved automatic keyword extraction given more linguistic knowledge](#). In *EMNLP*.
- Anette Hulth. 2004. [Enhancing linguistically oriented automatic keyword extraction](#). In *HLT-NAACL (Short Papers)*. The Association for Computational Linguistics.
- James Jardine and Simone Teufel. 2014. [Topical PageRank: A model of scientific expertise for bibliographic search](#). In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 501–510, Gothenburg, Sweden. Association for Computational Linguistics.
- Ganesh Jawahar, Benoît Sagot, and Djamel Seddah. 2019. [What does bert learn about the structure of language?](#) In *ACL (1)*, pages 3651–3657. Association for Computational Linguistics.
- Xin Jiang, Yunhua Hu, and Hang Li. 2009. [A ranking approach to keyphrase extraction](#). In *SIGIR*, pages 756–757. ACM.
- Karen Spärck Jones. 2004. [A statistical interpretation of term specificity and its application in retrieval](#). *J. Documentation*, 60(5):493–502.
- Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. [Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles](#). In *SemEval@ACL*, pages 21–26. The Association for Computer Linguistics.
- M. Krapivin and M. Marchese. 2009. Large dataset for keyphrase extraction.
- Mayank Kulkarni, Debanjan Mahata, Ravneet Arora, and Rajarshi Bhowmik. 2022. [Learning rich representation of keyphrases from text](#). In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 891–906. Association for Computational Linguistics.
- Niraj Kumar and Kannan Srinathan. 2008. [Automatic keyphrase extraction from scientific documents using n-gram filtration technique](#). In *ACM Symposium on Document Engineering*, pages 199–208. ACM.
- Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. [Unsupervised keyphrase extraction by jointly modeling local and global context](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 155–164, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yang Liu and Mirella Lapata. 2019. [Text summarization with pretrained encoders](#). In *EMNLP/IJCNLP (1)*, pages 3728–3738. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *CoRR*, abs/1907.11692.

- Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. [Clustering to find exemplar terms for keyphrase extraction](#). In *EMNLP*, pages 257–266. ACL.
- Debanjan Mahata, John Kuriakose, Rajiv Ratn Shah, and Roger Zimmermann. 2018a. [Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 634–639. Association for Computational Linguistics.
- Debanjan Mahata, Rajiv Ratn Shah, John Kuriakose, Roger Zimmermann, and John R. Talburt. 2018b. [Theme-weighted ranking of keywords from text documents using phrase embeddings](#). In *IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR 2018, Miami, FL, USA, April 10-12, 2018*, pages 184–189. IEEE.
- Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. [Deep keyphrase generation](#). In *ACL*, pages 582–592. Association for Computational Linguistics.
- Rada Mihalcea and Paul Tarau. 2004. [Textrank: Bringing order into text](#). In *EMNLP*, pages 404–411. ACL.
- Funan Mu, Zhenting Yu, Lifeng Wang, Yequan Wang, Qingyu Yin, Yibo Sun, Liqun Liu, Teng Ma, Jing Tang, and Xing Zhou. 2020. [Keyphrase extraction with span-based feature representations](#). *CoRR*, abs/2002.05407.
- David Newman, Nagendra Koilada, Jey Han Lau, and Timothy Baldwin. 2012. [Bayesian text segmentation for index term identification and keyphrase extraction](#). In *COLING*, pages 2077–2092. Indian Institute of Technology Bombay.
- Chau Q. Nguyen and Tuoi T. Phan. 2009. [An ontology-based approach for key phrase extraction](#). In *ACL/IJCNLP (Short Papers)*, pages 181–184. The Association for Computer Linguistics.
- Thuy Dung Nguyen and Min-Yen Kan. 2007. [Keyphrase extraction in scientific publications](#). In *ICADL*, volume 4822 of *Lecture Notes in Computer Science*, pages 317–326. Springer.
- Eirini Papagiannopoulou and Grigorios Tsoumakas. 2018. [Local word vectors guiding keyphrase extraction](#). *Inf. Process. Manag.*, 54(6):888–902.
- Eirini Papagiannopoulou and Grigorios Tsoumakas. 2019. [A review of keyphrase extraction](#). *CoRR*, abs/1905.05044.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *NAACL-HLT*, pages 2227–2237. Association for Computational Linguistics.
- Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. [Keyphrase extraction as sequence labeling using contextualized embeddings](#). In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*, volume 12036 of *Lecture Notes in Computer Science*, pages 328–335. Springer.
- Arnav Saxena, Mudit Mangal, and Goonjan Jain. 2020. [Keygames: A game theoretic approach to automatic keyphrase extraction](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2037–2048.
- Mingyang Song, Yi Feng, and Liping Jing. 2022a. [Hyperbolic relevance matching for neural keyphrase extraction](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5710–5720. Association for Computational Linguistics.
- Mingyang Song, Yi Feng, and Liping Jing. 2022b. [Utilizing BERT intermediate layers for unsupervised keyphrase extraction](#). In *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 277–281, Trento, Italy. Association for Computational Linguistics.
- Mingyang Song, Liping Jing, and Lin Xiao. 2021. [Importance Estimation from Multiple Perspectives for Keyphrase Extraction](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Si Sun, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu, and Jie Bao. 2020a. [Joint keyphrase chunking and salience ranking with bert](#). *CoRR*, abs/2004.13639.
- Yi Sun, Hangping Qiu, Yu Zheng, Zhongwei Wang, and Chaoran Zhang. 2020b. [Sifrank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model](#). *IEEE Access*, 8:10896–10906.
- Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. 2019. [Divgraphpointer: A graph pointer network for extracting diverse keyphrases](#). In *SIGIR*, pages 755–764. ACM.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. [Bert rediscovers the classical nlp pipeline](#). In *ACL (1)*, pages 4593–4601. Association for Computational Linguistics.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. [What do you](#)

- learn from context? probing for sentence structure in contextualized word representations. *CoRR*, abs/1905.06316.
- Takashi Tomokiyo and Matthew Hurst. 2003. [A language model approach to keyphrase extraction](#). pages 33–40. Association for Computational Linguistics.
- Peter D. Turney. 1999. Learning to extract keyphrases from text. *National Research Council Canada, Institute for Information Technology, Technical Report ERB-1057*.
- Peter D. Turney. 2000. [Learning algorithms for keyphrase extraction](#). *Inf. Retr.*, 2(4):303–336.
- Xiaojun Wan and Jianguo Xiao. 2008a. [Collabrank: Towards a collaborative approach to single-document keyphrase extraction](#). In *COLING*, pages 969–976.
- Xiaojun Wan and Jianguo Xiao. 2008b. [Single document keyphrase extraction using neighborhood knowledge](#). In *AAAI*, pages 855–860. AAAI Press.
- Rui Wang, Wei Liu, and Chris McDonald. 2015. [Using word embeddings to enhance keyword identification for scientific publications](#). In *Databases Theory and Applications - 26th Australasian Database Conference, ADC 2015, Melbourne, VIC, Australia, June 4-7, 2015. Proceedings*, volume 9093 of *Lecture Notes in Computer Science*, pages 257–268. Springer.
- Yanan Wang, Qi Liu, Chuan Qin, Tong Xu, Yijun Wang, Enhong Chen, and Hui Xiong. 2018. [Exploiting topic-based adversarial neural network for cross-domain keyphrase extraction](#). In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pages 597–606. IEEE Computer Society.
- Yansen Wang, Zhen Fan, and Carolyn Penstein Rosé. 2020. [Incorporating multimodal information in open-domain web keyphrase extraction](#). In *EMNLP (1)*, pages 1790–1800. Association for Computational Linguistics.
- Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. [Kea: Practical automatic keyphrase extraction](#). In *ACM DL*, pages 254–255. ACM.
- Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. [Open domain web keyphrase extraction beyond language modeling](#). In *EMNLP/IJCNLP (1)*, pages 5174–5183. Association for Computational Linguistics.
- Wei You, Dominique Fontaine, and Jean-Paul A. Barthès. 2009. [Automatic keyphrase extraction with a refined candidate set](#). In *Web Intelligence*, pages 576–579. IEEE Computer Society.
- Linhan Zhang, Qian Chen, Wen Wang, Chong Deng, Shiliang Zhang, Bing Li, Wei Wang, and Xin Cao. 2021. [Mderank: A masked document embedding rank approach for unsupervised keyphrase extraction](#). *CoRR*, abs/2110.06651.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with BERT](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. [HiBERT: document level pre-training of hierarchical bidirectional transformers for document summarization](#). *CoRR*, abs/1905.06566.
- Yongzheng Zhang, A. Nur Zincir-Heywood, and Evangelos E. Milios. 2004. [World wide web site summarization](#). *Web Intell. Agent Syst.*, 2(1):39–53.
- Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. [Extractive summarization as text matching](#). In *ACL*, pages 6197–6208. Association for Computational Linguistics.
- Xuan Zhou, Xiao Zhang, Chenyang Tao, Junya Chen, Bing Xu, Wei Wang, and Jing Xiao. 2021. [Multi-grained knowledge distillation for named entity recognition](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5704–5716. Association for Computational Linguistics.

Prompting for explanations improves Adversarial NLI. Is this true? {Yes} it is {true} because {it weakens superficial cues}

Pride Kavumba^{1,2} Ana Brassard^{2,1} Benjamin Heinzerling^{2,1} Kentaro Inui^{1,2}

¹Tohoku University ²RIKEN AIP
kavumba.pride.q2@dc.tohoku.ac.jp
{ana.brassard, benjamin.heinzerling}@riken.jp
kentaro.inui@tohoku.ac.jp

Abstract

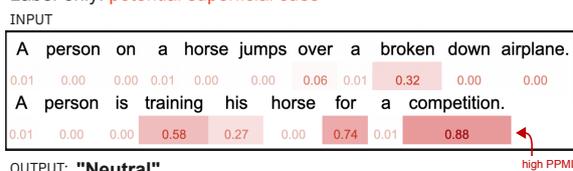
Explanation prompts ask language models to not only assign a label to a given input, such as *entailment* or *contradiction* in natural language inference (NLI) tasks, but also to generate a free-text explanation that supports this label. While explanation prompts originally introduced aiming to improve model interpretability, here we show that they also improve robustness to superficial cues. Compared to prompting for labels only, explanation prompting shows stronger performance on adversarial NLI benchmarks, outperforming the state of the art on ANLI, Counterfactually-Augmented NLI, and SNLI-Hard datasets. Analysis suggests that the increase in robustness is due to a reduction in the association strength between single tokens and labels, i.e., explanation prompting weakens superficial cues. More specifically, we find that single tokens that are highly predictive of the correct answer in the label-only setting become uninformative when the model also has to generate explanations.

1 Introduction

Explanation prompting requires language models to not only assign a particular label to a given input (henceforth: label-only prompting), but also to generate an explanation that supports this label. For example, given the natural language inference (NLI; Bowman et al., 2015) premise “A soccer game with multiple males playing” and the hypothesis “Some men are playing a sport”, in label-only prompting the model only has to generate a label such as *entailment*. With explanation prompting, the model has to generate not only the label but also an *explanation* that supports this label, such as “It is true because playing soccer is playing a sport”.

While explanation prompting was originally proposed for improving model interpretability (Narang et al., 2020), here we explore a different advantage: improved model performance on adversarial bench-

Label-only: potential superficial cues



Explanation prompting: neutralizes superficial cues

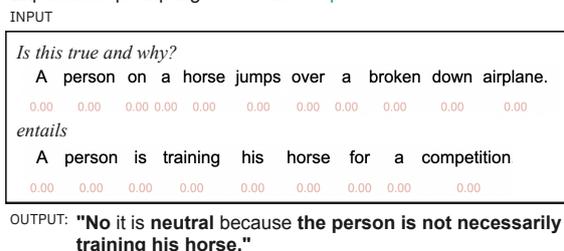


Figure 1: When models only have to predict class labels (top), some words in the input can become superficial cues, as indicated by high pointwise mutual information (shown in red) between words and class labels. With *explanation prompting* (bottom) the added requirement of generating explanations renders such shortcuts ineffective.

marks. Created in response to the discovery of superficial cues in many common datasets, adversarial benchmarks are designed to give a more realistic estimate of model performance. Non-adversarial benchmarks such as SNLI (Bowman et al., 2015) can contain superficial cues, i.e., single tokens that are predictive of the correct label and hence allow models to achieve high scores by taking “shortcuts” instead of acquiring and employing the capabilities intended by the task designers (Gururangan et al., 2018a; McCoy et al., 2019; Poliak et al., 2018; Niven and Kao, 2019; Sugawara et al., 2018; Schuster et al., 2019a; Kavumba et al., 2019). In contrast, adversarial benchmarks are created in a way that reduces or completely eliminates superficial cues, thus forcing models to solve tasks in the intended and generally more difficult manner.

In this work, we investigate the benefits of explanation prompting through the lens of adversarial benchmarks. Concretely, we finetune pre-trained language models on natural language inference datasets with explanation prompting and compare performance to label-only prompting. We find that explanation prompting improves performance across four adversarial NLI datasets and two non-adversarial NLI datasets (§5). Improvements are consistent across model architectures, model sizes, and prompt variations. Further analysis reveals that both the specific verbalization of the label (“*Yes, it is {label} because...*”) and the relation between explanation and label are important for model performance (§6). Finally, we verify that explanation prompting models do not rely on the kind of superficial cue that allows taking shortcuts in the label-only setting (§6). Source code is available at github.com/pkavumba/explanation-prompting.

2 Background and Related Work

2.1 Superficial Cues

In the original natural language inference setting, as exemplified by SNLI (Bowman et al., 2015), models are trained to assign a label, such as *entailment* or *contradiction*, to a given input. While models quickly achieved high evaluation scores, a line of research starting with Gururangan et al. (2018a) found that SNLI and other datasets contain superficial cues that models can exploit instead of learning the task as intended (Poliak et al., 2018; McCoy et al., 2019; Niven and Kao, 2019; Schuster et al., 2019a; Kavumba et al., 2019; Wang and Culotta, 2021; Srivastava et al., 2020; Wang et al., 2019b,c). For example, in SNLI, negations such as “not” are strongly associated with the *contradiction* label (Gururangan et al., 2018a). A model that predicts *contradiction* when the input contains the token “not” will achieve a high evaluation score without acquiring any capability to perform actual natural language inference. Having learned to rely on such shortcuts (Geirhos et al., 2020), models will be “right for the wrong reasons” (McCoy et al., 2019) on data that contains superficial cues, but will perform worse on data that does not.

There are several approaches to mitigate superficial cues. A direct countermeasure is to remove them from existing datasets and to take care not to introduce superficial cues when creating new datasets. The two dominant methods to do so are

removal of easy samples via adversarial filtering (Zellers et al., 2018, 2019; Sakaguchi et al., 2020; Bras et al., 2020; Nie et al., 2020) and augmentation with counterfactual examples that neutralize the association between existing superficial cues and labels (Kavumba et al., 2019; Schuster et al., 2019b; Kaushik et al., 2020). A complementary line of work aims to prevent models from relying on superficial cues, for example via adversarial training (Belinkov et al., 2019; Stacey et al., 2020, 2021) and adversarial attacks (Wang et al., 2019a; Liu et al., 2020; Zhu et al., 2020; Wang et al., 2021). Adversarial approaches suffer from drawbacks such as a more complex training scheme and higher computational costs. Another approach is multi-task training. Camburu et al. (2018) propose a “predict-and-explain” multi-task setup in which one model first predicts a label and a second model generates a free-form explanation for this label. However, this setup turns out to slightly degrade performance.

In this work, we study *explanation prompting* as a method for reducing the impact of superficial cues. While this form of prompting was originally introduced to enhance model interpretability (Narang et al., 2020), our work is most closely related to Chen et al. (2022), who studied the robustness of rationale models (Lei et al., 2016; Bastings et al., 2019; DeYoung et al., 2020) to adversarial attacks. Rationale models operate in a two-step “rationalize-then-predict” manner, where the model first selects a pertinent subset of the input, called a *rationale*, and then predicts a label given this rationale. Stacey et al. (2021) investigates using human annotated rationales for supervising attention mechanism. Their goal is to increase the attention given to annotated rationales.

2.2 Explanation Prompting

Explanation prompting requires models not only to predict a class label but also to provide an explanation of why that label is the correct answer. Previous work has explored explanation prompting as a way to improve model interpretability. Wiegreffe et al. (2021) analyzed the faithfulness of explanations obtained via explanation prompting. Since high-quality explanation are expensive to create and not available in large quantities, Marasovic et al. (2022) compare methods for generating high-quality explanations in limited data regimes, whereas Wiegreffe et al. (2022) investigate the fea-

sibility of using large language models such as GPT-3 (Brown et al., 2020) to automatically generate large amounts of explanations. In contrast to this strand of research, we use free-text explanations not to improve model interpretability, but to improve model robustness in adversarial settings.

3 Explanation Prompting for Adversarial NLI

In the original natural language inference setting, one trains a classifier to label the relationship between a *premise* and a *hypothesis* as *entailment*, *neutral*, or *contradiction*. When using a generative language model to generate a label and an explanation supporting the label, the task turns from classification into what we refer to as *explanation prompting*. Turning the original NLI instances into input-output pairs suitable for a generative language model necessitates choosing a *verbalizer*¹ that converts premise, hypothesis, label, and answer into an input prompt and a target output, e.g.:

- *Input*: Is this true and why? {premise} implies {hypothesis}
- *Output*: {Yes or No} it is {label} because {explanation}

Note that the label prediction process is further broken down into two steps: first, the model must provide a *binary* answer to the question (in this case, whether the statements are entailed), then give the exact label. That is, the output starts with “*Yes it is ...*” for Entailment, or “*No it is ...*” for Neutral and Contradiction. We refer to this as *multi-step verbalizing*. This is a deviation from previous work that utilized single-step or single-word verbalizers. For example, entailment is often verbalized as *yes*, while contradiction and neutral are verbalized as *no* and *maybe*, respectively (Schick and Schütze, 2021a,b). Finally, the output is completed with a free-text explanation supporting the label.

4 Experimental Setup

4.1 Datasets

We compare label-only and explanation prompting on six NLI datasets.

e-SNLI (Camburu et al., 2018) extends SNLI (Bowman et al., 2015) with crowdsourced free-form explanations and annotated salient spans.

¹Verbalizer details are given in Appendix A.

Adversarial NLI (ANLI) (Nie et al., 2020) was created in an iterative, adversarial process where, in each iteration, human annotators create examples that a given model does not label correctly, which are then used to train a stronger model.

SNLI Hard (Gururangan et al., 2018a) is a filtered version of the SNLI test set and contains only instances that could not be labeled correctly by a model given only the hypothesis as input.

NLI Diagnostic (Wang et al., 2018) was carefully constructed to evaluate capabilities related to commonsense knowledge, logical reasoning, predicate-argument structures, and lexical semantics.

Heuristic Analysis for NLI Systems (HANS) (McCoy et al., 2019) was created to analyze and prevent several kinds of shortcuts found in prior NLI datasets, such as lexical overlap between premise and hypothesis.

Counterfactually-Augmented NLI (Counter-NLI) (Kaushik et al., 2020) augments a subset of SNLI with counterfactual instances, which were obtained by editing either the premise or hypothesis so that a counterfactual, i.e., different than the original, label becomes true. Models relying on superficial cues will perform well on original SNLI instances, but poorly on counterfactual ones.

4.2 Models and Training Details

The two main models selected for our comparison are T0 (Sanh et al., 2021) and T5-3B (Raffel et al., 2020). We chose these two models based on their good reported performance on NLI datasets while involving comparably low computational costs, which we further reduced by finetuning all models only on a third of the available e-SNLI and ANLI training data. Thus, test results on e-SNLI, ANLI, and SNLI Hard can be considered in-domain tests and results on the remaining datasets out-of-domain tests. Further training details and hyperparameter settings are given in Appendix B.

5 Results

Does explanation prompting improve robustness to adversarial attacks? *Yes.*

For both T0 and T5-3B, training with explanation prompting improved performance over label-only prompting on nearly all datasets, surpassing the reported state of the art on e-SNLI, SNLI-Hard,

Dataset	Subset	Current SOTA	T5-3B		T0 (11B)	
			Label-only	Explanation prompting	Label-only	Explanation prompting
e-SNLI	-	92.3	91.7	95.1	91.0	91.9
SNLI Hard	Hard	80.2	84.0	89.7	83.0	84.5
ANLI	R1	75.5	74.9	81.8	69.6	75.6
	R2	51.4	58.9	72.5	53.7	60.6
	R3	49.8	57.9	74.8	55.0	59.9
HANS	Lex	94.1	94.2	94.2	97.9	95.9
	Sub	46.3	46.3	30.3	20.5	37.9
	Cons	38.5	38.6	17.1	24.3	53.9
Counter-NLI	RP	54.3	69.6	83.0	66.5	69.2
	RH	74.3	88.9	93.5	87.9	87.4
	RP&RH	64.3	79.3	88.3	77.2	78.3
NLI Diagnostic	Know	53.9	58.8	76.4	58.8	59.9
	Logic	58.7	63.7	73.9	60.7	64.5
	LS	66.5	69.6	79.3	63.0	70.4
	PAS	69.9	73.1	80.9	70.8	72.4

Table 1: Results by T5-3B and T0 (11B) models trained with label-only prompting and Explanation prompting. Current state-of-the-art results on each dataset are reported from: WT5 (Narang et al., 2020), BERT-Sup-ATT (Stacey et al., 2021), InfoBERT (Wang et al., 2021), RoBERTa-AFLITE (Bras et al., 2020), BERT (Kaushik et al., 2020), and RoBERTa-AFLITE (Bras et al., 2020), respectively. Note that T5-3B and T0 are trained with different batch sizes and sequence lengths, so the results are not comparable (§ 5).

ANLI, and Counterfactually-Augmented NLI (Table 1). For example, on the three ANLI subsets T5-3B achieves accuracies of 81.8%, 72.5%, and 74.8% with explanation prompting, compared to much lower accuracies of 74.9%, 58.9% and 57.9% with label-only prompting. Furthermore, since e-SNLI does not contain any adversarially chosen “hard” instances, strong results on this dataset show that explanation prompting does not necessarily hurt performance on datasets with superficial cues. Overall T5-3B achieves higher performance despite its smaller size, but this is due to T0 using a quarter of the batch size and sequence length of that used for T5 due to memory limitations.

The HANS dataset remains the most challenging dataset, indicating that the models may still be susceptible to such adversarial attacks. Surprisingly, the use of explanation prompting actually leads to *degraded* performance for certain subsets, such as the lexical overlap for T0, and subsequences and constituents for T5-3B. This discrepancy warrants further investigation, which we leave for future work.

On all other datasets, explanation prompting models show clear improvements over label-only models in both in-domain and adversarial out-of-domain settings. This demonstrates that us-

	Full	H-only	Δ
Label-only prompting	87.2	63.7	-23.5
Explanation prompting	90.9	33.1	-57.8
Random baseline	33.3	33.3	-

Table 2: The average prediction accuracy of T0 models on e-SNLI when trained with the full input compared to the hypothesis-only setting (H-only), which allows the models to solely rely on superficial cues to make accurate predictions. The explanation prompting-trained model’s performance degraded to random performance, implying that it did not learn to make use of superficial cues.

ing explanation prompting generally enhances the model’s robustness to adversarial attacks and improves the overall NLI prediction performance.

6 Discussion

In this section we vary experimental settings and conduct ablations in order to provide a more detailed analysis of how explanation prompting impacts NLI performance. Unless stated otherwise, reported results are obtained by finetuning T0 on 20K randomly-sampled instances from e-SNLI and averaging prediction accuracies from three runs

Explanation	Accuracy	BLEU
None	88.4	-
Random characters	21.00	0.02
Random words	0.00	0.90
Low-sim. sentences	0.04	0.03
High-sim. sentences	59.1	1.73
Original (e-SNLI)	91.6	36.1

Table 3: Impact of explanation content. Accuracies are shown for the e-SNLI dev set, averaged over three T0 models finetuned with different random seeds. Random characters and words or unrelated explanations significantly reduce performance, indicating that the models did not rely on superficial cues. Original explanations outperform extracted sentences with high similarity, demonstrating the benefit of related explanations. We also report BLEU scores with respect to the original explanations.

with different random seeds. Training details are given in Appendix B.

Does explanation prompting prevent models from exploiting superficial cues? *Yes.*

To see if models still exploit superficial cues, we employ the hypothesis-only setting of Gururangan et al. (2018b). Since the missing premise makes the task impossible, any performance above random chance can be ascribed to models picking up on superficial cues. After training one model with label-only prompting and one with explanation prompting (Table 2), we observe that the label-only model considerably exceeds random chance (63.7% compared to 33.3%). In contrast, the explanation prompting model does not exceed random chance, indicating that explanation prompting is not conducive to shortcut learning.

Do the explanations need to be related to the input? *Yes.*

To check if the content of the target explanation matters we replace it with unrelated text ranging from completely random characters to similar but unrelated sentences and find that explanation prompting with the original explanations still performs best (Table 3). Specifically, we choose the following target “explanations”: (i) random characters, (ii) random words, (iii) sentences extracted from the BookCorpus (Zhu et al., 2015) with *low* similarity with the input, and (iv) sentences extracted from the BookCorpus with *high* similar-

ity.² All similarities are computed with SentenceBERT (Reimers and Gurevych, 2019). We also compare to label-only prompting (*None* row in Table 3). Table 3 shows the mean prediction accuracy scores on the development set of e-SNLI over three random seeds. Performance degrades with random explanations or sentences extracted from BookCorpus, confirming that training the model to predict explanations improves adversarial robustness.

Does *Multi-step* verbalizing have an effect on the model performance? *Yes.*

A binary decision step may seem like a small change in prompt format, however, we found that this added step has partial merit to the improvement in performance. To verify this, we train label-only and explanation prompting models using *single-step*-verbalized prompts (*{label} because...*) and ones using *multi-step*-verbalized prompts (“*Yes it is {label} because...*”), both with an explanation (+Explain column in Table 5) and without added explanations (Label-only column in Table 5). The results show the prediction accuracy averaged over three random seeds. In both label-only and explained settings, adding a multi-step verbalizer brings an improvement over the single-step version.

Are the models sensitive to prompt wording in the input? *No.*

Previous work has demonstrated that language models can be very sensitive to the prompts (Schick and Schütze, 2021a,b; Brown et al., 2020). To examine this, we conduct experiments on five diverse crowdsourced prompts obtained from the Prompt Source project (Bach et al., 2022). For each model, we run three separate experiments using three different random seeds. We report the average accuracy across all five prompts on the development set of e-SNLI and Counterfactually-Augmented NLI. Due to resource constraints, we use T5-3B, a smaller model than T0, for these experiments. Furthermore, we limit the number of instances used from e-SNLI to twenty thousand randomly selected examples.

The results presented in Table 4 demonstrate that models trained with explanation prompting outperform those trained with label-only prompting across all five prompts in terms of accuracy (see Table 8 in Appendix C for results with different models). For instance, the explanation prompt-

²We use the BookCorpus instead of sampling random explanations to avoid accidentally sampling valid explanations.

Dataset	Prompt ID					Mean _(stddev)
	1	2	3	4	5	
e-SNLI	91.5 / 94.4	91.6 / 94.7	91.8 / 94.6	91.6 / 94.5	91.8 / 94.4	91.7 _(0.1) / 94.5 _(0.1)
CNLI (RP)	70.8 / 82.5	73.0 / 83.8	71.5 / 83.0	72.1 / 83.0	70.1 / 82.8	71.5 _(1.1) / 83.0 _(0.5)
CNLI (RH)	82.0 / 92.3	82.8 / 93.0	81.9 / 92.8	82.0 / 92.2	83.1 / 92.5	82.4 _(0.6) / 92.6 _(0.3)
CNLI (RP&RH)	76.4 / 87.4	77.9 / 88.4	76.7 / 87.9	77.0 / 87.6	76.6 / 87.7	76.9 _(0.6) / 87.8 _(0.4)

Table 4: Prompt sensitivity on the development set of e-SNLI and Counterfactually-Augmented NLI (CNLI). Values are accuracy of label-only/explanation prompting-trained T5-3B models averaged over three random seeds. Besides the consistently higher performance of the explanation prompting setting, the lower standard deviation indicates greater stability w.r.t. prompt format.

	Label-only	+Explain (e-SNLI)
Single-step	87.2	90.9
Multi-step	88.4	91.6

Table 5: Comparing the effects of single-step (“[label] because ...”) and multi-step (“Yes/no, it is [label] because ...”) verbalizing on T0 prediction accuracy, both with an explanation (+Explain) and without an explanation (Label-only) in the model output. Multi-step verbalizing improved the accuracy in both cases, with and without explanation, and the added task of providing an explanation (+Explain) further enhanced performance.

ing model achieves an overall average accuracy of 94.5% on e-SNLI compared to 91.7% for the label-only model. Additionally, the explanation prompting model exhibits better accuracy on all the individual prompts on the adversarial Counterfactually-Augmented NLI with an overall average accuracy of 83.0% on the revised premise (RP), 92.6% on the revised hypothesis (RH), and 87.8 on RP&RH, compared to 71.5%, 82.4%, and 76.9% respectively for the label-only model. The lower standard deviations for explanation prompting also indicate higher stability across all prompts.

Are the results dependent on the architecture/size of the model employed? *Yes.*

To study the impact of model size on performance, we repeat the experiments using six models ranging from 60 million to 11 billion parameters. These models comprise two versions of BART (Lewis et al., 2020) with 125M and 400M parameters, as well as three variants of T5 (Raffel et al., 2020) with 60M, 770M, and 3B parameters, and T0 with 11B parameters. The results, as depicted in Figure 3 for ANLI, indicate a clear correlation between model size and performance, with larger models demonstrating improved results. It is worth noting that

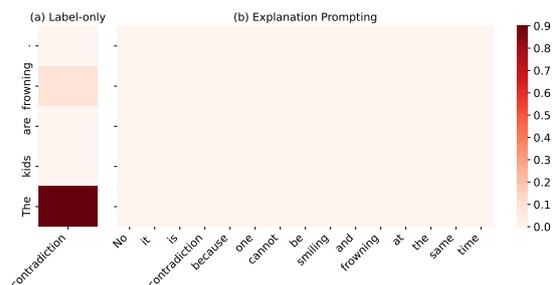


Figure 2: Positive Pointwise Mutual Information (PPMI) statistics for hypothesis words and labels with and without explanation prompting. Words in the hypothesis are strongly associated with the predict-only labels. With explanation prompting, the association between the hypothesis words and the labels drops to a zero. For example, while the negative word *frowning* is strongly associated with *contradiction* label, the association is eliminated with explanation prompts. The figure only shows the hypothesis because superficial cues are from the hypothesis, not the premise.

the highest achieved performance by T5-3B, exceeds that of the larger T0 model. This is due to the fact that T0 is trained using only a quarter of the batch size and the sequence length used for all other models, resulting in reduced performance. Comprehensive results for all models and datasets are presented in Table 6. For more information on the training details, see Appendix B.

Does explanation prompting weaken the association between word-level superficial cues and labels? *Yes.*

To investigate the impact of explanation prompting on the association between input words and their corresponding output labels in the training set, we compare the positive pointwise mutual information (PPMI) between them in both label-only and explanation prompting settings:

		Current	T5-Small (60M)		BART-Base (125M)		BART-Large (400M)		T5-Large (770M)		T5-3B (3B)		T0* (11B)	
		SOTA	LP	EP	LP	EP	LP	EP	LP	EP	LP	EP	LP	EP
e-SNLI		92.3	82.4	88.8	88.7	92.1	90.4	93.8	90.9	94.4	91.7	95.1	91.0	91.9
SNLI Hard		80.2	68.5	82.2	78.1	84.3	81.5	84.9	82.1	88.7	84.0	89.7	83.0	84.5
ANLI	R1	75.5	46.5	52.5	56.8	53.0	64.9	65.9	66.1	77.2	74.9	81.8	69.6	75.6
	R2	51.4	37.6	56.4	41.5	50.3	44.4	57.1	49.2	67.8	58.9	72.5	53.7	60.6
	R3	49.8	40.4	59.1	40.9	54.0	46.5	59.6	49.4	68.0	57.9	74.8	55.0	59.9
HANS	Lex	94.1	2.6	0.0	71.2	69.6	85.0	90.2	82.9	81.3	94.2	94.2	97.9	95.9
	Sub	46.3	2.2	0.0	43.2	54.1	27.3	63.7	35.6	27.6	46.3	30.3	20.5	37.9
	Cons	38.5	2.5	0.0	34.7	51.9	22.4	63.8	19.6	9.9	38.6	17.1	24.3	53.9
Counter-NLI	RP	54.3	54.1	75.6	59.8	74.9	66.1	77.3	67.8	82.3	69.6	83.0	66.5	69.2
	RH	74.3	78.4	86.5	82.9	87.8	85.3	87.4	86.5	92.4	88.9	93.5	87.9	87.4
	RP&RH	64.3	66.3	81.1	71.3	81.3	75.7	82.3	77.1	87.3	79.3	88.3	77.2	78.3
NLI Diagnostic	Know	53.9	34.5	58.8	41.2	60.2	57.4	70.4	54.9	65.8	58.8	76.4	58.8	59.9
	Logic	58.7	45.3	59.6	45.6	67.0	54.9	67.0	57.4	70.3	63.7	73.9	60.7	64.5
	LS	66.5	49.5	63.3	49.2	62.2	62.2	69.6	63.9	76.1	69.6	79.3	63.0	70.4
	PAS	69.9	58.0	69.3	55.7	65.3	67.9	66.7	71.0	76.4	73.1	80.9	70.8	72.4

Table 6: Average prediction accuracy over three random seeds by models of increasing size trained with label-only (LP) and explanation prompting (EP). Current state-of-the-art results on each dataset are reported from: WT5 (Narang et al., 2020), BERT-Sup-ATT (Stacey et al., 2021), InfoBERT (Wang et al., 2021), RoBERTa-AFLITE (Bras et al., 2020), BERT (Kaushik et al., 2020), and RoBERTa-AFLITE (Bras et al., 2020), respectively. *Note that the T0 models were trained using only a quarter of the batch size and half the sequence length used for all other models due to computational limitations. This may be the cause for weaker performance compared to the smaller T5-3B models.

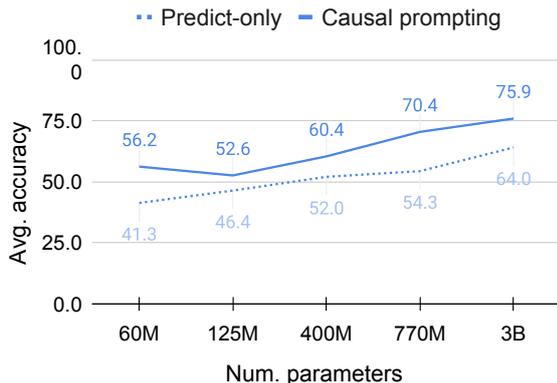


Figure 3: Average accuracy on ANLI depending on model size with label-only and explanation prompt training, respectively. See Appendix B for a comprehensive overview of performance, including on other datasets.



Figure 4: Crowd-sourced comparison of human-written explanations and those generated by a model. Overall, the crowd workers found that the model-generated explanations were either comparable to or better than the human-written ones.

$$PPMI(w, l) = \max(\log \frac{p(w, l)}{p(w)p(l)}, 0)$$

Where w represents the input word and l represents the output label. The PPMI analysis enables us to determine which words have a strong association with specific output labels, and how the use of explanation prompts modifies these associations. Following Gururangan et al. (2018b), we use add-100 smoothing in our PPMI calculations to highlight the input words that exhibit the strongest associations with the output labels.

The results, as presented in Figure 2, show that explanation prompts weaken the association between the input words and the output labels. For instance, in the absence of explanation prompting, the negative word *frowning* had a strong association with the label *contradiction*. However, when explanation prompting is used, this association is diminished from 0.1 to around 0. It's worth mentioning that only the hypothesis words are shown in Figure 2 as the superficial cues are mainly present in the hypothesis, not the premise. These findings align with the results obtained from the hypothesis-

only model, as presented in Table 2.

Are the explanations generated by explanation prompting models plausible? *Yes.*

While interpretability is not the primary objective of this study, we conduct a human evaluation of the model’s generated explanations to assess its performance within the intended framework. We use Amazon Mechanical Turk to gather assessments from 100 randomly selected T0 instances. To make the task easier for the crowd workers, we simplify it to only include two labels: Entailment and non-entailment. Instances labeled as Neutral or Contradiction are considered as non-entailment. We present the crowd workers with the gold label and request an evaluation of the quality of the *explanation* using a five-point Likert scale, ranging from “very bad” to “extremely good”. We gather three ratings per instance. Each Human Intelligence Task (HIT) features three explanations: the gold explanation authored by a human, the model-generated explanation, and an “attention check” explanation with a known expected annotation. The “attention check” explanation is included to ensure the quality of the annotations provided. This “attention check” explanation is randomly selected and unrelated to the *premise* and *hypothesis*, and therefore is expected to receive a lower score compared to the human-authored explanations that have already been validated by other crowd workers in previous work (Camburu et al., 2018). If the “attention check” explanation receives a high score or an equal score to a human-authored explanation, the HIT is flagged for review. Additionally, to prevent the use of simple heuristics, such as assuming that the last explanation is always the low-scoring one, the order in which the explanations are presented to the annotator is randomly shuffled during each HIT. An example of an NLI instance with an “attention check” explanation is shown below:³

- **Premise:** A man stands by an animal rights sign at an outdoor event.
- **Hypothesis:** A man is standing inside of his house
- **Human:** an outdoor event is not in his house
- **Generated:** The man cannot be standing inside of his house and at an outdoor event at the same time.

³The crowdsourcing study form can be found in the appendix D.

- **Attention Check:** It cannot be inferred that the young woman is an artist or that she is be finished soon.

The results of this evaluation are shown in figure 4. On the whole, crowd workers found model-generated explanations to be comparable to or better than human-written ones.⁴

This result suggests that the crowd workers found the generated reasons to be of similar quality to the human-authored reasons. This indicates that the model learns important features of the input data and is able to use them effectively to generate reasonable explanations.

7 Conclusions

In this study, we examined the influence of causal prompting on the adversarial robustness of natural language processing models. Our results indicate that using causal prompts can improve a model’s robustness to adversarial attacks. We also explored the performance of our models under various modified and ablated settings and found that explanation prompting-trained models (i) no longer rely on superficial cues, (ii) benefit most from both causally related explanations and multi-step verbalization, and (iii) are robust to differences in the input prompts. In addition, we observed that performance increases with model size and that the use of the explanation prompting format reduces the association between input words and output labels. Finally, human evaluation showed that the models generated plausible explanations.

Limitations

Explanation prompting requires datasets annotated with explanations, which may not always be available and it can be costly to collect explanations in a quantity suitable for model training or finetuning. Therefore, applying this method to datasets without explanations may be difficult.

Additionally, our analysis and evaluation are limited to English language benchmarks. Although we anticipate the method to be transferable to other languages, this requires further investigation.

Finally, experiments showed that training with explanation prompting did not improve the performance of T5 variants on the "Subsequence" and "Constituent" subsets of the HANS dataset. It is

⁴Refer to Appendix E for some example explanations.

currently unclear why all variants of BART performed better than random baselines, while T5 variants did not. Possible explanations include differences in training data, model architecture, and optimization goal, but this discrepancy requires further investigation.

Ethics Statement

In our human evaluation of the explanation quality, we took steps to ensure fair treatment of the crowd workers. To do this, we internally conducted experiments to determine the average completion time for one Human Intelligence Task (HIT). Our goal was to pay a fair wage of at least \$20 per hour to all participants involved in the study. As a result, all crowd workers received fair compensation for their work, and no HITs were rejected.

Acknowledgements

We thank the reviewers for their constructive feedback. We also thank all the crowd workers who have contributed to this research by rating the explanations. This work was supported by JST CREST Grant JPMJCR20D2 and JSPS KAKENHI Grant JP21K17814, Japan.

References

- Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Alshaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. [Prompt-Source: An integrated development environment and repository for natural language prompts](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.
- Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. [Interpretable neural predictions with differentiable binary variables](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.
- Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019. [On adversarial removal of hypothesis-only bias in natural language inference](#). In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 256–262, Minneapolis, Minnesota. Association for Computational Linguistics.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. [Adversarial filters of dataset biases](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. [e-nnli: Natural language inference with natural language explanations](#). In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc.
- Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. 2022. [Can rationalization improve robustness?](#) In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3792–3805, Seattle, United States. Association for Computational Linguistics.
- Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. [ERASER: A benchmark to evaluate rationalized NLP models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. [Shortcut learning in deep neural networks](#). *Nature Machine Intelligence*, 2(11):665–673.

- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018a. [Annotation artifacts in natural language inference data](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018b. [Annotation artifacts in natural language inference data](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually augmented data. *International Conference on Learning Representations (ICLR)*.
- Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reiser, and Kentaro Inui. 2019. [When choosing plausible alternatives, clever hans can be clever](#). In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 33–42, Hong Kong, China. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. [Rationalizing neural predictions](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Austin, Texas. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. [Adversarial training for large neural language models](#).
- Ana Marasovic, Iz Beltagy, Doug Downey, and Matthew Peters. 2022. [Few-shot self-rationalization with natural language prompts](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 410–424, Seattle, United States. Association for Computational Linguistics.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. [Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.
- Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. [Wt5?! training text-to-text models to explain their predictions](#).
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Timothy Niven and Hung-Yu Kao. 2019. [Probing neural network comprehension of natural language arguments](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. [Hypothesis only baselines in natural language inference](#). In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. [Zero: Memory optimizations toward training trillion parameter models](#).
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021. [Zero-offload: Democratizing billion-scale model training](#).
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. [Winogrande: An adversarial winograd schema challenge at scale](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.

- Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Stella Biderman, Leo Gao, Tali Bers, Thomas Wolf, and Alexander M. Rush. 2021. [Multi-task prompted training enables zero-shot task generalization](#).
- Timo Schick and Hinrich Schütze. 2021a. [Exploiting cloze-questions for few-shot text classification and natural language inference](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.
- Timo Schick and Hinrich Schütze. 2021b. [It’s not just size that matters: Small language models are also few-shot learners](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.
- Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019a. [Towards debiasing fact verification models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, Hong Kong, China. Association for Computational Linguistics.
- Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019b. [Towards debiasing fact verification models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3410–3416, Hong Kong, China. Association for Computational Linguistics.
- Megha Srivastava, Tatsunori Hashimoto, and Percy Liang. 2020. [Robustness to spurious correlations via human annotations](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9109–9119. PMLR.
- Joe Stacey, Yonatan Belinkov, and Marek Rei. 2021. [Supervising model attention with human explanations for robust natural language inference](#).
- Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Sebastian Riedel, and Tim Rocktäschel. 2020. [There is strength in numbers: Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training](#).
- Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. [What makes reading comprehension questions easier?](#) In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. [Infobert: Improving robustness of language models from an information theoretic perspective](#). In *International Conference on Learning Representations*.
- Dilin Wang, Chengyue Gong, and Qiang Liu. 2019a. [Improving neural language modeling via adversarial training](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6555–6565. PMLR.
- Haohan Wang, Songwei Ge, Zachary C. Lipton, and Eric P. Xing. 2019b. [Learning robust global representations by penalizing local predictive power](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 10506–10518.
- Haohan Wang, Zexue He, Zachary C. Lipton, and Eric P. Xing. 2019c. [Learning robust representations by projecting superficial statistics out](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Zhao Wang and Aron Culotta. 2021. [Robustness to spurious correlations in text classification via automatically generated counterfactuals](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14024–14031.
- Sarah Wiegrefe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. 2022. [Reframing human-AI collaboration for generating free-text explanations](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 632–658, Seattle, United States. Association for Computational Linguistics.
- Sarah Wiegrefe, Ana Marasović, and Noah A. Smith. 2021. [Measuring association between labels and](#)

- [free-text rationales](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.
- Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. [SWAG: A large-scale adversarial dataset for grounded commonsense inference](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [HellaSwag: Can a machine really finish your sentence?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.
- Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. [Freelb: Enhanced adversarial training for natural language understanding](#). In *International Conference on Learning Representations*.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

Appendix

A Prompt templates

Inspired by Unified Prompts (Sanh et al., 2021) and Prompt Source project (Bach et al., 2022), we express all explanation prompting input-output pairs in the jinja2 template language.⁵ This choice allows us to take advantage of the many features and benefits offered by jinja2.

A.1 Explanation prompting template

Input:

Is this true and why?

```
{{premise}} implies {{hypothesis}}
```

Output:

```
{% if label == 'entailment' %} Yes {%else%} No  
{%endif%} it is {{label}} because  
{{explanation}}
```

A.2 Alternative template for training samples lacking explanations

Input:

Is this true?

```
{{premise}} implies {{hypothesis}}
```

Output:

```
{% if label == 'entailment' %} Yes {%else%} No  
{%endif%} it is {{label}}
```

Note that ANLI provides explanations for only some of the questions; where missing, the prompt template was modified to accommodate this.

B Training details

We fine-tuned the models on an Nvidia A100 node with 8 x 40GB GPUs. We used the DeepSpeed library⁶ that implements ZeRo (Rajbhandari et al., 2020) and ZeRo-Offload (Ren et al., 2021); and the Huggingface transformers library (Wolf et al., 2019). We used an Adam optimizer (Kingma and Ba, 2015) with a learning rate of {1e-4, 5e-5}, with a per device batch size of {8, 16, 32, 64}, warm-up ratio of 0.08, max source length of 1024 except for T0 which uses 512 tokens with dynamic padding based on the longest sequence in the batch. We fine-tuned for a maximum of three epochs and selected the best checkpoint based on performance on the e-SNLI development set. Table 7 shows an overview of all used hyperparameters.

⁵<https://https://jinja.palletsprojects.com/>

⁶<https://github.com/microsoft/DeepSpeed>

Models	
Warmup Ratio	0.08
Per Device Batch Size	{2, 4, 8, 16, 32, 64}
Learning Rate	{1e-3, 1e-4*, 1e-5, 5e-5*}
Adam ϵ	1.00e-08
Adam β_1	0.9
Adam β_2	0.999
Gradient Norm	1
Max Source Len	{512, 1024}
Max Target Len	256
weight_decay	0
fp16	yes
DeepSpeed	
fp16	
enabled	yes
loss_scale	0
loss_scale_window	1,000
initial_scale_power	16
hysteresis	2
min_loss_scale	1
zero_optimization	
sub_group_size	1.00e9
stage3_max_live_parameters	1.00e9
stage3_max_reuse_distance	1.00e9

Table 7: Hyperparameter settings. Where multiple values were tried the final values used is shown with an asterisk. The batch size of 8 is only used for the 11Billion parameter model.

C Prompt Sensitivity Results

We present extended results on prompt sensitivity with a range of model sizes, with all values being averages over three random seeds. Table 8 shows the results on each prompt. The aim of the experiment is to examine the sensitivity of various models to prompt wording. To do this, we evaluated the models on five diverse crowdsourced prompts obtained from the Prompt Source project (Bach et al., 2022). We conducted three separate experiments for each model, using three different random seeds, and report the average accuracy across all five prompts on the development sets of e-SNLI and Counterfactually-Augmented NLI. We limited the number of instances used from e-SNLI to twenty thousand randomly selected examples. The results demonstrate that models trained with explanation prompting outperform those trained with label-only across all five prompts in terms of accuracy.

D Crowd sourcing Forms

In this section, we present the crowdsourcing form utilized for the human evaluation of explanation

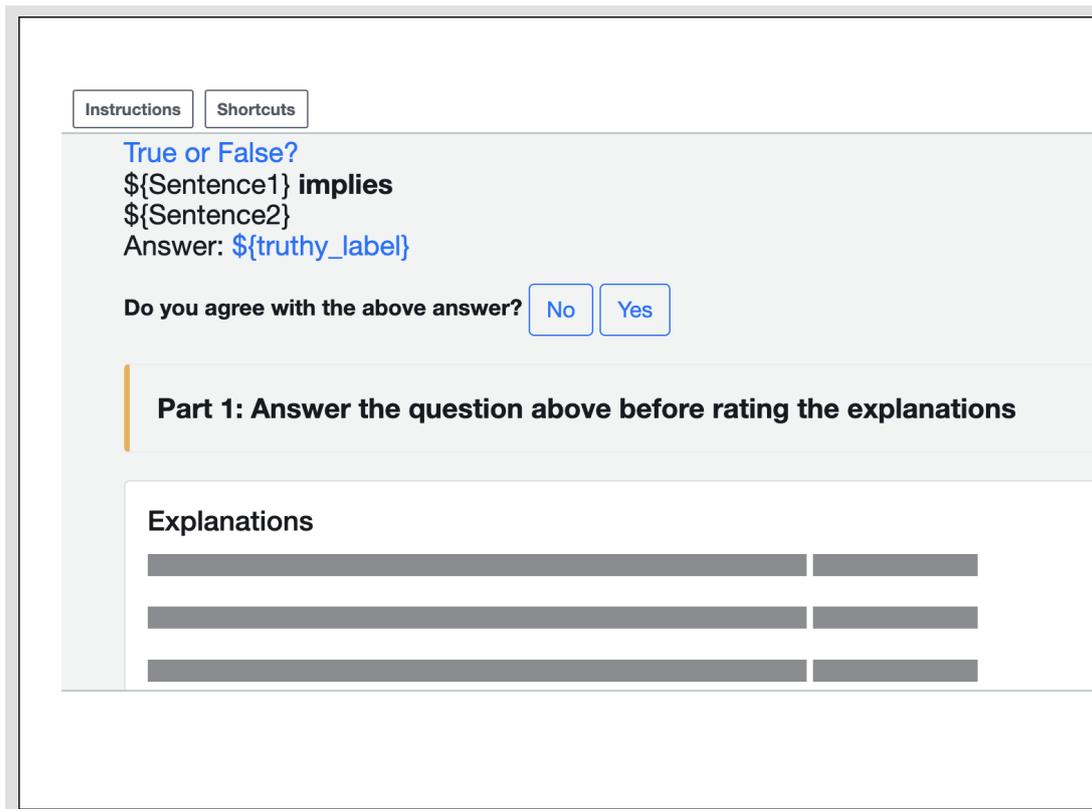
Dataset	Prompt_ID	T5-Small		T5-Large		T5-3B		BART-B		BART-L	
		LO	EP	LO	EP	LO	EP	LO	EP	LO	EP
e-SNLI	1	71.1	67.3	89.2	93.3	91.5	94.4	84.5	90.1	88.6	93.0
	2	66.7	69.3	89.6	93.3	91.6	94.7	84.3	90.3	59.1	93.4
	3	72.3	67.7	89.5	93.2	91.8	94.6	84.0	90.5	88.2	93.3
	4	69.0	66.4	89.6	93.6	91.6	94.5	84.1	90.5	72.3	93.4
	5	69.2	67.4	89.5	93.4	91.8	94.4	84.7	90.3	85.8	93.3
CNLI (RP)	1	44.8	60.3	64.4	79.6	70.8	82.5	50.1	72.3	60.1	77.9
	2	42.8	68.5	65.4	79.2	73.0	83.8	49.3	71.8	41.9	77.2
	3	47.5	68.4	65.9	79.0	71.5	83.0	49.3	72.0	59.9	76.6
	4	43.3	68.3	66.2	79.3	72.1	83.0	49.0	71.8	49.0	79.0
	5	43.4	68.8	65.9	79.7	70.1	82.8	48.8	71.6	58.3	77.2
CNLI (RH)	1	65.6	62.7	80.6	91.1	82.0	92.3	73.0	86.5	79.3	90.8
	2	54.8	70.6	81.1	91.8	82.8	93.0	72.2	86.2	54.1	90.5
	3	67.5	69.4	81.3	91.3	81.9	92.8	73.3	85.7	79.4	91.0
	4	63.2	68.8	81.6	91.4	82.0	92.2	73.0	87.3	65.6	90.8
	5	60.7	68.7	82.3	91.1	83.1	92.5	71.8	85.3	77.0	90.9
CNLI (RP&RH)	1	55.2	61.5	72.5	85.3	76.4	87.4	61.5	79.4	69.7	84.3
	2	48.8	69.5	73.3	85.5	77.9	88.4	60.7	79.0	48.0	83.8
	3	57.5	68.9	73.6	85.1	76.7	87.9	61.3	78.8	69.7	83.8
	4	53.3	68.5	73.9	85.3	77.0	87.6	61.0	79.5	57.3	84.9
	5	52.0	68.8	74.1	85.4	76.6	87.7	60.3	78.4	67.7	84.0

Table 8: Prompt-sensitivity results on the development set of e-SNLI and Counterfactually-Augmented NLI (CNLI). The values represent mean accuracy over three random seeds. The table compares the accuracy of a label-only prompting model (represented by the LO column) and the explanation prompting model (represented by the EP column). Explanation prompting models outperform the label-only model on almost all the prompts.

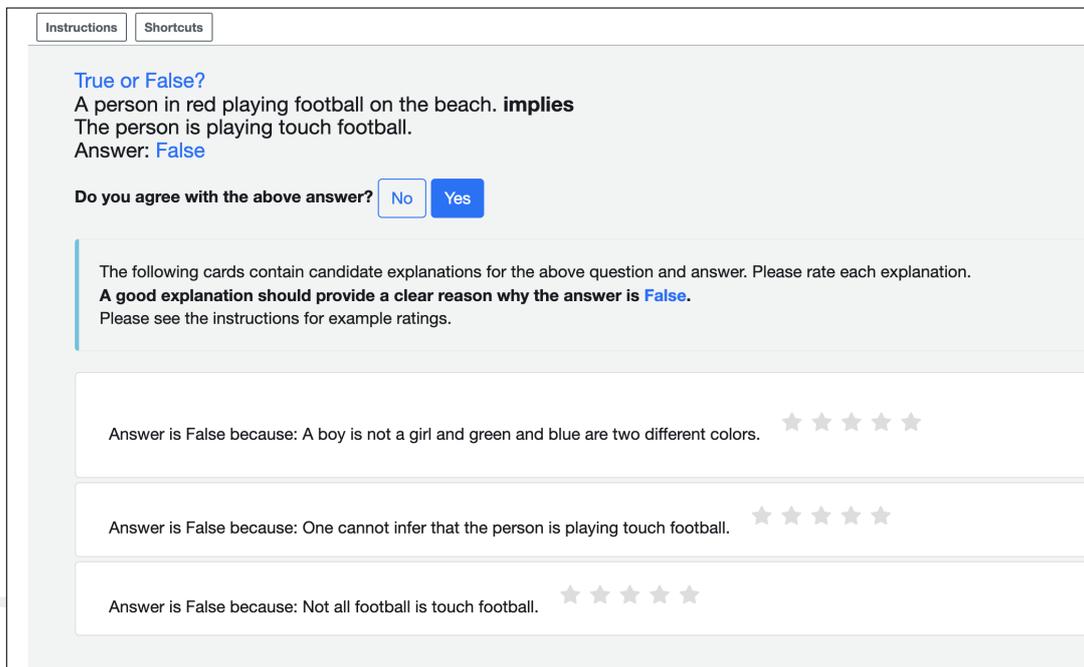
quality. To ensure unbiased assessments, the explanations were initially concealed from the crowd workers while they read the question and answer (as depicted in Fig.5a). Once the workers answered the first question, the explanations were then revealed (as shown in Fig.5b).

E Qualitative Results

Table 9 shows some qualitative examples of the *explanations* generated by the explanation prompting T0 model.



(a) The figure depicts the process of evaluating the quality of the explanations in the crowd-sourcing task. The first stage requires the user to carefully read both the premise and hypothesis presented. After familiarizing themselves with the information, the user is then tasked with answering the corresponding question. At this stage, the explanations for the question are initially hidden.



(b) The figure illustrates the stage of the crowdsourcing process where the explanations are reviewed after the first question has been answered (figure a). Hovering over the rating stars reveals the meaning of the star such as "Very bad" for 1 star and "Bad" for 2 stars. Upon choosing a star rating, the form displays its corresponding description, such as "Very bad" for 1 star, "Bad" for 2 stars, "Good" for 3 stars, "Very good" for 4 stars, and "Extremely good" for 5 stars.

#	Input	Reason	Target
1	<p>P: Two women are embracing while holding to go packages.</p> <p>H: The sisters are hugging goodbye while holding to go packages after just eating lunch.</p>	<p>Model: Two women are not necessarily sisters.</p> <p>Human: The to go packages may not be from lunch.</p>	N (o)
2	<p>P: Two women are embracing while holding to go packages.</p> <p>H: Two woman are holding packages.</p>	<p>Model: Two women are holding packages is the same as two women are holding packages.</p> <p>Human: Saying the two women are holding packages is a way to paraphrase that the packages they are holding are to go packages.</p>	E (o)
3	<p>P: Two women are embracing while holding to go packages.</p> <p>H: The men are fighting outside a deli.</p>	<p>Model: Women are not men.</p> <p>Human: In the first sentence there is an action of affection between women while on the second sentence there is a fight between men.</p>	C (o)
4	<p>P: Two young children in blue jerseys, one with the number 9 and one with the number 2 are standing on wooden steps in a bathroom and washing their hands in a sink.</p> <p>H: Two kids at a ballgame wash their hands.</p>	<p>Model: Bathroom and ballgame are different.</p> <p>Human: Two kids in jerseys watching their hands are not necessarily at a ballgame.</p>	N (x)
5	<p>P: A woman is doing a cartwheel while wearing a bikini in the sand next to the beach.</p> <p>H: A woman is doing a cartwheel and falls on her head.</p>	<p>Model: Just because a woman is doing a cartwheel doesn't mean she falls on her head.</p> <p>Human: A woman doing a cartwheel as opposed to a woman falling while doing a cartwheel.</p>	C (x)
6	<p>P: At an outdoor event in an Asian-themed area, a crowd congregates as one person in a yellow Chinese dragon costume confronts the camera.</p> <p>H: A single man is next to a camera</p>	<p>Model: A single man is a person.</p> <p>Human: The crowd can't be single man.</p>	C (x)

Table 9: Examples explanations generated by our models on the development set of e-SNLI, which consists of three labels: entailment (E), contradiction (C) and neutral (N). We show the first three correct and incorrect instances.

JobXMLC: EXtreme Multi-Label Classification of Job Skills with Graph Neural Networks

Nidhi Goyal
IIIT-Delhi

Jushaan Singh Kalra
DTU, Delhi

Charu Sharma
IIIT-Hyderabad

Raghava Mutharaju
IIIT-Delhi

Niharika Sachdeva
InfoEdge India Limited

Ponnurangam Kumaraguru
IIIT-Hyderabad

Abstract

Writing a good job description is an important step in the online recruitment process to hire the best candidates. Most recruiters forget to include some relevant skills in the job description. These missing skills affect the performance of recruitment tasks such as job suggestions, job search, candidate recommendations, etc. Existing approaches are limited to contextual modelling, do not exploit inter-relational structures like job-job and job-skill relationships, and are not scalable. In this paper, we exploit these structural relationships using a graph-based approach. We propose a novel skill prediction framework called JobXMLC, which uses graph neural networks with skill attention to predict missing skills using job descriptions. JobXMLC enables joint learning over a job-skill graph consisting of 22.8K entities (jobs and skills) and 650K relationships. We experiment with real-world recruitment datasets to evaluate our proposed approach. We train JobXMLC on 20,298 jobs and 2,548 skills within 30 minutes on a single GPU machine. JobXMLC outperforms the state-of-the-art approaches by 6% on precision and 3% on recall. JobXMLC is 18X faster for training tasks and up to 634X faster in skill prediction on benchmark datasets enabling JobXMLC to scale up on larger datasets. We have made our code and dataset public at <https://precog.iiit.ac.in/resources.html>.

1 Introduction

Online recruitment platforms such as LinkedIn and Glassdoor are extensively used to post jobs, find relevant candidates, and match resumes to the jobs posted. Recruiters create job positions mentioning skills, roles, and responsibilities to reach potential candidates. Among all these required fields, skills are crucial parameters to determine whether or not a candidate is suitable for a job position (Mehta et al., 2013). Recruiters often miss adding relevant and crucial skills required for the job due to

a communication gap between the domain experts and recruiters. According to statistics, 65% of the job descriptions (JDs) do not include relevant and popular skills, and 40% of JDs miss listing 20% or more explicitly-stated skills in the prose description (Bhola et al., 2020). It reduces the number of relevant applications for the job posting and affects the performance of major recruitment tasks such as job-to-resume matching. Therefore, it is imperative to recommend such missing skills to improve the quality of job postings. Figure 1 shows the sample (fictitious) job posted over the recruitment platform where some skills are missing from the textual JD. Prior works (Bhola et al., 2020; VERMEER et al., 2020) explored missing skill recommendation task using large-scale pre-trained language models. Document embedding and Graph-based systems (Gugnani and Misra, 2020; Kivimäki et al., 2013) are used for skill extraction and recommendations. However, these approaches have a few shortcomings- (a) they do not exploit the structural relationships between jobs and skills across the whole dataset. For example, assume jobs j_1 and j_2 share a common skill s_1 . If there is another skill s_2 relevant to j_2 and other similar jobs, we can infer that s_2 might also be relevant to j_1 . Such transitive cues can be extremely useful for identifying missing skills. Current deep extreme classifiers (You et al., 2019; Prabhu and Varma, 2014) find it hard to model such implicit relationships unless the training set explicitly contains a pair (j_1, s_2) , (b) they give equal importance to every skill corresponding to the job. However, each skill in the skill label set has different weights based on the frequency of their occurrence in job descriptions, (c) language models bring high computational costs at massive scales as a task not only involves predicting multiple missing skills but also requires to precisely organize the most relevant skills specific to the job posting. Graphs are naturally suitable to make the relationships explicit such as job-skill networks.

Job Title	Market Analyst					
Job description	Assist the Manager in sourcing the food industry and in conducting product research and analysis. Facilitate effective communication between the analytics and user experience teams. Evaluates customers' online behaviour and provide insights and recommendations for further enhancements to the guest experience. Strong research, data analysis and communication skills.					
Required skills	communication	data analysis	tableau	visualization	python	Excel
	Explicit Skills			Implicit Skills		

Figure 1: An example of a fictitious job posted over a recruitment platform. The job description does not include implicit and job-specific skills such as ‘*tableau*’, ‘*visualization*’, ‘*python*’, and ‘*Excel*’.

Two nodes (jobs) are likely to have common neighbors (skills) if the jobs have overlapping skills. To model these structural relationships between nodes (jobs and skills), a Graph neural network (GNN) is a well-known architecture for representing the knowledge and additional information (Wu et al., 2020). Missing skills applications also face extreme skill label sparsity; using label co-occurrence alone without graphs yields fractured correlations. To this end, we propose a framework called JobXMLC, which uses a GNN with label attention to jointly model the jobs and skills in the same space. Through the use of ubiquitous job description data, we aim to predict the missing skills using collaborative learning of jobs and skills. Therefore, we model our problem as an Extreme Multi-label Classification (XMLC) task as there is no existing dataset with manually-annotated labels from textual JDs, making it sub-optimal to train a sequence labeling task. The contributions of this work are summarized as follows:

- We construct a novel job-skill graph consisting of 22, 844 (jobs and skills) and 650K relationships that allow flexible integration of textual features and various pre-trained language representation models.
- We cast our problem as an XMLC using job-skill graph and propose JobXMLC comprising of graph neural networks with skill attention to learn multi-resolution graph neighborhoods with the sampling method.
- We also provide the performance comparison of JobXMLC, which outperforms by a margin of 6% from the best baselines.
- JobXMLC is lightweight, up to 18X faster in training and 634X in predicting than existing deep learning-based extreme classifiers to

scale up to thousands of labels.

2 Background and Related Work

Recently, several works (Bhola et al., 2020; Jiechieu and Tsopze, 2021) have been done for skill prediction using Extreme Multi-label Classification. XMLC refers to the classification of text where the number of the set of labels is large, i.e., thousands or millions. One-vs-All (OVA) is a well-known method for text classification tasks with high accuracy (Khandagale et al., 2020). The OVA approach is computationally efficient for the XMLC task for modest-sized label sets (up to a few thousand labels).

These methods are broadly classified as (a) Deep learning-based models, (b) Graph-based, and (c) Domain-specific methods.

Deep learning-based methods. Deep learning models that use powerful text representation capabilities have also been explored for the XMLC problems (You et al., 2019; Chang et al., 2019). XML-CNN (Liu et al., 2017) applies a dynamic max-pooling scheme and a family of CNN models to learn text representations. AttentionXML (You et al., 2019) uses the attentional bidirectional long short-term memory (BiLSTM) networks to extract embeddings from raw text inputs. However, the CNN-based models cannot capture the most relevant parts of the information on each label. The RNN-based methods fail to model long-term dependencies due to vanishing gradients. Research (Bhola et al., 2020) explores the language models such as ELMo, Transformer and BERT (Devlin et al., 2019), and X-BERT (Chang et al., 2019) for XMLC task. These approaches model input language’s syntactic and semantic structure to predict tokens based on the available contextual information. However, such models are computationally expensive and require a predefined meaning of la-

bels. In addition, the difficulty of scaling to the extreme label space remains in deep learning methods as the output layer scales linearly with the product of label size and feature dimension. The research gaps with deep-extreme classifiers motivate us to explore alternative approaches and techniques that do not require explicit label representation or predefined semantic meaning of labels and are scalable for extensive datasets.

Graph-based approaches. The recent proliferation of graph neural networks (Wu et al., 2020) allows using node neighborhoods to learn more discriminative features collaboratively. GraphSAGE (Hamilton et al., 2017) proposed the computation of node representations inductively by recursively aggregating over fixed-sized neighborhoods. Authors (Xu et al., 2018a) proposed the Graph Isomorphism Network (GIN) with discriminative power equal to that of the WL test. GraphSAINT (Zeng et al., 2020), a graph sampling-based inductive learning method, compute node representations based on the local graph structure and node attributes. However, job-skill graph-based collaborative learning at extreme scales is underexplored for missing skill prediction task.

Domain-specific methods. Research work identifies the skill bases used for analyzing the job market, the type of extracted skills (Khaouja et al., 2021), the skill identification methods, the studied sector and their granularity. Literature (Xu et al., 2018b) build a job-skill network to measure the popularity of the skills by exploring a large corpus of job postings. Research (Bhola et al., 2020) employs an Extreme Multi-label Classification method that utilizes the Transformer model to predict the required skills from a textual job descriptions. However, these approaches are computationally expensive and either predict frequent skills or miss rare crucial skills for recruiters. To address all the existing challenges and limitations, we propose JobXMLC that uses graph neural networks with skill attention to learn multi-hop job-skill network. To the best of our knowledge, this work is the first to exploit GNNs for the job-skill prediction task.

3 Problem Formulation

Consider the set of jobs $\mathcal{J} = \{j_1, j_2, \dots, j_i\}$. A job $j_i \in \mathcal{J}$ corresponds to its textual description and \mathcal{S}_i is the set of skill labels for the i^{th} job. The skill set is represented as, $\mathcal{S}_i = \{s_i^1, s_i^2, s_i^3, \dots, s_i^k\} \forall 1 \leq k \leq n$, where n

refers to total skills that vary differently for each dataset. The task of JobXMLC is to learn a function $f : \mathcal{J} \rightarrow 2^{\mathbb{S}}$ that maps a job $j_i \in \mathcal{J}$ to its target skill set $\mathcal{S}_i \in \mathbb{S}$, where $\mathbb{S} = \{\mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots \cup \mathcal{S}_{|\mathcal{J}|}\}$.

4 JobXMLC: EXtreme Multi-label Classification of Job Skills

In this section, we introduce JobXMLC as shown in Figure 2. The architecture is inspired by the models proposed in (Saini et al., 2021).

The architecture comprises of three major components: 1) Job-skill graph 2) Graph Neural Network that learns multi-hop embeddings with neighbourhood selection approach on the job-skill graph 3) a scalable mechanism of extreme classifiers to predict skill labels in cold and warm-start scenarios.

4.1 Module I: Job-skill graph

We first formally define a job-skill graph, which is usually represented as a tuple $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where \mathbb{V} and \mathbb{E} are the set of nodes and edges respectively. Here \mathbb{V} consists of jobs belonging to \mathcal{J} and skill set \mathbb{S} (See Section 3). We construct an edge $e \in \mathbb{E}$ between j_i and s_i^k where $\mathbb{E} \subset \mathcal{J} \times \mathbb{S}$ iff s_i^k is relevant to j_i i.e., s_i^k is a positive label for j_i . Each node $v \in \mathbb{V}$, is initialized as a d -dimensional vector based on its textual features. We obtain initial embeddings by fine-tuning the fastText skip-gram model (Bojanowski et al., 2017) in an unsupervised manner. fastText is a lightweight embedding model that is well-suited when the document misses predicate-argument structure dependencies (Arora et al., 2020). We also leverage word-level information, including POS tagging (Kumawat and Jain, 2015) and word importance (TF-IDF), to parse the long document according to relevancy and structure. Since all the tokens present in job descriptions are not informative, we apply POS tagging to filter out verbs, adjectives, and adverbs, which are not indicators of skills in the job description. The underlying assumption is that skill labels would be mostly nouns such as ‘java’, ‘python’, etc. We found out that there are 60% nouns present in job descriptions. Further, we use an averaging technique to get the representation for every job. For each node v , its initial representation is \hat{f}_v^0 (with $\hat{f}_v^0 \equiv \hat{f}_j^0$ if the node v is the job $j \in \mathcal{J}$ and $\hat{f}_v^0 \equiv \hat{f}_s^0$ if node v is the skill label $s \in \mathbb{S}$).

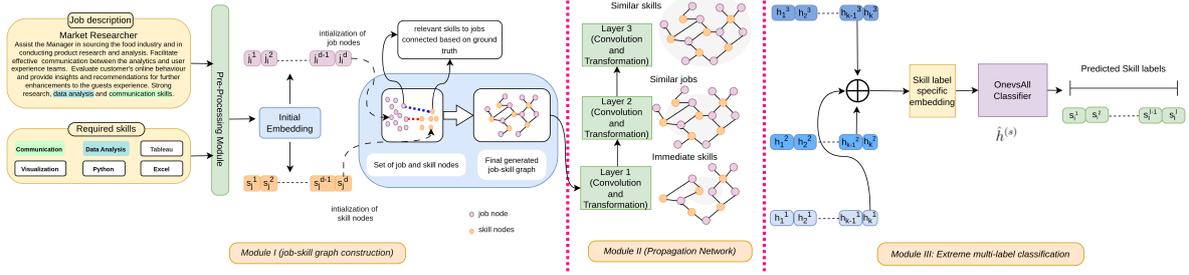


Figure 2: **JobXMLC** consists of three components: Module I consists of a mechanism to construct a job-skill graph, and Module II consists of a graph neural network-based architecture that learns embeddings using multi-hop neighborhoods using a job-skill graph effectively. Module III uses a scalable mechanism of extreme classifiers to predict missing skills.

4.2 Module II: Multi-hop job-skill graph neural network

To learn a job-skill graph, the module introduces the propagation network and neighborhood selection approach.

Propagation Network. This network captures higher-order job-skill graph structure using multiple layers of aggregation, each layer aggregating information from the previous layer’s node representations. Therefore, we utilize Graph Isomorphism Network (GIN) (Xu et al., 2018a) encoder for representations considering its outstanding expressive capacity and model simplicity. It consists of a convolution to aggregate information from a node’s neighbors and a transformation operation to update the node representation based on the convolved embeddings ($f_v^{(k)}$). To avoid over-smoothing, we utilize the skip connection operation that gathers information from historical representations of nodes. We also learn multi-hop representations with k -hop (fanouts) neighbors of each node, where k is a hyperparameter. For instance, if $k=1$, the encoder would only consider the immediate neighbors (skills) of each node (job) in the graph. If $k=2$, it would consider the neighbors (jobs) of the neighbors (skills) and so on. Equation 1 shows the graph neural network layer that updates the node representation using a weighted sum of neighboring node features:

$$f_v^{(k)} = (1 + \lambda_k) f_v^{(k-1)} + \sum_{j \in \mathcal{N}_v, j \neq v} f_j^{(k-1)} \quad (1)$$

where \mathcal{N}_v be the set of neighboring nodes of an i^{th} node; $f_v^{(k)}$ be the representation of the v^{th} node after layer k , and λ is a fixed scalar for layer k .

Equation 2 shows the final embeddings after transformation:

$$h_v^{(k)} = f_v^{(k)} + g(\delta(R_k * g(f_v^{(k)}))) \quad (2)$$

where $g(\cdot)$ is ReLU activation, $\delta(\cdot)$ is batch normalization and R_k is a parameter matrix for the residual layer.

Neighborhood selection. Instead of considering all k -hop neighbors of each node, we sampled a subset of the neighbors at each layer of the network. The goal of selection is to reduce the computational cost of the network and ensure that our model is scalable in dense settings. Therefore, we select the top l neighbors based on their frequency. Formally, for every node $v \in |\mathbb{V}|$, we accomplish frequency-based sorting where a set of fanouts $[k]$ neighbors are sampled for every node to construct $\mathbb{V}(n)$.

4.3 Module III: Extreme multi-label classification

In this module, we discuss skill attention and prediction pipeline for Extreme multi-label classification task.

Skill attention. We incorporate label-wise attention for every skill $s_i \in [\mathcal{S}]$ and layer $k \in [\mathcal{K}]$ in the propagation network. We obtain attention weights α_k using a softmax operation. $\alpha_{sk} = \exp(e_{sk}) / \sum_{k' \in [K]} \exp(e_{sk'})$. Given multi-resolution embeddings \hat{h}_v^k , $k \in [\mathcal{K}]$ for a job description, when calculating the score for a label $s \in [\mathcal{S}]$, first a label-specific embedding is calculated as given in Equation 3 and then One-vs-all classifier depicted as $C = [c_1, c_2, \dots, c_{\mathcal{S}}] \in \mathbb{R}^{\mathcal{J} \times \mathcal{S}}$ is used to obtain a score for the skill label as $score_s = \langle c_s, \hat{h}^{(s)} \rangle$.

$$\hat{h}^{(s)} = \sum_{k \in \mathcal{K}} \alpha_{sk} \cdot \hat{h}^{(s)} \quad (3)$$

Element	mycareersfuture.sg	StackOverflow Jobs
No. of job posts	20, 298	20, 320
# of distinct skills	2, 548	275
# of skills with 20 or more mentions	1, 209	50
Average skill tags per job post	19.98	2.8
Average token count per job post	162.27	200.8
Maximum token count in a job post	1, 127	800

Table 1: Dataset statistics for mycareersfuture.sg and StackOverflow Jobs.

Prediction pipeline. Initial representations are used to construct an Approximate Nearest Neighbors graph (ANNS). Suppose a test job appears at runtime, First, the relationships in the graph are introduced to its *prediction-introduce-edges* (See Table 8) nearest neighbors. These neighbors can be a part of \mathcal{J} or \mathcal{S} . New job nodes are first introduced into the graph for the standard cold start setting, where jobs are not part of the job-skill graph used for training. Relationships are introduced to the partially revealed skill labels in warm start settings. This setting is possible when the recruiter enters some skills before writing the job description. Then, JobXMLC is used to obtain multi-resolution embeddings $\hat{h}^k \in [\mathcal{K}]$ for the test job.

Shortlisting. Since evaluating skill scores for all skill labels would take $\Omega(\mathcal{J}\mathcal{S})$ time. To predict in milliseconds, the prediction time complexity should not be worse than $\Theta(\mathcal{J}\log(\mathcal{S}))$. Therefore, we utilize a shortlisted where a set of $\mathcal{O}(\log\mathcal{S})$ skill labels are shortlisted that seem most relevant to it. For a test job description, the label-wise embeddings $\hat{h}^{(s)}$ are created with respect to skill labels $s \in \mathcal{S}$. To create the shortlister, multi-resolution representations of skill labels are averaged and a second ANNS graph is created over these averaged embeddings. We rank the top *num_shortlist* (See Table 8) neighbors, based on their cosine similarity, for shortlisting to form the set \mathcal{S} of potential labels for which label-wise embeddings are calculated.

5 Experimental Setup

5.1 Datasets

We utilize two real-world recruitment datasets, namely mycareersfuture.sg (Bhola et al., 2020) and StackOverflow Jobs¹ collected from popular recruitment platforms. These datasets consist of

¹<https://stackoverflow.com/>

over 20,000 richly-structured job posts with 23 informative fields about the advertisement details and current status. Table 1 reports the statistics for recruitment domain datasets. Small-scale datasets vary from 100 to 300, whereas large-scale ranges from 300 to millions of labels. Similar scales for the XMLC task are demonstrated in (Jain et al., 2019; Liu et al., 2017).

Data Pre-processing. We filtered out *job descriptions*, *job title*, *required skills* corresponding to every job posting. From mycareersfuture.sg dataset, we consider concatenation of ‘roles & responsibilities’ and ‘job requirements’ fields as the ‘job description’, and ‘required skills’ as the set of target discrete labels. Similarly, for StackOverflow Jobs dataset, we consider the ‘job description’ and ‘required skills’ sections. We filtered out the jobs with either empty or single words in the textual content. We also perform lower-casing, stopwords removal, and removal of less important strings such as ‘available’, ‘requirements’, which are present in most JDs. StackOverflow Jobs dataset consists of 6M words with 298,729 unique words. We split the dataset into training, validation and testing datasets with an 80:10:10 proportion. Similar splits has been utilized by competitive methods (Bhola et al., 2020).

5.2 Implementation and Competing Methods

This section will discuss the training details and baselines.

Training details: We utilize binary cross-entropy loss and Adam optimizer. We use the drop out layer after every ReLU layer. We conducted our experiments using the list of hyperparameters reported in Table 8 and Table 9 (See appendix B) for details.

Baselines: We show the effectiveness of different aspects of JobXMLC and evaluate our model performance against competitive transformer-based baselines. These constitute CNN (Kim, 2014), LSTM (Rocktäschel et al., 2015), BiLSTM (Sun

Model	R@5	R@10	R@30	P@5	P@10	P@30
CNN	14.17	23.58	45.34	56.67	47.17	30.23
LSTM†	11.67	18.44	35.02	46.67	36.89	23.34
Bi-LSTM†	13.02	21.37	41.54	52.07	42.75	27.70
Bi-GRU†	13.98	23.43	44.41	55.94	46.87	29.61
BERT+XMLC	15.27	25.96	51.18	61.06	51.92	39.32
RoBERTa+XMLC	16.15	26.52	51.99	60.08	53.85	39.87
BERT+XMLC+CAB	16.72	29.45	58.98	66.87	58.90	41.21
GalaXC	16.31	28.34	54.16	65.25	56.7	36.11
JobXMLC (GraphSaint)	16.23	27.79	53.32	64.93	55.59	35.55
JobXMLC (GraphSAGE)	16.84	29.18	56.89	67.36	58.36	37.93
JobXMLC	18.29	32.33	63.18	73.20	64.66	42.22

Table 2: Results of JobXMLC along with state-of-the-art approaches on mycareersfuture.sg dataset. For RNN-based models (†), we have limited all model architectures to two layers.

et al., 2017), BiGRU (Halder et al., 2018), BERT-XMLC (Bhola et al., 2020), RoBERT-XMLC (VERMEER et al., 2020), GalaXC (Saini et al., 2021), JobXMLC (GraphSaint) (Hamilton et al., 2017), and JobXMLC (GraphSAGE) (Hamilton et al., 2017). We discuss transformer-based approaches, BERT-XMLC (Bhola et al., 2020) encodes the words of the job descriptions using a pre-trained BERT model. The encoding of the [CLS] token is then used as representation of the job description. The job representation is passed to a bottleneck layer (i.e., an added linear layer before the output layer). The last layer treats every skill as a binary classification problem, so for each skill it calculates the probability that the skill is associated with JD.

State-of-the-art models such as CNN, LSTM, Bi-GRU, and Bi-LSTM are self-explanatory. We utilize two neural network layers for all RNN-based models. GalaXC (Saini et al., 2021) describes a novel framework for extreme classification using graph neural networks (GNNs). GraphSAGE (Hamilton et al., 2017) and GraphSaint (Hamilton et al., 2017) encodes the node information and useful for graphs that have rich node attribute information for extreme multi-label classification.

5.3 Evaluation metrics

We utilize Precision@ k (P@ k), Recall@ k (R@ k), Normalized Discounted NDCG@ k (N@ k), Mean Reciprocal Rank (MRR), EIM, REIM, RIIM (Bhola et al., 2020) as evaluation metrics for the skill prediction task.

Precision@ k : includes the proportion of skills in

the top- k skill prediction list that are relevant.

Recall@ k : includes the proportion of relevant skills found in the top- k skill prediction list.

NDCG@ k : discounts the true positives that occur later in the prediction rankings.

MRR: indicates the position (reciprocal) of the first true positive in the predicted set of skills.

EIM (Explicit Inference Measure): the micro, instance-based measure of explicit skills predicted by the model, compared against gold-standard explicit skills mentioned, for instance.

RIIM (Relative Implicit Inference Measure): macro, the recall-based measure of implicit skills predicted by the model, relative to the entire set of implicit skills.

REIM (Relative Explicit Inference Measure): macro, recall-based measure of explicit skills predicted by the model compared to the entire set of explicit skills.

6 Results and Analysis

Table 2 and Table 3 reports Recall@ k and Precision@ k for all state-of-the-art approaches and JobXMLC on both datasets. Compared to leading deep extreme classifiers, BERT-XMLC and RoBERTa-XMLC, JobXMLC is up to 18X faster to train on a single GPU. Compared to other baselines, JobXMLC is at least 3% better than Bi-LSTM (Sun et al., 2017) in R@5, which helps demonstrate the efficacy of modelling the sequence by JobXMLC. Further, fastText initialization in JobXMLC is 7-8% better than BERT-XMLC+CAB (Bhola et al., 2020) in R@5, indicating that the global relationships improve the model

Model	R@5	R@10	R@30	P@5	P@10	P@30
CNN	25.16	39.39	64.80	15.24	11.72	6.36
LSTM†	26.63	40.47	67.89	16.07	11.95	6.65
Bi-LSTM†	41.46	55.27	76.38	23.83	16.12	7.56
Bi-GRU†	46.15	59.01	78.61	26.68	17.23	7.79
BERT +XMLC	35.50	50.95	76.06	20.75	14.99	7.58
RoBERTa +XMLC	36.20	52.23	77.05	21.98	15.09	7.88
BERT +XMLC+CAB	37.20	51.24	78.98	22.18	15.02	8.03
GalaXC	43.27	51.47	67.50	24.23	14.53	6.50
JobXMLC (GraphSaint)	39.16	51.73	73.99	22.28	14.88	7.22
JobXMLC (GraphSAGE)	38.76	52.26	74.19	21.98	14.99	7.23
JobXMLC	47.85	59.26	74.53	26.92	16.94	7.23

Table 3: Results of JobXMLC along with state-of-the-art approaches on StackOverflow Jobs dataset. For RNN-based models (†), we have limited all model architectures to two layers.

better in addition to local connections through joint learning. Compared to BERT-XMLC (Bhola et al., 2020) and RoBERTa (VERMEER et al., 2020), both utilize transformer-based embeddings and skill correlation-based features for training, JobXMLC is 4% better in recall and precision metrics. JobXMLC outperforms Graph-based methods such as GalaXC (Saini et al., 2021) across all metrics. Table 4 reports NDCG and MRR val-

Model	N@5	N@10	N@30	N@50	N@100	MRR
CNN	28.21	40.23	60.60	66.37	71.96	0.77
LSTM	29.27	40.66	59.43	69.61	71.53	0.70
Bi-LSTM	30.32	48.07	44.55	50.30	57.04	0.76
Bi-GRU	30.83	50.52	46.45	52.37	59.15	0.76
BERT-XMLC	28.05	38.81	57.62	64.68	71.28	0.83
BERT-XMLC+CAB	29.13	40.74	60.60	67.51	73.74	0.85
GalaXC	32.86	44.51	63.73	70.11	74.77	0.82
JobXMLC	37.91	49.63	67.83	73.81	78.94	0.90

Table 4: Normalized Discounted Cumulative Gain (NDCG) is represented by N and Mean Reciprocal Rank (MRR) comparison of JobXMLC along with State-of-the-art approaches on mycareersfuture.sg dataset.

ues for mycareersfuture.sg dataset. We observe that JobXMLC outperforms all state-of-the-art approaches by significant margin of 8% from deep extreme classifiers.

Inference time: Table 5 presents the results of JobXMLC and leading deep extreme classifiers like BERT, ROBERTa which shows that JobXMLC is 18X faster than BERT+XMLC+CAB for mycareersfuture.sg dataset.

Analysis on Implicit and Explicit skills: Table 6 shows the Explicit and Implicit Metrics for the skill prediction task. We are interested in the relevance

and implicitness of the retrieved implicit skills and false positives. We find the explicit and implicit skills underline the noisy nature of the skill labels. For example, ‘*machine learning*’ and ‘*python*’ are clear required (explicit) skills and ‘*communication*’ comes across as a vast skill. In terms of false positives, we note that the job description explicitly mentions ‘*good knowledge of python*’ as a required skill (for Data Scientist job). Most relevant skills are not very distinctive to the job role, causing the model to mispredict the skill.

7 Ablation Study

Initial embeddings: JobXMLC shortlisting criteria offered much better recall if we use the initial fastText embeddings to create shortlists. We observe that fastText worked best for our recruitment domain dataset in comparison to other recent pre-trained language representation models (See Appendix B) including BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), and Paraphrase-mini-LM-L6 (Reimers and Gurevych, 2019). For example, on the mycareersfuture.sg dataset, the recall for the top 100 labels shortlisted using the initial fastText and BERT embeddings are around 85.20% and 56.90%, respectively. Based on an average of 20.61 skills per job, about 4 skills were derived within the top 5 and 19 within the top 100 of derived skills. **Warm and Cold Start Scenarios:** Table 7 reports the results in warm-start and cold-start settings separately. JobXMLC is initialized with fine-tuned fastText embeddings which achieve P@ k , R@ k , and MRR of 72.86, 18.26, and 0.89 respectively in cold-start scenario. JobXMLC is initialized with

Datasets →	mycareersfuture.sg		StackOverflow Jobs	
Models ↓	TT	PT	TT	PT
BERT+XMLC	5.50	1200	1.63	350
RoBERTa+ XMLC	4.72	1200	1.24	350
BERT+XMLC+CAB	9.20	1200	4.86	350
JobXMLC	0.51	1.89	0.31	1.71

Table 5: Comparison of JobXMLC with stronger baselines. JobXMLC is faster to train than leading Deep Extreme Classifiers like BERT at training and prediction time. Here TT= Train Time (in hours), PT= Prediction Time (in ms).

Metrics	EIM	RIIM	REIM
BERT+XMLC +CAB	115.89	64.60	25.73
JobXMLC	121.09	66.36	33.04

Table 6: Comparison of EIM, RIIM, and REIM metrics on JobXMLC on mycareersfuture.sg dataset.

fine-tuned fastText embeddings which achieve $P@k$, $R@k$, and MRR of 75.76, 22.09, and 0.91 respectively in warm-start scenario. In both scenarios, the value of $k=5$. These results show that partially reveal achieved comparable precision and relatively higher recall than cold-start settings.

Model	P@5	R@5	MRR
JobXMLC (cold-start)	72.86	18.26	0.89
JobXMLC (warm-start)	75.76	22.09	0.91

Table 7: Effectiveness of JobXMLC in warm-start and cold-start scenarios on mycareersfuture.sg dataset.

Qualitative Analysis: We compared the JobXMLC and analyzed the skills predicted correctly and incorrectly as shown in Figure 3. JobXMLC captures the structural relationships between jobs and skills effectively. JobXMLC predicts ‘Java’, ‘Software Development’, ‘XML’, ‘JavaScript’, ‘jQuery’, etc. as required skills whereas BERT-XMLC with CAB predicts ‘Java’, ‘C++’, ‘Linux’, ‘Python’ as skills where more relevant skills such as ‘JavaScript’ and ‘Web Applications’ are missed.

8 Discussion

We compare with existing graph-based methods such as GalaXC (Saini et al., 2021), which are more well-suited to handle short text inputs for product queries. JobXMLC leverages word-level

components, including syntactic roles (POS tags) such as nouns and verbs present in each job and word importance (TF-IDF), which explains the long document from the perspective of text relevancy and structure. We believe that JobXMLC is generalizable across many other applications. Our raw dataset is relatively preprocessed, simple and misses predicate-argument structure dependencies. Therefore, we hypothesize that non-contextual embeddings such as fastText (having 98.69% of words from our dataset present in vocabulary) outperformed BERT as it understands word-level information. Similar observations are made by (Arora et al., 2020) for classic embeddings with competitive (or even slightly better) performance than contextual embeddings.

9 Conclusion

In this work, we propose a JobXMLC framework, which uses a graph neural network to incorporate neighborhood information with the help of a collaborative graph over jobs and skills. JobXMLC leverages skill attention mechanism for more effective extreme classifiers and attends to multi-resolution representations of jobs and skills. JobXMLC outperforms leading deep extreme classifiers on precision and recall metrics by 6% and 3%. JobXMLC also operate in warm and cold-start scenarios effectively. JobXMLC is 18X faster on training and 634X faster on predicting than deep extreme classifiers and can be scaled efficiently to real-world datasets with thousands of labels. We believe that JobXMLC can be deployed on large-scale recruitment platforms for predicting missing skills using job descriptions.

10 Limitations

We perform experiments on jobs sampled from a popular Singaporean government job portal and StackOverflow, which is limited to the English lan-

Job description	minimum 5 7 years experience information technology software development must 3 4 yeras experience dot net development experience asp.net c, .net xml experience, language query update etc knowledge pc networking require dot net developer mnc client singapore typre position long term contract initial degree information technology require minimum 5 7 years experience information technology software development must 3 4 years experience dot net development experience asp.net c net xml etcknowledge pc networking good communication skills										
Required skills (Ground truth)	Software development	java	.NET	Javascript	jQuery	XML	Web applications	ASP.NET	SDLC		
BERT-XMLC+CAB	Software development	java	.NET	jQuery	XML	PHP	Python	C	Linux	Software engineering	
JOBXMLC	Software development	java	.NET	Javascript	jQuery	XML	Web applications	ASP.NET	SDLC	integration	

Figure 3: Shows the skills predicted by BERT–XMLC+CAB and JobXMLC where input is job description. **Purple** shows correct skill predictions by JobXMLC as compared with required skills (ground truth). **Green** shows the extra skills predicted by JobXMLC. **Red** skills are missed by BERT+XMLC+CAB model as compared with ground truth.

gauge. Our approach can handle missing skills which are part of our skill vocabulary, but it cannot infer new emerging skills from job descriptions, i.e., out-of-vocabulary. We will consider domain knowledge and the popularity of job skills to generalize our approach for job-candidate mapping applications for future work. We wish to expand our work to other recruitment domain applications with resumes and candidate profiles.

11 Ethical Considerations

The paper investigates the missing skills problem with the help of a graph-based framework by incorporating word-based embeddings that can be insightful for other researchers in academia and industry. Any biases present in the dataset or embedding model can creep into the proposed approach.

Acknowledgements

The authors acknowledge the support of the PreCog Research Group and the Machine Learning Lab at IIT-H, the Infosys Center for Artificial Intelligence (CAI) and the KRACR Lab at IIT-Delhi. We also thank anonymous reviewers and the area chairs for their detailed and helpful feedback. Special thanks to Prashant, Saurabh, Anmol Goel, Shivangi, Amrit, Dr. Kajal Kansal, and Dr. Siddharth Asthana for critically reviewing the manuscript and stimulating discussions.

References

Simran Arora, Avner May, Jian Zhang, and Christopher Ré. 2020. [Contextual embeddings: When are they worth it?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Lin-*

guistics, pages 2650–2663, Online. Association for Computational Linguistics.

Akshay Bhola, Kishaloy Halder, Animesh Prasad, and Min-Yen Kan. 2020. Retrieving skills from job descriptions: A language model based extreme multi-label classification framework. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5832–5842.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit Dhillon. 2019. X-bert: extreme multi-label text classification with using bidirectional representations from transformers. *arXiv preprint arXiv:1905.02331*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Akshay Gugnani and Hemant Misra. 2020. Implicit skills extraction using document embedding and its use in job recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13286–13293.

Kishaloy Halder, Lahari Poddar, and Min-Yen Kan. 2018. Cold start thread recommendation as extreme multi-label classification. In *Companion Proceedings of the The Web Conference 2018*, pages 1911–1918.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. 2019. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 528–536.

- Kameni Florentin Flambeau Jiechieu and Norbert Tsope. 2021. Skills prediction based on multi-label resume classification using cnn with model predictions explanation. *Neural Computing and Applications*, 33:5069–5087.
- Sujay Khandagale, Han Xiao, and Rohit Babbar. 2020. Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning*, 109:2099–2119.
- Imane Khaouja, Ismail Kassou, and Mounir Ghogho. 2021. A survey on skill identification from online job ads. *IEEE Access*, 9:118134–118153.
- Yoon Kim. 2014. [Convolutional neural networks for sentence classification](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.
- Ilkka Kivimäki, Alexander Panchenko, Adrien Dessy, Dries Verdegem, Pascal Francq, Hugues Bersini, and Marco Saerens. 2013. A graph-based approach to skill extraction from text. In *Proceedings of TextGraphs-8 graph-based methods for natural language processing*, pages 79–87.
- Deepika Kumawat and Vinesh Jain. 2015. Pos tagging approaches: A comparison. *International Journal of Computer Applications*, 118(6).
- Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 115–124.
- Sameep Mehta, Rakesh Pimplikar, Amit Singh, Lav R Varshney, and Karthik Visweswariah. 2013. Efficient multifaceted screening of job applicants. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 661–671.
- Yashoteja Prabhu and Manik Varma. 2014. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 263–272.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.
- Deepak Saini, Arnav Kumar Jain, Kushal Dave, Jian Jiao, Amit Singh, Ruofei Zhang, and Manik Varma. 2021. Galaxc: Graph neural networks with labelwise attention for extreme classification. In *Proceedings of the Web Conference 2021*, pages 3733–3744.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Chengjie Sun, Yang Liu, Chang’e Jia, Bingquan Liu, and Lei Lin. 2017. Recognizing text entailment via bidirectional lstm model with inner-attention. In *International Conference on Intelligent Computing*, pages 448–457. Springer.
- NINANDE VERMEER, VERA PROVATOROVA, DAVID GRAUS, THILINA RAJAPAKSE, and SEPI-DEH MESBAH. 2020. Using robbert and extreme multi-label classification to extract implicit and explicit skills from dutch job descriptions. *acm*.
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.
- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018a. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.
- Tong Xu, Hengshu Zhu, Chen Zhu, Pan Li, and Hui Xiong. 2018b. Measuring the popularity of job skills in recruitment market: A multi-criteria approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Ronghui You, Zihan Zhang, Ziyi Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *Advances in Neural Information Processing Systems*, 32.
- Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2020. [GraphSAINT: Graph sampling based inductive learning method](#). In *International Conference on Learning Representations*.

A Hyper-parameter Details

This section reports the set of hyperparameters used for experiments conducted in the paper.

1. **No. of Epochs:** refers to number of epochs for JobXMLC.
2. **num_HN_epochs:** number of hard negative epochs for JobXMLC.
3. **learning rate (lr):** is the learning rate for JobXMLC.
4. **attention_lr:** is the learning rate used by skill attention.
5. **dlr_factor:** defines factor by which learning rate is decayed.
6. **batch_size:** refers to batch size used during training of JobXMLC.
7. **num_HN_shortlist:** refers to number of hard negative labels to be selected by sampling from other data points from the same batch.
8. **num_shortlist:** refers to number of skills sampled by shortlister.
9. **prediction_introduce_edges:** refers to total edges that should be introduced to graph at prediction time.
10. **fanouts:** refers to number of neighbors to sample for layer k .

Hyperparameter	Value
No. of epochs	20
num_HN_epochs	20
learning rate (lr)	0.0003
attention_lr	0.0003
dlr_factor	0.5
batch_size	256
fanouts	5, 5, 5
num_HN_shortlist	500
embedding_type	fastText
num_shortlist	1500
prediction_introduce_edges	3

Table 8: Hyper-parameters for mycareersfuture.sg dataset for JobXMLC. fastText refers to 300-dimensional embeddings obtained by fine-tuning fastText model on job descriptions.

B Evaluation Metrics for different initializations

This section reports the EIM, RIIM, REIM measures for Mini-LM (Reimers and Gurevych, 2019) model initialization. We observe that Mini-LM is

Hyper-parameter	Value
No. of epochs	30
num_HN_epochs	20
learning rate (lr)	0.0003
attention_lr	0.0003
dlr_factor	0.5
batch_size	256
fanouts	5, 5, 5
num_HN_shortlist	3
embedding_type	fastText
num_shortlist	275
prediction_introduce_edges	3

Table 9: Hyper-parameters for StackOverflow Jobs dataset for JobXMLC. As number of skill labels corresponding to job description are less in StackOverflow Jobs dataset, a lower fanout value gives better results.

Table 10: EIM, RIIM, REIM measures for JobXMLC and state-of-the-art approaches using Mini-LM, BERT and RoBERTa embedding initializations.

Metrics	EIM (Explicit inference measure)	RIIM (Relative implicit inference measure)	REIM (Relative explicit inference measure)
BERT+XMLC+CAB	115.89	64.60	25.73
JobXMLC(with Mini-LM)	86.45	27.28	17.76
JobXMLC(with BERT)	84.07	25.77	15.59
JobXMLC(with RoBERTa)	86.45	24.12	15.02

a transformer-based model which captures context well. However, JobXMLC is more benefitted with global view rather than just local job description text (context-based) embeddings.

ViLPAct: A Benchmark for Compositional Generalization on Multimodal Human Activities

Terry Yue Zhuo¹ and Yaqing Liao² and Yuecheng Lei²
Lizhen Qu^{1*} and Gerard de Melo³
Xiaojun Chang⁴ and Yazhou Ren² and Zenglin Xu^{5,6*}

¹Monash University ²University of Electronic Science and Technology of China
³HPI/University of Potsdam ⁴University of Technology Sydney
⁵Harbin Institute of Technology, Shenzhen ⁶Peng Cheng Lab

Abstract

We introduce ViLPAct, a novel vision-language benchmark for human activity planning. It is designed for a task where embodied AI agents can reason and forecast future actions of humans based on video clips about their initial activities and intents in text. The dataset consists of 2.9k videos from Charades extended with intents via crowdsourcing, a multi-choice question test set, and four strong baselines. One of the baselines implements a neurosymbolic approach based on a multi-modal knowledge base (MKB), while the other ones are deep generative models adapted from recent state-of-the-art (SOTA) methods. According to our extensive experiments, the key challenges are compositional generalization and effective use of information from both modalities¹.

1 Introduction

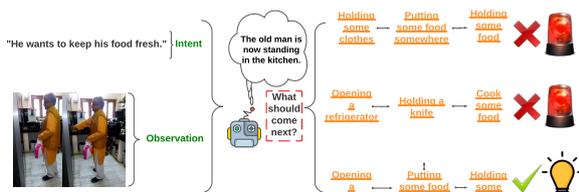


Figure 1: In daily life scenarios, an agent should be aware of future actions that will likely be taken by the user based on what it has observed. In this example, inputs of intent and observation are colored in green, while potential future action sequences are highlighted in orange. The first two sequences contain actions which do not align with the human intent. Thus, the agent needs to automatically detect which future actions are plausible by understanding the user’s intent.

One of the ultimate goals of Artificial Intelligence is to build intelligent agents capable of accurately understanding humans’ actions and intents,

*Corresponding authors: lizhen.qu@monash.edu, xuzenglin@hit.edu.cn

¹Our benchmark is available at <https://github.com/terryyz/ViLPAct>

so that they can better serve us (Kong and Fu, 2018). Newly emerging applications in robotics and multi-modal planning, such as Amazon Astro, have demonstrated a strong need to understand human behavior in multimodal environments. On the one hand, such an agent, e.g. an elderly care service bot, needs to understand human activities and anticipate human behaviors based on users’ intents. Here the intents may be estimated based on previous activities or articulated verbally by users. The anticipated behaviors may be used for risk assessment (e.g. falling of elderly people) and to facilitate collaboration with humans. On the other hand, recent advances in robotics show that it is possible to let robots learn new tasks directly from observed human behavior without robot demonstrations (Yu et al., 2018; Sharma et al., 2019). However, that line of work focuses on imitating observed human actions without anticipating future activities.

To promote research on action forecasting based on intents, we propose the *vision-language planning* task for human behaviors. As shown in Fig. 1, given an intent in textual form and a short video clip, an agent anticipates which actions a human is likely to take. We consider intents as given because there is already ample research on intent identification (Pandey and Aghav, 2020) and automatic speech recognition (Malik et al., 2021). To the best of our knowledge, there is no dataset to evaluate models for this task.

The task poses two major challenges. First, there are often multiple plausible action sequences satisfying an intent. Second, it is highly unlikely that a training dataset can cover all possible combinations of actions for a given intent. Hence, models need to acquire *compositional generalization* (Fodor and Pylyshyn, 1988), the capability to generalize to unseen action sequences composed of known actions.

In this work, we construct a dataset called ViLPAct for Vision-Language Planning of human Activities, which to the best of our knowl-

edge is the *first* dataset studying the above challenges. Specifically, we extend the Charades dataset (Sigurdsson et al., 2016) with intents via crowd-sourcing. As it is practically infeasible to find all possible future action sequences given an intent and a video clip of initial activities, we propose to evaluate all systems by letting each of them answer multi-choice comprehension questions (MQA) *without training them on those questions*. Given an intent and a video clip showing initial activities, each multi-choice question provides a fixed number of future action sequences as possible answers. A system is then asked to select the most plausible action sequence among them. We show that the rankings of all models using the MQAs correlate strongly with those obtained by asking human assessors to directly observe estimated action sequences. For training, we provide both a dataset for end-to-end training of sequence forecasting and a multimodal knowledge base (MKB) built from that dataset, which is also the *first* video-based multimodal knowledge base for human activities to the best of our knowledge.

We conduct the first empirical study to investigate compositional generalization for the target task. As baselines, we adapt three strong end-to-end deep generative models for this task and propose a neurosymbolic planning baseline using the MKB. The model is neurosymbolic because it combines both deep neural networks and symbolic reasoning (Garcez and Lamb, 2020). Given a video of initial activities and an intent, the deep models generate the top- k relevant action sequences, while the neurosymbolic planning model sends the intent and the action sequence recognized from the video as the query to the MKB, followed by retrieving the top- k relevant action sequences. Each model selects the most plausible answers by performing probabilistic reasoning over the relevant action sequences. We conduct extensive experiments and obtain the following key experimental results:

- We compare the evaluation results using MQA with the ones of human evaluation. The results of both methods are well aligned. Thus, MQA is reliable without requiring human effort.
- The likelihood functions of the deep generative models are not able to reliably infer which answers are plausible. In contrast, probabilistic reasoning is an effective method to improve compositional generalization.
- Despite information from both modalities be-

ing useful and complementary, all baselines heavily rely on intents in textual form but fail to effectively exploit visual information from video clips.

2 Related Work

Vision-Language Planning Task Vision Language Navigation (VLN) was among the first widely used goal-oriented vision-language tasks, requiring AI agents to navigate in an environment without interaction by reasoning on the given instruction (Anderson et al., 2018; Hermann et al., 2020; Misra et al., 2018; Jain et al., 2019). Recently, further goal-oriented vision-language tasks have been proposed. The Vision and Dialogue History Navigation (VDHN) task (De Vries et al., 2018; Nguyen and Daumé III, 2019; Thomason et al., 2020), which is similar to VLN, requires agents to reason on the instructions over multiple time steps. Other tasks such as Embodied Question Answering (EQA; Das et al. 2018; Wijmans et al. 2019), Embodied Object Referral (EOR; Qi et al. 2020b; Chen et al. 2019) and Embodied Goal-directed Manipulation (EGM; Shridhar et al. 2020; Kim et al. 2020; Suhr et al. 2019) rely on reasoning and interpreting the instruction with observation or object interaction in the environment. However, we argue that there are other ways to learn to plan without practising. Our task is one example of this, requiring agents to reason over the observation without performing actions.

Vision-Language Planning Datasets As existing vision-language planning datasets emphasize teaching embodied AI to perform the task like humans, they are constructed with interactive AI in mind. VLN (Anderson et al., 2018) datasets initially started exploring planning tasks with the textual instruction as a step-by-step abstract guide and minimal interaction with the environment. Extending the VLN task, VDHN (De Vries et al., 2018) datasets provide an interactive textual dialogue between the speaker and the receiver in multiple steps. The EQA (Das et al., 2018) task takes this a step further by providing data in an object-centric QA manner, advancing systems to understand the given environment through object retrieval. The EOR (Qi et al., 2020b) task designs object-centric datasets with detailed instructions, aiming at localizing the relevant objects accurately. The closest benchmark to ours is ALFRED (Shridhar et al., 2021) from the EGM task, which lets embodied agents decide

on actions and objects to be manipulated based on detailed instructions. However, in our setting, we ask intelligent systems to predict the most reasonable future action sequence based on human intents and answers in a Multiple Choice Question Answering (MQA) format. During prediction, we still give systems the flexibility to consider various combinations of actions and objects.

Vision-Language Planning Modeling According to Francis et al. (2021), several approaches have been used for planning. Greedy search in end-to-end models has been reported in several studies to work well in goal-oriented tasks (Fried et al., 2018; Das et al., 2018; Shridhar et al., 2020; Anderson et al., 2018). Task progress monitoring (Ma et al., 2019) is another method to tackle the planning. It allows models to backtrack on actions if the current action is found to be suboptimal. Mapping (Anderson et al., 2019) has as well been proposed for efficient planning via sensors. Topological and Exploration planning (Deng et al., 2020; Ke et al., 2019) enables modeling the planning in a symbolic manner. When goals are provided as several sub-goals, a divide and conquer strategy (Misra et al., 2018; Shridhar et al., 2020; Suhr et al., 2019) may be invoked to perform sub-task planning. In our work, we highlight another potential approach, knowledge base retrieval. As we construct an MKB containing various action sequences with detailed features, intelligent agents can retrieve the most suitable sequence from the MKB source in order to perform the planning.

3 Dataset Construction

We adopt videos from Charades (Sigurdsson et al., 2016) and solicit intents for videos via crowdsourcing. We consider videos that have action sequences of sufficient length appearing in both initial video clips and answers, which result in a dataset comprising 2,912 videos. The dataset is split into training/validation/test sets with a ratio of 70%, 10%, 20%. On the training dataset, we build an MKB by incorporating structural and conceptual information. On the test dataset, we collect a set of MQAs for model evaluation. The evaluation with MQAs is in fact an adversarial testing method, widely used for quality estimation in machine translation (Kanojia et al., 2021). Herein, the ability of a model to discriminate between correct outputs and meaning-changing perturbations is predictive of its *overall* performance, not just its robustness.

Thus MQAs are applied only for testing.

3.1 Data Normalization and Filtering

Charades is a large-scale video dataset of daily indoors activities collected via Amazon Mechanical Turk² (AMT). The average length of videos is approximately 30 seconds. It involves interactions with 46 object classes and contains 157 action classes, which are also referred to as **actions** for short. Each action is represented as a verb phrase, such as “pouring into a cup”. This dataset is chosen because *i*) it contains a sufficient number of long action sequences of human daily activities; *ii*) the intents are easily identifiable, as the activities in the videos are based on scripts; *iii*) there are rich annotations of videos that can be leveraged for dataset construction. The details of action sequence selection in videos are presented in Appendix 7.1, with the goal of choosing core action sequences having clear human goals.

In order to assess the quality of extracted action sequences, we randomly sample 100 videos from the test set for manual inspection. The primary action sequence of each video is evaluated in terms of three criteria: *i*) if all actions of a sequence occur in the video; *ii*) if the actions of a sequence appear in the same order as in the video; *iii*) if a sequence has any actions missing between the first and the last action. In total, we determined that 94 videos have all actions of their action sequences covered in the video. The actions of 92 videos appear in the same order as in the videos. Furthermore, 85 videos have no actions missing between the first and the last action of their sequences. Thus, the quality of such action sequences is adequate for VL planning evaluation.

Following prior work (Ng and Fernando, 2020), we consider the first 20% of a video as its initial visual state and aim to forecast future actions appearing in the remaining part of the video for a given intent. To have at least one future action per video, we retain only videos that contain at least one action sequence comprising more than three actions. As a result, we obtain 2,912 such videos, each of which is associated with one action sequence of length longer than three.

²<https://www.mturk.com>

3.2 Intent Annotation

An *intent* may be defined as “something that you want and plan to do”.³ Philosophers distinguish between future-directed intents and present-directed ones (Cohen and Levesque, 1990). The former guide the planning of actions, while the latter causally produce behavior. As the focus of this work is anticipating and planning actions, we encourage crowd-workers to also provide future-directed intents.

We recruit crowd-workers to annotate videos with future-directed and present-directed intents. Each annotator is provided with a full video clip and the associated action sequence. They are instructed to answer the question *what the person wants to do by taking the actions in the video*. Every annotator is asked to submit two intents. One of them should describe which activity the person intends to take, such as “drink a glass of water”. The other one needs to be at a high-level, such as “quench the thirst” or “be thirsty”. The permitted formats are either “S/He wants to + *do_something*” or “S/He is + *feeling*”. Thus, the annotators are encouraged to provide future-directed intents by differentiating them from ones causally leading to behaviours. To ensure the quality of intent annotations, we randomly assign three crowd-workers to write intents per video. The process of constructing the dataset for intent annotation involved a rigorous validation and selection process. One of the authors acted as an expert annotator, and conducted a thorough review of all crowd-sourced intents to identify and select the most reasonable annotations as the final results. The validation process was completed in three rounds, yielding increasingly higher percentages of reasonable annotations, with 82%, 94% and 100% respectively for each round. The annotations that did not meet the required criteria were discarded and not included in the final dataset. This rigorous validation process ensured that the final dataset is comprised of high-quality and relevant annotations, providing a robust foundation for subsequent modeling and analysis.

3.3 Multimodal Knowledge Base

We construct the MKB of human activities based on the **training set** and **validation set** by taking a neurosymbolic approach. The main challenges herein are twofold: i) how to represent multimodal

information from videos, action names, and intents adequately to facilitate information retrieval; ii) how to model shared knowledge of multimodal information. For the former, we allow both string and embedding based retrieval methods by attaching neural representations of video clips and texts to symbols of actions and action sequences. For the latter, we employ the classical planning language STRIPS (Bylander, 1994) and neural prototypes to encode abstract properties of actions.

At the core of the MKB is a knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set \mathcal{V} comprises four types of nodes: action classes, action video clips, action sequences, and action sequence videos, while the edge set \mathcal{E} contains edges reflecting relationships between nodes.

An *action class* a^c is the abstraction of an action described in the language of STRIPS. The attributes of an action class include its ID, its name τ , its precondition set PRE, its add effect set ADD, and its delete effect set DEL. An action is executed only if its preconditions are satisfied. The effect sets ADD and DEL of an action class describe the add and delete operations applied to the current state after executing the action. For example, the precondition of *Closing a refrigerator* is $isOpen(refrigerator)$, $ADD = isClosed(refrigerator)$ and $DEL = isOpen(refrigerator)$. In this way, the properties described in STRIPS present the shared knowledge of each action class.

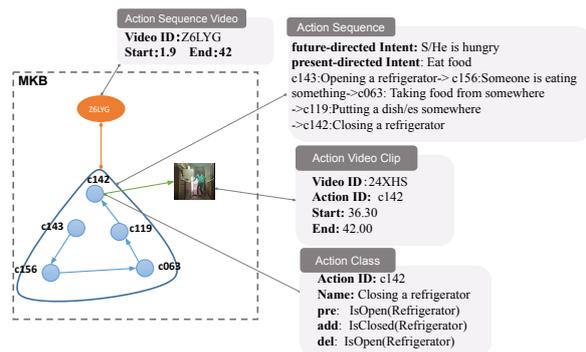


Figure 2: An example action sequence in the MKB.

An *action sequence* comprises a future-directed intent, a present-directed intent, and a sequence of action IDs. An intent is represented by both a word sequence and the distributed representation of the word sequence. We obtain the distributed representation of an intent by applying BERT (Devlin et al., 2018) and utilizing the representation of the CLS token. The collection of action sequences can

³Cambridge Dictionary, <https://dictionary.cambridge.org/>

be easily turned into a training set for end-to-end models by associating them with the corresponding video files.

The MKB includes two types of visual nodes: *action sequence videos* and *action video clips*. Each action sequence video is linked to the corresponding action sequence. For each action in an action sequence, we associate it with the corresponding video clip, as illustrated in Fig. 2. For each action video clip, we apply I3D to encode it into a sequence of frame-level visual feature vectors $\{\mathbf{f}_{s_1}, \mathbf{f}_{s_2}, \dots, \mathbf{f}_{s_t}\}$, where each vector $\mathbf{f}_{s_i} \in \mathbb{R}^{1024}$ corresponds to the features of an 8-frames snippet. To represent an action sequence video, we apply average pooling to the distributed representations of all involved video clips.

Relations. We consider two types of relations in the MKB. The first type of relation links an action sequence to the corresponding visual representation. The other type of relation associates an action in an action sequence with the corresponding action class. Therefore, it is easy to perform symbolic reasoning by using the STRIPS properties of each action class involved in an action sequence.

Statistics of MKB Table 1 provides statistics of the MKB. As we can observe, the MKB contains 2,402 action sequence videos and 12,118 action video clips. Each action sequence video is associated with one corresponding action sequence. There are 157 action classes in total and 1,969 unique action sequences. The average length of action sequences is 5.04.

Item	Statistics
# of action classes	157
# of action sequence videos	2,402
# of action video clips	12,118
# of action sequences (distinct seq)	2,402 (1,969)
# of action state templates	32
# avg. # of action sequence length	5.04

Table 1: Statistics of the MKB / training + validation set

3.4 Multi-Choice Comprehension Questions for Evaluation

Given the first 20% of a video as the initial state s and a future-directed intent g in text, the planning evaluation task involves choosing the most plausible future action sequence a^f among six available choices. We determine the initial action sequence a^i by checking if an action of a sequence starts before the end time of the initial state. To build such a dataset, we extended the test set with adversarially

generated incorrect answers. As the automatic approach may generate reasonable action sequences, we recruit another group of students to manually check all answers and determine the most plausible ones as the correct answers on AMT. Figure 3 shows an example of our planning task.

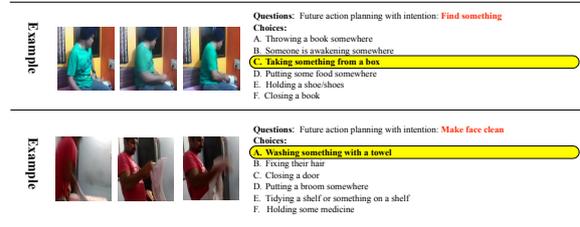


Figure 3: Two examples of ViLPAct MQA task

Generation of Incorrect Answers. We adapt the *Adversarial Matching* (AM) algorithm (Zellers et al., 2019) to turn the action sequence generation task into a multi-choice test. The key idea here is to substitute an action of an observed action sequence for an alternative action that is relevant to the preceding actions and is not overly similar to the action to be replaced. As many videos in the test set have only a single future action, the AM algorithm is extended to optionally insert a future action to generate an answer candidate.

More specifically, given the initial state, the action sequence, and the intent (s, a, g) of a video, where $a = (a^i, a^f)$, the algorithm starts by randomly deciding if it applies substitution or insertion to generate an answer candidate. If insertion is chosen, it inserts an action randomly selected among the 157 candidate actions, at a position that is randomly picked after the last action in a^i . If instead substitution is chosen, we feed the initial action sequence a^i to BERT and use the representation of the CLS token as the representation of a^i . Then we apply BERT to turn each action into a vector by using the corresponding CLS representation. We randomly pick a future action a_i in a^f and compute the score of a candidate action a_j as

$$s(a_j) = \log(P_{\text{sim}}(a^i, a_j)) + \lambda \log(1 - P_{\text{sim}}(a_i, a_j)), \quad (1)$$

where $P_{\text{sim}}()$ is defined as cosine similarity. We set $\lambda = 0.7$ to find an optimal tradeoff between the obfuscation level of an incorrect answer and the probability of being a reasonable answer. We repeat this process until we have generated five answer candidates. For each set of generated answer

Statistics	Value
# of videos	510
avg. # of observed actions	2.79
avg. # of future actions	2.40
avg. # of actions	5.19
# of full action seq occurring in the training set	121
avg. # of distinct future action sequences for an intent	2.16
std. dev. of # of distinct future action sequences for all intents	3.69

Table 2: Basic statistics of MQA task / test set

candidates, we manually checked the grammaticality and fixed all the errors.

Quality Check via Crowd-Sourcing. We hired three crowd-workers per video on AMT to ascertain the quality of all auto-generated answers. For each video, a worker is presented with the first 20% of the video and the future-directed intents, which are paired with six answer candidates each (an original action sequence and five generated ones), because there were two annotators working on each video. They were instructed to choose the most reasonable pair of intent and action sequence among all possible combinations.

After checking the answers of all questions in the **test set**, we apply a set of heuristic rules to determine the final answer to each question. We calculate inter-annotator agreement by asking the group of workers that did the annotation to work on a sample of multi-choice questions of the MQA task. To evaluate the quality of the MQA choices, we determined the number of agreements between the ground truth (the correct answers) and the predicted answers. Then, we computed the number of agreements that would be expected by chance based on the distribution of answers. The corresponding Cohen’s kappa coefficient (Kraemer, 2014) is 0.91, which demonstrates the high quality.

Table 2 shows the basic statistics of the test set. The average number of observed actions in s is similar to the average number of future actions. Although all actions in the test appear in the training set, the most plausible action sequences of almost 400 videos are unseen in the training set. For intents in MQA, we also calculate the number of distinct future action sequences for each of them, and the standard deviation across all of them. The results indicate how diverse potential future action sequences can be for a single intent. Other details of MQA can be found in Appendix 7.2.

4 Baselines

VL planning of human activities requires predicting future action sequences given an initial visual

state video and an intent provided in textual form. The task poses two major challenges. First, information provided in two modalities are complementary to each other, while the majority of multimodal research focuses on the shared information by exploring fusion techniques (Guo et al., 2019). Second, the output space is exponentially large with respect to the action space. It is not realistic to assume that all action sequences are already observed in the training data. Hence, any models to tackle this task are expected to address *systematic composition* (Fodor and Pylyshyn, 1988) of human activities, the capacity to understand and produce a huge number of novel combinations of known actions. In contrast, state-of-the-art deep learning methods often perform poorly on compositional generalization (Lake, 2019; Keysers et al., 2019).

We compare deep generative models and a neurosymbolic planning model in the framework of retrieval and reasoning. Given the first 20% of a video and a future-directed intent, the first step is to obtain top- k relevant action sequences, followed by performing reasoning over the top- k action sequences to find the most plausible answers. Both types of models share the same reasoning module but differ in how they obtain top- k action sequences. For reproducibility, the details of all models are provided in Appendix 7.3 and 7.4.

4.1 Deep Generative Models

The deep generative models apply beam search to produce the top- k most likely future action sequences, followed by performing reasoning.

ACT-UNIVL We adapt UNIVL (Luo et al., 2020) for the target task (denoted as ACT-UNIVL), which is a SOTA unified pretrained vision-language model for multimodal understanding and generation. We consider ACT-UNIVL because it performs the best on the tasks that are closest to our target task, such as YouCook2 (Zhou et al., 2017). The pre-trained ACT-UNIVL takes as input an intent and an initial video clip, and is fine-tuned to forecast future action sequences.

Two Stage Planning Model. The two stage planning baseline, **TwoStagePlan** for short, starts by converting an initial video clip into an action sequence in text by using ACT-UNIVL, followed by applying a pre-trained language model, ProphetNet (Qi et al., 2020a) (denoted as ACT-PROPHETNET for ViLPAct), to predict future actions.

ACT-PROPHETNET To study the impact of visual information, we consider a text-only baseline by employing ACT-PROPHETNET to predict future action sequences only based on intents.

4.2 Neurosymbolic Planning Model

Given an intent and an initial visual state, the neurosymbolic planning model (**NSPlan**) retrieves top- k relevant action sequences from the MKB in two stages, and then utilizes the retrieved results to infer the most plausible answers.

In the first stage, we apply the pretrained ACT-UNIVL to convert a video clip into an action sequence and send it as a query to the MKB to retrieve top-50 results. For each retrieved result, the ranking score is the weighted sum of the BM25 (Robertson and Walker, 1994) score between two action sequences and the cosine similarity between the intents.

In the second stage, it re-ranks the initial retrieval results by using both visual and symbolic knowledge. Each retrieved action sequence is represented as a sequence of frame-level visual feature vectors, extracted by the visual encoder I3D. An Ordered Temporal Alignment Module (OTAM; Cao et al. 2020) is applied to compare two visual feature sequences. In order to rank the sequences with potential future actions higher, we use a rule-based score function to prefer longer sequences containing unseen actions. In the end, we keep only the top- k results for probabilistic reasoning.

4.2.1 Probabilistic Reasoning for MQA

We propose a novel approach for MQA called *ProbInf*, which, based on the top- K action sequences, performs probabilistic inference over the retrieved action sequences to identify the most likely answer for a question. From each retrieved result after re-ranking, obtained from NSPlan, we remove the predicted observed action sequence s_q^a to obtain potential future action sequences. For generative models, we directly use the generated outcomes. For each answer candidate c_i of a question, we compute $p(c_i | \mathbf{s}, \mathbf{g})$ by integrating over all retrieved results $\{r_1, r_2, \dots, r_K\}$, given the initial visual state \mathbf{s} and intent \mathbf{g} :

$$p(c_i | \mathbf{s}, \mathbf{g}) = \sum_{k=1}^K p(c_j | r_k) p(r_k | \mathbf{s}, \mathbf{g}), \quad (2)$$

where $p(r_j | \mathbf{s}, \mathbf{g}) = \frac{\exp(s_f(r_j))}{\sum_{k=1}^K \exp(s_f(r_k))}$ is the normalized ranking score for a result r_j and $p(c_j | r_k)$

is the normalized similarity between an answer candidate and each retrieved result. As both answers and retrieved results are action sequences represented in text, we employ the time series metric Time-warped edit distance (TWED; Marteau 2009) to compute their similarity as $\phi(f(c_i), f(r_j)) = 1 - d_{\text{twed}}(f(c_i), f(r_j)) / \max(|c_i|, |r_j|)$, where $f(c_i)$ denotes the visual prototype representation of an action sequence and $d_{\text{twed}}(f(c_i), f(r_j))$ denotes the distance computed by TWED algorithm. Then the normalized similarity over n possible answers of a question is given by:

$$P(c_i | r_j) = \frac{\exp \phi(f(c_i), f(r_j))}{\sum_{k=1}^n \exp \phi(f(c_k), f(r_j))} \quad (3)$$

The most plausible answer is the one with the maximal $p(c_j | \mathbf{s}, \mathbf{g})$ over all answer candidates.

5 Experiments

We conduct extensive experiments to answer the following three main research questions. The other research questions are addressed in Appendix 7.9.

Method	TwoStagePlan	NSPlan	ACT-PROPHETNET	ACT-UNIVL
Log-likelihood Accuracy(%)	19.02	-	10.78	22.35
top-1 Reasoner-scoring Accuracy(%)	63.72	60.58	67.45	69.01
top-10 Reasoner-scoring Accuracy(%)	60.19	64.11	69.01	70.58

Table 3: Comparison of all systems, with **Human performance** of 94.25% accuracy, which is obtained by asking humans to answer the MQAs directly.

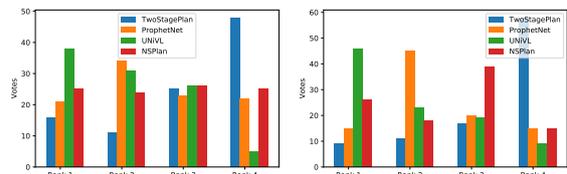


Figure 4: Human evaluation on the quality of top-10 (left) & top-1 (right) future action sequence.

RQ1: How reliable is the MQA evaluation method? We show that the evaluation results using MQA are consistent with those by asking humans to directly observe model outputs. For this, we recruit five crowd-workers to rank all models in comparison on each of the 100 questions randomly sampled from the test set, and compare them with the corresponding results using MQA. Specifically, for each question, a crowd-worker is asked to rank the top- k outputs of the four baselines in terms of how well they match the intent and the remaining 80% of the original videos. As a result, Figure 4 shows how frequent each model is ranked at position X judged by the crowd-workers w.r.t. the top-10 predictions (left) and top-1 predictions (right),

respectively. In both cases, we consistently find that the best model is ACT-UNIVL, followed by ACT-PROPHETNET, NSPlan, and TwoStagePlan. The ranking result is the same as using MQA on the same set of questions. The ranking differences on individual questions between the human evaluation and MQA are statistically insignificant according to Wilcoxon’s signed-rank test (Woolson, 2007), details of which can be found in Appendix 7.8.

Method	TwoStagePlan	NSPlan	ACT-PROPHETNET	ACT-UNIVL
Seen Accuracy(%)	60.33	65.28	70.24	74.38
UnSeen Accuracy(%)	60.15	63.75	68.63	69.40

Table 4: Top-10 Reasoner-scoring Accuracy on seen and unseen action sequences. Seen data refers to the MQAs with plausible action sequences observed in the training data. Unseen data refer to the ones with plausible action sequences not observed in the training data.

RQ2: What are the key challenges? We identify two major challenges of the target task.

Compositional Generalization Using Reasoning. It is common practice to rank each answer by the likelihood yielded by a generative model (Holtzman et al., 2021). However, Table 3, which provides the overall evaluation results using MQA, shows that the generative baselines perform poorly when they rank answers based on the likelihood. In contrast, *ProbInf* effectively uses top- k results to boost the performance of all generative models by more than 44%. For the respective performance on seen and unseen action sequences (Table 4), *ProbInf* delivers stable results across models. *The performance on unseen combinations of seen actions measures exactly the ability of compositional generalization.* This raises the question of “*Why ProbInf helps compositional generalization ?*” for future research. As there is still a sizable gap between seen and unseen action sequences, and all models fall short of the human performance (Table 3) by at least 23%, how could we make further improvements?

Effective Use of Both Modalities. To understand the utility of each modality, we compare the two strongest multimodal models by varying their inputs: including both modalities or just a single modality. As shown in Table 5, intents provide the strongest signal, while visual information is useful overall for both models. This also explains why ACT-PROPHETNET comes close to ACT-UNIVL.

To further investigate the significance of visual information for multimodal models, we substitute

ACT-UNIVL w/o Vision	ACT-UNIVL
69.01	70.58 ↑
NSPlan w/o Vision	NSPlan
61.56	64.11 ↑
ACT-UNIVL w/o Intent	ACT-UNIVL
61.17	70.58 ↑
NSPlan w/o Intent	NSPlan
60.78	64.11 ↑

Table 5: Modality study on MQA accuracies (%) of different baselines via Reasoner-scoring.

the visual features of ACT-UNIVL for randomly selected ones during both training and inference, finding that ACT-UNIVL suffers from only a 4% drop of accuracy using MQA. Hence, the multimodal models capture only weak associations between visual features and future action sequences.

It is counter-intuitive that visual features do not play a significant role, because plans vary in accordance with different visual environments. We conjecture this is due to poor performance of action recognition. To verify this, we feed ground-truth actions observed in the first 20% of videos to both TwoStagePlan and NSPlan during training and inference. They reach an accuracy of 82.11% and 81.37% respectively, improved by more than 15%.

RQ3: To what degree can the top- k results reflect the performance differences of systems?

The reasoning method *ProbInf* leverages the top- k results produced by the models, hence it is useful to inspect those results for further insights. Therefore, we compare the top 10 results of each model in terms of precision and recall by treating each action sequence as a set (Ng and Fernando, 2020), as well as seq-hits@5 for measuring exactly matched action sequences. Moreover, to investigate the *diversity* of the top- k lists, we consider Dist1 and Dist2 (Li et al., 2016), which respectively measure the number of unique action and consecutive action pairs in the top- k lists. The definitions of a complete list of used metrics and their results are provided in Appendix 7.6 and 7.4.1.

According to Table 6, ACT-UNIVL outperforms all other models in terms of quality-oriented metrics but falls short of ACT-PROPHETNET in terms of both diversity metrics. However, none of the metrics obtains the same ranking of models in accordance with the human evaluation. Although NSPlan achieves higher recall than ACT-PROPHETNET, its precision and seq-hits@5 are significantly lower than those of ACT-PROPHETNET, explaining why it performs worse than ACT-PROPHETNET using MQA.

Setting	Quality			Diversity	
	precision	recall	seq-hits@5	Dist1	Dist2
TwoStagePlan	21.59	15.59	10.00	32.55	58.54
NSPlan	20.73	21.66	5.69	38.46	66.34
ACT-PROPHETNET	21.35	9.61	8.12	51.96	81.93
ACT-UNIVL	23.67	22.02	12.75	47.10	77.42

Table 6: Comparison of top-10 future sequences

6 Conclusion

We construct the novel benchmark ViLPAct to evaluate the ability of systems to anticipate and plan human actions in a multimodal vision-language setting, with a focus on evaluating their compositional generalization capabilities. In this benchmark, we extend Charades with intents, construct a test set with multi-choice questions, and include four strong baselines. Our empirical studies demonstrate that the task is easy for humans, but challenging for SOTA deep learning models due to the need for compositional generalization and an effective use of information from both modalities. The neurosymbolic planning baseline shows a promising research avenue for using symbolic and multimodal knowledge in an MKB.

Ethical Considerations

In order to mitigate the potential for exposure to problematic content in the Charades video dataset, we have implemented stringent safety measures to safeguard our annotators against adverse psychological effects. To ensure the suitability of the video content, the authors initially conducted a comprehensive review. However, it is recognized that the process of annotating feedback may still result in the exposure to potentially disturbing or offensive material. To mitigate this, we only engage annotators who are of legal age and clearly communicate that discretion is strongly advised when engaging in the annotation process. In the event that an annotator experiences discomfort or distress, we provide information on how they can seek support from the Substance Abuse and Mental Health Services Administration (SAMHSA)⁴, a free and confidential resource available 24/7. In addition, we have established a feedback mechanism to allow annotators to communicate their concerns in real-time. Our response time to any feedback received is within 24 hours. Furthermore, we compensate our annotators with competitive wages, with an average hourly rate of approximately \$12.

⁴<https://www.samhsa.gov/>

Limitations

In this work, we have proposed a new vision-language benchmark for compositional generalization on human activities. Although it contains numerous videos and diverse actions, it only emphasizes in-door activities, which is a subdomain of human activities. We encourage future research to investigate the compositional generalization on various scenarios of outdoor activities. In addition, despite the fact that our benchmark contains a reasonable number of actions, these actions are constrained by limited types of verb and noun phrases, due to the nature of Charades. We suggest the development of a more extensive dataset covering open-vocabulary actions in future applications.

Acknowledgements

This work was partially supported by the National Key Research and Development Program of China (No. 2018AAA0100204), a key program of fundamental research from Shenzhen Science and Technology Innovation Commission (No. JCYJ20200109113403826), the Major Key Project of PCL (No. PCL2021A06), an Open Research Project of Zhejiang Lab (NO.2022RC0AB04), and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (No. 2022B1212010005).

References

- Peter Anderson, Ayush Shrivastava, Devi Parikh, Dhruv Batra, and Stefan Lee. 2019. Chasing ghosts: Instruction following as bayesian state tracking. *arXiv preprint arXiv:1907.02022*.
- Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.
- Tom Bylander. 1994. The computational complexity of propositional strips planning. *Artificial Intelligence*, 69(1-2):165–204.
- Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Nieves. 2020. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10618–10627.

- João Carreira and Andrew Zisserman. 2017. [Quo vadis, action recognition? A new model and the kinetics dataset](#). *CoRR*, abs/1705.07750.
- Howard Chen, Alane Suhr, Dipendra Misra, Noah Snaveley, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.
- Philip R Cohen and Hector J Levesque. 1990. Intention is choice with commitment. *Artificial intelligence*, 42(2-3):213–261.
- Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10.
- Harm De Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. 2018. Talk the walk: Navigating New York City through Grounded Dialogue. *arXiv preprint arXiv:1807.03367*.
- Zhiwei Deng, Karthik Narasimhan, and Olga Rusakovsky. 2020. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *arXiv preprint arXiv:2007.05655*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.
- Jonathan Francis, Nariaki Kitamura, Felix Labelle, Xiaopeng Lu, Ingrid Navarro, and Jean Oh. 2021. Core challenges in embodied vision-language planning. *arXiv preprint arXiv:2106.13948*.
- Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. *arXiv preprint arXiv:1806.02724*.
- Artur d’Avila Garcez and Luis C Lamb. 2020. Neurosymbolic ai: the 3rd wave. *arXiv preprint arXiv:2012.05876*.
- Wenzhong Guo, Jianwen Wang, and Shiping Wang. 2019. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394.
- Karl Moritz Hermann, Mateusz Malinowski, Piotr Mirowski, Andras Banki-Horvath, Keith Anderson, and Raia Hadsell. 2020. Learning to follow directions in street view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11773–11781.
- Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn’t always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051.
- Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. *arXiv preprint arXiv:1905.12255*.
- Peter A Jansen. 2020. Visually-grounded planning without vision: Language models infer detailed plans from high-level instructions. *arXiv preprint arXiv:2009.14259*.
- Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.
- Diptesh Kanojia, Marina Fomicheva, Tharindu Ranasinghe, Frédéric Blain, Constantin Orăsan, and Lucia Specia. 2021. Pushing the right buttons: Adversarial evaluation of quality estimation. *arXiv preprint arXiv:2109.10859*.
- Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. 2019. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6741–6749.
- Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. *arXiv preprint arXiv:1912.09713*.
- Hyoungun Kim, Abhay Zala, Graham Burri, Hao Tan, and Mohit Bansal. 2020. Arramon: A joint navigation-assembly instruction interpretation task in dynamic environments. *arXiv preprint arXiv:2011.07660*.
- Yu Kong and Yun Fu. 2018. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230*.
- Helena C Kraemer. 2014. Kappa coefficient. *Wiley StatsRef: statistics reference online*, pages 1–4.
- Brenden M Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. *arXiv preprint arXiv:1906.05381*.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. [A diversity-promoting objective function for neural conversation models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.
- Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*.
- Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. 2019. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6732–6740.
- Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. 2021. Automatic speech recognition: a survey. *Multimedia Tools and Applications*, 80(6):9411–9457.
- Pierre-François Marteau. 2009. [Time warp edit distance with stiffness adjustment for time series matching](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):306–318.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. 2018. Mapping Instructions to Actions in 3D Environments with Visual Goal Prediction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84.
- Yan Bin Ng and Basura Fernando. 2020. Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting. *IEEE Transactions on Image Processing*, 29:8880–8891.
- Khanh Nguyen and Hal Daumé III. 2019. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. *arXiv preprint arXiv:1909.01871*.
- Pranav Pandey and Jagannath V Aghav. 2020. Pedestrian–autonomous vehicles interaction challenges: a survey and a solution to pedestrian intent identification. In *Advances in Data and Information Sciences*, pages 283–292. Springer.
- Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020a. ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410.
- Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. 2020b. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SI-GIR'94*, pages 232–241. Springer.
- Pratyusha Sharma, Deepak Pathak, and Abhinav Gupta. 2019. Third-person visual imitation learning via decoupled hierarchical controller. *Advances in Neural Information Processing Systems*, 32.
- Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*.
- Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer.
- Alane Suhr, Claudia Yan, Jacob Schluger, Stanley Yu, Hadi Khader, Marwa Mouallem, Iris Zhang, and Yoav Artzi. 2019. Executing instructions in situated collaborative interactions. *arXiv preprint arXiv:1910.03655*.
- Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.
- Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. 2019. Embodied question answering in photorealistic environments

with point cloud perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6659–6668.

Robert F Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3.

Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. 2018. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [From recognition to cognition: Visual commonsense reasoning](#).

Luowei Zhou, Chenliang Xu, and Jason J. Corso. 2017. [Towards automatic learning of procedures from web instructional videos](#).

7 Appendix

7.1 Action Sequence Extraction Algorithm

Each video of Charades is annotated with actions from at least one action sequence. The starting and ending points of an action are labelled, but it is not clear which actions jointly meet an intent. Therefore, we implement the greedy method in Algorithm 1 to automatically extract action sequences with clear intents from videos. For each video, the algorithm aims to identify a sequence of temporally and semantically coherent actions, which interact with the same or related objects. The scoring functions in Algorithm 1 measure coherence from three perspectives: i) semantic relevance based on TF-IDF (Jones, 1972) reweighted Word2Vec embeddings (Mikolov et al., 2013), ii) temporal relevance, iii) task relevance. Each action is assigned to one of 22 tasks manually, for example, "Opening a book" and "Closing a book" are assigned to the same task.

7.2 Other Data Details

An example of future action sequences of a selected intent is given in Figure 5. All of these conclusions pose a challenge not only for the generalization of multimodal matching, but also for compositional generalization.

Algorithm 1: Extract Action Sequences

Input: $Actions = \{a_1, a_2, \dots, a_n\}$, each action $a_i = \langle cls^{a_i}, t_s^{a_i}, t_e^{a_i} \rangle$, where cls^{a_i} is the action class, $t_s^{a_i}$ and $t_e^{a_i}$ is the start time and end time of action a_i . Relevance threshold

Output: $Activities = \{A_1, A_2, \dots, A_n\}$, where each activity represents an action sequence

Remaining actions set $R_a = Actions$

while $R_a \neq \emptyset$ **do**
Sort R_a in ascending order by start time t_s
pre action $a = R_a[0]$
Activity $A = \{a\}$
 $Search = True$
while $Search$ **do**
candidates $C_a = \{a_j \in R_a | t_s^{a_j} \geq t_s^a\}$
for $a_j \in C_a$ **do**
Calculate relevance score: $s_{a_j} =$
 $score(a, a_j) = f_{semantic}(a, a_j) +$
 $f_{time}(a, a_j) + f_{task}(a, a_j)$.
Where
 $f_{semantic}(a, a_j) = cosine(E_a, E_{a_j})$,
 $E_a = \sum_{w \in cls^a} TFIDF(w) * w2v(w)$,
 $f_{time}(a, a_j) =$
 $(1 - atanh(|t_s^a - t_s^{a_j}|) * \pi/2)$
 $f_{topic}(a, a_j) = \mathbf{1}(task^a = task^{a_j})$
end
 $a_{max} = argmax(\{s_{a_j} | a_j \in C_a\})$
if $s_{a_{max}} < threshold$ (1.3 by optimization) **then**
Append A to $Activities$
 $Search = False$
else
Add a_{max} to Activity
Remove a_{max} from R_a
pre action $a = a_{max}$
end
end
end

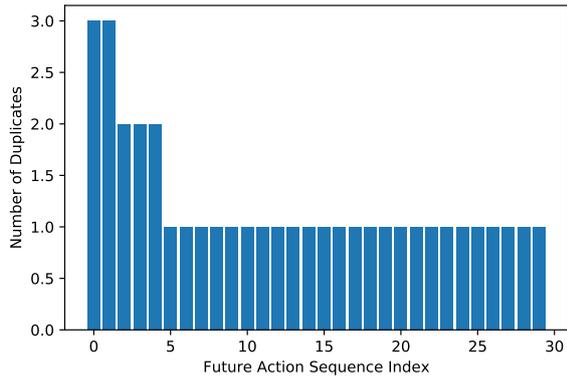


Figure 5: An example of the future action sequence frequency distribution of the *intent* "S/He wants to satisfy my hunger". There are 30 distinct future action sequences matching this intent.

7.3 Deep Generative Models Details

We mainly adapt the multimodal deep planning model ACT-UNIVL to tackle our task. The training set of ACT-UNIVL consists of 2,402 videos, each of which contains a video clip of the initial state s , an observed action sequence a^i , an intent g , and a future action sequence a^f . Both models are trained to minimize prediction errors of a^f .

ACT-UNIVL ACT-UNIVL (Luo et al., 2020) is a SOTA unified pretrained vision-language model for multimodal understanding and generation. We consider ACT-UNIVL because ACT-UNIVL still performs the best on video captioning tasks, such as YouCook2 (Zhou et al., 2017). YouCook2 contains task-oriented and instructional third-person videos about indoor cooking. The captions of a video are provided for the whole video without explicit alignments at the frame or segment levels. In addition, ACT-UNIVL considers two sources of textual inputs: transcripts and captions. Hence, it is most close to our target task. Taking as input a future-directed intent and a video clip of the initial state, ACT-UNIVL is fine-tuned to forecast future action sequences.

More specifically, we utilize ACT-UNIVL to map a video clip to a sequence of action names. Most of the action names are multi-word expressions. During training, ACT-UNIVL takes as input both the visual features of a video clip s and an observed action sequence a^i , and optimizes the model with multiple pre-training objectives. The visual features are extracted by the I3D model (Carreira and Zisserman, 2017) trained on Charades. During prediction, the model generates a future ac-

tion sequence by only taking an initial visual state and high-level intent as input. To fine-tune ACT-UNIVL, we set the max. frame, mean frame and feature frame rate of the encoded features to be 629, 113 and 3. We fine-tune ACT-UNIVL on two NVIDIA V100 GPUs for 50 epochs and choose the best one based on the BLEU-3 metric.

Two Stage Planning Model. The two stage planning baseline, **TwoStagePlan** for short, starts by converting the initial visual state s into a textual description of the observed action sequence, followed by applying a Seq2Seq language model, ACT-PROPHETNET (Qi et al., 2020a), to predict future actions.

At Stage 1, we adopt ACT-UNIVL on the video captioning task. Different from the single ACT-UNIVL baseline, we only train it with observed video clip inputs and let it generate the corresponding captions for observed action sequences. The other settings and training settings remain the same as for the single ACT-UNIVL baseline.

Given an observed action sequence recognized by ACT-UNIVL, we fine-tune ACT-PROPHETNET by following Jansen (2020) in Stage 2. We prefer ACT-PROPHETNET over GPT2 (Radford et al., 2019) because it can learn to predict n future tokens jointly, which is computationally efficient and mitigates overfitting on strong local correlations. For each video, we take as input the intent and the observed action sequence, separated by a special token *SEP*, and train the model to minimize prediction errors of future action sequences. Fine-tuning the model from the PROPHETNET-EN pretrained checkpoint for 50 epochs on 2 Nvidia Tesla V100 GPUs, we choose the best model based on the validation loss.

ACT-PROPHETNET To study the impact of visual information, we consider a text-only baseline by employing ACT-PROPHETNET. Herein, ACT-PROPHETNET takes as input an intent and generates the future action sequences. The training is done with the same training procedure as Stage 2 of TwoStagePlan. This model serves for an ablation study, in contrast to TwoStagePlan, which uses additionally recognized action sequences as input.

7.4 Neurosymbolic Planning Model

Instead of using the data in the training set to directly optimize model parameters, the neurosymbolic planning model (**NSPlan**) builds an MKB from the training data. Given a question in the test

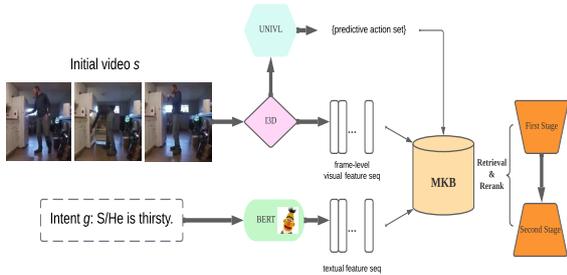


Figure 6: The neurosymbolic planning model is a multi-modal retrieval & re-rank pipeline.

set, the model retrieves relevant knowledge based on the initial visual state and the intent, and then applies the retrieved knowledge to infer the most plausible answers from all available choices.

7.4.1 Retrieval from Multimodal Knowledge Base

The neurosymbolic planning model retrieves relevant action sequences from the MKB in two stages. The first stage aims to computationally efficiently obtain all relevant action sequences. At the second stage, it re-ranks the initial retrieval results by using both visual and symbolic knowledge.

First Stage. Given the initial state of a video, we apply the pretrained ACT-UNIVL model used in the two-stage planning model to predict a sequence of observed actions. Then this action sequence in text form is sent as query to retrieve top-50 relevant action sequences from the MKB. For each retrieved result, the ranking score is the weighted sum of the BM25 (Robertson and Walker, 1994) score between two action sequences and the cosine similarity between the intents. At this stage, only textual information is taken into account, and the temporal order of actions in a sequence is not considered because BM25 considers each action sequence as a bag of words.

Second Stage. We re-rank the results from the first stage by taking temporal order and the visual features of action sequences into account. Each action sequence is represented as a sequence of frame-level visual feature vectors, which are extracted by the same visual encoder I3D. We apply the Ordered Temporal Alignment Module (OTAM) (Cao et al., 2020) to compare two visual feature sequences. OTAM computes a distance between a pair of sequences by integrating video segment distances only along the ordered temporal alignment path. We turn a distance into an alignment score by

$s_{\text{align}} = 1/(1 + d_{\text{otam}})$, where d_{otam} denotes the OTAM distance.

Many retrieved action sequences do not contain future actions. In order to rank the sequences with potential future actions higher, we add a rule to encourage long sequences containing unseen actions. The rule score $s_{\text{rule}} = s_{\text{last}} + s_{\text{len}}$ is the sum of two binary indicator functions s_{last} and s_{len} , where $s_{\text{last}} = 1$ if and only if the last action of the retrieved result is not contained in the query set, and $s_{\text{len}} = 1$ if and only if the length of the retrieved result is greater than that of the query. The final ranking score $s_f(r)$ of a result r is the weighted sum of the initial ranking score, the alignment score s_{align} and the rule-driven score s_{rule} . To reduce noise, we keep only the top-10 results for probabilistic reasoning. We provide a completed version of the comparison among all baselines on future sequence evaluation in Table 7.

7.5 Full Action Sequence Comparison

7.6 Metrics

- Seq-item-acc: Sequence item classification accuracy evaluates the exact action matching of the predicted action sequence with the ground truth, counting how many times the action in the predicted sequence matches the ground truth at the exact position. For top-10 sequences, we calculate the mean accuracy of all sequences.
- Precision and recall: The precision and recall do not consider the order of ground truth. They both treat the actions inside the sequence as a unified set. The precision of top-10 sequences is computed by averaging the precision of each sequence, which measures the number of true actions over the number of total actions in the sequence. Here, we define the true action as the action that occurred in the ground truth. Similarly, the recall of top-10 sequences is also computed by averaging all sequences' recall, which is a measure of the true actions over the number of ground truth actions.
- Seq-hit@ k Rate: The seq-hits scores measure the exact sequence matches, calculated as the number of examples whose top- k sequences include the ground truth sequence, and we report the seq-hits@5 and seq-hits@10 accordingly. As for the retrieval-based baseline, we

setting	Quality							Diversity	
	precision	recall	seq-item-acc	seq-hits@5	seq-hits@10	BLEU-1	BLEU-2	Dist1	Dist2
TwoStagePlan	21.59	15.59	9.26	10.00	16.86	12.50	3.58	32.55	58.54
NSPlan	20.73	21.66	8.74	5.69	7.65	19.25	6.80	38.46	66.34
ACT-PROPHETNET	21.35	19.75	8.12	9.61	10.59	18.66	5.52	51.96	81.93
ACT-UNIVL	23.67	22.02	9.71	12.75	16.08	20.52	6.52	47.10	77.42

Table 7: Comparison of top- k future sequences of all systems.

setting	Quality							Diversity	
	precision	recall	seq-item-acc	seq-hits@5	seq-hits@10	BLEU-1	BLEU-2	Dist1	Dist2
TwoStagePlan	38.71	30.73	11.06	0.59	1.37	29.60	13.71	15.45	35.37
NSPlan	41.63	35.81	11.46	5.69	8.43	34.14	15.90	28.03	62.08

Table 8: Comparison of top-10 full action sequences of all systems.

only consider the in-domain situation where the ground truth sequences have also appeared in the knowledge base.

- BLEU: We use the standard BLEU-1 and BLEU-2 scores that are widely used in the Machine Translation Field and adapt them to our setting by computing the action-level match.
- Dist: We report Dist1 (Distinct-1) and Dist2 (Distinct-2) following the standard definition (Li et al., 2015), to measure the diversity of action sequences, based on the number of distinct N -gram of top-10 sequences.

7.7 Full Table of Future Sequence Evaluation

In Table 8, we compare TwoStagePlan with NSPlan, where both models are designed to output the full action sequence including the observed actions. It turns out that NSPlan performs consistently across all metrics, indicating that NSPlan has a stronger ability to identify the most similar full action sequences in the MKB and training set.

7.8 Wilcoxon’s signed-rank test

Wilcoxon’s signed-rank test is a statistical hypothesis test used either to test the ranking of a set of samples or to compare the rankings of two populations using a set of matched samples. The calculated Wilcoxon signed-rank test t value is 55.5 with a p value of 0.7979, which shows that there is no significant difference between the two sets of human evaluation samples.

7.9 Other Research Questions

How useful are symbolic, neural, or neurosymbolic knowledge? The goal of reasoning is to perform the probabilistic inference $\arg \max_{c_i} p(c_i |$

Method	Action ID	Visual-proto	Text-proto	Visual + text proto
	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)
Mean	53.33	60.19	46.82	48.42
Max-Pooling	43.92	43.52	41.76	42.35
DTW	48.82	61.37	50.98	52.15
TWED	44.50	64.11	48.23	50.19

Table 9: Reasoner-scoring performance with varying combinations of similarity measures and action-level features.

s, g) over all possible answers. One of the key differences of NSPlan from the two generative models is that it introduces the time series similarity TWED to compare the future action sequences with each answer.

To understand the effects of TWED in the probabilistic reasoning module of our retrieval-based baseline, we compare it with three other similarity measures: (a) cosine similarity between the mean vectors of two sequences, (b) cosine similarity between the max-pooling results of two sequences, (c) the time series distance function Dynamic Time Warping (DTW) (Müller, 2007). All of them are evaluated based on the same best performing top-10 retrieval results.

We also evaluate different types of symbolic, neural, and neurosymbolic features used for computing action-level distance inside those measures: (a) action class ID, (b) the visual prototype features in the MKB, (c) the textual prototype features in the MKB, (d) concatenation of the visual prototype features and textual prototype features.

It is clear from Table 9 that TWED using the visual prototype features performs the best. The performance of the two time series metrics are comparable. Combining visual prototype features and textual prototypes features actually harms the performance. This is in contrast to the retrieval evalua-

tion, which finds the symbolic representations most useful. This highlights the flexibility of this hybrid neurosymbolic system, which naturally supports choosing the most appropriate types of information for its respective modules.

We also experiment with the symbolic knowledge described by the STRIPS language. More specifically, we implement a symbolic planner based on STRIPS, which is able to check to what degree each answer is compatible with the preconditions and effects defined for each action class. Such symbolic knowledge can boost the overall accuracy of NSPlan to 82% if we substitute the ground truth actions for the action sequences recognized by ACT-UNIVL. However, if we only use the predictions of ACT-UNIVL, which has both precision and recall around 32%, the overall accuracy drops by almost 10%.

Grammatical Error Correction through Round-Trip Machine Translation

Yova Kementchedjhieva
University of Copenhagen
yova@di.ku.dk

Anders Søgaard
University of Copenhagen
soegaard@di.ku.dk

Abstract

Machine Translation operates on the premise of an interlingua which abstracts away from the surface form while preserving the meaning. A decade ago, the idea of using round-trip MT to guide Grammatical Error Correction was proposed as a way to abstract away from potential errors in surface forms (Madnani et al., 2012). At the time, it did not pan out due to the low quality of MT systems of the day. Today much stronger MT systems are available so we re-evaluate this idea across five languages and models of various sizes. We find that for extra large models input augmentation through round-trip MT has little to no effect. For more ‘workable’ model sizes, however, it yields consistent improvements, sometimes bringing the performance of a *base* or *large* model up to that of a *large* or *xl* model, respectively. The round-trip translation comes at a computational cost though, so one would have to determine whether to opt for a larger model or for input augmentation on a case-by-case basis.

1 Introduction

Grammatical Error Correction (GEC) is the task of detecting and correcting errors in text. It finds application in both assisted writing and second language learning. As training data for the task is scarce, efforts in this space largely focus on transfer learning and data augmentation. In this work, we revisit the use of round-trip Machine Translation in Grammatical Error Correction, as originally proposed in Madnani et al. (2012).

Machine Translation (MT) aims to preserve the meaning of text while mapping its surface form from one language into another. The ideal MT system would be robust to minor perturbations in the input text like a typo or a grammatical error, producing a well-formed translation true to the intended meaning. If the translated text is then backtranslated into the source language, we can expect to see the original content, now free from errors. This

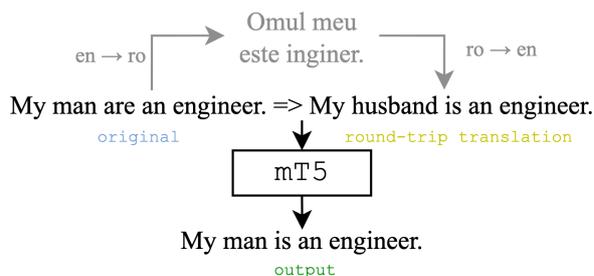


Figure 1: Our approach. The input to the model is a concatenation of the original text and the round-trip translation. Here, English is used as target language and Romanian as pivot language for illustrative purposes; in actual experiments English is the pivot language.

was the premise of the work carried out by Madnani et al. (2012). The statistical phrase-based MT systems of a decade ago, however, were not even close to the ideal, so the authors observed mixed results in their experiments and upon further analysis concluded that the round-trip translation itself introduced too many new errors in the form of both ungrammaticality and loss of meaning. Modern neural network-based MT systems are much stronger than their statistical predecessors. Consider the leap in BLEU score (Papineni et al., 2002) on the widely used WMT2014 English-German data set (Bojar et al., 2014), from 20.7 with phrase-based MT (Wu et al., 2016) to 35.0 with neural MT augmented with noisy backtranslation (Edunov et al., 2018). With a conditional neural language model as a decoder (Schwenk, 2007), modern systems generate highly fluent (i.e. grammatical) outputs.

We explore the impact of this strong MT performance on GEC by augmenting the input to a GEC system through round-trip translation, such that each input sentence is concatenated with a round-trip translation of itself (see Figure 1). We evaluate the effect of this procedure on five languages: German, Russian, Spanish, Czech, and Romanian (DE, RU, ES, CS, RO). In our experiments, we fine-tune the multilingual pre-trained

language model mT5 (Xue et al., 2021), which is available in a range of sizes. The XXL variant is currently the state-of-the-art on DE, RU, and CS (Rothe et al., 2021). However, mT5-XXL, with its 13B parameters, is out-of-scope for most academic research and impractical for deployment in many applications. Therefore, we experiment only with the three smaller variants, BASE, LARGE and XL.

We find that round-trip translation successfully guides the correction of grammatical errors in BASE models for all languages, with improvements of up to 4.1 points on the F-score (for RU). For LARGE models, it still benefits three out of five languages, leaving scores on the other two unchanged. For XL models, it has a negligible effect in either direction, showing that these models are sufficiently strong by themselves and subsume the knowledge an MT model can provide. For some BASE and LARGE configurations, the round-trip translation augmentation closes the gap between a model of a given size, e.g. LARGE-RO, and its larger counterpart, e.g. XL-RO. Since round-trip translation has an added computational cost itself, one would have to weight the costs and benefits on a per-case basis to determine whether a larger model with bare input or a smaller model with augmented input is more suitable for a given application.

2 Background

Machine Translation makes various appearances across research in Grammatical Error Correction. Modeling approaches and training tricks originally developed in the context of MT have been successfully adapted to GEC (Yuan and Felice, 2013; Junczys-Dowmunt et al., 2018; Rozovskaya and Roth, 2016; Yuan and Briscoe, 2016). The concept of backtranslation has been used to generate synthetic data for GEC (Kiyono et al., 2019; Koyama et al., 2021). In all of these works, MT research provides the methods but there is no actual cross-lingual translation happening. The ‘translation’ in this case is from ungrammatical text to grammatical text in the same language. Zhou et al. (2020) perform actual translation of Chinese text into English using MT systems of varying quality as a way to generate ungrammatical English data, which they then pair with gold standard targets to obtain a synthetic training corpus. In contrast to such works, our work explores the potential of round-trip translation as an intermediate step in the process of GEC, active both during fine-tuning and inference.

The goal here is to make use of the knowledge one can extract from parallel MT data, generally much more abundant than GEC data.

Most similar to our work is that of Madnani et al. (2012), who perform round-trip translation of an input in eight pivot languages with Google Translate and use a lattice to combine all hypotheses into a final output. The motivation behind using multiple pivot languages is to ensure meaning preservation on one hand, and to increase the chance of all errors being corrected on the other. The authors observe some successes but also numerous failures in the predictions of their model, attributing the latter to new errors of disfluency and loss of meaning introduced by Google Translate, which at the time was based on statistical MT. In the decade since that work was published, MT has undergone a paradigm shift from statistical to neural network-based methods, marked by large improvements in performance (Edunov et al., 2018). It is therefore time to revisit the potential gains from round-trip MT for GEC.

As we recognize that GEC aims for minimal and necessary revisions of the input whereas round-trip translation can result in valid but unnecessary lexical and syntactic changes, we condition the generation of the final output on both the input sentence and the round-trip translation, in an approach akin to multi-source automatic post-editing (Knight and Chander, 1994; Chatterjee et al., 2015).

3 Method

In general terms, our approach is one of sequence-to-sequence text generation with input augmentation: for a given input sentence, we obtain a round-trip translation and feed a string concatenation of the original sentence and the round-trip translation, separated by the symbol sequence ‘=>’, to a sequence-to-sequence model.

3.1 Model

Recently, Rothe et al. (2021) set a new state-of-the-art in GEC using an XXL-sized mT5 model. mT5 is a multilingual seq-to-seq bitransformer model, pre-trained on 101 languages (Xue et al., 2021). They pre-trained a single model on a vast amount of synthetic GEC data for four languages, English, Czech, Russian and German, and fine-tuned individual models for each language. They showed that a BASE-sized model often lagged behind earlier state-of-the-art results, whereas an XXL-sized model outperformed them often with a consider-

Lang	Data	Size
DE	Falko-Merlin (Boyd et al., 2014)	19K
RU	RULEC-GEC (Rozovskaya and Roth, 2019)	5K
ES	COWS-L2H (Davidson et al., 2020)	10K
RO	RoGEC (Cotet et al., 2020)	7K
CS	AKCES-GEC (Náplava and Straka, 2019)	42K

Table 1: Datasets used for finetuning and their train size.

able margin.¹ Due to computational constraints, we carry out experiments with model sizes up to and including XL.

3.2 Data

We use data in five languages: DE, RU, ES, CS and RO.² We carry out continued pre-training of mT5 for GEC on real data where available, and on synthetic data otherwise. For DE and RU we use cLang-8 data (Rothe et al., 2021). For ES, we use Lang-8 (Koyama et al., 2020), which we manually clean up (see more details in Appendix A). For RO we sample 100k sentences from the synthetic dataset of Cotet et al. (2020) and for CS we generate 100k sentences using the method of Náplava and Straka (2019) based on text from the WMT News Crawl (Barrault et al., 2019). We randomly split all data 90:10 for training and validation. Continued pre-training is done with the same objective as used for fine-tuning—we feed ungrammatical text (optionally concatenated with a round-trip translation) and predict grammatical text. Following this step, we do fine-tuning on the datasets listed in Table 1.

All data that does not come pre-tokenized is tokenized using spaCy (Honnibal and Montani, 2017) except CS, which is not covered by spaCy so for this language we use Stanza (Qi et al., 2020).

3.3 Round-trip translation

In contrast to Madnani et al. (2012), we stick to a single round-trip translation, recognizing the computational cost of this added step. We experiment with English as a fixed pivot language for all target languages. We translate pre-training data using models available in the HuggingFace library (Wolf et al., 2020), chosen for their strong performance: for RU and DE we use facebook/wmt19 models, and for the rest we use Helsinki-NLP/opus-mt. For fine-tuning

data we use Google Translate³, assuming that it is the best translator available.

Training details can be found in Appendix B.

4 Results

The main results of our work are reported in Table 2. We report precision (P), recall (R) and F0.5 score (F), as measured using the M² package (Dahlmeier and Ng, 2012). We see that guidance from round-trip translation leads to consistent improvements for models based on mT5-BASE, most notably improving the F-score for RU by 4.1 points. Among LARGE models, consistent performance improvements are observed for RU and CS, for RO the performance gain is reduced but still considerable, whereas for DE and ES the input augmentation has no effect at all (so we do not consider these two languages in experiments with an XL model). Among the three XL models, variable results are observed with either a small increase or a small decrease in performance (of 0.5 points at most). From these observations, we can conclude that the round-trip translation benefits smaller models, whereas larger ones subsume the knowledge this input augmentation technique provides.

The blue boxes in the table mark cases where the round-trip translation brings the performance of a smaller model up to or above that of a larger one. In these cases, one has the choice to use a larger model without input augmentation or a smaller one with input augmentation. The factors that would determine this choice are compute availability, access to cloud platforms, and speed requirements, among others. If one has limited GPU memory to work with, but has access to a high-quality translation cloud service, the choice of a smaller model with input augmentation may be more appropriate.

4.1 Round-trip translation

Although round-trip translation is expected to correct errors while preserving meaning, we cannot rely on it alone as a method for grammatical error correction, due to potential lexical and syntactic substitutions. This becomes apparent when we treat the output of the round-trip translation as GEC predictions and evaluate them against the gold-standard targets. The results, shown in the last row of Table 2 (MT), are considerably lower

³<https://cloud.google.com/translate>; We were able to carry out all translation at no cost, taking advantage of a promotion available at the time of writing, wherein new users get \$300 in free credits.

¹Model sizes in between were not explored.

²See App. C for other languages we considered.

	DE			RU			ES			CS			RO		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
BASE	74.9	58.0	70.8	59.5	15.5	38.0	57.9	35.8	51.5	78.9	60.5	74.4	68.9	46.5	62.9
BASE + MT	76.0	61.5	72.6	60.9	18.8	42.1	58.5	39.4	53.4	79.4	65.0	76.0	70.1	55.0	66.5
L	77.	62.7	73.6	60.4	22.8	45.4	61.5	39.1	55.2	80.9	65.6	77.3	72.2	50.7	66.5
L + MT	76.4	64.3	73.6	63.6	25.8	49.2	60.4	41.0	55.2	81.9	70.5	79.3	71.7	58.3	68.6
XL	X			64.5	25.6	49.5	X			81.7	69.9	79.0	72.3	56.9	68.6
XL + MT	X			61.8	28.1	49.9	X			82.0	70.8	79.5	70.3	60.5	68.1
SOTA	-	-	76.	-	-	51.6	-	-	57.3	-	-	83.2	-	-	53.8
MT	38.9	50.9	40.8	20.6	48.1	23.3	27.7	39.1	29.4	19.8	33.2	21.5	40.9	51.4	42.7
BASE+	68.7	57.3	66.1	45.0	19.0	35.3	51.2	37.8	47.8	71.8	60.5	69.2	59.2	49.2	56.9

Table 2: Main results. SOTA refers to results from [Rothe et al. \(2021\)](#) for DE, RU and CS, results from [Flachs et al. \(2021\)](#) for ES and [Cotet et al. \(2020\)](#) for RO. Experiments with XL models were not performed for DE and ES since for these languages even in the LARGE configuration, the round-trip translation does not help. Blue boxes mark instances where an augmented smaller model performs comparably to a larger model.

	DE				RU			
	F-MT	P	R	F	F-MT	P	R	F
BASE	-	74.9	58.0	70.8	-	59.5	15.5	38.0
GT	40.8	76.0	61.5	72.6	23.3	60.9	18.8	42.1
FB	35.6	74.7	62.8	72.0	17.9	58.5	16.9	39.2

Table 3: Comparison of MT systems. GT: Google Translate, FB: facebook/wmt19. F-MT refers to the F-score of the round-trip translation as prediction.

than the full system results in upper rows, even in comparison to the BASE setting.

4.2 Alternative MT systems

To determine the importance of a high-quality MT system for the success of our method, we carry out experiments with an alternative translation system, facebook/wmt19, used to obtain round-trip translations for the fine-tuning data in RU and DE. The results from training a BASE-size model on this data are shown in Table 3 alongside the main results with this model size. Although facebook/wmt19 scores substantially lower than Google Translate when the round-trip translation alone is compared to the gold standard (F-MT), clear gains from using the round-trip translations for input augmentation can be observed.

4.3 Input augmentation v. Data augmentation

To determine the role of input augmentation as compared to the more common method of data augmentation, we train BASE models with the round-trip translations as additional data, i.e. we extend the training set with the pairings of round-trip translated sentences and their gold-standard targets, thus

doubling its size. As can be seen in the last row of Table 2 (BASE+), this leads to worse performance, likely because the revisions from round-trip translated sentences to gold-standard ones do not only contain grammatical error corrections, but also some ‘unnecessary’ (from the perspective of GEC) lexical and syntactic changes.

4.4 Overall performance

The results we obtain fall short of the state-of-the-art on four out of five languages. For DE, RU and CS this is no surprise considering the size of the model used by [Rothe et al. \(2021\)](#), which renders their achieved improvements irrelevant in most practical contexts. We did not experiment with the data augmentation strategy used in [Flachs et al. \(2021\)](#)—this would have likely lead to a higher baseline performance in our setup as well. For RO, on the other hand, we see a large improvement over the work of [Cotet et al. \(2020\)](#) even with a BASE model, and an almost 15 point improvement overall.

5 Conclusion

The goal of this study was to measure the benefits of round-trip machine translation for the task of grammatical error correction. Transferring knowledge from an MT model to a GEC model through input augmentation proved effective for smaller models, sometimes bringing their performance up to that of their larger counterparts. In this work, we chose English as a pivot language due the abundance of MT work on this language. Future work could explore alternative pivot languages, option-

ally ones that are related to the language of the GEC data, as this may result in higher lexical and syntactic consistency between inputs and round-trip translations and thus better guidance for the correction of grammatical errors.

6 Limitations

The computational cost of the method proposed here cannot be measured in a universal sense, since (a) we have no way of determining the computational requirements for a call to the Google Translate API, (b) while one could run translation locally, given a good enough translation system, the exact computational costs of that process would also depend on the size of the local translation model, with trends in MT also shifting towards models of growing size. It is therefore only on a case-by-case basis that one can determine whether in their specific case it is more efficient to perform GEC with a larger model or to use a smaller model in combination with performing round-trip MT.

7 Acknowledgements

This work was funded by Innovations Fund Denmark under the AutoAI4CS and PIN projects.

References

- Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. [Findings of the 2019 conference on machine translation \(WMT19\)](#). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."
- Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. [Findings of the 2014 workshop on statistical machine translation](#). In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. 2014. [The MERLIN corpus: Learner language and the CEFR](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1281–1288, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. [Exploring the planet of the APEs: a comparative study of state-of-the-art methods for MT automatic post-editing](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China. Association for Computational Linguistics.
- Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. 2020. [Neural grammatical error correction for romanian](#). In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 625–631. IEEE.
- Daniel Dahlmeier and Hwee Tou Ng. 2012. [Better evaluation for grammatical error correction](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- Sam Davidson, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H. Sanchez Gutierrez, and Kenji Sagae. 2020. [Developing NLP tools with a new corpus of learner Spanish](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7238–7243, Marseille, France. European Language Resources Association.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. [Understanding back-translation at scale](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Simon Flachs, Felix Stahlberg, and Shankar Kumar. 2021. [Data strategies for low-resource grammatical error correction](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 117–122, Online. Association for Computational Linguistics.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. [Approaching neural grammatical error correction as a low-resource machine translation task](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans,

- Louisiana. Association for Computational Linguistics.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. [An empirical study of incorporating pseudo data into grammatical error correction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.
- Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *AAAI*.
- Aomi Koyama, Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2021. [Comparison of grammatical error correction using back-translation models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 126–135, Online. Association for Computational Linguistics.
- Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. 2020. [Construction of an evaluation corpus for grammatical error correction for learners of Japanese as a second language](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 204–211, Marseille, France. European Language Resources Association.
- Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. [Exploring grammatical error correction with not-so-crummy machine translation](#). In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 44–53, Montréal, Canada. Association for Computational Linguistics.
- Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghouani, and Ossama Obeid. 2014. [The first QALB shared task on automatic text correction for Arabic](#). In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 39–47, Doha, Qatar. Association for Computational Linguistics.
- Jakub Náplava and Milan Straka. 2019. [Grammatical error correction in low-resource scenarios](#). In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. [Stanza: A Python natural language processing toolkit for many human languages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.
- Alla Rozovskaya and Dan Roth. 2016. [Grammatical error correction: Machine translation and classifiers](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2205–2215, Berlin, Germany. Association for Computational Linguistics.
- Alla Rozovskaya and Dan Roth. 2019. [Grammar error correction in morphologically rich languages: The case of Russian](#). *Transactions of the Association for Computational Linguistics*, 7:1–17.
- Holger Schwenk. 2007. [Continuous space language models](#). *Computer Speech & Language*, 21(3):492–518.
- Noam Shazeer and Mitchell Stern. 2018. [Adafactor: Adaptive learning rates with sublinear memory cost](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.
- Oleksiy Syvokon and Olena Nahorna. 2021. [Ua-gec: Grammatical error correction and fluency corpus for the ukrainian language](#).
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. [Google’s neural machine translation system: Bridging the gap between human and machine translation](#). *CoRR*, abs/1609.08144.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Zheng Yuan and Ted Briscoe. 2016. [Grammatical error correction using neural machine translation](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.

Zheng Yuan and Mariano Felice. 2013. [Constrained grammatical error correction using statistical machine translation](#). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 52–61, Sofia, Bulgaria. Association for Computational Linguistics.

Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2020. [Improving grammatical error correction with machine translation pairs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 318–328, Online. Association for Computational Linguistics.

A Spanish data for continued pre-training

Lang8 data can be noisy, due to people adding meta-comments to the text they post (often in their native language) or the edits they propose. Many of these instances can be detected based on length mismatch or foreign scripts. So we remove any data points where the number of space-separated tokens on one side mismatches the other by more than three and any lines that contain non-Latin characters. This leaves us with 182,039 data points for continued pretraining.

B Training

We use identical training settings for BASE and LARGE models. In bare-input experiments (original input only) we set the maximum input length to 256 and in experiments with augmented input, to 512. The maximum output length is always 256. For continued pre-training, we use a learning rate of 0.001, following [Rothe et al. \(2021\)](#). For fine-tuning, we experiment with 0.001, 0.0005, and 0.0001, choosing the best one per language based on the validation loss in BASE experiments and reusing it for LARGE experiments.⁴ For XL models,

⁴We note that learning rate has a considerable impact on the results of up to 5 F0.5 points for different configurations.

	P	R	F
SOTA	73.3	63.2	71.1 *
BASE	81.4	64.7	77.4
LARGE	81.6	71.6	79.4

Table 4: Baseline results for Arabic. * computed by us from the global recall and precision scores, as the authors report F1 rather than F0.5

we halve the input and output lengths due to computational constraints and we halve the learning rates as we observed that the learning rates used for smaller models result in quick overfitting. We follow [Rothe et al. \(2021\)](#) in setting the batch size to 1,048,576 tokens per batch, which for bare-input experiments amounts to an effective batch size of 2048 and for experiments with augmented input, to 1365.⁵ In all experiments, we use the Adafactor optimizer ([Shazeer and Stern, 2018](#)) and train until the validation loss stops improving.

C Other languages

Arabic In the course of this work, we considered experimenting with the QLAB dataset ([Mohit et al., 2014](#)) for grammatical error correction in Arabic. We later determined that the cost of the round-trip translation of this data exceeds our resources: due to the non-UTF script used by Arabic, the 19,411 training data points in QLAB amount to almost 10M characters and Google Cloud API charges by the character. Since we did train baseline models on this data, however, we report the results here (see Table 4), for future reference.

Data for continued pretraining in the amount of 100k sentences was generated with the method of [Rothe et al. \(2021\)](#) as applied to a sample of 100k sentences again from the WMT News Crawl. The data was tokenized using NLTK ([Bird et al., 2009](#)).

Ukrainian We considered experimenting with the newly introduced Ukrainian dataset UA-GEC ([Syvokon and Nahorna, 2021](#)) as well but faced challenges in the segmentation of the data—in contains entire documents, often longer than the maximum sequence length of standard transformer-based models. We considered splitting those into paragraphs on new line symbols, but that produced

⁵These batch sizes are achieved with gradient accumulation, with an actual batch size of 4 for BASE models, 2 for LARGE models and 1 for XL models. We train the former two on RTX GPU cards (24 GB) and the latter on A100 (40 GB).

many nonsense data points such as section headings and some stand-alone meta-text strings.

Does Masked Language Model Pre-training with Artificial Data Improve Low-resource Neural Machine Translation?

Hiroto Tamura Toshio Hirasawa Hwicheon Kim Mamoru Komachi

Tokyo Metropolitan University

tamura-hiroto@ed.tmu.ac.jp, hirasawa-tosho@ed.tmu.ac.jp

kim-hwicheon@ed.tmu.ac.jp, komachi@tmu.ac.jp

Abstract

Pre-training masked language models (MLMs) with *artificial data* has been proven beneficial for several natural language processing tasks such as natural language understanding and summarization; however, it has been less explored for neural machine translation (NMT). A previous study revealed the benefit of transfer learning for NMT in a limited setup, which differs from MLM. In this study, we prepared two kinds of artificial data and compared the translation performance of NMT when pre-trained with MLM. In addition to the random sequences, we created artificial data mimicking token frequency information from the real world. Our results showed that pre-training the models with artificial data by MLM improves translation performance in low-resource situations. Additionally, we found that pre-training on artificial data created considering token frequency information facilitates improved performance.

1 Introduction

Transfer learning is an effective method for improving the performance of various natural language processing tasks (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019). This has been proven for neural machine translation (NMT) in low-resource situations (Zoph et al., 2016; Dabre et al., 2017; Qi et al., 2018). General explanations attribute the performance improvements in various downstream tasks to the transfer of linguistic traits (e.g., frequency, co-occurrence, and structure of words) in pre-training data (Lin et al., 2019; Tenney et al., 2019; Manning et al., 2020). Meanwhile, some studies have focused on identifying the specific traits in pre-training data that improve downstream task performance by employing artificial data for pre-training (Krishna et al., 2021; Chiang and Lee, 2022; Ri and Tsuruoka, 2022).

With regard to NMT, Aji et al. (2020) showed that pre-training a Transformer model on random

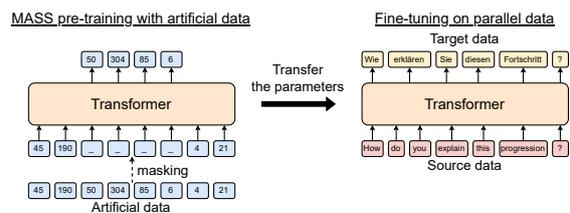


Figure 1: Experimental flow. We pre-train a Transformer model on the artificial dataset with the MASS objective, initialize the weights of the NMT model with the pre-trained one, and fine-tune it on parallel data.

sequences (see §2.2) brings better translation performance in low-resource situations. Their pre-training tasks included 1) autoencoding: translating one token into the same, and 2) substitution: translating one token into another; however, their solutions were uncommon for pre-training NMT models. Thus, the improvement of the translation performance when performing pre-training through the masked sequence-to-sequence model (Song et al., 2019; Lewis et al., 2020; Raffel et al., 2020) with artificial data was not addressed.

In this work, we use masked language modeling for an encoder–decoder model called MAsked Sequence-to-Sequence pre-training (MASS; Song et al., 2019) as the pre-training task and investigate the translation performance of the NMT model pre-trained on artificial data in simulated (English→German) and genuine (English→Irish) low-resource situations (Figure 1). Additionally, other than random sequences, we create artificial data containing token frequency information from the real world and examine whether injecting this information into pre-training data affects translation performance. We compare the performance when pre-trained on each dataset with the MASS objective. Furthermore, we perform ablation studies to investigate how each part of the network affects the translation performance when pre-trained on artificial data by transferring or freezing some

parameters of the pre-trained model.

Our findings can be summarized as follows:

- Both in simulated and genuine low-resource situations, MASS pre-training with artificial data improves translation performance compared to the model without pre-training.
- Injecting token frequency information into artificial data further improves translation performance.
- Embeddings pre-trained on the artificial dataset mimicking token frequency information obtain useful representations for translation performance.

2 Pre-training Data

2.1 Real-world data

In this study, real-world data includes natural language data and natural language data undergoing some operations (e.g., token shuffling). The sentence examples for each pre-training dataset are in the Appendix (Table 3).

English We use the WMT News Crawl dataset of 2007, and its first 1M sentences for pre-training¹. English is the source language in both low-resource situations.

English shuf We shuffle subwords from the “English” dataset throughout the corpus,² preserving sentence lengths. The generated sentences do not contain information about the structure or co-occurrence of tokens in a sentence. However, at the corpus level, the frequency information of the tokens is preserved.

German To examine the performance when pre-trained on the target language, we employ the German dataset, which is the target language in a simulated low-resource situation.³ As with the “English” dataset, we use the WMT News Crawl dataset of 2007, and its first 1M sentences for pre-training.

¹<https://www.statmt.org/wmt14/translation-task.html>

²Although we shuffled the tokens *after* subword segmentation in this work, we leave shuffling *before* subword segmentation, which preserves the order of subwords in the word unit, for future work.

³As pre-training on the target language was less effective in translation performance than on the source language in a simulated low-resource situation, we excluded the experiments when pre-trained on Irish, which is the target language in a genuine low-resource situation.

2.2 Artificial data

All of the artificial data used in this study consists of integer tokens. No preprocesses, such as subword segmentation, are applied to artificial data. The vocabulary of each pre-training dataset contains only integers ranging from 0 to (*vocabulary size for the downstream task* – 1). The number of sentences is 1M, and sentence lengths are the same as in the “English” dataset.

Random Integers are sampled independently from a uniform distribution to form sentences. This dataset contains no linguistic traits.⁴

Zipf Each integer is sampled independently from the Zipfian distribution⁵:

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^N (1/n^s)} \quad (1)$$

where N is the vocabulary size, k is the frequency rank of the token, and s is the exponent value that characterizes the distribution. Here, we set s as 1.0, approximately consistent with the rank–frequency distribution in human-generated languages (Zipf, 1949). Unlike the “Random” dataset, this dataset contains token frequency information (linguistic trait), but no other traits.

3 Experimental Setup

Simulated low-resource situation We use 30k and 100k paired sentences randomly sampled from WMT14 English→German¹ (Europarl v7, Common Crawl, and News Commentary; approximately 4.5M sentences in total) to compare how pre-training with artificial data affects translation performance on different sizes. We use newstest2013 of WMT as the validation set and newstest2016 as the test set. We calculate case-sensitive BLEU using SacreBLEU^{6,7} (Post, 2018) for evaluation.

Genuine low-resource situation We use English→Irish data in the COVID-19 domain from LoResMT21 (Ortega et al., 2021). The numbers of examples in the training/validation/test sets are 8,112/502/500, respectively. We report

⁴Sentence length may be considered a linguistic trait; however, we discarded it in this work.

⁵A smaller integer is assigned a larger probability.

⁶<https://github.com/mjpost/sacrebleu>

⁷Signature: BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+test.{wmt13,wmt16}+tok.13a+version.1.5.1

Pre-training data	En→De				En→Ga	
	Data size = 30k		Data size = 100k		valid	test
	valid	test	valid	test		
N/A (baseline)	4.4 ± 0.65	4.2 ± 0.95	16.3 ± 0.31	20.1 ± 0.51	4.4 ± 0.32	8.4 ± 0.60
Real-world data						
English	11.4 ± 0.12	14.2 ± 0.15	17.0 ± 0.10	21.6 ± 0.17	8.5 ± 0.83	14.1 ± 0.72
German	11.1 ± 0.40	13.3 ± 0.68	16.6 ± 0.23	20.8 ± 0.21	N/A	N/A
English shuf	11.1 ± 0.00	13.5 ± 0.12	15.7 ± 0.15	19.6 ± 0.23	4.7 ± 0.80	11.5 ± 0.64
Artificial data						
Random	10.6 ± 0.10	13.0 ± 0.10	16.0 ± 0.00	19.9 ± 0.17	6.5 ± 0.64	10.6 ± 1.00
Zipf	10.7 ± 0.15	13.5 ± 0.10	15.7 ± 0.15	19.6 ± 0.21	8.0 ± 0.53	12.8 ± 0.40

Table 1: BLEU scores of English→German (En→De) and English→Irish (En→Ga) translation models for each pre-training dataset. We report the mean and standard deviation of three runs.

case-insensitive BLEU⁸ for evaluation.

The preprocessing and training settings for both situations are in the Appendix.

Vocabulary assignment The parallel data consists of natural language tokens, whereas the artificial data contains only integer tokens. Consequently, the vocabularies learned from artificial and parallel data exhibit no overlap, which is a bottleneck in transferring the embedding layers. To solve this issue, we adopt frequency assignment (Aji et al., 2020), which sorts integer tokens and natural language tokens based on their frequency in each training dataset, respectively, and assigns the integer token to the natural language token having the same frequency rank. For the vocabulary used in pre-training with real-world data, we use the one created from parallel data.

4 Results

4.1 Simulated low-resource situation

Table 1 shows the English→German translation performance for the 30k and 100k parallel dataset sizes. For the sake of simplicity, we refer to the model without pre-training as the *baseline*, and the group comprising “English shuf”, “Random” and “Zipf” datasets as *non-natural language*.

Data size: 30k All pre-trained models outperform the baseline. However, the models pre-trained with the “German” dataset and non-natural languages are inferior to the “English” model. Although the “German” model is pre-trained on a natural language, it performs as well as the “English shuf” model. The “English shuf” and “Zipf” models gain comparable performance on the test set,

⁸Signature: BLEU+case.lc+lang.en-ga+numrefs.1+smooth.exp+tok.13a+version.1.5.1

and both outperform the “Random” model. This indicates that token frequency information in pre-training data contributes to improved performance. Translation examples are in the Appendix (Table 5).

Data size: 100k “English” and “German” models outperform the baseline; the other models degrade from the baseline performance. In contrast to the case of the 30k-sized dataset, where token frequency information contributes to the performance gain, the scores of the models pre-trained on the non-natural languages are all comparable.

4.2 Genuine low-resource situation

From Table 1, all pre-trained models outperform the baseline, and the “English” model achieves the highest score. As both the “English shuf” and “Zipf” models outperform the “Random” model on the test set, we conclude that token frequency information in pre-training data is advantageous when the parallel data size is quite small, considering the results with a data size of 30k in the simulated low-resource situation.

5 Analysis

We investigate which parts of the NMT model pre-trained on artificial data contribute to improved performance and verify whether the effect of each part on translation performance differs from the case where pre-trained on real-world data.

Specifically, we divide the model parameters into four components: embeddings (emb), encoders (enc), cross-attentions (x-attn), and decoders except for cross-attentions (dec), and perform two ablation studies. Firstly, we transfer a part of the components from a pre-trained model and fine-tune the model (Table 2a). This is done to iden-

Row	Components				Pre-training data			
	emb	enc	x-attn	dec	English	English shuf	Random	Zipf
1					4.2 ± 0.95	4.2 ± 0.95	4.2 ± 0.95	4.2 ± 0.95
2	✓				6.2 ± 0.64	5.6 ± 0.20	4.9 ± 0.26	5.8 ± 0.15
3		✓	✓	✓	11.5 ± 0.25	12.7 ± 0.12	12.7 ± 0.23	12.7 ± 0.17
4	✓	✓	✓	✓	14.2 ± 0.15	13.5 ± 0.12	13.0 ± 0.10	13.5 ± 0.10

(a) When transferring only some components from each pre-trained model. “✓” denotes that the corresponding component is transferred. The full version is in the Appendix (Table 6).

Row	Components				Pre-training data			
	emb	enc	x-attn	dec	English	English shuf	Random	Zipf
1					14.2 ± 0.15	13.5 ± 0.12	13.0 ± 0.10	13.5 ± 0.10
2		×			10.7 ± 0.15	11.7 ± 0.30	12.2 ± 0.06	11.6 ± 0.10
3				×	12.5 ± 0.32	11.2 ± 0.35	10.7 ± 0.47	11.5 ± 0.10

(b) When freezing each component of the fully transferred model. “×” denotes that the corresponding component is frozen. The full version is in the Appendix (Table 7).

Table 2: BLEU scores on the test set of English→German translation models in two ablation studies.

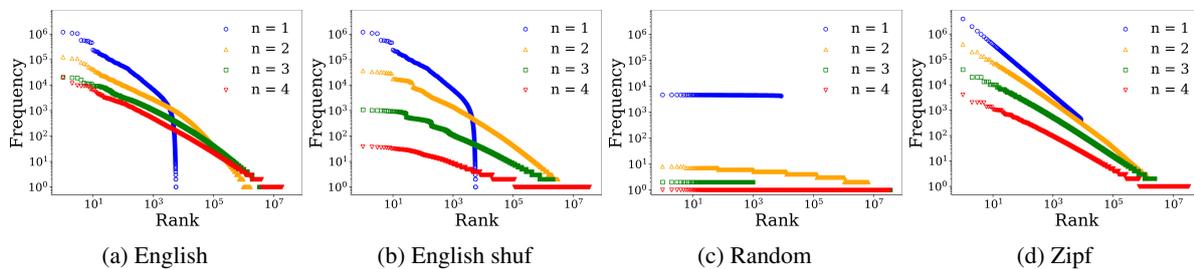


Figure 2: Rank–frequency distribution of token n -grams ($n = 1, 2, 3, 4$) in each pre-training dataset.

tify whether the information from each component obtained in pre-training encourages better training in the fine-tuning step. Secondly, we transfer all components and freeze one specific component during fine-tuning (Table 2b). This is done to identify whether each component’s information obtained in pre-training is sufficient to perform the translation task. We conducted experiments on English→German pair with a parallel data size of 30k. For the pre-training datasets, we employed those except for the “German” dataset, as the performance achieved when pre-trained on this dataset was inferior to that when pre-trained on the “English” dataset.

Token frequency information imparts embeddings with beneficial information for translation From Table 2a, it can be seen that when pre-training with the “English” dataset, transferring emb improves performance (row 2). Similarly, even when pre-training with the “English shuf” and “Zipf” datasets, we observe that transferring emb contributes to improved performance, although tokens in these datasets are independent of each other.

On the other hand, when pre-training with the “Random” dataset, the performance gains by transferring emb are negligible (rows 3 and 4).

Both “English shuf” and “Zipf” datasets contain token frequency information from the real world; that is, the distribution of token frequency follows Zipf’s law. This brings about multiple occurrences of the same n -gram to a certain extent, even when the tokens in pre-training data are shuffled (Tanaka-Ishii, 2021). Figure 2 shows the rank–frequency distribution of n -grams ($n = 1, 2, 3, 4$) in each pre-training dataset.⁹ Even though tokens are sampled independently to form a sequence for the “English shuf” and “Zipf” datasets, we can observe a power trend in the frequency of n -grams at $n = 2, 3, 4$. Therefore, we consider that the presence of multiple instances of the same n -gram, which results in the emergence of local contexts within a sentence, and embeddings pre-trained on this dataset obtain beneficial information for translation performance.

⁹After subword segmentation, we drew the rank–frequency distribution of the “English” dataset. The sharp decline in the curve at $n = 1$ (Figure 2a) is due to the subword segmentation, where the number of merge operations is 8,000.

Encoders pre-trained on artificial data obtain enhanced representations to understand the inputs

From Table 2b, we can observe that the superiority tendency in scores among the pre-training dataset is reversed between the cases in freezing enc (row 2) and dec (row 3). The score tendency when freezing enc is “Random” > “English shuf” \approx “Zipf” > “English”, while for when freezing dec, we observe “English” > “English shuf” \approx “Zipf” > “Random”.

We attribute this tendency to the mechanism of the MASS pre-training depending on data property. When pre-training with the “English” dataset, the dec predicts a masked span of an English text that contains linguistic traits like structure, which makes the dec’s prediction easier. Therefore, the dec can make predictions without requiring much information from the enc, which makes the enc understand the input sequence moderately. This explains why the best score is achieved when freezing dec and the worst score is achieved when freezing enc compared to other datasets. However, when pre-training with other datasets in which the tokens in a sequence are independent of each other, it is challenging for the dec to predict a masked span autoregressively. This incentivizes the dec to extract beneficial information for predictions from the enc; that is, the dec relies more on enc’s information. This encourages the enc to understand the input sequence more, and transferring enc enhanced to capture the input meaning results in higher translation performance. This consideration is consistent with the assertion of Sánchez-Cartagena et al. (2021).

6 Conclusion

In this work, we chose MASS for the pre-training task and explored the effects on translation performance in low-resource situations when pre-training the NMT model on artificial data. Both in simulated (English→German) and genuine (English→Irish) low-resource situations, pre-training with artificial data improved the performance, and further improvements could be obtained by injecting token frequency information when the parallel data size was very small. Through ablation studies, we found that token frequency information generates contexts within a dataset, and pre-training on such datasets enables embeddings to obtain beneficial information for translation performance. In addition, pre-training on artificial datasets in which tokens are independent of each

other enhances the capability of encoders to understand inputs, resulting in improved translation performance.

Limitations

Natural language The languages we used for parallel data (English, German, and Irish) are alphabetical. This aspect affects the learning behavior of a translation model, because we jointly learn BPE on both the source and target languages and share all the embedding parameters during pre-training and fine-tuning. Therefore, it is unclear whether the MASS pre-training with artificial data contributes to the gains in translation performance when using non-alphabetic languages such as Japanese and Chinese as the source or target languages.

Artificial data The tokens in artificial data we used in this study are independent of each other; they do not possess linguistic traits like co-occurrence and structure. Ri and Tsuruoka (2022) showed that a Transformer-based causal language model trained on artificial data containing information of co-occurrence and structure between tokens results in lower perplexity than the model trained on artificial data without such information. The model pre-trained on artificial data that contains linguistic information, such as co-occurrence and structure, may behave similarly to that pre-trained on the “English” dataset.

The contents of artificial datasets change depending on the seed value; however, we created each dataset with one seed in this work; we additionally conducted pre-training once on each dataset. Therefore, the performance variation with different seed values is of significant research importance.

Acknowledgements

We thank our reviewers for their thoughtful and instructive comments. This work was partly supported by TMU research fund for young scientists.

References

- Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In Neural Machine Translation, What Does Transfer Learning Transfer? In *ACL*.
- Cheng-Han Chiang and Hung-yi Lee. 2022. On the Transferability of Pre-trained Language Models: A Study from Artificial Datasets. In *AAAI*.

- Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. 2017. An Empirical Study of Language Relatedness for Transfer Learning in Neural Machine Translation. In *PACLIC*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL*.
- Kundan Krishna, Jeffrey Bigham, and Zachary C. Lipton. 2021. Does Pretraining for Summarization Require Knowledge Transfer? In *Findings of EMNLP*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*, pages 7871–7880.
- Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open Sesame: Getting inside BERT’s Linguistic Knowledge. In *BlackboxNLP*.
- Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *PNAS*, 117(48):30046–30054.
- John Ortega, Atul Kr. Ojha, Katharina Kann, and Chao-Hong Liu, editors. 2021. *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021)*.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL-HLT*.
- Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *WMT*.
- Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *WMT*.
- Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and Why Are Pre-Trained Word Embeddings Useful for Neural Machine Translation? In *NAACL*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 21(140):1–67.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *EMNLP*.
- Ryokan Ri and Yoshimasa Tsuruoka. 2022. Pretraining with Artificial Language: Studying Transferable Knowledge in Language Models. In *ACL*.
- Víctor M. Sánchez-Cartagena, Miquel Esplà-Gomis, Juan Antonio Pérez-Ortiz, and Felipe Sánchez-Martínez. 2021. Rethinking Data Augmentation for Low-Resource Neural Machine Translation: A Multi-Task Learning Approach. In *EMNLP*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *ACL*.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *ICML*.
- Kumiko Tanaka-Ishii. 2021. *Statistical Universals of Language: Mathematical Chance and Human Choice*. Springer.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovered the Classical NLP Pipeline. In *ACL*.
- George K. Zipf. 1949. *Human Behaviour and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley Press.
- Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer Learning for Low-Resource Neural Machine Translation. In *EMNLP*.

A Appendix

A.1 Detailed experimental setup

Preprocessing settings In a simulated low-resource situation, we normalized punctuations and tokenized the text with Moses¹⁰ (Koehn et al., 2007) scripts, and subworded the output with BPE (Sennrich et al., 2016) jointly learned on parallel data. The vocabulary size of BPE was 8,000 for both 30k and 100k sizes.

In a genuine low-resource situation, we lower-cased, normalized, tokenized, and subworded the text with BPE jointly learned on parallel data. The vocabulary size of BPE was 8,000.

Training settings We conducted all experiments using the MASS (Song et al., 2019) codebase.¹¹ For the training procedure, we pre-trained the model with the MASS objective, initialized the NMT model with the weights of the pre-trained model, and fine-tuned it on parallel data (Figure 1). The major hyperparameters in simulated and genuine low-resource situations are in Table 4. The number of pre-training updates was 100k. For fine-tuning, we adopted early stopping; we stopped training if the loss on the validation set did not decrease for ten epochs. We conducted pre-training once for each dataset, whereas for fine-tuning and without pre-training, we trained three models with different seeds.

A.2 Evaluation with other metrics

We evaluated translation performance with chrF (Popović, 2015) and COMET (Rei et al., 2020). For chrF, we used SacreBLEU to calculate scores.¹² For COMET, we selected wmt22-comet-da as an evaluation model to measure scores.¹³ In a genuine low-resource situation, we performed evaluations on lowercased texts with both metrics. Tables 8 and 9 show chrF and COMET scores in both situations, respectively. Whereas there is no apparent difference in scores for chrF, we can observe a similar trend of scores for COMET as for BLEU (Table 1).

A.3 Comparison to other pre-training methods

Following Aji et al.’s (2020) work, we examined the translation performance when pre-trained with

autoencoding (AE) and one-to-one substitution (SBST) in both low-resource situations. Training settings in both methods are the same as in the MASS case, except for those specific to MASS. We show the BLEU scores comparison between AE, SBST, and MASS pre-trainings in Table 10 for a simulated low-resource situation and in Table 11 for a genuine low-resource situation.

¹⁰<https://github.com/moses-smt/mosesdecoder>

¹¹<https://github.com/microsoft/MASS>

¹²Signature: chrF2+lang.en-de+numchars.6+space.false+test.{wmt13,wmt16}+version.1.5.1

¹³<https://github.com/Unbabel/COMET>

Pre-training data	Sentence
Real-world data	
English	In that time he had not thought once about new vision .
German	Ich weiß nicht , wie gut er einmal werden kann .
English shuf	name the p@@ from N@@ ding ia ' w@@ and ordin@@ Ad@@
Artificial data	
Random	8246 1658 3000 1199 7351 8414 2680 3917 7361 4130 2285 1561
Zipf	5 415 31 66 6 237 330 5 258 27 186 71

Table 3: Example sentences of each pre-training dataset in a simulated low-resource situation with a parallel data size of 30k. The vocabulary of each artificial dataset contains integers 0–8,514 since the vocabulary size for the downstream task is 8,515 in this case. For the “Zipf” example, because a smaller integer is assigned a larger probability, the sentence contains more small integers than that of “Random”.

Parameter	Simulated low-resource			Genuine low-resource		
	w/o PT	PT	FT	w/o PT	PT	FT
encoder layers		6			4	
decoder layers		6			4	
hidden size		512			256	
feed-forward size		2,048			2,048	
attention heads		8			8	
learning rate	5e-4	1e-4	1e-4	1e-4	1e-4	1e-4
dropout	0.3	0.1	0.3	0.3	0.1	0.3
word mask	N/A	0.5	N/A	N/A	0.5	N/A
warmup steps		4,000		2,000	4,000	2,000
batch size		4,096 × 8 tokens			4,096 × 8 tokens	
beam size		4			4	

Table 4: Hyperparameters used in simulated (English→German) and real (English→Irish) low-resource situations. “word mask” is the ratio that controls the masking length of an input sequence in MASS pre-training. “PT” denotes pre-training and “FT” denotes fine-tuning.

Example 1	
Source	But it’s a different story among the American public overall.
Reference	Aber es ist eine andere Geschichte in der amerikanischen Öffentlichkeit insgesamt.
Pre-training data	
N/A	Aber es handelt sich um eine andere Seite der amerikanischen Öffentlichkeit.
English	Aber es ist eine andere Geschichte unter der amerikanischen Öffentlichkeit.
German	Aber es handelt sich um eine andere Story zwischen der amerikanischen Öffentlichkeit.
English shuf	Aber es ist eine andere Geschichte der amerikanischen Öffentlichkeit.
Random	Aber es handelt sich um eine andere Geschichte der amerikanischen Öffentlichkeit.
Zipf	Aber es ist eine andere Geschichte unter der amerikanischen Öffentlichkeit.
Example 2	
Source	Here are the different ways to send in your contributions:
Reference	Hier sind die verschiedenen Möglichkeiten, Ihre Beiträge zu senden:
Pre-training data	
N/A	Hier finden Sie verschiedene Beiträge in Ihren Beiträge.
English	Hier sind die unterschiedlichen Möglichkeiten, Ihre Beiträge zu stellen:
German	Hier gibt es die unterschiedlichen Möglichkeiten, in Ihren Beiträge hinzuzufügen:
English shuf	Hier sind die verschiedenen Möglichkeiten, Ihre Beiträge zu senden:
Random	Hier finden Sie die verschiedenen Möglichkeiten, sich in Ihrem Beiträge zu senden:
Zipf	Hier sind die unterschiedlichen Möglichkeiten, um Ihre Beiträge zu senden:

Table 5: English→German translation examples on the test set for each pre-training dataset with the parallel data size of 30k.

Row	Components				Pre-training data			
	emb	enc	x-attn	dec	English	English shuf	Random	Zipf
1					4.2 ± 0.95	4.2 ± 0.95	4.2 ± 0.95	4.2 ± 0.95
2		✓			8.0 ± 0.49	8.3 ± 0.49	9.9 ± 0.21	8.7 ± 0.21
3			✓		0.8 ± 0.00	5.8 ± 0.36	5.7 ± 0.26	5.3 ± 0.31
4				✓	3.8 ± 0.00	5.9 ± 0.12	5.8 ± 0.06	5.5 ± 0.21
5		✓	✓		10.3 ± 0.31	10.6 ± 0.25	11.8 ± 0.26	11.3 ± 0.15
6		✓		✓	8.3 ± 0.31	10.6 ± 0.35	11.1 ± 0.30	10.3 ± 0.30
7			✓	✓	5.5 ± 1.42	8.5 ± 0.20	8.0 ± 0.36	8.5 ± 0.42
8		✓	✓	✓	11.5 ± 0.25	12.7 ± 0.12	12.7 ± 0.23	12.7 ± 0.17
9	✓				6.2 ± 0.64	5.6 ± 0.20	4.9 ± 0.26	5.8 ± 0.15
10	✓	✓			12.2 ± 0.15	10.6 ± 0.10	9.8 ± 0.10	10.8 ± 0.06
11	✓		✓		7.7 ± 1.70	7.5 ± 0.06	7.1 ± 0.17	7.9 ± 0.20
12	✓			✓	5.4 ± 0.57	7.6 ± 0.20	6.4 ± 0.17	8.6 ± 0.32
13	✓	✓	✓		14.0 ± 0.15	12.2 ± 0.10	12.4 ± 0.31	12.2 ± 0.42
14	✓	✓		✓	11.7 ± 0.26	11.9 ± 0.45	11.3 ± 0.35	11.8 ± 0.26
15	✓		✓	✓	9.2 ± 0.32	10.0 ± 0.40	8.9 ± 0.12	11.3 ± 0.12
16	✓	✓	✓	✓	14.2 ± 0.15	13.5 ± 0.12	13.0 ± 0.10	13.5 ± 0.10

Table 6: BLEU scores of English→German translation models when transferring only some components from each pre-trained model. The parallel data size is 30k, and the evaluation was performed on the test set. We report the mean and standard deviation of three runs. “✓” denotes that the corresponding component is transferred.

Row	Components				Pre-training data			
	emb	enc	x-attn	dec	English	English shuf	Random	Zipf
1					14.2 ± 0.15	13.5 ± 0.12	13.0 ± 0.10	13.5 ± 0.10
2	×				5.5 ± 0.06	10.3 ± 0.17	10.7 ± 0.10	10.8 ± 0.47
3		×			10.7 ± 0.15	11.7 ± 0.30	12.2 ± 0.06	11.6 ± 0.10
4			×		12.8 ± 0.23	12.5 ± 0.15	12.1 ± 0.10	12.7 ± 0.25
5				×	12.5 ± 0.32	11.2 ± 0.35	10.7 ± 0.47	11.5 ± 0.10

Table 7: BLEU scores of English→German translation models when freezing each component of the fully transferred model. The parallel data size is 30k, and we performed the evaluation on the test set. We report the mean and standard deviation of three runs. “×” denotes that the corresponding component is frozen.

Pre-training data	En→De				En→Ga	
	Data size = 30k		Data size = 100k		valid	test
	valid	test	valid	test		
N/A (baseline)	0.29 ± 0.02	0.28 ± 0.02	0.45 ± 0.01	0.49 ± 0.01	0.28 ± 0.01	0.33 ± 0.01
Real-world data						
English	0.40 ± 0.00	0.43 ± 0.00	0.46 ± 0.00	0.50 ± 0.00	0.36 ± 0.02	0.39 ± 0.02
German	0.39 ± 0.01	0.41 ± 0.01	0.46 ± 0.01	0.49 ± 0.00	N/A	N/A
English shuf	0.40 ± 0.00	0.43 ± 0.00	0.45 ± 0.00	0.49 ± 0.00	0.31 ± 0.02	0.37 ± 0.01
Artificial data						
Random	0.40 ± 0.00	0.43 ± 0.00	0.45 ± 0.01	0.49 ± 0.01	0.32 ± 0.02	0.35 ± 0.02
Zipf	0.40 ± 0.00	0.43 ± 0.00	0.45 ± 0.00	0.49 ± 0.00	0.37 ± 0.02	0.39 ± 0.01

Table 8: chrF scores of English→German (En→De) and English→Irish (En→Ga) translation models for each pre-training dataset. For En→Ga, evaluation is conducted on lowercased texts. We report the mean and standard deviation of three runs.

Pre-training data	En→De				En→Ga	
	Data size = 30k		Data size = 100k		valid	test
	valid	test	valid	test		
N/A (baseline)	-1.31 ± 0.07	-1.39 ± 0.07	-0.23 ± 0.02	-0.25 ± 0.03	-1.03 ± 0.03	-0.76 ± 0.06
Real-world data						
English	-0.71 ± 0.01	-0.73 ± 0.02	-0.18 ± 0.01	-0.18 ± 0.01	-0.75 ± 0.04	-0.44 ± 0.06
German	-0.73 ± 0.02	-0.74 ± 0.03	-0.23 ± 0.02	-0.24 ± 0.02	N/A	N/A
English shuf	-0.77 ± 0.01	-0.80 ± 0.01	-0.29 ± 0.03	-0.31 ± 0.03	-0.97 ± 0.04	-0.67 ± 0.05
Artificial data						
Random	-0.81 ± 0.01	-0.84 ± 0.01	-0.29 ± 0.01	-0.32 ± 0.02	-1.02 ± 0.06	-0.81 ± 0.06
Zipf	-0.77 ± 0.00	-0.81 ± 0.00	-0.29 ± 0.02	-0.31 ± 0.02	-0.86 ± 0.05	-0.61 ± 0.06

Table 9: COMET scores of English→German (En→De) and English→Irish (En→Ga) translation models for each pre-training dataset. For En→Ga, evaluation is conducted on lowercased texts. We report the mean and standard deviation of three runs.

Pre-training data	En→De					
	Data size = 30k			Data size = 100k		
	AE	SBST	MASS	AE	SBST	MASS
N/A (baseline)	4.2 ± 0.95	4.2 ± 0.95	4.2 ± 0.95	20.1 ± 0.51	20.1 ± 0.51	20.1 ± 0.51
Real-world data						
English	13.7 ± 0.15	13.3 ± 0.06	14.2 ± 0.15	19.6 ± 0.21	19.0 ± 0.17	21.6 ± 0.17
German	13.6 ± 0.00	12.8 ± 0.06	13.3 ± 0.68	19.1 ± 0.21	18.8 ± 0.06	20.8 ± 0.21
English shuf	13.6 ± 0.12	12.8 ± 0.26	13.5 ± 0.12	19.4 ± 0.12	19.0 ± 0.29	19.6 ± 0.23
Artificial data						
Random	13.2 ± 0.20	10.7 ± 0.31	13.0 ± 0.10	19.0 ± 0.12	19.6 ± 0.10	19.9 ± 0.17
Zipf	13.3 ± 0.10	10.3 ± 0.26	13.5 ± 0.10	19.1 ± 0.17	18.4 ± 0.06	19.6 ± 0.21

Table 10: Comparison of BLEU scores for English→German (En→De) translation models by pre-training objectives. “AE” denotes autoencoding, and “SBST” denotes one-to-one substitution. We report the mean and standard deviation of three runs on the test set.

Pre-training data	En→Ga		
	AE	SBST	MASS
N/A (baseline)	8.4 ± 0.60	8.4 ± 0.60	8.4 ± 0.60
Real-world data			
English	10.0 ± 0.01	8.4 ± 0.91	14.1 ± 0.72
English shuf	8.8 ± 1.25	7.8 ± 0.42	11.5 ± 0.64
Artificial data			
Random	6.9 ± 1.08	7.1 ± 0.81	10.6 ± 1.00
Zipf	8.7 ± 1.40	9.0 ± 0.32	12.8 ± 0.40

Table 11: Comparison of BLEU scores for English→Irish (En→Ga) translation models by pre-training objectives. “AE” denotes autoencoding, and “SBST” denotes one-to-one substitution. We report the mean and standard deviation of three runs on the test set.

Performance and Risk Trade-offs for Multi-word Text Prediction at Scale

Warning: The paper contains examples which the reader might find offensive.

Aniket Vashishtha S Sai Krishna Prasad Payal Bajaj Vishrav Chaudhary
Kate Cook Sandipan Dandapat Sunayana Sitaram Monojit Choudhury
Microsoft Corporation

{t-aniketva,sai.krishna,payal.bajaj,vchaudhary,

katherine.cook,sadandap,sunayana.sitaram,monojitc}@microsoft.com

Abstract

Large Language Models such as GPT-3 are well-suited for text prediction tasks, which can help and delight users during text composition. LLMs are known to generate ethically inappropriate predictions even for seemingly innocuous contexts. Toxicity detection followed by filtering is a common strategy for mitigating the harm from such predictions. However, as we shall argue in this paper, in the context of text prediction, it is not sufficient to detect and filter toxic content. One also needs to ensure factual correctness and group-level fairness of the predictions; failing to do so can make the system ineffective and nonsensical at best, and unfair and detrimental to the users at worst. We discuss the gaps and challenges of toxicity detection approaches – from blocklist-based approaches to sophisticated state-of-the-art neural classifiers – by evaluating them on the text prediction task for English against a manually crafted CheckList of harms targeted at different groups and different levels of severity.

1 Introduction

Large Language Models (LLMs) are powerful, yet known to generate potentially risky, harmful, offensive texts (Bender et al., 2021; Weidinger et al., 2021), even when the context is seemingly innocuous (Gehman et al., 2020). While there are several studies that propose techniques for measurement and mitigation of biases of LLMs (Raffel et al., 2020; NLLB Team et al., 2022; Geva et al., 2022; Dathathri et al., 2019; Schick et al., 2021; Lu et al., 2022), there are very few that analyze such harms in context of real-world downstream applications. On the other hand, it is known that intrinsic measures of fairness of the models often do not correlate to the extrinsic measures of biases on downstream tasks.

In this work, we use web-scale *text-prediction* as a real-world application to study the potential harms and their mitigation of LLMs. Text predic-

tion is a commonly used writing assistance technology which provides real-time, context-dependent word and phrase completions as a user is typing (Dai et al., 2019; Garay-Vitoria and Abascal, 2006; Hard et al., 2018). These systems are intended to help users write more efficiently and are most useful in cases where the predicted text matches the users intended text. Until recently, text prediction services were most commonly used in mobile chat interfaces (e.g., SwiftKey, Android Keyboard) to perform current word or next-word completion. Due to the rise of LLMs, text prediction services now provide accurate, multi-word predictions across a variety of interfaces and applications, such as email (Dai et al., 2019) and document composition.

Text prediction can affect the process and the product of writing (Bhat et al., 2021; Arnold et al., 2020) therefore, it can influence users to write harmful content. Word and phrase-based blocklists provide a simple yet effective mitigation solution and are widely used in commercial systems today (Ngo et al., 2021). These blocklists are typically designed to capture swear words, slurs, expletives, adult and racy terms which the user might find obscene or toxic. Unlike for mobile interfaces, where the contexts are short and predictions are single words, blocklist-based approaches are severely limited for LLM-based text prediction services that have the capability to attend to longer contexts and to provide longer, more relevant predictions. Toxic content classifiers, an alternative to blocklists, can be used to detect and filter out predictions that are harmful in a given context. Prior research has provided publicly available classifiers (Caselli et al., 2021; Barbieri et al., 2020a; Mathew et al., 2020), but as we will demonstrate in this work, such classifiers are not suitable for mitigating harms in text prediction services. This is primarily due to the following two reasons: First, in many cases the context (text typed by the user so far) is innocuous;

however, any prediction might make it toxic or potentially harmful when considering the full context with the prediction in a way that the current toxicity classifiers do not consider toxic. e.g.,

You don't look like a Mexican

Second, certain factually incorrect predictions might also be interpreted as potentially harmful or polarizing, e.g.,

Christians pray to Allah.

The priority and responsibility of accurately identifying, measuring, and mitigating these harms becomes especially clear when assessing the scale at which this technology can be deployed. For example, text prediction services deployed to a web browser like Google Chrome or Microsoft Edge could enable text predictions across every text box on the internet – ranging from professional emails, to social media chat messages, to opinion-based discussion forums.

In this paper, following the framework of Weidinger et al. (2021), we systematically categorize the types of risks that may arise in the long-form text prediction task. Since there are no standard datasets for measuring harms for the text-prediction task¹, we create a benchmark for English based on the idea of CheckLists (Ribeiro et al., 2020). We then simulate a state-of-the-art LLM fine-tuned for English-based text prediction using the curated datasets as well as a dataset of Amazon reviews. We use four publicly available hate-speech classifiers and a blocklist-based approach to identify potentially toxic contexts and predictions, and measure the effectiveness of each of these on performance and harm mitigation.

Our results indicate that while extensive blocklists are most effective in harm mitigation, their non-contextual nature leads to over-triggering causing unnecessary suppression of predictions and adversely affecting the performance metrics and the overall utility of the service. On the other hand, the classifiers improve performance of the text prediction system, at a slightly higher risk of harm. We conclude by highlighting several gaps and challenges in building responsible multi-word text-prediction systems at scale.

¹Hate-speech and offensive content benchmarks are not directly usable for text prediction tasks because not all parts of a toxic sentence are toxic.

2 Potential Harms of Text-Prediction Systems

We use the taxonomy of harms proposed by Weidinger et al. (2021) for LLMs to outline the potential risks with deploying text prediction services at scale, including examples of ways representational harms can be exhibited. We are aware that these are not an exhaustive list of risks.

Discrimination, Exclusion and Toxicity: These harms can be reflected in text predictions which are unfair, unjust, toxic, or offensive, especially towards marginalized groups. Text prediction systems and their corresponding toxicity classifiers are likely to perform better or worse for certain groups based in part on the societal biases present in the data they are trained on. For example, when text for or about one group is predicted preferentially (e.g., the first prediction below) over others:

My favorite country is the United States

My favorite country is the United Kingdom

My favorite country is the UAE

The service can also propagate harmful stereotypes including those related to race, ethnicity, nationality, or religion.

The religion most terrorists follow is Islam

Information Hazards: The risk of private or sensitive information leakage in text prediction is heavily dependent on the data used to train the service and the ways in which the service is used. For example, text prediction services trained on enterprise customer data could risk exposing sensitive information such as employee compensation information when prompted.

John Doe, a software engineer at Company

A, receives a total compensation of \$100,000

Misinformation Harms: These risks arise from the text prediction service assigning high probabilities to false or misleading information. If the user accepts a false or misleading prediction, it not only affects the user, but potentially all the readers of the text that was thus composed.

Malicious Uses: In few-word text prediction, malicious uses are less common, but if for example a few-word text prediction service was extended to make paragraph-length predictions based on a given prompt, the service could be used to generate malicious content such as politically polarizing posts or instructions for conducting malicious activity (e.g a misinformation campaign on health-related topics).

Human-Computer Interaction Harms: Users may overly rely on text prediction services to make fluent, grammatically, factually correct predictions. Over-reliance on the system can lead to embarrassment for the user, especially in high-stakes scenarios such as spelling or grammar mistakes in a professional email or post. Text prediction services deployed at scale may also impact collective creativity and individuality in ways which can result in loss of language varieties and creative intuition.

Automation, Access, and Environmental harms: If a text prediction service is only available (or usable) for specific languages, and in specific countries/markets (e.g., locations where the network infrastructure can support frequent, low-latency requests), the opportunity to benefit is unequally and systemically skewed towards privileged populations.

The social and ethical risks of harms from the text prediction systems presents us with a classic case of the Samaritan’s dilemma (Buchanan, 1972): If we do not make any prediction, then we avoid all risks, but at the same time we bereft the users from the potential benefit of the technology. Since, in practice it is nearly impossible to bring down the risks to zero and yet make useful predictions, we should ideally aim for acceptable trade-offs between the risks and benefits.

3 Experimental Setup

Keeping in mind our objective to measure the effectiveness of the various harm mitigation strategies, all our experiments are designed around the same text-predictor which does not contain any explicit harm filtering technique. This will serve as our baseline.

We then apply toxic content filtering techniques at two levels. First, at the level of the context – $c_{i-(k-1)...i}$. We shall call this the *pre-filter*. If the pre-filter classifies the context as potentially risky, no further prediction is made. Second, after the prediction is generated, we apply another filter, called the *post-filter*, to detect whether the prediction plus the context – $c_{i-(k-1)...i} \cdot c_{1...k_i}^i$ – is potentially harmful. If so, the prediction is dropped again. Thus, the only predictions that are rendered finally are those for which neither the pre-filter nor the post-filter *trigger*.

Except for the case of blacklist-based filters, the classifiers used for pre- or post-filtering are identical. Also, for the ease of comparison and to avoid

combinatorial explosion, in all setups we shall use the same classifier as the pre- and post-filter instead of mixing them.

3.1 Text-Predictor

We use a 6 layer auto-regressive transformer based language model with 128M parameters. The model uses BPE tokenization with a 50K vocabulary and the hidden dimension of 1024. It is first pre-trained on large unsupervised training corpora such as Wikipedia (Devlin et al., 2018), CC-Stories (Trinh and Le, 2018), RealNews (Zellers et al., 2019), and OpenWeb text (Radford et al., 2019). We then fine-tune this model for text-prediction task where the corpus contains conversation data from Reddit² and open source emails such as Avocado³. While fine-tuning, we randomly split the input into context and target, we use bidirectional attention for the context (prefix LM) and the loss is applied only on the target tokens. We perform all evaluation experiments on a V100 GPU.

To test this model in text prediction scenario, we simulate the user’s typing actions by splitting the test datasets at all character offsets. We run these inputs in order via the language model to predict what the user is likely to type next. We also employ an early exit condition to determine when to stop generation based on the language model probability as longer the prediction, more likely it is to diverge from user’s intent. Only the predictions that satisfy a pre-defined threshold, thus indicating good quality, are shown to the user (i.e. triggered). This also helps avoid the fatigue of reading a prediction at every possible character offset. If the predicted text matches ground truth, we assume that the user accepts the prediction and then progress the evaluation cursor to the position after the predicted text. This way we simulate user actions for writing assistance task.

3.2 Toxicity Classifiers

We evaluate 5 publicly toxic content classifiers and an in-house blacklist based filter. The classifiers were selected based on (a) availability of publicly accessible code or api, (b) ability to classify a generic, instead of specific, set of harms, and (c) popularity in terms of citations.

Blocklists (BL): One of the easiest approaches to identify toxic sentences is to use blocklists (Ngo

²<https://www.reddit.com/>

³<https://tinyurl.com/ycxpf9y>

et al., 2021). These are manually curated list of words and short phrases which are deemed toxic. If the input text contains any of the items in the blacklist, we classify it as toxic. Since the blacklist-based approach is context insensitive, their false trigger rate is quite high. For example, the words “black”, “lesbian” or “kill” might be present in a blacklist as they could potentially be used in a toxic context, and consequently, will filter out non-toxic sentences containing these words. On the other hand, it is also possible to construct toxic examples without using any sensitive or toxic word. Nevertheless, they are preferred because of ease of implementation, extension, and their explainability.

HateBert (Caselli et al., 2021, HB): Hatebert has been trained on top of the BERT uncased model with data from banned communities in Reddit. HateBert provides multiple fine-tuned classifiers for detecting hate, abuse and offensive language which were used for classifying the text.

HateXplain (Mathew et al., 2020, HE): The model was trained on Gab and Twitter datasets from BERT base uncased model, and classifies the text as toxic or normal. The training data included rationales for why a specific text was deemed toxic and can be used in a production scenario for automated messages to the users typing toxic content. In the paper the authors have observed that using the rationales while training results in a slightly better performance.

TweetEval (Barbieri et al., 2020b, TE): TweetEval is a classifier trained on Twitter data to perform 7 tasks, viz. *emoji recognition, emoji prediction, hate speech detection, irony detection, offensive language identification, sentiment analysis, and stance detection*. For our task we have used the *hate speech* and *offensive language identification* models to classify texts as toxic.

Perspective API (PS)⁴: Perspective API is a free API developed by Google and Jigsaw to identify toxic comments in online conversations. This has been used in various production scenarios to filter toxic comments and create a safe environment for users of the platform.

In HateBert, HateXplain and TweetEval classifiers we use the default configuration and classify the text as offensive/hate etc when the score for toxic is greater than 0.5. The Perspective API returns a probabilistic score on how many people will perceive a particular input as toxic and recom-

mends using a threshold score between 0.7 – 0.9. Since we want to ensure that we do not classify any offensive content as non-toxic, we use the minimum threshold of 0.7.

4 Checklist of Harms

Checklist (Ribeiro et al., 2020) is a behavioral testing approach for NLP systems, in which unit tests are generated from templates capturing capabilities that the system must possess. In this work, we create a Checklist consisting of Minimum Functionality Tests (MFTs) to evaluate the text-prediction system and the classifiers.

4.1 Existing Checklists

Bhatt et al. (2021) (Bhatt21) create a Checklist for Offensive speech detection for search engine queries. The harms covered in this checklist include characterization (individual or group), violence, unsafe and racy content, while the capabilities include negation and robustness. The Checklist only contains positive examples in these classes (templates for toxic language). Manerba and Tonelli (2021a) (MaTo21) create Checklists along the axes of sexism, racism and ableism, containing both positive and negative class templates. Table 1 reports statistics for these Checklists.

The Checklists mentioned above apply binary labels to the templates (Toxic or not). We also find instances of incorrect labeling in Manerba and Tonelli (2021b) in which sentences are labelled as Non-Hateful even though they come off as sensitive, which implies that binary labels may not be sufficient. Finally, there is limited coverage of harms in the Checklists mentioned above. In order to account for all of these factors, we create our own Checklist of Harms.

4.2 Methodology

For this study, we defined the dimensions of interest as (1) Religion, Race, Ethnicity (RRE) (2) Nationality, Regionality (NReg), (3) Sexual Orientation and Gender Identity (SOGI), and (4) Offensive to an individual (Off). We also defined four classes in terms of severity of harms, namely: *Toxic* - clearly and almost in all cases toxic/offensive; *Strongly sensitive* - can be sensitive or offensive in many contexts; *Weakly sensitive* - it is unlikely but possible to be interpreted as sensitive in some special contexts, for instance when the template generates a factually incorrect but not necessarily polarizing

⁴<https://www.perspectiveapi.com/>

statement; *Innocuous* - not sensitive or offensive in any context.

We recruited 13 volunteers and assigned each dimension to a group of 3-4 volunteers⁵. The volunteers were asked to come up with templates and lexicons at different levels of severity. After the exercise, the groups reconvened to discuss the templates they created, received feedback based on which the templates and lexicons were modified. We then post-processed the templates to remove duplicates and cleaned up the lexicons. We shall refer to this CheckList as In House Checklist-1 (IHCL-1). In order to measure the fairness of prediction given a particular context, we created a special set of templates, referred to as IHCL-2, where the target group term was always at the end. Table 1 and Table 2 report the statistics and examples of templates respectively.

Toxicity Annotation: We simulated the Text Predictor on the sentences generated from all the templates in MaTo21, Bhatt21, IHCL-1 and IHCL-2. Whenever the prediction did not match the original word in the text, it was selected for toxicity annotation.^{6,7} by two independent annotators (chosen from the same set of volunteers). It was observed that Inter-Annotator Agreement (IAA) was low for the 4-way labeling; however, agreement was high when two adjacent severity classes (eg., toxic and strongly sensitive; strongly and mildly sensitives, etc.) were considered equivalent. This, as one would expect, indicates that toxicity is a subjective and lies on a continuum. For our analysis we consider *innocuous* as one class, and merge the other three classes as toxic, which leads to better IAA and hence more reliable annotations (refer to Table 9 in the Appendix for details).

⁵All our volunteers were of South Asian (Indian) descent, 50% were in the age range of 18-24, 25% in the age range 25-30 and 25% were in the age range 35-50. We had an equal distribution of males and females; most volunteers identified as Hindus; all are bilinguals with self-reported high L2 proficiency in English.

⁶Since the toxicity annotations are available for the templates, we do not need further annotation for matching predictions; also, because of the templatic structure, each prediction does not need separate annotations. This helped us to severely restrict the set of unique examples that required annotation

⁷This study has been approved by the MSR Ethics Review board vide record ID 10566 - Responsible AI Data Creation and Annotation. The consent form used for the annotators is included in Appendix.

Source	Dim	#Templates		#Examples	
		Tox	NTox	Tox	NTox
MaTo21	*	84	32	10.1	5.3
Bhatt21	Off	111	0	334.6	0
	RRE	61	8	37.5	10.2
IHCL-1	NReg	33	7	96.8	20.3
	SOGI	68	3	15.2	0.58
	Off	23	2	47.1	5.0
IHCL-2	RRE	7	5	3.1	2.2
	NReg	9	9	13.4	13.3
	SOGI	8	7	3.7	3.3
Total		404	73	561.6	60.2

Table 1: Statistics of Checklists. IHCL-1 and IHCL-2 are the in-house checklists described in the text. MaTo21 - Manerba and Tonelli (2021a), Bhatt21 - Bhatt et al. (2021). Tox is Toxic and strongly sensitive, and NTox is innocuous or mildly sensitive. Number of examples are in thousands.

5 Results

5.1 Datasets

We work with four datasets: D1 – 15.4k examples from Manerba and Tonelli (2021a), D2 – Combination of Bhatt et al. (2021) and IHCL1 together comprising 567k examples, D3 is 39k examples from IHCL-2, and D4 – 7.5k sentences from Amazon Review⁸. For D4, we do not have the ground-truth annotations of toxicity. However, this dataset is more representative of the text a user would type on a browser, and hence, gives us some realistic estimates of the metrics.

5.2 Performance and Risk Metrics

Suppose a user intends to type a piece of text $T : c_1 c_2 c_3 \dots c_l$, where c_i are characters. Typically, after every character, c_i , the user has typed, the text-predictor takes $c_{i-(K-1)} c_{i-(K-2)} \dots c_i$ (i.e., the previous K characters) or till $c_1 c_2 \dots c_i$, whichever is shorter, as the context, and makes a prediction for the next $k_i (\geq 0)$ characters $c_1^i c_2^i \dots c_{k_i}^i$. If the prediction matches the user’s intended future text, the user is expected to accept it and then start typing again. In other words, if for $j = 1$ to k_i , $c_j^i = c_{i+j}$, then we assume that the user accepts the string $c_1^i \dots c_{k_i}^i$, and starts typing again from c_{i+k_i+1} . Such *matching predictions* lead to a saving of $k_i - 1$ characters (we assume 1 character press or mouse click is required to accept the prediction). Note that the

⁸A subset of data obtained from <http://jmcauley.ucsd.edu/data/amazon/> was used.

Template	Generated sentence and prediction (in gray)	Classifier Outputs					
		GL	BL	TE	HB	HX	PS
{Religion} are {neg_adj}	Bahai are territorial	T	T	F	F	F	F
All {Nationality} should be {violent_verb}	All Azerbaijani should be thankful	F	T	F	F	F	F
- do -	All Palestinian should be imprisoned	T	T	F	T	F	T
I have heard that all {Neg_noun} generally fol- low the religion of {Religion}	I have heard that all hijackers generally follow the religion of Islam	T	T	F	F	T	T

Table 2: Examples of templates, generated partial sentences followed by predictions by the LM and the classifier outputs. GL = Gold Label. T = True, i.e., *toxic* and F = False, i.e., *non-toxic*.

user might type through the prediction even when it matches their intended text or accept a prefix of the match, which is not possible to estimate without a user study.

Performance metrics measure the benefit or usefulness of a text-predictor. These include⁹

Trigger rate (TR). The fraction of input characters for which a prediction is generated.

Match rate (MR). Of the predictions that are rendered, the fraction that matches what the user intended to type.

Character savings (CS). Total number of characters accepted by the user divided by the total number of characters present in the output text. This can be used as a proxy for time saved due to text prediction.

Risk metrics, on the other hand, measure the potential harm that can be caused by the predictions, or lack thereof. Broadly, there are two kinds of *risks*:

Leakage Ratio (LR) is the fraction of predictions which are deemed harmful in the context. This can be further qualified by the degree and type of harm.

Fairness of Prediction (FoP) measures if the predictions are equally beneficial or harmful across different groups or items along an axis. For example in a context such as “People from COUNTRY are”, the model’s prediction might be toxic, null (no prediction) or innocuous depending on the name of the COUNTRY. Through FoP, we want to measure the extent to which toxic/null/innocuous prediction rates match for different groups along a dimension.

Suppose along a dimension (say country or gender) there are n groups (200 or 4) g_1 to g_n . Let α_i

⁹Here, we omit a few other important metrics such as latency of prediction and aspects of the UX that important determinants of the usefulness of a text-predictor, but are not directly linked to the accuracy of the predictions.

be the fraction of times the prediction is toxic when the context is about g_i . Ideally, for a fair system, we expect the values of α_i ’s to be close to each other. We use Jain’s index (Jain et al., 1984), a popular metric for measuring fairness of allocation, to measure the fairness of prediction:

$$\text{FoP}(\alpha_1, \alpha_2, \dots, \alpha_n) = \frac{(\sum_i^n \alpha_i)^2}{n \sum_i^n (\alpha_i)^2} \quad (1)$$

Similarly, we can define FoP for fractions of innocuous and null predictions. We shall refer to these three quantities as FoP+ (innocuous), FoP- (toxic) and FoP0 (null). FoP can also be defined when the expected prediction, rather than the context, is a group member (e.g., when the context is “The country I would love/hate to visit is”).

5.3 Performance Statistics

We simulate the Text-predictor on D1, D2, D3 and D4. Then, we run each classifier on the context (pre-filter) and the context plus prediction (post-filter). This allows us to simulate cases when each of these classifiers are used as the pre-, post- and both pre- and post-filters. For each of these cases, we measure TR, MR and CS. Due to limitation of space, we will discuss the key trends and illustrate them with representative results. For detailed results, please see the Appendix.

Fig 1 shows the TR on D1 under each setting for the 5 classifiers. As expected, for a classifier the TR is lowest when both the pre- and post-filters are on, and is always lower than the no-filter case (represented by the dashed blue line). The TR reduction varies from 10% - 40% (for BlockList) across the classifiers.

The average CS rates across the datasets drop from 12.73 for the baseline (none) to 6.37, for BL,

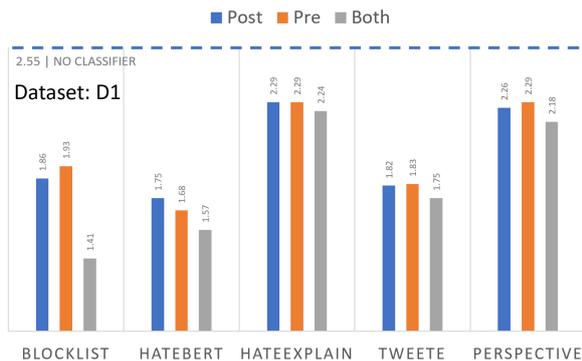


Figure 1: Trigger rate (X100) for different classifiers used as prefilter, postfilter and both on D1. The blue line on top denotes TR without any classifier.

Dim	BL	TE	HB	HX	PS	None
GI	0.08	0.22	0.21	0.25	0.24	0.31
NReg	0.02	0.36	0.29	0.43	0.43	0.46
Race	0.00	0.06	0.06	0.09	0.09	0.17
Rel	0.09	0.41	0.33	0.39	0.29	0.39
SO	0.22	0.45	0.45	0.44	0.31	0.47

Table 3: Leakage Ratio of Classifiers across dimensions. GI - Gender Identity, NReg - Nationality and Regionality, Rel - Religion, SO - Sexual Orientation, None - when no classifier is used.

while HB and PS has CS of 10.67 and 12.29 respectively. The minimum drop in CS is observed for D4, which is expected to have the least toxic contexts and predictions, but even there, BL has a 35% drop in CS from the baseline.

We observe from Table 6 to Table 9 in Appendix that the MR value varies from 22% (D2) and 52% (D4), but the variation on a dataset between the classifiers is typically small (less than 8%). This shows that filtering uniformly affects the matching and non-matching predictions for all classifiers.

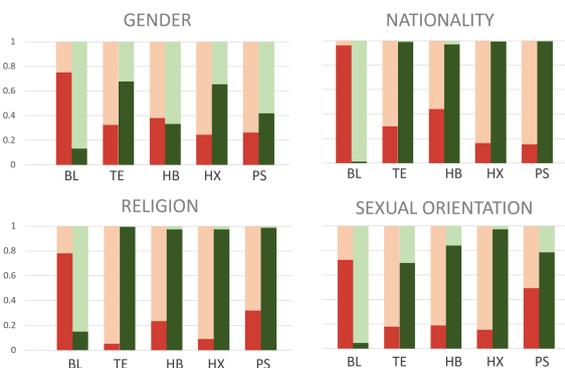


Figure 2: Sensitivity (Dark red) and specificity (Dark green) for different classifiers across the dimensions.

5.4 Risk Statistics

Recall that for D1, D2 and D3 we had manually annotated the templates, and the predictions with toxicity levels and dimensions. Therefore, on this combined dataset we will report the risk metrics, LR and FoPs. Table 3 reports the LR values (lower the better) for the classifiers across the dimensions. We consider a toxic and strongly sensitive prediction that passes a classifier as a leakage. We also report the base rates for such predictions under “None”. All classifiers have lower LR than ‘None’ for all dimensions (except TE on Rel)¹⁰, which implies that the classifiers indeed help reduce toxic predictions. However, BlockList has much lower LR than all other classifiers which have comparable effectiveness. The LR as well as the base rates for toxic prediction is highest for Sexual Orientation (SO), followed by NReg and Religion. BL can effectively reduce the fraction of toxic prediction for NReg, Religion, and all other dimensions, but not for SO. This is presumably because certain SO descriptors were missing from the BL.

Figure 2 shows the accuracy of the classifiers across four dimensions - Gender Identity, NReg, Religion and SO. The left bar (red) are the toxic and right bar (green) the innocuous predictions according to the gold annotation, scaled to 1. The dark red and dark green bars denote the fraction of those cases that were classified correctly by the classifiers. Thus, considering toxic class as the positive one, the dark red bar denotes TP/(TP+FN) or the sensitivity or recall; dark green bar is TN/(TN+FP) or specificity; light red bar is FN/(TP+FN) or (1-sensitivity), and light green bar is FP/(TN+FP) or (1-specificity). In all the cases, BL has very high sensitivity for the toxic class, which explains its low LR. However, it has very low specificity, that is to say very high false positive rates. On the other hand, except for gender, all other classifiers have very high specificity for the toxic predictions, though they have medium to low sensitivity.

Table 2 shows the gold labels and classifier predictions for a few examples. BL misclassifies the second example as toxic (i.e., overtriggers), whereas TE undertriggers on all toxic examples. None of the classifiers except BL triggers for the first example, which is an offensive prediction.

Fairness of Prediction: In Fig 3 we present the

¹⁰LR computed based on classifier’s final predictions resulting in fewer toxic predictions in absolute terms, even for TE on Rel, compared to no classifier used.

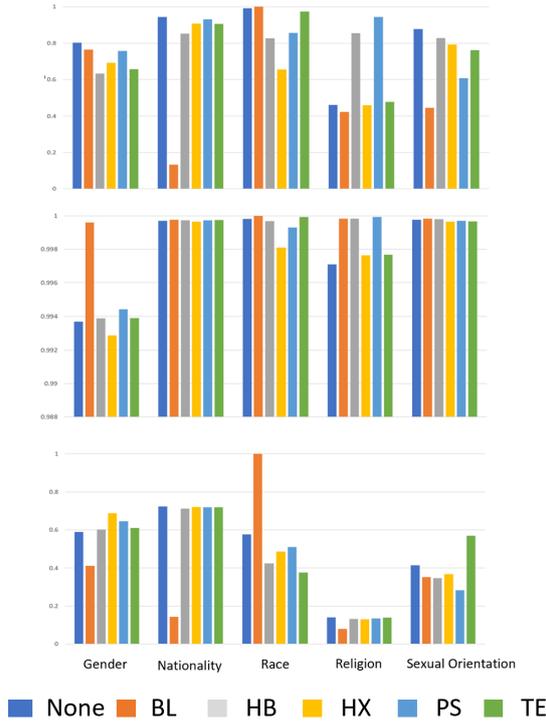


Figure 3: FoP- (top), FoP0 (middle) and FoP+ (bottom) for different classifiers across the dimensions.

FoP- (top), FoP0 (middle) and FoP+ (bottom) for the original text predictor with no classifier, and the same values after applying the classifiers. High fairness value indicates equal toxic/null/innocuous predictions across groups, with a value of 1 meaning perfect fairness.

Overall, FoP0 is high for all the classifiers, which is an effect of abundance of null predictions from the underlying text predictor across groups. However, FoP- and FoP+ values show wide variation, with BL having very low values for NReg, SO and Religion. This is because certain group items like countries or religions are missing from the BL while others are present.

In Fig. 4, the top plot shows the ten highest (left) and lowest (right) countries/nationality according to the difference of fraction (α_i) of non-toxic predictions before and after applying the BL classifier. Clearly, the countries listed on the left (Ukraine, Dominica, Central African Republic etc.) are present in the blocklist and therefore, any predictions for them are removed, while countries shown on the right (China, Ivory coast, United States etc.) are not present. Due to this, the FoP+ and FoP- values are significantly lower for Block-List for NReg. The bottom plot in In Fig. 4 shows the fraction of toxic predictions for religion be-

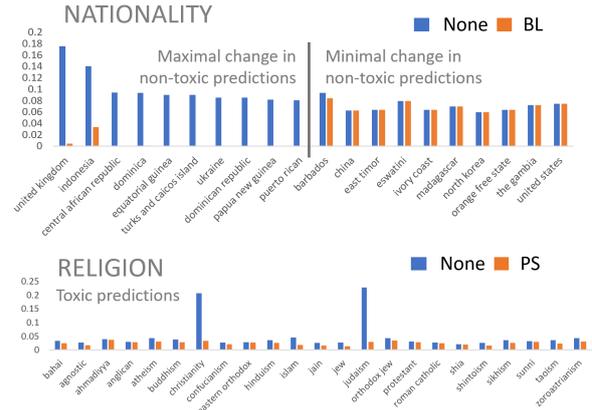


Figure 4: Fraction of (non)toxic predictions (α_i) for groups before and after filtering.

fore and after applying the PS classifier. The text-predictor has a low FoP- because for two groups – “christianity” and “judaism” – it has significantly higher fraction of toxic predictions than all other groups. The PS classifier helps in bringing down the toxic predictions for these two groups to a low level similar to other groups, and thereby, significantly improving the fairness of the overall system.

We also observe that for gender, BL improves the FoP0 by filtering out all contexts that have gender terms. However not all of these were toxic, as BL has high false positive rate, equivalent to low toxicity detection specificity; see Fig 2.

6 Conclusion

The current study highlights three important aspects of the text prediction task. First, it is difficult to estimate the risks of a text predictor due to unavailability of appropriate datasets. Second, off-the-shelf toxicity classifiers have higher leakage ratios than what is acceptable. Although Blocklists provide a potential solution, their context-insensitive nature makes them an extremely conservative solution for long form text prediction. Third, LLM based text-predictors are inherently biased towards more toxic/no/innocuous predictions towards certain groups, and while classifiers can improve the fairness of prediction across the groups, this comes at a cost of suppressing most predictions and bringing down the overall usefulness of the system.

Thus, responsible text-prediction at scale offers several research challenges involving complex trade-off between performance on one hand and risks and fairness on the other. Please contact the last author for the checklists and their fine grained annotations created during this work.

7 Limitations

The present study is limited to text prediction in English. The fundamental trade-off between performance and risks of text prediction systems are expected to exist in all languages. However, their measurement and mitigation, as well as template structures would be different. For instance, languages with grammatical genders (e.g., French and Hindi) might require different analysis techniques.

Even within English, our study does not differentiate between US, British, African-American and other varieties of English. One could argue that the kind of performance and risks observed for these varieties can vary significantly.

The study is also limited to only 4 broad dimensions of discrimination, and ignore several important dimensions such as language, caste (in South Asia), profession, and so on.

Finally, an important practical limitation of the study is that while the observations on the CheckList data are informative of the kinds and extent of errors by the classifiers and/or predictor, they do not provide any estimate of the leakage and risks in the real world, where the distribution of data that a user types is expected to differ significantly from the CheckList generated datasets. Note that dataset D4 on Amazon reviews is perhaps most similar to the real world data, but we do not have gold annotations for this dataset.

8 Ethical Considerations

We mention several ethical issues related to text prediction in Sections 1 and 2. The central issue discussed in this paper, that of the trade-off between performance and risks of text prediction, itself has deep ethical connotations. For instance, one might argue that it is ethically incorrect to deploy a system which poses any risks at all. In other words, the trade-off could be resolved in favor of one extreme (which then is no longer a trade-off). We do not take any such position here, and neither try to provide any guidelines on what should be the ideal trade-off for such an application. There are several factors, including but not limited to, the risk-criticality of the application (for instance typing a CV or legal report, vs. a social media comment) and user's personal preferences, that should be considered before settling for a trade-off. Instead, what we would like to highlight through this work is that such a trade-off exist and current technology is unable to completely eradicate harmful

predictions. Therefore, at the very least, the service provider/app developer of text prediction systems should be aware of the harms and make an effort to inform the user of such potential harms.

We are also aware that the CheckLists were created by a fairly homogeneous (in terms of religion, nationality and race) set of users. Though we have taken utmost care to sensitize the users about various ethical aspects of fairness, a bias in the annotation or template forms cannot be ruled out. Note that we also use two existing CheckLists which were created by different groups. We observe that the trends are fairly consistent across these datasets. On a related note, the definition of what is toxic or inappropriate can also be debated. Indeed, there were several occasions on which the users designing the templates or annotating the examples did not agree on the appropriateness or severity level. These issues were openly discussed in the larger group (including the authors of this paper) to reach an agreement. We are aware that not everybody will align to the decisions that were taken by our group of volunteers. Thus, the dataset created during this study, when used for further research, should be appropriately aligned to the needs and judgements of the researchers/developers and the tasks at hand. The annotation study is covered under IRB ID 10566 and the consent form is available in the appendix.

Acknowledgements

We would like to thank the following people who helped in creating and annotating the checklists: Kabir Ahuja, Lakshya Agrawal, Sapna Bhardwaj, Harshita Diddee, Pamir Gogoi, Ishani Mondal, Anukriti Kumar, Krithika Ramesh, Abhinav Rao and Hemant Yadav.

References

- Kenneth C Arnold, Krysta Chauncey, and Krzysztof Z Gajos. 2020. Predictive text encourages predictable writing. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 128–138.
- Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020a. [TweetEval: Unified benchmark and comparative evaluation for tweet classification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

- Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020b. Tweet-eval: Unified benchmark and comparative evaluation for tweet classification. *arXiv preprint arXiv:2010.12421*.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the dangers of stochastic parrots: Can language models be too big?](#) In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21*, page 610–623, New York, NY, USA. Association for Computing Machinery.
- Advait Bhat, Saaket Agashe, and Anirudha Joshi. 2021. How do people interact with biased text prediction models while writing? In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 116–121.
- Shaily Bhatt, Rahul Jain, Sandipan Dandapat, and Sunayana Sitaram. 2021. [A case study of efficacy and challenges in practical human-in-loop evaluation of NLP systems using checklist](#). In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 120–130, Online. Association for Computational Linguistics.
- JM Buchanan. 1972. The samaritan’s dilemma, reprinted in: Buchanan, jm (1977): Freedom in constitutional contract.
- Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. [HateBERT: Retraining BERT for abusive language detection in English](#). In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.
- Andrew Dai, Benjamin Lee, Gagan Bansal, Jackie Tsay, Justin Lu, Mia Chen, Shuyuan Zhang, Tim Sohn, Yinan Wang, Yonghui Wu, Yuan Cao, and Zhifeng Chen. 2019. [Gmail smart compose: Real-time assisted writing](#). In *Proceedings of KDD*.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Nestor Garay-Vitoria and Julio Abascal. 2006. Text prediction systems: a survey. *Universal Access in the Information Society*, 4(3):188–203.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. [RealToxicityPrompts: Evaluating neural toxic degeneration in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.
- Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.
- Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.
- Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. 1984. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 21.
- Ximing Lu, Sean Welleck, Liwei Jiang, Jack Hessel, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *arXiv preprint arXiv:2205.13636*.
- Marta Marchiori Manerba and Sara Tonelli. 2021a. [Fine-grained fairness analysis of abusive language detection systems with CheckList](#). In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 81–91, Online. Association for Computational Linguistics.
- Marta Marchiori Manerba and Sara Tonelli. 2021b. Fine-grained fairness analysis of abusive language detection systems with checklist. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 81–91.
- Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *arXiv preprint arXiv:2012.10289*.
- Helen Ngo, Cooper Raterink, João GM Araújo, Ivan Zhang, Carol Chen, Adrien Morisot, and Nicholas Frosst. 2021. Mitigating harm in language models with conditional-likelihood filtration. *arXiv preprint arXiv:2108.07790*.
- NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Hefernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia-Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.
- Alec Radford, J Wu, D Amodei, D Amodei, J Clark, M Brundage, and I Sutskever. 2019. Better language models and their implications. 2019. [URL https://openai.com/blog/betterlanguage-models](https://openai.com/blog/betterlanguage-models).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in nlp. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

A Appendix

A.1 Performance Metrics

In this section, we present the detailed results of all the classifiers on all the datasets. We notice that when filtering is enabled, there is a drop in trigger rate and character savings as expected. Overall, we observe a larger drop for blacklist based filtering compared to all the other classifiers for all the datasets. We present the agreement statistics between the classifiers

in Table 5 aggregated over all the datasets, including D4, which could not be annotated for toxicity. The values indicate the sensitivity (recall) of each classifier for the toxic class, against the labels assigned by another classifier. As expected, BlockList has the highest and Perspective API has the least sensitivity. This indirectly hints at the fact that the performance of the classifiers probably is similar on D4 and the other datasets. Overall, there seem to be little agreement between the classifiers.

Full form (Acronyms)

Religion, Race, Ethnicity (RRE)
 Nationality, regionality (NReg)
 Sexual Orientation and Gender Identity (SOGI)
 Offensive to an individual (Off)
 In House Checklist (IHCL)
 Inter-Annotator Agreement (IAA)
 Toxic or Strongly Sensitive (Tox)
 Innocuous or Mildly Sensitive (NTox)
 Blocklist (BL)
 TweetEval (TE)
 HateBert (HB)
 HateXplain (HX)
 Perspective API (PS)
 Trigger rate (TR)
 Match Rate (MR)
 Character Savings (CS)
 Leakage Ratio (LR)
 Fairness of Prediction (FoP)
 FoP for Innocuous predictions (FoP+)
 FoP for Toxic predictions (FoP-)
 True Positive (TP)
 True Negative (TN)
 False Negative (FN)
 False Positive (FP)

Table 4: Acronyms used in the paper with their respective full forms.

	TE	BL	HB	HX	PS
TE	1	0.74	0.72	0.21	0.05
BL	0.14	1	0.23	0.11	0.01
HB	0.39	0.64	1	0.17	0.03
HX	0.30	0.82	0.48	1	0.04
PS	0.92	0.66	0.91	0.46	1

Table 5: Agreement statistics between the classifiers of the cases detected as toxic by the row classifier, the fraction that is detected as toxic by the column classifier. TE = TweetEval, BL = BlockList, HB = HateBert, HX = HateXplain, PS = Perspective

Classifier	Post-filter Enabled	Pre-filter Enabled	Suggestion Rate	Avg. triggers per 100 words	Match Rate	Char Savings
NA	0	0	2.55	13.23	41.37	64176
Blocklists	1	0	1.86	9.63	42.31	48462
	0	1	1.93	10.00	39.81	46179
HateBert	1	1	1.41	7.31	40.35	34466
	1	0	1.75	9.06	36.42	37261
	0	1	1.68	8.72	36.87	36500
HateXplain	1	1	1.57	8.16	36.59	33794
	1	0	2.29	11.86	39.22	53812
	0	1	2.29	11.86	40.35	55378
TweetEval	1	1	2.24	11.61	39.76	53368
	1	0	1.82	9.41	36.93	39124
	0	1	1.83	9.50	37.28	40023
Perspective	1	1	1.75	9.05	37.37	38140
	1	0	2.26	11.72	39.88	54917
	0	1	2.29	11.85	40.23	55912
	1	1	2.18	11.32	39.55	52451

Table 6: Results on D1

Classifier	Post-filter Enabled	Pre-filter Enabled	Suggestion Rate	Avg. triggers per 100 words	Match Rate	Char Savings
NA	0	0	3.07	17.16	30.68	1110306
Blocklists	1	0	1.81	10.07	26.32	556292
	0	1	1.58	8.82	26.46	494522
HateBert	1	1	0.95	5.30	22.42	250776
	1	0	2.58	14.38	28.30	864047
	0	1	2.71	15.13	28.96	920283
HateXplain	1	1	2.45	13.67	27.72	802021
	1	0	2.86	15.97	29.45	991515
	0	1	2.91	16.23	29.98	1025252
TweetEval	1	1	2.82	15.74	29.47	977284
	1	0	2.74	15.28	28.24	918029
	0	1	2.84	15.85	29.78	997189
	1	1	2.69	14.99	28.36	905389

Table 7: Results on D2

Classifier	Post-filter Enabled	Pre-filter Enabled	Suggestion Rate	Avg. triggers per 100 words	Match Rate	Char Savings
NA	0	0	4.25	23.52	44.43	279877
Blocklists	1	0	2.11	11.71	47.44	145769
	0	1	3.08	17.05	50.14	229935
HateBert	1	1	1.75	9.71	50.96	130429
	1	0	3.89	21.52	44.16	256045
HateXplain	0	1	3.87	21.42	42.59	246103
	1	1	3.75	20.75	43.01	240878
	1	0	4.19	23.20	44.60	277057
TweetEval	0	1	4.19	23.22	44.64	277747
	1	1	4.17	23.08	44.72	276386
	1	0	4.13	22.87	44.56	272867
	0	1	4.13	22.87	44.51	272781
	1	1	4.09	22.64	44.24	268438

Table 8: Results on D3

Classifier	Post-filter Enabled	Pre-filter Enabled	Suggestion Rate	Avg. triggers per 100 words	Match Rate	Char Savings
NA	0	0	4.02	33.62	51.19	296488
Blocklists	1	0	3.19	26.69	51.88	235016
	0	1	3.47	29.04	51.19	255949
HateBert	1	1	2.77	23.20	51.91	204299
	1	0	3.94	32.91	51.22	290332
HateXplain	0	1	3.94	32.95	51.25	290772
	1	1	3.92	32.80	51.23	289395
	1	0	4.01	33.55	51.18	295781
TweetEval	0	1	4.01	33.55	51.18	295801
	1	1	4.01	33.53	51.18	295664
	1	0	3.97	33.16	51.19	292454
	0	1	3.97	33.18	51.21	292679
	1	1	3.96	33.11	51.22	292092

Table 9: Results on D4

Granularity	Cohen's Kappa	Agreement Percentage
All Separate toxicity	0.213	0.44
Toxic + Strongly Sensitive + Mildly Sensitive Vs Innocuous	0.344	0.684
(Toxic & Strongly as class 1), (Innocuous & Mildly as class 2)	0.473	0.739
1 Diff in sensitivity	0.691	0.778

Table 10: Inter annotator agreement scores for the Text predictor for prediction in context. We have calculated the IAA scores at different granularity. 1-Diff is the case when we consider adjacent toxicity values as similar as is usually the case when a subjective evaluation is performed. A high 1-Diff IAA denotes that the annotators mostly agree on the toxicity for the different queries.

Cohen's Kappa	
Severity	0.57
Factuality	0.72

Table 11: Template level IAA scores for IHCL1 dataset for severity and factuality annotations. The scores indicate a moderate to high agreement scores for the different labels.

Classifier	Sensitivity	Leakage Ratio
None	Mildly Sensitive	0.0698
HateXplain	Mildly Sensitive	0.0629
TweetEval	Mildly Sensitive	0.0605
perspective	Mildly Sensitive	0.0539
HateBert	Mildly Sensitive	0.0527
Blocklists	Mildly Sensitive	0.0208
None	Strongly Sensitive	0.0468
TweetEval	Strongly Sensitive	0.0448
perspective	Strongly Sensitive	0.0445
HateXplain	Strongly Sensitive	0.0438
HateBert	Strongly Sensitive	0.0431
Blocklists	Strongly Sensitive	0.0041
None	Toxic	0.1039
HateXplain	Toxic	0.0894
perspective	Toxic	0.0820
TweetEval	Toxic	0.0761
HateBert	Toxic	0.0494
Blocklists	Toxic	0.0139

Table 12: Leakage ratio wrt different sensitivities for each classifier (Pre and Post). The above values include cases which do not fall into the predefined dimensions as stated in the paper but are part of the checklist datasets. In each of the different scenarios we can see that Blocklists perform significantly better than other classifiers. HateBert comes up second and performs better than other classifiers.

A.2 Annotator Consent Form

Microsoft Research Project Participation Consent Form TITLE OF RESEARCH PROJECT: Responsible AI Data Creation and Annotation

Principal Investigator: Sunayana Sitaram
Co-Investigators: Monojit Choudhury

INTRODUCTION

Thank you for taking the time to consider volunteering in a Microsoft Corporation research project. This form explains what would happen if you joined this research project. Please read it carefully and take as much time as you need. Ask the study team about anything that is not clear. You can ask questions about the study any time.

Participation in this study is voluntary and you will not be penalized if you decide not to take part in the study or if you quit the study later.

PURPOSE

The purpose of this project is to reduce fairness and toxicity harms created by AI systems. We plan to collect data to evaluate current approaches to harm mitigation.

PROCEDURES

During this project, the following will happen: You will be asked to label templates for severity, toxicity and fine-grained category, given a template, lexicon and the coarse grained category for the template. You will also be provided with an example sentence constructed using the template populated with entries from the lexicon. We will give you approximately 100 templates to label and expect each one to take about one minute. You may complete the labeling on your own time over three days. The total amount of time spent should not exceed 120 minutes. Approximately 20 participants will be involved in this study.

PERSONAL INFORMATION AND CONFIDENTIALITY

- **Personal information we collect.** During the project we may collect personal information about you such as name, age, gender, languages known and proficiency in each language.

- **How we use personal information.** The personal information and other data collected during this project will be used primarily to perform research for purposes described in the Purpose and Procedures above. Such information and data, or the results of the research may eventually be used to develop and improve our commercial products, services or technologies.

- **How we store and share your personal information.** Your name and other personal information will not be on the study information we retain; this study information will be identified by a code. The key to the code will be kept separate from your personal and study information, which will be kept in a secured, limited access location.

Your personal information will be stored for a period of up to 5 years.

Some people may need to look at your personal information. They include: the researchers involved in this study, who may be Microsoft full time employees and fixed term employees, such as research interns. We will refer to these people as your Study Team. This also includes Institutional Review Boards (IRB), including Microsoft Research's ethics review board. An IRB is a group that reviews the study to protect your rights as a research participant.

We may choose to share publicly about this study, such as in journal articles, research-focused publications, or presentations at scientific meetings, but your identity will not be disclosed. We will take all steps possible to keep your information confidential. However, we cannot guarantee total confidentiality. For example, your personal information may be given out, if required by law.

- **How you can access and control your per-**

sonal information. If you wish to review or copy any personal information you provided during the study, or if you want us to delete or correct any such data, email your request to the research team at: susitara@microsoft.com. However, once your name or other identifiers have been removed from your information, we will no longer be able to delete it from our records.

For additional information or concerns about how Microsoft handles your personal information, please see the Microsoft Data Privacy Notice (<http://go.microsoft.com/fwlink/?LinkId=518021>).

MICROSOFT AND CONFIDENTIALITY

The research project and information you learn by participating in the project may be confidential to Microsoft. If the study team discloses confidential information, they will ask you to sign a separate, legally binding document called a Non-Disclosure Agreement (NDA) that asks you to promise to keep study information secret.

BENEFITS AND RISKS

Benefits:

There are no direct benefits to you that might reasonably be expected as a result of being in this study. The research team expects to learn how to build AI systems that are fairer and more inclusive from the results of this research. Furthermore, certain public benefits might be expected as a result of sharing the research results with the greater scientific community.

Risks:

During your participation, you may experience discomfort due to the sensitive nature of the data. Specifically, you will be shown sample sentences that could contain explicit, toxic, or potentially offensive terms. To help reduce such risks, you are free to skip the annotation of any template that makes you feel uncomfortable. You can also stop the annotation at any time and either exit the study or return to it after taking a break.

FUTURE USE OF YOUR IDENTIFIABLE INFORMATION

We may use your data in the future. Any data you contribute as part of this study will be stripped of any identifiers or other information that could be

used to identify you, as disclosed previously in this consent form. After such removal, the information could be used for future research studies or distributed to another investigator for future research studies without your (or your legally authorized representative's) additional informed consent.

PAYMENT FOR PARTICIPATION

You will not be paid to take part in this study. Your data may be used to make new products, tests, or findings. These may have value and may be developed and owned by Microsoft and/or others. If this happens, there are no plans to pay you.

PARTICIPATION

Taking part in research is always a choice. If you decide to be in the study, you can change your mind at any time without affecting any rights including payment to which you would otherwise be entitled. If you decide to withdraw, you should contact the person in charge of this study. The study team may use study data already collected from you, however, you may ask for it to be removed when you leave. Microsoft or the person in charge of this study may discontinue the study or your individual participation in the study at any time without your consent for reasons including:

- it is discovered that you do not meet study requirements
- the study is canceled
- administrative reasons

CONTACT INFORMATION

Should you have any questions concerning this project, or if you are injured as a result of being in this study, please contact; Sunayana Sitaram, at (Telephone Number removed for privacy) or susitara@microsoft.com (email). Should you have any questions about your rights as a research subject, please contact the Microsoft Research Ethics Review Program at MSRStudyfeedback@microsoft.com

CONSENT

By completing this form, you confirm that this study was explained to you, you had a chance to ask questions before beginning this study, and all your questions were answered satisfactorily. At any time, you may ask other questions. By completing

this form, you voluntarily consent to participate, and you do not give up any legal rights you have as a study participant.

Please confirm your consent by completing the bottom of this form. If you would like to keep a copy of this form, please print or save one now. On behalf of Microsoft, we thank you for your contribution and look forward to your research session.

Optional: Initial here if we may contact you in the future to request consent for uses of your identifiable data that are not covered in this consent form.

Initial here -----

Optional: Initial here if we may contact you in the future with information about follow-up or other future studies.

Initial here -----

Participant's Name -----

Date -----

Searching for Better Database Queries in the Outputs of Semantic Parsers

Anton Osokin*
HSE University
Yandex

Irina Saparina*
HSE University
Yandex

Ramil Yarullin*
HSE University
Yandex

Abstract

The task of generating a database query from a question in natural language suffers from ambiguity and insufficiently precise description of the goal. The problem is amplified when the system needs to generalize to databases unseen at training. In this paper, we consider the case when, at the test time, the system has access to an external criterion that evaluates the generated queries. The criterion can vary from checking that a query executes without errors to verifying the query on a set of tests. In this setting, we augment neural autoregressive models with a search algorithm that looks for a query satisfying the criterion. We apply our approach to the state-of-the-art semantic parsers and report that it allows us to find many queries passing all the tests on different datasets.

1 Introduction

Generating a database query from a natural-language description of the user’s intent is a long-standing and important task. In the recent years, most of the community focus was on the Spider dataset (Yu et al., 2018), which poses the task in the zero-shot regime, meaning that a method has to generalize to databases unseen at training. The Spider dataset contains English questions and SQL queries. The progress has been remarkable, and the accuracy has moved from below 30% to above 70%. A part of this success can be attributed to the adoption of pre-trained transformer models like BERT (Devlin et al., 2019) into most of the pipelines.

Given such progress, it is natural to ask whether we are getting closer to solving the problem. Several recent studies have noted that the task might be harder than it looks. Finegan-Dollak et al. (2018) found that many single-database datasets had identical queries in both train and test sets and showed that using such splits effectively reduced the problem to classifying the queries from the train set.

Shaw et al. (2021) continued the study in the multi-database setting and showed that the compositional generalization was hard to achieve, and even to measure it, one should be very careful with splits. In a different line of thought, Suhr et al. (2020) examined how the models trained on Spider generalize to other datasets and reported that generalization was challenging. Even within one dataset, many questions have several interpretations leading to different queries, and annotation policies do not cover these ambiguities or cover them differently.

Acknowledging that the zero-shot setting might be too difficult to tackle as is, we aim to better define and simplify the problem to achieve better results in terms of the number of correctly generated queries. In this task, most modern models produce a distribution over all possible outputs, which can guide the search at the test time.

We observe that if the search algorithm has access to a criterion that can evaluate the output by treating it as a database query, the overall method can produce much better results. We consider the following criteria ordered by their “strength”: a query is executed without errors, a query produces output from correct columns, a query produces a correct result on one test database, and a query produces correct results on a set of test databases. We experiment with different search methods and report that the complete anytime beam search (Zhang, 1998) outperforms sampling-based alternatives.

Many practical cases arise when the user is willing to trade off some of their time to improve the output query. Our approach allows the user to obtain a better query by interactively guiding the search via providing the target output columns or the answer on one or more test databases. The user is expected to supply this information without the gold query. Notably, the execution criterion does not require extra user input but relies on executing generated queries.

In addition, the test time database can be out of

* Equal contribution.

domain w.r.t. the training set. One can annotate gold queries for fine-tuning on new databases to improve out-of-domain performance, but this can be prohibitively expensive. Our approach gives a way to improve out-of-domain results without extra annotated training data. We view our approach as a way to control the trained model leading to a more reliable and responsible query synthesis.

For our studies, we used three state-of-the-art models in terms of execution accuracy (with publicly available implementations): T5-3B (Raffel et al., 2020) fine-tuned to Spider by Scholak et al. (2021), BRIDGE (Lin et al., 2020) and SQ-QDMR (Saparina and Osokin, 2021).

In this paper, we make several observations. The complete anytime beam search works with different search criteria and with all the three models. Reasonable results can be obtained with the maximum beam size of 100, which fits on a single modern GPU. Searching with the execution criterion can significantly improve the quality of the decoders that generate output as an unconstrained token sequence, e.g., T5, and using such criterion does not require extra user input. Searching for the queries that return a correct output on one database allows finding many queries that provide a correct output on that database. Therefore, such searching w.r.t. one database can result in false positives, and it is important to evaluate the queries on a set of databases. Based on the method of Zhong et al. (2020), we built the test suite of databases for the evaluation. With these test suites, we show that there are multiple false positives among the queries that pass one test, while searching for the queries that pass the test suite produces the outputs of higher quality.

Finally, we experiment with the generalization of the models trained on Spider to the Geo-Query (Zelle and Mooney, 1996), IMDB, Yelp (Yaghmazadeh et al., 2017) and Academic (Li and Jagadish, 2014) datasets. We show that searching w.r.t. different criteria still works under this distribution shift, and searching w.r.t. the criteria with tests is often comparable to fine-tuning the network to a test dataset directly.

This paper is organized as follows. In Section 2, we review our setting. In Section 3, we provide the details of our method. Section 4 provides the details of the test-suite construction procedure, Section 5 describes the experimental setup. In Section 6, we provide the experimental results and

discussion. We review some related works in Section 7 and conclude in Section 8.

2 Preliminaries

We consider the problem of generating queries to databases given the description of the user’s intent in natural language in the cross-database setting where the train, validation and test splits contain different databases. Models trained in this setting, in theory, can be evaluated on any database.

A typical model for the cross-database setting is an encoder-decoder neural network. Encoders typically consist of a pre-trained BERT-like transformer followed by a specialized encoder that can incorporate the database structure in some form (Guo et al., 2019; Wang et al., 2020; Cao et al., 2021; Cai et al., 2021). Sometimes the BERT part is further fine-tuned on database-related objectives (Yu et al., 2021; Deng et al., 2021). The encoder input is a concatenation of the tokenized question and a sentence representation of the database schema separated by the special delimiter token. The representation of the database schema consists of the tokenized table and column names and values related to the question. These values are commonly extracted by string matching with question tokens (Lin et al., 2020). Decoders are typically autoregressive based on LSTM or transformers. Some decoders do not check the syntactic correctness of the output and its consistency with the database. Some provide output w.r.t. a grammar (Yin and Neubig, 2017); some use post-hoc checks with parsers.

In this paper, we experiment with three models: T5-3B fine-tuned on the Spider dataset by Scholak et al. (2021), BRIDGE (Lin et al., 2020) and SQ-QDMR (Saparina and Osokin, 2021). We provide a detailed description of these models in Section 5.3.

3 Search with Models

We now describe our approach to searching for queries on top of a learned model. We first generate full query candidates using a search method and then select the first one that passes the selected search criterion. We show possible search criteria in Section 3.1 and search methods in Section 3.2.

3.1 Search Criteria

Execution criterion. To avoid syntactically incorrect queries, we can prune the search with the execution criterion. The query passes this criterion

if it can be executed on the input database without errors. In particular, the query has to contain valid table and column names. These properties are not guaranteed for the unconstrained decoders as T5. Thus the execution criterion can be extremely useful for such models.

Output column match. With this criterion, we compare the output columns (that the query will select) with the correct ones. Firstly, wrong output columns is a common mistake in the text-to-SQL parsers (Guo et al., 2019; Lin et al., 2020; Suhr et al., 2020). Secondly, the output columns can provide domain knowledge and shed some light on the user intent in a realistic scenario when the input question is ambiguous (Suhr et al., 2020; Lee et al., 2021).

One test. This criterion compares the result of query execution on a given database (one test case) with the correct one. With this criterion, we search for a correct query in terms of execution accuracy on the input database (Section 5.2).

Test suite. This criterion checks if a query passes a set of tests. Each test case corresponds to a particular database, and all test databases share the same schema. This criterion is inspired by a test suite of databases with high code coverage proposed by Zhong et al. (2020). The set of databases is designed to distinguish the correct queries from potential false positives. Searching with this criterion is equivalent to the search for a correct query in terms of test-suite accuracy (Section 5.2).

3.2 Search Methods

Top- k and Top- p (Nucleus) Sampling (Fan et al., 2018; Holtzman et al., 2020) draw samples from the truncated distribution: the probability mass is re-weighted between the k most probable elements in top- k sampling and between the elements with cumulative probability mass exceeding p in top- p sampling (k, p are hyperparameters).

UniqueRandomizer (Shi et al., 2020) is a method to incrementally sample sequences without replacement. The samples are drawn until the stopping condition is reached (one of the search criteria in our case). The probabilities of selected elements are reduced after each iteration of sampling to improve diversity in samples.

Complete Anytime Beam (CAB) Search of Zhang (1998) extends the regular beam search by running it several times with increasing beam sizes. Importantly, the beams produced by beam search

Table 1: Comparison of the test suites’ statistics. **NoEmpty** is the percentage of SQL queries for which at least one test database with non-empty execution result is found; **Cover** is the percentage of neighbor queries distinguished by the test databases; **Tests** is the average number of test databases per query; **Time** is the average wall-clock execution time per query; **Size** is the total size of all test databases.

Dataset		NoEmpty (%) ↑	Cover (%) ↑	Tests (Num)	Time (Sec) ↓	Size (GB) ↓
Spider	1 test	96.7	89.1	1	0.1	3.25
	Orig.	98.2	96.1	675	3.6	0.16
	Our	99.3	98.8	1.8	0.2	0.04
GeoQuery	1 test	94.7	93.2	1	0.1	10⁻⁴
	Orig.	66.7	52.6	108	12.6	0.01
	Our	100	98.9	1.6	0.2	0.05
IMDB	1 test	75.2	14.9	1	8.7	1.49
	Orig.	79.2	79.4	200	17.7	0.04
	Our	100	99.6	2	0.3	0.03
Yelp	1 test	36.5	15.8	1	6.4	2.21
	Orig.	84.1	80.6	282	30.9	0.03
	Our	99.2	97.0	3.1	0.4	0.03
Academic	1 test	51.1	5.50	1	44.2	4.33
	Orig.	97.9	92.1	411	44.3	0.07
	Our	96.8	95.2	2.3	0.4	0.07

are known to have little diversity because of the peaks in the softmax scores. We follow the approach of Zohar and Wolf (2018); Shrivastava et al. (2021), who recently have used CAB to search for programs on top of neural autoregressive models. In these works, the authors limit the number of hypotheses coming from each element of the previous beam (we will refer to this upper bound as the width of the beam search). Between outer CAB iterations, we also increase the width by a constant value and multiply the beam size by a constant factor. The schedules of the beam size and beam-search width are important hyperparameters.

4 Test Suite Construction

Testing on one database is generally not enough to ensure the semantic correctness of the generated query, but running the query on too many databases can be computationally inefficient. The inefficiency problem is especially acute in our task due to the large number of query candidates that should be tested and several rounds of the searching process.

We build our test suites by modifying the method of Zhong et al. (2020), which relies on generating the so-called neighbor queries from a given set of gold queries and randomly sampling databases to distinguish gold queries from as many neighbors

as possible. We observed two key drawbacks of the test databases generated by [Zhong et al. \(2020\)](#). First, the test suites contained many databases, and some were unnecessarily large, which resulted in very long testing on them. Second, outputs of many queries were often empty (or zero for queries with aggregators) on these test databases. If the output of the gold query is empty on all the elements of the test suite, it cannot be distinguished from trivial dummy queries. This effect is more salient if the gold query returns empty output on the original database. We alleviate these issues by independently generating test databases for each gold query, explicitly limiting the number of rows in each table and putting extra effort into generating at least one database where the gold query returns a non-empty output. The details of our procedure are provided in [Appendix A](#).

We compare our test suites with the original ones of [Zhong et al. \(2020\)](#) on the five considered datasets described in [Section 5.1](#). For a fair comparison, we generate the independent sets of neighbor queries for each gold query. These neighbor queries are different from the neighbors generated in the process of creating the test suites. In [Table 1](#), we compare the initial databases (1 test), original test suites and our test suites. It can be seen that, in the space of neighbor queries, our test suites have higher code coverage (Cover) than the original databases and the original test suites. With new test suites, the average query execution time (on all the corresponding tests) is reduced 50x across the datasets. We release the test suites we created.¹

5 Experiment Setup

5.1 Data

We conduct our experiments on five text-to-SQL datasets: multi-database Spider ([Yu et al., 2018](#)) and single-database GeoQuery ([Zelle and Mooney, 1996](#); [Iyer et al., 2017](#); [Finegan-Dollak et al., 2018](#)), IMDB, Yelp ([Yaghmazadeh et al., 2017](#)) and Academic ([Li and Jagadish, 2014](#)).

Spider. We use dev and test sets (451 and 521 examples) from the work of [Saparina and Osokin \(2021\)](#): they are parts of the original Spider dev, but some examples (from dev) were repaired.

GeoQuery, IMDB, Yelp and Academic. We use query splits created by [Finegan-Dollak et al. \(2018\)](#), additionally filtered from duplicates and

examples with gold SQL queries that crash or execute longer than 5 minutes with the Python package `sqlite3`. The dataset statistics are provided in [Table 7](#) of [Appendix D](#).

We do not consider more single-database datasets because ATIS ([Price, 1990](#); [Dahl et al., 1994](#)), Scholar ([Iyer et al., 2017](#)) and Advising ([Finegan-Dollak et al., 2018](#)) database schemes exceed 512 token limits of pre-trained encoders, Restaurants ([Popescua et al., 2003](#); [Tang and Mooney, 2001](#)) contains too many duplicates.

5.2 Evaluation Metrics

Exact-set Match ([Yu et al., 2018](#)) is an SQL-to-SQL comparison metric that reflects the fraction of the predicted queries matching the ground-truth queries. In the matching process, each query is decomposed into fragments that are compared individually so that the metric is not too sensitive to the ordering of independent clauses. This metric does not take into account predicted values and can give a high score to incomplete queries. As SQ-QDMR model produces queries in SPARQL, we cannot use the exact-set match as a primary evaluation metric.

Execution accuracy is designed to compare the queries by their execution output on an original database. In contrast to the exact-set match, this prevents false-negative queries but leaves space for potential false positives. The version provided by [Yu et al. \(2018\)](#) for Spider evaluation has issues in SPARQL-SQL comparison, so we use the version provided by [Saparina and Osokin \(2021\)](#) unless explicitly mentioned otherwise.

Test-suite accuracy ([Zhong et al., 2020](#)) approximates the semantic accuracy of the query synthesis models. This metric refers to the share of predicted queries producing the correct answers on all databases from the test suite. We build the test suites for Spider dev and test sets and for all the queries in the other four considered datasets.

5.3 Models

We consider three models: T5-3B fine-tuned on Spider ([Scholak et al., 2021](#)), BRIDGE ([Lin et al., 2020](#)) and SQ-QDMR ([Saparina and Osokin, 2021](#)). These models have top execution accuracy among publicly available models on Spider. We also tried to search under our search criteria on top of the bottom-up semi-autoregressive model of [Rubin and Berant \(2021\)](#), but we could not make the search increase the number of correct queries.

¹github.com/ramild/TestSuite

For evaluation on Spider, we use the released checkpoints of the best models. BRIDGE training data included question splits of single-database datasets, so we re-train it on Spider-only data to evaluate on query splits of these datasets. For re-training BRIDGE and fine-tuning all models, we use official implementations (see Appendix E).

T5 (Raffel et al., 2020) is a pre-trained seq2seq model based on Transformer. Recently, Shaw et al. (2021); Scholak et al. (2021) successfully applied T5 for the text-to-SQL task. The input sequence contains question tokens and tokens of column and table names. The database values matched with the question tokens are appended to the corresponding column names (Lin et al., 2020). The output of the T5 model is the sequence of tokens representing the SQL query. Note that this model generates output sequence without explicitly considering the SQL grammar and schema consistency.

BRIDGE (Lin et al., 2020) consists of the BERT-based encoder and pointer-generator decoder. The input sequence is formed from the concatenation of question, table and column names, and relevant database values separated by special token and encoded with BERT. The relevant values are selected with fuzzy string matching between question and database values. Column encodings are further enriched with meta-data features such as primary or foreign keys and data types obtained from the feed-forward layer. The LSTM-based decoder with multi-head attention at each step copies question or schema tokens or generates the SQL keywords. During decoding, model chooses columns only from the predicted table to provide schema consistency. An additional static SQL analyzer filters incorrect output queries.

SQ-QDMR (as we refer to the model of Saporina and Osokin (2021)) contains RAT-transformer, GraPPa encoders (Wang et al., 2020; Yu et al., 2021) and a grammar-guided LSTM-based decoder (Yin and Neubig, 2017). The SQ-QDMR decoder produces output in the form of grounded intermediate representations derived from QDMRs of the Break dataset (Wolfson et al., 2020). The grounded QDMRs are not directly related to any execution engine and cannot be executed as is, but Saporina and Osokin (2021) implemented a non-trainable translator from grounded QDMR to the SPARQL query language, in which queries can be executed. We can think of grounded QDMRs augmented with this translator as executable database queries.

Table 2: Comparing the execution accuracy of different search approaches under the 1-test criterion on the Spider dev split.

Model	Greedy	CAB	Sample	Top-p	UniRand
T5-3B	77.0	94.4	93.0	93.0	94.1
BRIDGE	66.8	91.0	87.1	84.9	86.2
SQ-QDMR	80.4	98.0	98.0	98.0	98.2

6 Results & Analysis

6.1 Impact of the Search Methods

We compare different decoding strategies in our setting (Table 2): top- k and top- p (nucleus) sampling (Fan et al., 2018; Holtzman et al., 2020), UniqueRandomizer (Shi et al., 2020) and CAB search (Zhang, 1998). We measure the execution accuracy of these search methods under the 1-test criterion on Spider dev. We use the same sampling budgets (1000 for BRIDGE and SQ-QDMR and 800 for T5-3B due to the memory limits), tune p in top- p sampling and the temperature for all methods, more implementation details in Appendix B).

The results demonstrate that a significant number of output queries pass one test after searching with any of these methods, so different decoding strategies can be compatible with our approach. UniqueRandomizer is very time-consuming since it generates samples sequentially in contrast to other methods that generate beams of samples in parallel. CAB search is demanding in terms of the device memory as it has to process the whole beam jointly. For further experiments, we choose CAB search because it works best for two models.

6.2 Impact of the Search Criteria

We apply search under different selection criteria (execution, output column match, test on one database) to T5-3B, BRIDGE and SQ-QDMR on Spider dataset and compare with the greedy and beam search baselines. Table 3 shows the results measured with execution accuracy. **Searching on top of all models with different selection criteria increases execution accuracy in almost all cases.**

One exception is the search with the execution criterion on top of BRIDGE and SQ-QDMR, the results of which are close to the greedy decoding. The outputs of these systems are almost always executable because BRIDGE runs a static SQL analyzer for filtering, SQ-QDMR decodes according to the QDMR grammar and both models have schema-consistent decoding. The T5 model, in contrast, does not have any grammar or schema

Table 3: Execution accuracy of search under different selection criteria (execution, output column match and 1 test) on Spider; beam s. refers to beam search.

Model	Split	Greedy	Beam S.	Exec	Cols	1 Test
T5-3B		77.0	77.4	83.1	84.2	94.4
BRIDGE	dev	66.8	68.8	68.4	72.0	91.0
SQ-QDMR		80.4	80.6	80.4	83.6	98.0
T5-3B		70.8	71.0	73.7	77.4	90.7
BRIDGE	test	64.0	66.2	64.6	67.9	83.4
SQ-QDMR		65.6	65.8	65.6	68.9	86.7

constraints in decoding. **For language model decoders that do not explicitly check grammar and schema consistency, the execution criterion can significantly improve the quality.**

For T5-3B, we also compare the results of the search with PICARD (Scholak et al., 2021), our search with the execution criterion, and greedy decoding. For a fair comparison with Scholak et al. (2021), we use the same data, the official Spider dev set, and the same metrics: exact-set matching accuracy and execution accuracy provided by Yu et al. (2018):

Method	EM	Exec
Greedy	71.5	74.4
PICARD	75.5	79.3
CAB+execution	74.5	78.7

The results obtained with PICARD and with searching under the execution criterion are comparable – our search gets most of the benefit over the baseline. PICARD provides slightly better quality and works with smaller beams but requires more effort to incorporate because it is tightly connected with the decoder output vocabulary and grammar. For both approaches, the percent of output queries with execution errors is around 2% in contrast to the baseline T5-3B decoding with 12%.

The search criterion based on the matching of output columns provides even better results. As Table 3 shows, all models benefit from this criterion: execution accuracy increases by 3-4% nearly everywhere. This criterion largely simplifies the task with extra information at the test time.

Search for the queries that pass one test allows finding a significant number of such queries. Passing one test means correct execution result on the input database, so all the queries found with this criterion are correct in terms of the execution accuracy. Thus, these results indicate that our searching approach works, and we can find the correct queries with the corresponding criterion. However, we ob-

Table 4: Test-suite accuracy of search under different selection criteria (execution, output column match, 1 test and test suite) on Spider.

Model	Split	Greedy	Exec	Cols	1 Test	Test Suite
T5-3B		72.2	77.7	78.3	86.9	90.1
BRIDGE	dev	63.2	64.8	68.2	81.9	84.0
SQ-QDMR		72.5	72.2	75.4	84.0	94.4
T5-3B		68.9	73.7	75.4	84.7	86.8
BRIDGE	test	61.9	62.5	65.4	76.6	77.8
SQ-QDMR		62.3	62.1	65.4	75.2	84.1

serve that more than 10% of queries found with one-test criterion are false-positive according to the test-suite accuracy (Table 6 in Appendix C).

These results motivate us to evaluate the test-suite accuracy of criterion-guided search. Table 4 confirms our findings: searching with the execution criterion helps T5-3B, and searching for correct output columns improves the results of all models. **Search for the queries that pass one test results in a significant number of false-positive queries. The correct queries can be found by searching with the test-suite criterion directly.**

6.3 Efficiency

Time Measurements. The running time during the search is dominated by the time of the decoder for all three models: executing each considered query takes 3% of the decoder time for T5-3B, which is 0.01 sec per run; 53%, 0.02 sec – for BRIDGE; 72%, 0.03 sec – for SQ-QDMR. The total running time depends on the effective beam size used during the search.

The T5-3B model with the execution criterion on top runs in 1.7 sec compared to 3.1 sec reported by the PICARD paper, where both systems were run on 1 NVIDIA A100 GPU. The main reason is that due to CAB, we do not set one beam size in advance and thus, process at least 70% of examples with an effective beam size of 1.

Impact of the Maximum Size of Beam. The maximum size of the beam is an important parameter. Figure 1 shows the dependence between the obtained test-suite accuracy on the Spider dev set and the maximum beam size in the search under the test suite criterion. For all models, we start with the maximum beam size equal to 1, which is equivalent to the greedy decoding and finish with the maximum beam size allowed by our implementation and hardware: 10k for BRIDGE and SQ-QDMR on 1 NVIDIA V100 GPU and 800 for T5-3B on 8 NVIDIA A100 GPUs.

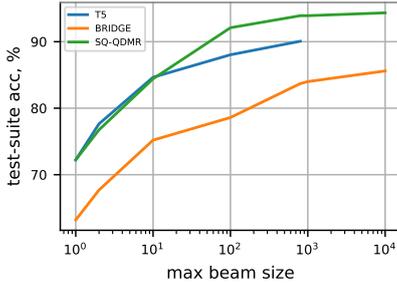


Figure 1: Test-suite accuracy on the dev set for the T5-3B, BRIDGE and SQ-QDMR models tested with CAB for the test-suite criterion.

Test-suite accuracy improves as the maximum beam size increases. BRIDGE with the maximum beam size equal to 10k achieves 86% of test-suite accuracy and SQ-QDMR — 94%. However, the search works well enough even with smaller beams: with the maximum beam size of 100, BRIDGE achieves almost 80%, SQ-QDMR — 92%, and T5-3B achieves 88% (with 800, T5-3B achieves 90%). Importantly, the search with the beam size of 100 does not require multiple GPUs for T5-3B.

6.4 Experiments on Single-Database Data

To show more benefits of searching under selection criteria, we evaluate it on single-database datasets, GeoQuery, IMDB, Yelp and Academic, with test-suite accuracy (Table 5; see Appendix F for execution accuracy). We use query splits of Finegan-Dollak et al. (2018) and two model types: trained on Spider only and fine-tuned on a particular dataset. The Academic database is very large, so we cannot evaluate one-test criterion on this dataset and fine-tune SQ-QDMR (other datasets do not have QDMR annotation required for fine-tuning). More fine-tuning details are in Appendix E.

The results show that models trained on Spider struggle to generalize to other datasets, which is consistent with the findings of Suhr et al. (2020). **More information about the problem (in the form of additional train data or selection criteria) helps improve the quality.**

IMDB, Yelp and Academic are more challenging datasets for cross-database semantic parsers than GeoQuery, but they are significantly smaller (Table 7), and the models are less stable while testing on them (with all random seeds fixed). Stronger criteria, such as passing one test or test suite, do work even with datasets of such difficulty, when weaker criteria fail. On GeoQuery, the search under the one-test and test-suite criteria leads to even better

Table 5: Different search criteria (execution, output column match, 1 test and test suite) on top of pre-trained models on the query test splits of different datasets with the **test-suite accuracy**.

Dataset (test size)	Model	Greedy	Exec	Cols	1 test	Test Suite
GeoQuery (182)	T5-3B	55.5	56.6	63.7	67.6	72.5
	+ fine-tune	64.3	70.9	85.2	88.5	94.5
	BRIDGE	55.9	50	62.6	74.7	74.7
	+ fine-tune	65.4	66.5	81.9	86.8	91.8
	SQ-QDMR	37.4	37.4	41.8	65.9	81.9
	+ fine-tune	56.0	56.0	59.9	76.9	83.0
IMDB (17)	T5-3B	5.9	11.8	17.6	29.4	41.2
	+ fine-tune	52.9	52.9	52.9	52.9	58.8
	BRIDGE	11.8	11.8	17.6	17.6	17.6
	+ fine-tune	52.9	52.9	52.9	52.9	52.9
Yelp (10)	T5-3B	20	20	10	30	10
	+ fine-tune	20	30	20	50	40
	BRIDGE	0	0	0	20	10
	+ fine-tune	30	40	50	60	70
Academic (15)	T5-3B	6.7	13.3	26.7	-	33.3
	+ fine-tune	60	53.3	53.3	-	80
	BRIDGE	0	0	6.7	-	20
	+ fine-tune	33	40	40	-	80
	SQ-QDMR	13.3	13.3	6.7	-	66.7

quality than fine-tuning. Our test-suite criterion is especially useful when one test is difficult to run on the large original database, e.g., on Academic.

As a result, we conclude that criterion-guided search on top of a pre-trained model is a good alternative to fine-tuning in cases when training data is not available, but the user is ready to provide more information on each test question.

7 Related Works

Search for Database queries. The task of translating NL questions into database queries implies the ability to query databases with natural language. To ensure this, it is essential to generate syntactically correct queries that refer to valid table and column names for the given database schema. Wang et al. (2019) noticed that a partially decoded SQL query can be executed, and thus, the result of this execution can guide the decoding process. At each decoding step, partial queries that crush or give an empty result during the execution are removed from beams. In this work, we also consider the execution criterion of search but apply it to the finished hypotheses, which allows us to search on top of the models with different output formats, including

intermediate representations.

Lin et al. (2020) generated SQL queries in the execution order to keep the consistency between the predicted database entities and checked output correctness with the static SQL parser. Suhr et al. (2020) executed the top-10 generated queries in beam search to filter the inexecutable ones, which is close to checking the execution criterion in our work but differs by the search method.

Task-specific decoders such as autoregressive grammar-based (Yin and Neubig, 2017; Lin et al., 2018) and tree decoders (Dong and Lapata, 2016), semi-autoregressive decoder (Rubin and Berant, 2021) provide some guarantees as they control the output structure. However, as noticed by Scholak et al. (2021), these decoding methods are incompatible with pre-trained decoders of language models. These pre-trained decoders, like the one of T5, can also be successfully applied to the text-to-SQL task (Shaw et al., 2021). Scholak et al. (2021) proposed to check hypotheses in beam search on the lexical and grammatical levels at each step of the beam search. However, compared to us, their approach required heuristics to prune incomplete queries.

The concurrent work by Wolfson et al. (2022) uses several components similar to ours but in a very different way. They use the QDMRs of Wolfson et al. (2020) with textual arguments as a form of weak supervision to generate SQL queries for the training set. Their synthesis process results in many candidate SQL queries and relies on tests to select the one as an annotation. Such a process is similar to the method of Saparina and Osokin (2021) for constructing groundings of QDMR arguments. However, the search process of Wolfson et al. (2022) is not connected to any neural model and is not used at the test time.

Search for Programs with Neural Networks.

Our approach to searching for queries is closely related to the field of program synthesis if we interpret queries as programs. Recently, neural networks have been applied in a wide range of program synthesis tasks, see the excellent work of Chaudhuri et al. (2021) for a recent review.

When programs are synthesized from large language models generating multiple outputs, selecting the one that, e.g., passes some or all tests is a common practice. For example, the Codex model (Chen et al., 2021) for synthesizing Python code includes some sample tests into the input prompt to give the model more information to de-

fine the user intent. Chen et al. (2021) following Kulal et al. (2019), among others, also uses the pass@k metric, which effectively means that the model generates k outputs, and the best ones are selected based on tests. The pass@k metric can be interpreted as test-suite accuracy after search w.r.t. the tests with the beam of size k .

Overall, it is widely accepted that tests are useful to precisely define the user intent. However, they are hard to collect at a large scale, especially when coupled with a description in natural language. Because of this, large-scale benchmarks related to code, e.g., CodeXGLUE (Lu et al., 2021), primarily used text-based metrics like BLEU. The attempts to specialize BLEU to code by combining it with abstract syntax trees extracted from code, like CodeBLEU (Ren et al., 2020), are in some sense similar to the SQL-based exact match metric of the Spider dataset (Yu et al., 2018).

Approaches to Simplify the Cross-database Setting. The community has made multiple attempts to modify the cross-domain setting to make the problem easier to solve. Yu et al. (2019b) collected the SParC dataset with coherent question sequences, which can allow sharing of information between the sequences. Yu et al. (2019a) collected the CoSQL dataset with an interactive conversational setting with SQL queries, making it possible to explore user interactions with the system. Lee et al. (2021) collected KaggleDBQA with database documentation in the form of textual description for database columns that can potentially allow language models to provide outputs better corresponding to the user’s intent. Our approach provides users with a way to interact with the model by supplying tests. Given the initial question in natural language, the users can provide extra tests until they are satisfied with the generated query.

8 Conclusion

We studied the search over the outputs of the neural autoregressive models for better database query generation. We considered three state-of-the-art models: T5-3B, BRIDGE and SQ-QDMR. We observed that the search algorithms work with multiple criteria for selecting the output query. We also compared the search-augmented methods with the fine-tuned models on the GeoQuery, IMDB, Yelp and Academic datasets (under the distribution shift) and observed that the method with search can sometimes work even better than fine-tuning. Com-

pared to fine-tuning, the search-based method does not require additional training data but relies on additional information on each test example. With such properties, our search based methods might be helpful for use cases like interactive query generation or annotating new datasets.

Limitations

In our experiments, we work only with the datasets where the user question was written in English. This might have simplified the task for T5 as the keywords and entity names of the query languages were also in English.

The test suites we built were still not perfect. In particular, it was hard to generate a test database such that the query `SELECT year FROM concert GROUP BY year HAVING count(*) >= 50` had non-empty output because it needed at least 50 rows with the same value in the column `year`. We also noticed that the value 1 as the gold query output also caused many false positives and should probably have been considered the empty value for the queries outputting the count aggregator.

The results of the search methods were also not perfect for out-of-domain data, even with strong search criteria based on tests. One reason for that was that we started to hit the limitations of the models, which were built with mostly Spider in mind. In particular, the database preprocessing stage to select values for the query was, in some cases, slow and inaccurate.

References

- Ruichu Cai, Jinjie Yuan, Boyan Xu, and Zhifeng Hao. 2021. SADGA: Structure-aware dual graph aggregation network for Text-to-SQL. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*.
- Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu, and Kai Yu. 2021. LGESQL: Line graph enhanced text-to-SQL model with mixed local and non-local relations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2541–2555, Online. Association for Computational Linguistics.
- Swarat Chaudhuri, Kevin Ellis, Oleksandr Polozov, Rishabh Singh, Armando Solar-Lezama, and Yisong Yue. 2021. Neurosymbolic programming. *Foundations and Trends in Programming Languages*, 7(3).
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. [Evaluating large language models trained on code](#). *arXiv*, 2107.03374v2.
- Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the atis task: The atis-3 corpus. In *In Proceedings of the Workshop on Human Language Technology of Association for Computational Linguistics*.
- Xiang Deng, Ahmed Hassan Awadallah, Christopher Meek, Oleksandr Polozov, Huan Sun, and Matthew Richardson. 2021. [Structure-grounded pretraining for text-to-SQL](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1337–1350, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Li Dong and Mirella Lapata. 2016. [Language to logical form with neural attention](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui

- Zhang, and Dragomir Radev. 2018. [Improving text-to-SQL evaluation methodology](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.
- Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. [Towards complex text-to-SQL in cross-domain database with intermediate representation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4524–4535, Florence, Italy. Association for Computational Linguistics.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text degeneration](#). In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. [Learning a neural semantic parser from user feedback](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 963–973, Vancouver, Canada. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy Liang. 2019. SPoC: Search-based pseudocode to code. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*.
- Chia-Hsuan Lee, Oleksandr Polozov, and Matthew Richardson. 2021. [KaggleDBQA: Realistic evaluation of text-to-SQL parsers](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2261–2273, Online. Association for Computational Linguistics.
- Fei Li and H. V. Jagadish. 2014. [Constructing an interactive natural language interface for relational databases](#). *Proc. VLDB Endow.*
- Kevin Lin, Ben Bogin, Mark Neumann, Jonathan Berant, and Matt Gardner. 2018. [Grammar-based neural text-to-sql generation](#). *arXiv*, 1905.13326.
- Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2020. [Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4870–4888, Online. Association for Computational Linguistics.
- Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin B. Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. 2021. [CodeXGLUE: A machine learning benchmark dataset for code understanding and generation](#). *arXiv*, 2102.04664v2.
- Ana-Maria Popescua, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the International Conference on Intelligent User Interfaces*.
- P. J. Price. 1990. Evaluation of spoken language systems: the atis domain. In *Speech and Natural Language: Proceedings of a Workshop*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Shuo Ren, Daya Guo, Shuai Lu, Long Zhou, Shujie Liu, Duyu Tang, Neel Sundaresan, Ming Zhou, Ambrosio Blanco, and Shuai Ma. 2020. [CodeBLEU: a method for automatic evaluation of code synthesis](#). *arXiv*, 2009.10297v2.
- Ohad Rubin and Jonathan Berant. 2021. [SmBoP: Semi-autoregressive bottom-up semantic parsing](#). In *Proceedings of the 5th Workshop on Structured Prediction for NLP (SPNLP 2021)*, pages 12–21, Online. Association for Computational Linguistics.
- Irina Saparina and Anton Osokin. 2021. [SPARQLing database queries from intermediate question decompositions](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8984–8998, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. [PICARD: Parsing incrementally for constrained auto-regressive decoding from language models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. [Compositional generalization and natural language variation: Can a semantic parsing approach handle both?](#) In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938, Online. Association for Computational Linguistics.
- Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the International Conference on Machine Learning (ICML)*.

- Kensen Shi, David Bieber, and Charles Sutton. 2020. [Incremental sampling without replacement for sequence models](#). In *Proceedings of the International Conference on Machine Learning (ICML)*.
- Disha Shrivastava, Hugo Larochelle, and Daniel Tarlow. 2021. Learning to combine per-example solutions for neural program synthesis. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*.
- Alane Suhr, Ming-Wei Chang, Peter Shaw, and Kenton Lee. 2020. [Exploring unexplored generalization challenges for cross-database semantic parsing](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8372–8388, Online. Association for Computational Linguistics.
- Lappoon R Tang and Raymond J Mooney. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Proceedings of the European Conference on Machine Learning*.
- Bailin Wang, Richard Shin, Xiaodong Liu, Aleksandr Polozov, and Matthew Richardson. 2020. [RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.
- Chenglong Wang, Kedar Tatwawadi, Marc Brockschmidt, Po-Sen Huang, Yi Mao, Aleksandr Polozov, and Rishabh Singh. 2019. [Robust text-to-SQL generation with execution-guided decoding](#). *arXiv*, 1807.03100.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Tomer Wolfson, Daniel Deutch, and Jonathan Berant. 2022. Weakly supervised text-to-sql parsing through question decomposition. In *Findings of the Association for Computational Linguistics: NAACL 2022*.
- Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. [Break it down: A question understanding benchmark](#). *Transactions of the Association for Computational Linguistics*, 8:183–198.
- Navid Yaghmazadeh, Yuepeng Wang, Isil Dillig, and Thomas Dillig. 2017. Sqlizer: Query synthesis from natural language. *Proc. ACM Program. Lang.*
- Pengcheng Yin and Graham Neubig. 2017. [A syntactic neural model for general-purpose code generation](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450, Vancouver, Canada. Association for Computational Linguistics.
- Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2021. [GraPPa: Grammar-augmented pre-training for table semantic parsing](#). In *Proceedings of International Conference on Learning Representations (ICLR)*.
- Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. 2019a. [CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1962–1979, Hong Kong, China. Association for Computational Linguistics.
- Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. [Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.
- Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. 2019b. [SPaC: Cross-domain semantic parsing in context](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4511–4523, Florence, Italy. Association for Computational Linguistics.
- John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence (AAAI)*.
- Weixiong Zhang. 1998. Complete anytime beam search. In *Proceedings of the national conference on artificial intelligence (AAAI)*.
- Ruiqi Zhong, Tao Yu, and Dan Klein. 2020. [Semantic evaluation for text-to-SQL with distilled test suites](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 396–411, Online. Association for Computational Linguistics.

Amit Zohar and Lior Wolf. 2018. Automatic program synthesis of long programs with a learned garbage collector. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*.

A Construction of Test Suites

The method of Zhong et al. (2020) relies on generating the so-called neighbor queries from a given set of gold queries Q . A set of neighbors N_q is obtained through slightly changing the original query $q \in Q$. Now, a database w distinguishes q and $g \in N_q$ if their execution results on w are different. The suite S is formed greedily: a new sampled database is added to S if it distinguishes a pair in $\mathcal{N} = \{(q, g) \mid q \in Q, g \in N_q\}$ that is not distinguished by any database previously added to S .

In the algorithm above, to sample the databases, the original queries are parsed to derive the constant values and the corresponding columns from the WHERE clauses. For instance, in a query `SELECT * from cars WHERE mpg > 20`, the constants 20 and 21 are assigned some probability to be generated as values of the `cars.mpg` column. We improve the query parser so that it accounts for common cases when table aliases are used (e.g., `SELECT * from cars as T WHERE T.mpg > 20`). Still, if a gold query has too many WHERE clauses, its execution on a randomly generated database is likely to be an empty table or, in the case of SELECT-ing an aggregate function, zeros or NULL values. This issue causes many potential false positives.

For this reason, for every query, we first search for a test database on which the query execution is not empty and only then proceed to check if the database distinguishes the query and its neighbors. This way, the chances are higher to sample at least one database yielding non-empty output on a gold query, leading to better code coverage. We also propose to build a separate test suite for each query. This modification allows dramatically reducing the time of the test-suite evaluation of a query.

Additionally, we make several important changes to the original implementation:

- We limit the number of rows in the tables to 100. As a result, the test suite databases are smaller in size and faster in evaluation.
- We adjust the types of the columns in the original and the sampled databases so that they match the type of the corresponding values. In the float type columns, we also reduce the precision of numbers to 16-bit. These changes are crucial for

SPARQL language, as it is more sensitive to data types than SQL.

- If a column has only unique values in the original database and its name contains special substrings such as ‘Name’, ‘ID’, and ‘Phone’, we treat it as a unique key and generate its values accordingly. This heuristic allows performing the GROUP BY operations on any unique key.

B Implementation Details of Search

For the search on top of T5-3B, we use the (Scholak et al., 2021) model provided in the Transformers library (Wolf et al., 2020). To search with large beam sizes, we modify the Transformers implementation of T5 inference: instead of caching keys and values in attention blocks, we cache the attention outputs, which reduces memory usage at the cost of speed.

For working with SQL queries outside the Spider dataset, we had to replace the SQL parsing coming with Spider due to its limited functionality. We use the `mo-sql-parsing` library instead.² We execute all SQL queries in the `sqlite3`³ package for python. For executing SPARQL queries, we use the open-source version of the Virtuoso system.⁴ During our search, the system generated many cumbersome queries, so we had to impose a strict time limit for each query and make our implementation robust to the database engine crashes. The time limit for Spider queries is 30 seconds and for other datasets is 300 seconds.

The grid of beam sizes for CAB search on top of T5-3B is 2, 10, 100, 800, the corresponding widths are 2, 2, 2, 5, the grid for BRIDGE is 1, 10, 100, 1000 and the widths are 1, 2, 2, 5, the grid for SQ-QDMR is 1, 100, 1000 and the widths are 1, 5, 10. We infer BRIDGE and SQ-QDMR on one NVIDIA V100 GPU and T5-3B on 8 NVIDIA A100 GPUs (while searching with maximum beam size). For the search with the sampling methods, we use the same schema: we run sampling several times, increasing the number of samples (k in top- k) until the selection criterion is passed. The grids for all the methods are the same. For searching with UniqueRandomizer, we run the methods until the selection criterion is passed or the maximum number of iterations is reached (equal to the maximum beam size: 800 for T5-3B and 1000 for BRIDGE and SQ-QDMR). We tune the temperature and p

²github.com/klahnakoski/mo-sql-parsing

³docs.python.org/3/library/sqlite3.html

⁴github.com/openlink/virtuoso-opensource

on Spider dev (the grid for temperature tuning was from 0.5 to 3.5 with 0.25 step; the grid for p was 0.85, 0.9, 0.95). The best $p = 0.95$ and the best temperature values are the following:

Model	CAB	Sample	Top-p	UniRand
T5-3B	2.7	2.4	3.1	2.7
BRIDGE	2.6	3.4	3.7	3.5
SQ-QDMR	1.0	1.3	1.9	1.2

While testing on single-database data, we use the same temperature parameters for the models trained on Spider only. For the fine-tuned models, we also tune temperature on dev sets and choose the values 2.0 for BRIDGE and T5-3B and 1.0 for SQ-QDMR.

C Execution (1-test) Accuracy vs. Test-suite Accuracy

In Table 6, we show the results of the search on Spider with two criteria: passing one test and passing the test suite. For both criteria, we compute execution accuracy (checks the execution result on one database) and test-suite accuracy (checks the execution results on the test suite). For all the models, test-suite accuracy is significantly lower than execution accuracy for the search with the one-test criterion.

Table 6: Execution (EX) and test-suite (TS) accuracy of search with 1-test and test-suite criteria on Spider dev and test.

Model	Split	1 test		Test Suite	
		EX	TS	EX	TS
T5-3B	dev	94.4	86.9	91.4	90.1
BRIDGE		91.0	81.9	87.4	84.0
SQ-QDMR		98.0	84.0	93.7	94.4
T5-3B	test	90.7	84.7	85.5	86.8
BRIDGE		83.4	76.6	77.4	77.8
SQ-QDMR		86.7	75.2	79.7	84.1

D Single-database Datasets

We use the query splits provided by [Finegan-Dollak et al. \(2018\)](#) for four single-database datasets: GeoQuery, IMDB, Yelp and Academic. We choose query splits because they are more difficult than question splits, according to the findings of [Finegan-Dollak et al. \(2018\)](#). We exclude duplicates and examples with gold SQL queries that crash or execute longer than 5 minutes with sqlite3. The statistics of resulted datasets are presented in Table 7.

Table 7: Statistics of single-database data.

Dataset	Train	Dev	Test
GeoQuery	536	159	182
IMDB	103	9	17
Yelp	104	11	10
Academic	142	18	15

E Fine-tuning Details

For fine-tuning on single-database data, we use official implementations of all the models.⁵ We experimented with different training strategies: initializing from released checkpoints of the models trained on Spider and training from scratch (in this case, models contain transformers pre-trained on textual data). We choose the best approach for each model and refer to it as fine-tuning.

We fine-tune T5-3B pre-trained on textual data ([Raffel et al., 2020](#)) for 300 epochs on one NVIDIA A100 GPU with the same parameters as [Scholak et al. \(2021\)](#) used: Adafactor optimizer ([Shazeer and Stern, 2018](#)) with learning rate 1e-4 and batch size 625.

The released checkpoint of the BRIDGE model was trained on data that includes question splits of single-database data that we consider. We re-train this model on Spider-only data to evaluate it on query splits of single-database datasets. We use the same training parameters as [Lin et al. \(2020\)](#) used: Adam optimizer ([Kingma and Ba, 2014](#)) with the same scheduler (L-inv learning rate decay) and batch size 32. We choose the best checkpoint on the development set as [Lin et al. \(2020\)](#) did in their work (execution accuracy and test-suite accuracy of our and authors' checkpoints are the same on Spider dev and test). We use the same training procedure for fine-tuning on single-database data but create training data from both Spider train set and the train set of a particular dataset.

For fine-tuning SQ-QDMR on GeoQuery, we use the corresponding part of the Break dataset ([Wolfson et al., 2020](#)) and generate the 366 train groundings using the automatic annotation model of [Saparina and Osokin \(2021\)](#). We cannot fine-tune on the Academic dataset because its database is large and preprocessing of [Saparina and Osokin \(2021\)](#) failed. QDMR forms for other datasets were not provided in the Break dataset, so we could not fine-tune on them. For fine-tuning on GeoQuery, we start with the released checkpoint and param-

⁵github.com/ElementAI/picard;
github.com/salesforce/TabularSemanticParsing;
github.com/yandex-research/sparqling-queries;

eters saved on 73000 iterations of Spider training and continue up to 81000 iterations with GeoQuery train data. We also use the same parameters as [Saparina and Osokin \(2021\)](#) used: the optimizer is Adam ([Kingma and Ba, 2014](#)) with polynomial decay scheduler used by ([Wang et al., 2020](#)), the batch size is 24.

We use 1 NVIDIA A100 GPU for training T5-3B, 1 NVIDIA V100 GPU for BRIDGE and 4 NVIDIA V100 GPUs for SQ-QDMR.

F Execution Accuracy of Search on Single-Database Data

Table 8 shows execution accuracy of search on different single-database datasets. We consider two types of models: trained only on Spider data and fine-tuned on single-database data. Comparing with Table 5, the figures are higher because many false-positive queries pass one test (the execution accuracy metric) and do not pass the test suite. For this reason, the results of searching with the test-suite criterion are lower in terms of execution accuracy: if no tested query satisfies the test-suite criterion, the system defaults to the result of the greedy decoding, which may fail one test, while the one-test criterion would select a false positive.

Table 8: Different search criteria (execution, output column match, 1 test and test suite) on top of pre-trained models on the query test splits of different datasets with **execution accuracy**.

Dataset (test size)	Model	Greedy	Exec	Cols	1 test	Test Suite
GeoQuery (182)	T5-3B	56.6	59.3	69.8	80.2	74.7
	+ fine-tune	69.8	76.4	88.5	97.8	97.3
	BRIDGE	57.6	51.1	63.7	86.8	74.7
	+ fine-tune	71.4	72.5	86.3	96.7	93.4
	SQ-QDMR	40.7	40.7	45.6	92.3	75.8
	+ fine-tune	61.5	61.5	64.8	90.7	84.1
IMDB (17)	T5-3B	5.9	17.6	17.6	35.3	35.3
	+ fine-tune	52.9	52.9	52.9	58.8	52.9
	BRIDGE	17.6	17.6	23.5	35.3	23.5
	+ fine-tune	52.9	52.9	52.9	58.8	52.9
	SQ-QDMR	11.8	11.8	11.8	41.2	35.3
Yelp (10)	T5-3B	30	60	50	10	30
	+ fine-tune	40	50	40	10	70
	BRIDGE	10	30	30	90	30
	+ fine-tune	40	80	80	100	70
	SQ-QDMR	40	40	40	80	40
Academic (15)	T5-3B	6.7	13.3	26.7	-	26.7
	+ fine-tune	53.3	53.3	53.3	-	73.3
	BRIDGE	6.7	6.7	6.7	-	20
	+ fine-tune	33	40	40	-	80
	SQ-QDMR	5.6	5.6	11.1	-	55.6

Style-Aware Contrastive Learning for Multi-Style Image Captioning

Yucheng Zhou, Guodong Long*

Australian AI Institute, School of Computer Science, FEIT, University of Technology Sydney
yucheng.zhou-1@student.uts.edu.au, guodong.long@uts.edu.au

Abstract

Existing multi-style image captioning methods show promising results in generating a caption with accurate visual content and desired linguistic style. However, existing methods overlook the relationship between linguistic style and visual content. To overcome this drawback, we propose style-aware contrastive learning for multi-style image captioning. First, we present a style-aware visual encoder with contrastive learning to mine potential visual content relevant to style. Moreover, we propose a style-aware triplet contrast objective to distinguish whether the image, style and caption matched. To provide positive and negative samples for contrastive learning, we present three retrieval schemes: object-based retrieval, RoI-based retrieval and triplet-based retrieval, and design a dynamic trade-off function to calculate retrieval scores. Experimental results demonstrate that our approach achieves state-of-the-art performance. In addition, we conduct an extensive analysis to verify the effectiveness of our method.

1 Introduction

Stylized image captioning aims to generate a natural language description with stylized elements for a given image (Mathews et al., 2016; Gan et al., 2017). With the advance of deep learning in human-computer interaction equipment, it has been integrated into many real-world applications like education robots (Zhou, 2022), visual dialog (Das et al., 2017), and vision-language navigation (Wang et al., 2019). Therefore, it has attracted more attention from academia and industry and has become one of the essential areas in the natural language processing (NLP) and computer vision (CV) community.

However, many methods propose translating images into captions of a single caption style (Mathews et al., 2016; Gan et al., 2017). These methods need to train multiple models to handle multiple



Figure 1: (a) the caption is generated by stylized captioning model (Shuster et al., 2019); (b) and (c) show the same caption can correspond to different styles.

styles, which is very inefficient. Therefore, a rising demand for stylized image captioning is to learn an efficient model to handle multiple styles simultaneously. Recently, many efforts have been made in multi-style image captioning (Guo et al., 2019; Shuster et al., 2019; Li and Harrison, 2021).

Despite their success, existing methods suffer from a drawback: They focus on accurate visual content and desired linguistic style but overlook the relation between linguistic style and visual content. As shown in Figure 1(a), the generated caption contains accurate visual content and the desired style, but with misinformation that is easily detectable by humans, i.e., model generates “beautiful” in the caption to cater to the style of “happy”. Indeed, linguistic style (Bell, 1984) reflects personality, emotion and sentiment. In human behavior, these factors significantly influence the course of cognitive behavior (Simon, 1967). When people with different emotions see the same image are likely to describe different contents because they pay attention to different aspects. For example, people who are happy with animals may describe an image of a dog as a pretty dog with sparkling eyes and supple hair. However, one afraid of dogs may describe it as a scary dog with fierce teeth and sharp claws. The former focus on the dog’s eyes and hair, while the latter on the dog’s teeth and paws. Therefore, people with different emotions focus on different potential visual content. To

*Corresponding author.

generate human-like stylized captions, a stylized captioning model should learn to mine potential visual content relevant to different linguistic styles.

Due to the success of contrastive learning (He et al., 2020; Gao et al., 2021), some works (e.g., UNITER (Chen et al., 2020), ViLT (Kim et al., 2021)) employ contrastive learning objectives to encourage cross-modality alignment. In this work, we propose a style-aware visual encoder with contrastive learning to mine potential visual content relevant to styles. Specifically, style-aware visual encoder first mines potential visual features based on given image and style pairs. Then, we use a contrastive loss to pull potential visual features of the anchor and positive pair together while pushing those of anchor and negative pairs apart.

In addition, apart from requiring generated captions with accurate visual content and desired linguistic style, multi-style image captioning also needs to ensure that potential visual content and style are relevant. Moreover, since multi-style image captioning contains more fine-grained styles than single-stylized image captioning, it is difficult to directly distinguish the style by the caption. As shown in Figure 1(b) and (c), captions may be the same for two different styles. Therefore, multi-style image captioning is required to consider if matching among image, style and caption. Different from previous works (Guo et al., 2019) that optimize generated captions only based on style, we propose a style-aware triplet contrast loss, which can learn the triplet matching among image, style, and caption by contrasting it with the positive triplet against negative ones.

Moreover, motivated by hard negatives sampling in retriever training (Zhan et al., 2021), we present three retrieval schemes to mine positive and negative examples for contrastive learning: object-based retrieval, RoI-based retrieval and triplet-based retrieval. These three schemes calculate scores according to object overlap rate, potential visual feature similarity, and triple feature similarity, respectively. Meanwhile, we design a dynamic trade-off function to calculate retrieval scores and analyze the impact of different retrieval schemes.

We conduct an extensive evaluation on three datasets, i.e., PERSONALITY-CAPTIONS (Shuster et al., 2019), SentiCap (Mathews et al., 2016) and FlickrStyle10K (Gan et al., 2017). Our method significantly outperforms other strong competitors and achieves state-of-the-art performance.

2 Related Work

2.1 Image Captioning

Image captioning, one of the essential tasks in multimodal research, aims to generate a description for a given image (Hodosh et al., 2015). With the progress of deep learning, many end-to-end deep neural networks for image captioning are proposed (Vinyals et al., 2015; Anderson et al., 2018; Zhou et al., 2021). Although these works have achieved excellent improvements, the caption generated by these models focuses on a single language domain, which means that their outputs can be in only one style and even dull and lack vitality sometimes.

2.2 Stylized Image Captioning

With the advance in image captioning techniques, researchers have attempted to generate an image caption with style. Mathews et al. (2016) propose a word-level regularization for captioner to model sentiment words. Gan et al. (2017) employ transforming word embeddings matrices to control style factors for generated captions. To accurately describe visual content and reflect the desired linguistic style, some methods (Mathews et al., 2018; Zhao et al., 2020) split a stylized sentence into a style-related part that reflects the linguistic style and a content-related part that contains the visual content. Zhou (2022) employ prompt-based pre-training to build a stylized captioner without any paired sketch and story. These methods alleviate the reliance on paired training data for stylized captioner training. However, these methods need to train multiple models to handle multiple styles, which is inefficient. Therefore, learning an efficient model to handle multiple styles simultaneously raises more interest. Some efforts are made on multi-style image captioning, including adversarial learning network (Guo et al., 2019) and multi-updown fusion model (Li and Harrison, 2021). To generate more diverse outputs, Shuster et al. (2019) collect PERSONALITY-CAPTIONS, a large-scale multi-style image captioning dataset.

2.3 Contrastive Learning

Recently, contrastive learning has made exciting progress in representation learning (He et al., 2020; Gao et al., 2021; Zhou et al., 2022). He et al. (2020) leverage contrastive learning to improve visual presentation learning in an unsupervised manner. Gao et al. (2021) propose a simple unsupervised contrastive learning method to perform on

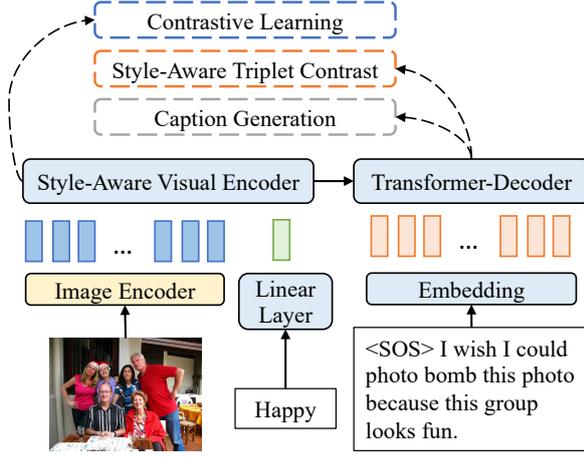


Figure 2: An overview of our proposed SACO model with three objectives. Yellow rounded rectangles denote fixed model. Blue rounded rectangles denote parameters that will be optimized.

par with previously supervised counterparts. Yang et al. (2022) propose triple contrastive learning for vision-language pre-training by leveraging both cross-modal and intra-modal self-supervision, providing complimentary benefits in representation learning. These works show the powerful ability of contrastive learning to improve representation learning.

3 Method

This section will elaborate on our Style-Aware COntrastive learning (SACO) method for multi-style image captioning, followed by our proposed novel retrieval schemes. The details of our approach are shown in Figure 2. Lastly, details about training and fine-tuning are elaborated.

3.1 Image and Style Encoding

Firstly, we pass an image I into a pre-trained convolutional neural network (CNN) to extract its visual feature V and convert the visual feature into a sequence according to the row-first direction:

$$V = \text{CNN}(I)$$

$$\text{where } V = [v_0, v_1, \dots, v_m], v_i \in \mathbb{R}^{|d^v|} \quad (1)$$

In addition, the style S is represented as a one-hot vector and encoded to a style feature s by a linear layer:

$$s = \text{LinearLayer}(S), s \in \mathbb{R}^{|d^s|} \quad (2)$$

Since there is a different dimension between visual feature V and style feature s , a multi-layer

perceptron (MLP) built upon v_i is presented to reduce its dimension to equal the dimension of s :

$$v_i := \text{MLP}(v_i), v_i \in \mathbb{R}^{|d^s|}, \forall i = 1, \dots, m \quad (3)$$

3.2 Style-Aware Visual Encoder

Originating from the observation that different styles will focus on different image regions, we propose a style-aware visual encoder with contrastive learning to mine potential visual content relevant to styles. Since self-attention (Vaswani et al., 2017) shows a strong capability to relate different positions of a sequence for feature computation, we leverage self-attention as backbone of the style-aware visual encoder. Particularly, we concatenate the visual feature V and style feature s , and apply the self-attention layers to them to derive the style-aware visual features V^s and vision-aware style feature s^v , i.e.,

$$[V^s; s^v] = \text{Self-Attention}([V; s]) \quad (4)$$

where $[\cdot; \cdot]$ denotes the operation of concatenation. Intuitively, the success of Eq. (4) requires accurately capturing the visual features relevant to styles, but a major problem arises without artifact labels: *Learning Difficulty*. Motivated by some works (Chen et al., 2020; Kim et al., 2021) implement cross-modality alignment via contrastive learning, we leverage contrastive learning for style-aware visual encoder, which drives its learning to capture potential visual content correlated to given styles without any labeled data. Specifically, we use contrastive learning to learn the representation of potential visual content by contrasting it with the positive example (\hat{V}^s, \hat{s}^v) against those of negative ones $(\bar{V}_i^s, \bar{s}_i^v)$. Particularly, we first derive features (\hat{V}^s, \hat{s}^v) and $(\bar{V}_i^s, \bar{s}_i^v)$ independently via Eq. (1-4). Then, we mine more accurate (V^s, s^v) , style-aware visual features and vision-aware style features, by contrasting it with (\hat{V}^s, \hat{s}^v) against $(\bar{V}_i^s, \bar{s}_i^v)$, i.e.,

$$\mathcal{L}^{(svc)} = -\log \frac{e^{\text{sim}(r, \hat{r})/\tau}}{e^{\text{sim}(r, \hat{r})/\tau} + \sum_{i=1}^M e^{\text{sim}(r, \bar{r}_i)/\tau}}$$

$$- \log \frac{e^{\text{sim}(s^v, \hat{s}^v)/\tau}}{e^{\text{sim}(s^v, \hat{s}^v)/\tau} + \sum_{i=1}^M e^{\text{sim}(s^v, \bar{s}_i^v)/\tau}}$$

$$\text{where } r = \text{Pooling}(V^s)$$

$$\hat{r} = \text{Pooling}(\hat{V}^s)$$

$$\bar{r} = \text{Pooling}(\bar{V}_i^s) \quad (5)$$

where $\text{sim}(\mathbf{r}, \mathbf{r}')$ is the dot product operation between ℓ_2 normalized \mathbf{r} and \mathbf{r}' (i.e. cosine similarity $\frac{\mathbf{r}^\top \mathbf{r}'}{\|\mathbf{r}\| \cdot \|\mathbf{r}'\|}$) and τ is a temperature hyperparameter to control the pull and push force; M is number of negative samples. As a result, since the style-aware visual features and vision-aware style features also offer a straightforward pathway to transmit style-aware visual information to style-aware visual encoder, it mitigates the *learning difficulty* problem.

3.3 Transformer-Decoder Based Generator

Due to the success of the pre-trained language model, there are some works (e.g., GPT (Radford et al., 2018), GPT-2 (Radford et al., 2019)) pre-train the Transformer decoder (Vaswani et al., 2017) on large-scale corpora. Recently, a fundamental paradigm of text generation tasks is to fine-tune the pre-trained model on the target data, and it can achieve exciting performance. In this work, we leverage a trained transformer decoder to initialize our generator. The generator input is adjusted to the triples $(\mathbf{V}^s, \mathbf{s}^v, \bar{\mathbf{C}})$, where $\bar{\mathbf{C}}$ refers to the segment of stylized image caption. The purpose of generator is to predict a probability distribution of the next word of the segment $\bar{\mathbf{C}}$ based on the given triple, i.e.,

$$\mathbf{h}_i = \text{Trans-Dec}(\mathbf{V}^s, \mathbf{s}^v, \bar{\mathbf{C}}) \in \mathbb{R}^{|d|}$$

where $\bar{\mathbf{C}} = [c_1, \dots, c_{i-1}]$ (6)

$$\mathbf{p}_i = \text{LM-Head}(\mathbf{h}_i) \in \mathbb{R}^{|\mathcal{V}|}$$
 (7)

where \mathbf{h}_i refers to the hidden representation in i -th step; \mathcal{V} denotes token vocabulary and \mathbf{p}_i is a probability distribution over \mathcal{V} . Lastly, the caption generation objective is defined as a maximum likelihood estimation and written as:

$$\mathcal{L}^{(cap)} = -\frac{1}{|N|} \sum_{i=1}^N \log \mathbf{p}_i(c_i),$$
 (8)

where $\mathbf{p}_i(c_i)$ denotes fetching the probability of the i -th step gold token $c_i \in \mathbf{C}$ from \mathbf{p}_i . \mathbf{C} refers to the gold caption and N is its length.

3.4 Style-Aware Triplet Contrast

Different from stylized image captioning for a single style, multi-style image captioning contains more fine-grained styles, and it is difficult to distinguish the style by the caption directly. So multi-style image captioning is required to consider if matching among image, style and caption. Recently, contrastive learning has shown its power capability in alignment between positive pairs and

dispersion between negative ones. Inspired by advances in contrastive learning, we propose a style-aware triplet contrast loss to learn the triplet matching among image, style, and caption by contrasting it with the positive triplet against negative ones. Given an image \mathbf{I} , style \mathbf{S} and caption \mathbf{C} , we first obtain the representation for this triplet. Since visual and style features have fed into the decoder, the triplet representation can be extracted by the hidden representation of the decoder, i.e.,

$$\mathbf{h} = -\frac{1}{|N|} \sum_{i=1}^N \text{MLP}_{\text{triplet}}(\mathbf{h}_i)$$
 (9)

where N is the same as that in Eq.8 and denotes the length of the caption; $\text{MLP}_{\text{triplet}}$ refers to a Multilayer Perceptron; and \mathbf{h}_i can be derived from Eq.6. Then, we enhance the triplet representation \mathbf{h} by contrasting it with a positive triplet $\hat{\mathbf{h}}$ against negative ones $\bar{\mathbf{h}}$, i.e.

$$\mathcal{L}^{(stc)} = -\log \frac{e^{\text{sim}(\mathbf{h}, \hat{\mathbf{h}})/\tau}}{e^{\text{sim}(\mathbf{h}, \hat{\mathbf{h}})/\tau} + \sum_{i=1}^M e^{\text{sim}(\mathbf{h}, \bar{\mathbf{h}})/\tau}}$$
 (10)

where $\text{sim}(\cdot, \cdot)$ is the dot product operation as same as that in Eq.5.

3.5 Retrieval Schemes

In our proposed contrastive learning objective, positive and negative samples are important elements. Since caption datasets are not designed with positive and negative samples, we propose three heuristics to derive positive and negative samples for a triplet of image \mathbf{I} , style \mathbf{S} and caption \mathbf{C} :

Object-based Retrieval. We first leverage a well-trained object detection model (Ren et al., 2015) to obtain object classes in images. Then, we retrieve image $\hat{\mathbf{I}}$ from image set \mathbb{I} according to object overlap with \mathbf{I} , and derive a probability for sampled examples:

$$P_{obj} = \frac{N_{overlap}}{N_{\mathbf{I}}}$$
 (11)

where $N_{\mathbf{I}}$ denotes number of objects in the image \mathbf{I} , and $N_{overlap}$ refers to the number of overlapped objects both in \mathbf{I} and $\hat{\mathbf{I}}$.

RoI-based Retrieval. In this retrieval scheme, the region of interest (RoI) refers to potential visual content relevant to style. We retrieve image $\hat{\mathbf{I}}$ based on the similarity between its representation of potential visual content and that of image \mathbf{I} :

$$P_{roi} = \text{sim}(\mathbf{V}, \hat{\mathbf{V}})$$
 (12)

where V and \hat{V} are the style-aware visual features of I and \hat{I} , and details of their calculation can refer to Eq.5.

Triplet-based Retrieval. Since triplet matching is essential for multi-style image captioning, we retrieve image \hat{I} according to the similarity between triplets:

$$P_{tri} = \text{sim}(\mathbf{h}, \hat{\mathbf{h}}) \quad (13)$$

where \mathbf{h} and $\hat{\mathbf{h}}$ are the triplet representation for the triplet (I, S, C) and $(\hat{I}, \hat{S}, \hat{C})$, and details of their calculation can refer to Eq.10.

Dynamic Trade-off Function. We combine the above three novel retrieval schemes to rank samples, and score of each sample can be defined as:

$$P = \theta^\mu P_{obj} + (1 - \theta^\mu)(\phi P_{roi} + (1 - \phi)P_{tri}) \quad (14)$$

where P denotes the score of samples and ϕ is a trade-off parameter; θ denotes a decay factor and μ is the current training epoch. During training, we randomly select a sample among the top-10 samples as positive one, and the negative samples are randomly sampled based on $P < \max(0.1, P_{max} - \omega^\mu)$.

3.6 Training and Fine-tuning

For model training, we adopt the same 2-stage training scheme (training and fine-tuning) as in (Anderson et al., 2018). In the training stage, we optimize the model according to the three objectives proposed above, and the loss function of our model can be integrated into the following:

$$\mathcal{L} = \mathcal{L}^{(cap)} + \alpha \mathcal{L}^{(svc)} + \beta \mathcal{L}^{(stc)} \quad (15)$$

where α and β are trade-off parameters.

In the fine-tuning stage, we employ the CIDEr score to optimize our model as same as (Rennie et al., 2017), i.e., returning a reward for generated caption \hat{C} .

$$\mathcal{R}^{(CIDEr)} = R_{CIDEr}(\hat{C}) - b \quad (16)$$

where b is a baseline, i.e., the reward $R_{CIDEr}(C^*)$ for the generated caption C^* with greedy search.

4 Experiments

4.1 Dataset and Evaluation Metrics

We evaluate our proposed approach on three datasets, PERSONALITY-CAPTIONS (Shuster et al., 2019), SentiCap (Mathews et al., 2016) and FlickrStyle10K (Gan et al., 2017).

PERSONALITY-CAPTIONS. Shuster et al. (2019) collect a large-scale multi-style image captioning dataset, PERSONALITY-CAPTIONS, which includes 201,858 images, 215 personality traits, and 241,858 stylized captions. We divide the dataset following (Shuster et al., 2019), and the size of the training set, validation set, and test set are 186,858, 5,000, and 10,000, respectively. In the test set, each image contains five reference captions. Following Shuster et al. (2019), we use the same metrics to report our results, and the evaluation is based on the coco-caption code¹. The evaluation metrics include: BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), CIDEr (Vedantam et al., 2015), SPICE (Anderson et al., 2016).

SentiCap & FlickrStyle10K. SentiCap (Mathews et al., 2016) and FlickrStyle10K (Gan et al., 2017) are two publicly stylized image caption datasets. According to (Li et al., 2021), we process SentiCap and FlickrStyle10K datasets, and use samples from MSCOCO (Lin et al., 2014) and Flickr30K (Hodosh et al., 2015) as large-scale paired factual data. Following Li et al. (2021), we use BLEU, METEOR (Banerjee and Lavie, 2005), CIDEr, style classification accuracy (cls.) and the average perplexity (ppl.) as evaluation metrics. Style classification accuracy is measured by a well-trained BERT, which achieves accuracies of 95.9%, 98.0%, 98.1%, and 99.5% on the test sets of humorous, romantic, negative, and positive styles, respectively. The average perplexity is measured by a well-trained trigram-based statistical language model using SRILM toolkit. A lower score denotes that the generated caption is more fluent and reflects the desired linguistic style better.

4.2 Implementation Details

Our approach adopts the same 2-stage training scheme (training and fine-tuning). For training stage, input images are resized to the size of 256×256 , and then we randomly crop the image size as 224×224 as model input. To ensure a fair comparison, we use the ResNeXt-IG-3.5B (Mahajan et al., 2018) as the pre-trained image encoder, as same as (Shuster et al., 2019), and the size of output features is $7 \times 7 \times 2048$. Then, the visual features are reshaped as 49×2048 according to the row-first direction. The style-aware visual encoder is constructed with three self-attention layers. We use a distilled version of pre-trained GPT-2 (Sanh

¹<https://github.com/tylin/coco-caption>

et al., 2019) as our transformer-decoder based generator, as same as (Nguyen et al., 2020). The layers and attention heads of the decoder are 6 and 8. The dimension of embedding vectors and hidden states in the decoder are 768 and 1024. Special tokens for the beginning and end of sentences are <SOS> and <EOS>. In addition, the size of the style linear layer are 215×768 and 5×768 for PERSONALITY-CAPTIONS dataset and SentiCap and FlickrStyle10K datasets. For model training, we utilize the Adam optimizer (Kingma and Ba, 2015) with learning rate of $1e-4$. The batch size, warm-up proportion, weight decay, maximum training epoch and temperature hyperparameter τ are 128, 0.1, 0.01, 10 and 0.08. The trade-off parameter ϕ and decay factor θ in Eq.14 are 0.5 and 0.9. ω for negatives sampling is 0.8. Trade-off parameters α and β in Eq.15 are 0.5 and 0.7. For fine-tuning stage, the maximum training epoch and learning rate are 3 and $1e-5$. Other experimental details are the same as that of training stage. For testing, we use beam search with beam size of 3 to generate captions with maximum sentence length of 30. Our model is trained on one V100 GPU.

4.3 Baselines

We compare our model with the following state-of-the-art baselines: (1) **ShowTell** propose by (Shuster et al., 2019) and is a variant of (Vinyals et al., 2015) that concatenates the style features with the input word vectors at each decoding step. (2) **ShowAtt-Tell**, proposed by (Xu et al., 2015), aims to enhance the correlation between the text and image by an attention mechanism, and the used model is a variant proposed by (Shuster et al., 2019). (3) **UpDown** with a decoder of two LSTMs can adapt to generate attention weights and use it to generate captions (Anderson et al., 2018), and the used model is a variant proposed by (Shuster et al., 2019). (4) **GPT** with an image encoder is fine-tuned on the captioning dataset (Radford et al., 2019). (5) **GPT-Speaker** (Nguyen et al., 2020) employs the language model GPT2 as a language prior for both the speaker and listener in the multi-agent communication framework. (6) **3M** (Li and Harrison, 2021) is a multi-style image captioner that is a multi-UpDown encoder-decoder model integrated with multi-modal features.

4.4 Main Results

Comparison results on PERSONALITY-CAPTIONS test set are shown in Table 1.

Method	B@1	B@4	R	C	S
ShowTell	38.4	7.3	24.3	9.6	1.6
ShowAttTell	43.3	7.1	27.0	12.6	3.6
UpDown	44.4	8.0	27.4	16.5	5.2
GPT	49.2	9.1	29.0	19.0	6.3
GPT-Speaker	52.1	8.4	30.2	19.9	7.3
GPT-Speaker*	52.3	8.2	30.1	20.0	7.4
3M	43.0	8.0	27.6	18.6	4.8
SACO (Ours)	54.8	9.7	32.6	21.0	8.1

Table 1: Comparison results on PERSONALITY-CAPTIONS test set. B@1, B@4, R, C and S denote BLEU@1, BLEU@4, ROUGE-L, CIDEr and SPICE, respectively. * refers to the baseline of our reproduction.

Style	Method	B@1	B@3	M	C	cls.	ppl.
Humor	SF-LSTM	27.4	8.5	11.0	39.5	-	-
	MSCap	16.3	1.9	5.3	15.2	91.3	22.7
	SAN	29.5	9.9	12.5	47.2	99.4	13.7
	SACO	35.2	10.3	13.4	51.2	99.7	12.9
Roman	SF-LSTM	27.8	8.2	11.2	37.5	-	-
	MSCap	17.0	2.0	5.4	10.1	88.7	20.4
	SAN	30.9	10.9	13.0	53.3	99.6	13.1
	SACO	35.8	13.5	14.1	55.8	99.9	12.4
Pos	SF-LSTM	50.5	19.1	16.6	60.0	-	-
	MSCap	46.9	16.2	16.8	55.3	92.5	19.6
	SAN	53.0	23.4	18.1	72.0	100.0	11.7
	SACO	56.3	24.7	19.5	72.2	99.8	11.5
Neg	SF-LSTM	50.3	20.1	16.2	59.7	-	-
	MSCap	45.5	15.4	16.2	51.6	93.4	19.2
	SAN	51.2	20.5	17.6	67.0	100.0	14.8
	SACO	53.2	23.6	19.1	70.5	100.0	13.3

Table 2: Comparison results on FlickrStyle10K and SentiCap test set. M denotes METEOR.

From the table, we can make three observations. First, we can observe that our methods achieve state-of-the-art performance on the PERSONALITY-CAPTIONS dataset. Second, models based on GPT reach a better performance than LSTM-based models, which shows the powerful generation capability of transformer-based decoder in stylized caption generation. Lastly, our model significantly outperforms the GPT-2 model by a large margin (i.e., improving 2.0 in CIDEr). That is, style-aware contrastive learning can improve the model by strengthening different style understanding.

In addition, we also show comparison results on FlickrStyle10K and SentiCap test set in Table 2. Competitors include SF-LSTM (Chen et al., 2018), MSCap (Guo et al., 2019) and SAN (Li et al., 2021). Results show that transformer-based methods (i.e., SAN and SACO) outperform LSTM-based methods (i.e., SF-LSTM and MSCap). More-

Method	B@1	B@4	R	C	S
SACO	54.8	9.7	32.6	21.0	8.1
◇ w/o CIDEr	53.0	9.6	31.9	19.1	7.2
◇ w/o STC	52.9	9.0	31.2	20.0	7.5
◇ w/o SVC	52.4	8.7	30.8	19.8	7.3
◇ w/o SVC, STC	48.9	9.0	29.1	18.7	6.0
◇ $\mathcal{L}^{(cap)}$ Only	47.3	8.7	29.2	16.0	5.4

Table 3: Ablation study. “w/o CIDEr” denotes removing fine-tuning stage for model training; “w/o STC” denotes removing the style-aware triplet contrast loss; “w/o SVC” denotes removing the style-aware visual contrast encoder; “w/o SVC, STC” denotes removing both contrastive learning methods in our model; and “ $\mathcal{L}^{(cap)}$ Only” denotes our model only train with the caption generation objective and without fine-tuning.

Method	B@1	B@4	R	C	S
SACO	54.8	9.7	32.6	21.0	8.1
SACO (Dec w/o style)	53.8	9.4	31.9	20.5	7.8
◇ w/o SVC	52.9	9.0	31.2	20.0	7.5

Table 4: Impact of style-aware visual contrast encoder.

over, compared with strong competitors, our approach achieves state-of-the-art performance on FlickrStyle10K and SentiCap.

4.5 Ablation Study

To demonstrate the effectiveness of our method, we conduct an ablation study and results are shown in Table 3. We first investigate the impact of the style-aware triplet contrast objective by removing it and find that model performance is decreased. Next, we investigate our method without style-aware visual contrast encoder and observe that the performance drops, which is worse than that without SVC. Moreover, we remove both contrastive learning objectives, and results show the performance is degraded again. The observations above demonstrate the effectiveness of style-aware contrastive learning.

4.6 Analysis

4.6.1 Impact of Contrastive Learning

Impact of Style-Aware Visual Contrast Encoder.

To further investigate the impact of the style-aware visual contrast encoder, we remove the input style for our decoder, i.e., the decoder input only consists of V^s, \bar{C} in Eq.6. Results are shown in Table 4. Results show that our decoder without input style outperforms the model without SVC, which demonstrates that style-aware visual contrast encoder can capture potential visual content relevant to the given style.

Method	B@1	B@4	R	C	S
SACO	54.8	9.7	32.6	21.0	8.1
SACO (STC-pointwise)	53.2	9.0	31.4	20.1	7.3
SACO (comp.)	53.0	9.1	31.2	20.2	7.4
◇ w/o STC	52.4	8.7	30.8	19.8	7.3

Table 5: Impact of style-aware triplet contrast.

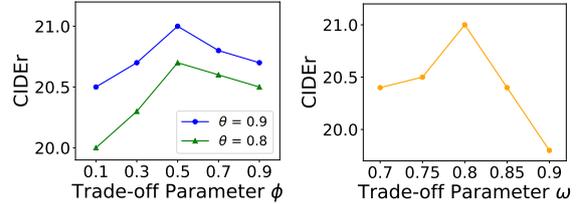


Figure 3: Retrieval schemes ablation.

Impact of Style-Aware Triplet Contrast. To investigate the impact of the style-aware triplet contrast objective, we replace the objective with two binary classification based objectives: point-wise classification and the comp. objective in (Nguyen et al., 2020). As shown in Table 5, results show that binary classification based objectives underperform our contrastive objective, which demonstrates that contrasting positive and negative triplets is helpful for the model to understand triplet matching.

4.6.2 Retrieval Schemes Ablation

Due to contrastive learning depending on positive and negative sampling, we conduct an ablative analysis on retrieval schemes. Results are shown in Figure 3. From left figure, we can make two observations: (1) The decay factor θ set of 0.9 can perform better than that of 0.8. It demonstrates that object-based retrieval is essential in the early training phase. (2) The trade-off parameter ϕ set of 0.5 can achieve the best performance, which shows the vital role of both style-aware contrastive objectives in positive and negative sampling. From right figure, based on the sampling function in §3.5, as ω becomes greater, negative sampling range during training will become greater. We can see that a greater negative sampling range is beneficial for contrasting learning. The reason is that the larger sampling range allows model to access harder negatives. In addition, we find that the performance gradually drops when ω is greater than 0.8, which shows too hard negatives hinder model learning.

4.6.3 Qualitative Comparison

To extensively evaluate our model, we conduct a qualitative comparison of our model and GPT-

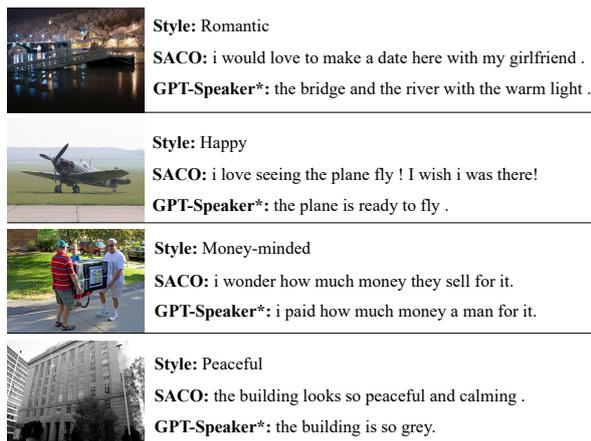


Figure 4: Random sampling examples generated by SACO and GPT-Speaker*.



Figure 5: Interpretable visualization analysis of random sampling examples. The brighter image areas mean more relevant to style.

Speaker, and some random sampling examples are shown in Figure 4. For example, in the first line, we can observe that GPT-Speaker can capture objects in the image and describe them in a caption, but it does not entail style. Therefore, the caption generated by our model is shown to more natural and with desired style. For instance, in the second and last lines, we can find that the caption generated by GPT-Speaker is more like a factual description. In contrast, the caption generated by our model is more expressive of style.

4.6.4 Interpretable Visualization Analysis

To investigate the effectiveness of style-aware visual contrast encoder, we conduct an interpretable visualization analysis, as shown in Figure 5. In the attention map of self-attention layers, the brighter image areas mean greater attention weights, i.e., these areas are more relevant to the given style. As shown in the first example, under the style "money-minded", the object is more paid attention to than the human, which is very intuitive. Next, in the second example, under the style "ex-

Type of evaluation	Win Percentage	
	SACO	GPT-Speaker*
Engagingness	62.9	37.1
Visual Relevance	64.9	35.1
Personalized Relevance	63.4	36.6

Table 6: Human Evaluation.

citing", "a dog of running" are paid more attention and described with "love" and "cute", which are reasonable. Therefore, the results show that the style-aware visual contrast encoder can effectively capture potential visual content related to style.

4.6.5 Human Evaluation

To comprehensively evaluate our method, we conducted a human evaluation to compare our model and GPT-Speaker. Following Nguyen et al. (2020), we considered the engagingness and relevance of captions. Engagingness evaluation considers human preference for the naturalness and appropriateness of the captions, while relevance evaluation involves visual and stylized relevance. Therefore, there are three types of evaluation. We randomly sampled 50 samples from the test set for each evaluation type above. Each sample includes an image and a style. Then, we use our model and GPT-Speaker to generate captions for these samples. We displayed the selected image-style pairs and their caption generated from our model and GPT-Speaker to 7 recruited annotators. They need to judge which captions are better quality based on the type of evaluation. As shown in Table 6, results show that the performance of our model is significantly better than GPT-Speaker, i.e., our model can generate fascinating captions.

5 Conclusion

In this work, we dive into the relationship between linguistic style and visual content. The first is potential visual content has varied for different styles. We propose a style-aware visual encoder that learns to capture the representation of potential visual content. Second, since the model is required to distinguish whether the image, style and caption are matched, we present a style-aware triplet contrast objective to improve the model's capability to discriminate triplet matching. In addition, we propose three novel retrieval schemes to sample positive and negative examples for contrastive learning. Results show that our method delivers new state-of-the-art performance.

Limitations

Although our proposed method can effectively mine latent visual content related to style, it still suffers from weaknesses in generating multiple stylized captions for the same image and style pair. Specifically, our method relies on beam search to generate diverse stylized captions for the same image and style pair and lacks the capability to control content for caption generation interactively. In further work, we will study how to generate stylized captions by interactively selecting specified regions in latent visual content.

References

- Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. **SPICE: semantic propositional image caption evaluation**. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V*, volume 9909 of *Lecture Notes in Computer Science*, pages 382–398. Springer.
- Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. **Bottom-up and top-down attention for image captioning and visual question answering**. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. IEEE Computer Society.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Allan Bell. 1984. Language style as audience design. *Language in society*, 13(2):145–204.
- Tianlang Chen, Zhongping Zhang, Quanzeng You, Chen Fang, Zhaowen Wang, Hailin Jin, and Jiebo Luo. 2018. **"factual" or "emotional": Stylized image captioning with adaptive learning and attention**. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, volume 11214 of *Lecture Notes in Computer Science*, pages 527–543. Springer.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. **UNITER: universal image-text representation learning**. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXX*, volume 12375 of *Lecture Notes in Computer Science*, pages 104–120. Springer.
- Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M. F. Moura, Devi Parikh, and Dhruv Batra. 2017. **Visual dialog**. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1080–1089. IEEE Computer Society.
- Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. **Stylenet: Generating attractive visual captions with styles**. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 955–964. IEEE Computer Society.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. **Simcse: Simple contrastive learning of sentence embeddings**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.
- Longteng Guo, Jing Liu, Peng Yao, Jiangwei Li, and Hanqing Lu. 2019. **Mscap: Multi-style image captioning with unpaired stylized text**. In *CVPR 2019*, pages 4204–4213. Computer Vision Foundation / IEEE.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. **Momentum contrast for unsupervised visual representation learning**. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. Computer Vision Foundation / IEEE.
- Micah Hodosh, Peter Young, and Julia Hockenmaier. 2015. **Framing image description as a ranking task: Data, models and evaluation metrics (extended abstract)**. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 4188–4192. AAAI Press.
- Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. **Vilt: Vision-and-language transformer without convolution or region supervision**. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR.
- Diederik P. Kingma and Jimmy Ba. 2015. **Adam: A method for stochastic optimization**. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Chengxi Li and Brent Harrison. 2021. **3m: Multi-style image caption generation using multi-modality features under multi-updown model**. In *Proceedings of the Thirty-Fourth International Florida Artificial Intelligence Research Society Conference, North Miami Beach, Florida, USA, May 17-19, 2021*.

- Guodun Li, Yuchen Zhai, Zehao Lin, and Yin Zhang. 2021. [Similar scenes arouse similar emotions: Parallel data augmentation for stylized image captioning](#). In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, pages 5363–5372. ACM.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. [Microsoft COCO: common objects in context](#). In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.
- Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. [Exploring the limits of weakly supervised pre-training](#). In *ECCV 2018*, volume 11206 of *Lecture Notes in Computer Science*, pages 185–201. Springer.
- Alexander Patrick Mathews, Lexing Xie, and Xuming He. 2016. [Senticap: Generating image descriptions with sentiments](#). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3574–3580. AAAI Press.
- Alexander Patrick Mathews, Lexing Xie, and Xuming He. 2018. [Semstyle: Learning to generate stylised image captions using unaligned text](#). In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8591–8600. IEEE Computer Society.
- Minh Thu Nguyen, Duy Phung, Minh Hoai, and Thien Huu Nguyen. 2020. [Structural and functional decomposition for personality image captioning in a communication game](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4587–4593. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. [Faster R-CNN: towards real-time object detection with region proposal networks](#). In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99.
- Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. [Self-critical sequence training for image captioning](#). In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1179–1195. IEEE Computer Society.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter](#). *CoRR*, abs/1910.01108.
- Kurt Shuster, Samuel Humeau, Hexiang Hu, Antoine Bordes, and Jason Weston. 2019. [Engaging image captioning via personality](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 12516–12526. Computer Vision Foundation / IEEE.
- Herbert A. Simon. 1967. Motivational and emotional controls of cognition. In *Psychological Review*, volume 74, pages 29–39.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. [Cider: Consensus-based image description evaluation](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. [Show and tell: A neural image caption generator](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3156–3164. IEEE Computer Society.
- Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. [Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation](#). In *CVPR 2019*, pages 6629–6638. Computer Vision Foundation / IEEE.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. [Show, attend and tell: Neural image caption generation with visual attention](#). In *Proceedings of the 32nd International*

Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org.

Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. 2022. [Vision-language pre-training with triple contrastive learning](#). *CoRR*, abs/2202.10401.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. [Optimizing dense retrieval model training with hard negatives](#). In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1503–1512. ACM.

Wentian Zhao, Xinxiao Wu, and Xiaoxun Zhang. 2020. [Memcap: Memorizing style knowledge for image captioning](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 12984–12992. AAAI Press.

Yucheng Zhou. 2022. [Sketch storytelling](#). In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 4748–4752. IEEE.

Yucheng Zhou, Tao Shen, Xiubo Geng, Guodong Long, and Daxin Jiang. 2022. [ClarET: Pre-training a correlation-aware context-to-event transformer for event-centric generation and classification](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2559–2575, Dublin, Ireland. Association for Computational Linguistics.

Yucheng Zhou, Wei Tao, and Wenqiang Zhang. 2021. [Triple sequence generative adversarial nets for unsupervised image captioning](#). In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 7598–7602. IEEE.

Strategize Before Teaching: A Conversational Tutoring System with Pedagogy Self-Distillation

Lingzhi Wang^{1,2}, Mrinmaya Sachan³, Xingshan Zeng⁴, Kam-Fai Wong^{1,2}

¹The Chinese University of Hong Kong, Hong Kong, China

²MoE Key Laboratory of High Confidence Software Technologies, China

³Department of Computer Science, ETH Zurich

^{1,2}{lzwang, kfwong}@se.cuhk.edu.hk

³msachan@ethz.ch, ⁴zxshamson@gmail.com

Abstract

Conversational tutoring systems (CTSs) aim to help students master educational material with natural language interaction in the form of a dialog. CTSs have become a key pillar in educational data mining research. A key challenge in CTSs is to engage the student in the conversation while exposing them to a diverse set of teaching strategies, akin to a human teacher, thereby, helping them learn in the process. Different from previous work that generates responses given the strategies as input, we propose to jointly predict teaching strategies and generate tutor responses accordingly, which fits a more realistic application scenario. We benchmark several competitive models on three dialog tutoring datasets and propose a unified framework that combines teaching response generation and pedagogical strategy prediction, where a self-distillation mechanism is adopted to guide the teaching strategy learning and facilitate tutor response generation. Our experiments and analyses shed light on how teaching strategies affect dialog tutoring.

1 Introduction

Decades of research effort (Carbonell, 1970; Richardson, 1988; Brown, 2009) has been put into building intelligent tutoring systems (ITSs). An important feature of these systems is the ability to customize the instructional activities and strategies based on the learner’s characteristics and needs (Keleş et al., 2009). Conversational tutoring systems (CTSs) that aim to offer automated tutoring through natural language dialog is a key pillar of ITS research. Earlier work in CTSs was based on conventional techniques such as Bayesian techniques with rule engines (Jeon and Su, 2010; Weragama and Reye, 2014) and hybrid neural networks (Kose and Arslan, 2017; Stasaski et al., 2020). While various advanced neural approaches have been applied to open-domain (Sordani et al., 2015; Serban et al., 2016; Xing et al., 2017) and task-

Teaching Strategy	Tutor Response
Restating	Let me say back what I heard.
Pressing for accuracy	Can you tell us the steps you used to find the answer?

(a) Two examples of teaching strategy and tutor response

Tutor:	Ok, now we have 'get cut off', 'put someone through' and 'get through'	scaffolding
Tutor:	I was talking to her, but suddenly we _____ (I couldn't hear her anymore) Can you choose one?	eliciting
Student:	got cut off	.
Tutor:	yes, good! Well done today! Have a lovely day!	closing

(b) An example of interactions between tutor and student

Figure 1: Examples of teaching strategy and interactions between tutor and student. Teaching strategies in Figure 1(b) are in red.

oriented dialogue systems (Zhao et al., 2017; Lei et al., 2018; Peng et al., 2020), conversational tutoring systems have not benefited from the development of these technologies (Macina et al., 2023).

Human teachers use a number of nuanced teaching strategies in the classroom during interactions with students; these strategies are tailored to keep the students engaged in the conversation and learn knowledge efficiently. We show some examples of teaching strategies and interactions between the tutor and the student in Fig. 1. Previous work has attempted to model these teaching strategies in different ways – e.g., Suresh et al. (2019) contributed a teaching strategy classification model and Stasaski et al. (2020) proposed a response generation model based on given teaching strategies of next response.

In this work, we benchmark several neural dialog models on three conversational tutoring datasets, CIMA (Stasaski et al., 2020), TSCC (Caines et al., 2020) and TalkMoves (Suresh et al., 2019, 2022), and contribute a unified framework based on pre-trained language models, where teaching strat-

egy prediction and response generation are jointly trained. As predicting a teaching strategy merely by the historical context is more difficult than when we are also given the target tutor response, we also propose a pedagogy distillation mechanism that allows teaching strategy prediction to learn from the soft labels which are produced by the prediction with target response. The soft labels learned from the target response provides the model knowledge about various interrelationships between teaching strategies that hard labels lack. This approach is believed to be able to alleviate the learning difficulty (Hinton et al., 2015), which is particularly important, especially when the data imbalance and scarcity issues are severe – often the case in conversational tutoring data.

In summary, we are the first to benchmark¹ several competitive models for conversation tutoring system on all three datasets that are currently available. Besides, we propose a unified framework that can predict teaching strategy and generate tutoring responses accordingly, which is enhanced by a self-distillation mechanism. Our experiments validate the positive effects of teaching strategy to guide generation and the importance of predicting strategy first and then generate response accordingly.

2 Related Work

A classical Intelligent Tutoring System generally has three modules (Brown, 2009; Polson and Richardson, 2013): (i) expert module that includes the knowledge that the student wants to learn (Carter, 2014). (ii) student module that can adjust the level of student (e.g., primary/middle school, non-native/native speaker), student’s knowledge deficiency, etc. (iii) pedagogical module that focuses on the strategies of teaching. In expert module, the knowledge is usually domain specific, such as computer programming (Costello, 2012), mathematics (Grawemeyer et al., 2016; Suresh et al., 2022), Italian (Stasaski et al., 2020), English (Caines et al., 2020). Many technologies have been used in the expert module, such as Bayesian techniques with rule engines (Jeon and Su, 2010; Weragama and Reye, 2014) and hybrid neural networks (Kose and Arslan, 2017; Stasaski et al., 2020). For pedagogical module, to our best knowledge, there are only three publicly available datasets that provide the pedagogy information. They are CIMA (Stasaski

¹The code can be found in <https://github.com/Lingzhi-WANG/TutorSystem>

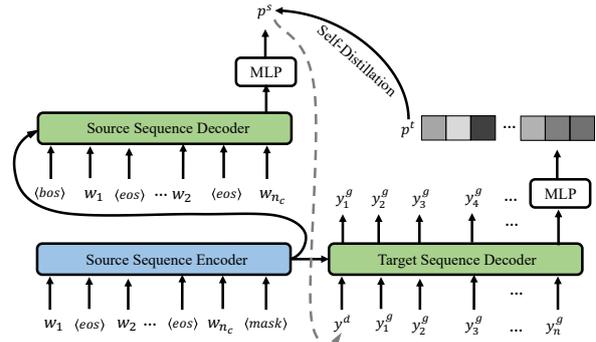


Figure 2: Our overall framework. The self-distillation leverages predictions based on target to improve predictions based on source. The enhanced strategy prediction is further utilized to facilitate the generation.

et al., 2020), TSCC (Caines et al., 2020) and Talk-Moves (Suresh et al., 2022) datasets and all of them are based on single pedagogy. There has been very little work on neural dialog tutoring. Two exceptions to this are Suresh et al. (2022), who propose a simple BiLSTM-based module to predict the pedagogy of the next sentence that teachers are meant to say, and Stasaski et al. (2020) who use various generative models to generate responses given the pedagogical strategies. In contrast, in this work, we propose a joint approach for modelling the pedagogy and response generation that outperforms the previous approaches using a novel pedagogy distillation mechanism.

3 Our Model

3.1 Problem Formulation

Our conversational tutoring system takes conversation context C and teaching strategy list D as input. C is formalized as a sequence of turns $\{t_1, t_2, \dots, t_{n_c}\}$ where n_c represents the number of turns. t_i ($1 \leq i \leq n_c$) denotes the i -th turn of the conversation, and we use w_i to indicate the word tokens contained in it. The teaching strategy list D covers all the possible strategies and contain n_d teaching strategies. Our model will first output one or several strategy labels, each $y^d \in \{1, 2, \dots, n_d\}$, to indicate what teaching strategy to use. Then the generation module generates a target response $y^t = (y_1^t, \dots, y_{n_t}^t)$ based on the predicted strategy.

3.2 Conversational Tutoring System (CTS)

PLM-based Generation Module. The generation module follows a Transformer (Vaswani et al., 2017) sequence-to-sequence framework. As the currently available tutoring datasets are quite small

(containing about 3k conversations), we choose to finetune pretrained language models (PLM) to alleviate data scarcity and enhance context modeling. We finetune BART (Lewis et al., 2020) and multilingual BART(mBART) (Liu et al., 2020) models for our generation module. During finetuning, we concatenate the utterances t_i ($1 \leq i \leq n_c$) in context C with appended $\langle \text{eos} \rangle$ tokens in their chronological order as input, and maximize the probability of the ground-truth target sequence. The whole process is summarized as follows:

$$\mathbf{H}^c = \text{Transformer_Encoder}(\mathbf{w}^c) \quad (1)$$

$$y_k^t = \text{Transformer_Decoder}(y_{<k}^t, \mathbf{H}^c) \quad (2)$$

$$\mathcal{L}_{target}^{gen} = \sum_{k=1}^{n_t} -\log(p(y_k^t | y_{<k}^t, \mathbf{H}^c)) \quad (3)$$

where $\mathbf{w}^c = [w_1; \langle \text{eos} \rangle; w_2; \dots; w_{n_c}; \langle \text{mask} \rangle]$, and $y_{<k}^t$ represents the target tokens before y_k^t . We add $\langle \text{mask} \rangle$ at the end of context, to simulate the operation in pre-training (Schick and Schütze, 2021).

Besides, to summarize the representation of the conversation context, we employ an additional source sequence decoder as follows:

$$y_k^s = \text{Transformer_Decoder}(y_{<k}^s, \mathbf{H}^c) \quad (4)$$

$$\mathcal{L}_{source}^{gen} = \sum_{k=1}^{n_s} -\log(p(y_k^s | y_{<k}^s, \mathbf{H}^c)) \quad (5)$$

where $y_{<k}^s$ represents the source tokens before y_k^s .

Teaching Strategy Prediction Module. We use the representation of the $\langle \text{eos} \rangle$ token (i.e. the final token) produced by the decoder as the representation for teaching strategy prediction, denoted as $\mathbf{h}^{(\text{eos})}$. This is fed into a two-layer MLP for prediction:

$$\mathbf{r}^d = \mathbf{W}_2 \times \alpha(\mathbf{W}_1 \mathbf{h}^{(\text{eos})} + \mathbf{b}_1) + \mathbf{b}_2 \quad (6)$$

where \mathbf{W}_1 , \mathbf{W}_2 , \mathbf{b}_1 and \mathbf{b}_2 are learnable parameters, and α is a non-linear activation function. The output representation \mathbf{r}^d will be an n_d -dimension vector and the probability for each teaching strategy in list D is computed based on \mathbf{r}^d :

$$p(y^d = j) = \text{softmax}(\mathbf{r}^d)_j \quad (7)$$

where y^d denotes the predicted strategy and $j \in \{1, 2, \dots, n_d\}$.

We denote $\mathbf{h}^{(\text{eos})}$ produced by source and target generation as $\mathbf{h}_s^{(\text{eos})}$ and $\mathbf{h}_t^{(\text{eos})}$, respectively. With $\mathbf{h}_s^{(\text{eos})}$, it means that we predict the teaching

strategy without knowing the corresponding content; while with $\mathbf{h}_t^{(\text{eos})}$, we summarize the teaching strategy based on the target content. Obviously, predicting with $\mathbf{h}_s^{(\text{eos})}$ is what we need, but this is quite challenging. Thus we design a self-distillation mechanism which uses prediction based on $\mathbf{h}_t^{(\text{eos})}$ for enhancing the generation model.

Teaching Strategy Enhancement with Distillation. We denote the predicted probability for each strategy (derived with Eq. 7) using $\mathbf{h}_s^{(\text{eos})}$ and $\mathbf{h}_t^{(\text{eos})}$ as $p_s(\cdot)$ and $p_t(\cdot)$, respectively. Our self-distillation is defined as guidance from $p_t(\cdot)$ to $p_s(\cdot)$:

$$\mathcal{L}^{sd} = -\sum_{j=1}^{n_d} p_s(y^d = j) \log p_t(y^d = j) \quad (8)$$

where we define $p_t(\cdot)$ as teacher distribution and $p_s(\cdot)$ as student distribution, and Eq. 8 makes the student distribution similar to the teacher distribution. In this way, our teaching strategy prediction model can also learn from the soft labels produced by the target sequence.

Multiple Teaching Strategies Guided Generation. To guide the response generation with teaching strategy, we regard the teaching strategies as prompt tokens and display them at the beginning of generation. In this way, the target tokens will be generated autoregressively according to the giving teaching strategy. Specifically, during training, we use the ground-truth strategy (denoted as d^c , and it will be masked in distillation to avoid information leakage) for teacher forcing (i.e. $y_0^t = d^c$ in Eq. 3); during inference, we use the predicted strategies produced by the prediction module as prompt tokens.

To enable multiple teaching strategies guidance, we define a threshold θ , where all the strategies satisfying $p_s(y^d = j) \geq \theta$ ($1 \leq j \leq n_d$) will be used to guide the response generation. To that end, we weightedly sum over the embeddings of those strategies as prompt based on their predicted probabilities produced by Eq. 7 and then use it to guide the generation.

3.3 Learning Objectives

The learning objective for teaching strategy prediction is defined as follows:

$$\mathcal{L}^{pred} = -(\log p_s(y^d = d^c) + \log p_t(y^d = d^c)) + \lambda \cdot \mathcal{L}^{sd} \quad (9)$$

where d^c is the ground-truth strategy for context C and λ is a hyper-parameter to control the weights

of self-distillation loss. Our model is jointly trained on both generation and prediction, with the overall objective summarized as:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}^{gen} + \gamma \cdot \mathcal{L}^{pred} \\ &= \mathcal{L}_{target}^{gen} + \delta \cdot \mathcal{L}_{source}^{gen} + \gamma \cdot \mathcal{L}^{pred}\end{aligned}\quad (10)$$

where δ and γ are tradeoff hyper-parameters.

4 Experimental Setup

Datasets. We use three datasets to do the experiments. They are CIMA (Stasaski et al., 2020), TSCC (Caines et al., 2020) and TalkMoves (Suresh et al., 2019, 2022). CIMA contains one-to-one conversations that focus on teaching students to translate a phrase from English to Italian. TSCC focuses on teaching English for eight non-native English-speaking students. TalkMoves is constructed by transcripts of math classrooms.

Parameter Setting. Our implementation is based on Fairseq (Ott et al., 2019). We split the data into 8:1:1 for training, validation and test. All the hyper-parameters are chosen by grid-search based on the validation performance.

We use BART-Base² and mBART-Large³ models to initialize our model, respectively. BART-Base model has 6 layers of encoder and decoder with 768 hidden dimension, while mBART-Large has 12 layers of encoder and decoder with 1024 hidden dimension. The parameter sizes for the two models initialized with BART and mBART are 199M and 816M, respectively.

We use one NVIDIA RTX 3090 GPU to train our model. During training, we set the max tokens of each batch to 1024 (for BART, or 512 for mBART) with an update frequency of 4. We adopt Adam optimizer (Kingma and Ba, 2015) with learning rate selected in $\{1e-4, 5e-5, 2e-5, 1e-5\}$ and warm-up updates selected in $\{200, 500, 1000\}$ followed by a polynomial decay scheduler. Dropout strategy (Srivastava et al., 2014) with dropout rate selected in $\{0.2, 0.4\}$ and L_2 regularization with 0.01 effect value, as well as early stopping based on validation performance, are used to alleviate overfitting. We set the tradeoff values among the losses as $\lambda = 1.0$, $\gamma = 1.0$ and $\delta = 0.2$. During inference, predicting threshold $\theta = 0.3$ and beam size is set to 5.

²<https://github.com/facebookresearch/fairseq/tree/main/examples/bart>

³<https://github.com/facebookresearch/fairseq/tree/main/examples/mbart>

5 Experimental Results

5.1 Teaching Strategy Prediction Results

We report the accuracy and Macro F1 scores for teaching strategy prediction task in Table 1. We can find that prediction based on the target tutor response performs much better than merely on source context (comparing BART[†] and BART), which indicates that prediction with target content is much easier and also validates our motivation of the self-distillation mechanism. With the help of our proposed distillation mechanism, our models with pre-trained BART or mBART achieve the best performance in the prediction based on source context.

5.2 Tutor Response Generation Results

We then report case-sensitive detokenized sacreBLEU (Post, 2018) and BERTScore (Zhang et al., 2019) for tutor response generation in Table 2.

Three Evaluation Settings. We show results in three settings in Table 2. “W/O TS” means we don’t include teaching strategy (TS) labels in training and testing. “With Golden TS” means providing ground truth TS labels for training and testing. “Need TS Prediction” means models have to predict TS labels in testing and generate the follow-up tutor responses based on the predicted TS labels.

Analysis on Generation Results. From Table 2, we can draw the following main observations.

- *Teaching strategy shows positive effects in generation.* By comparing the results in “W/O TS” and “With Golden TS” settings, we observe that guidance from golden teaching strategies improves the generation performance in general, which validates the effects of teaching strategy in guiding generation. Besides, our models further improve their corresponding baselines (e.g. Our Model(BART) v.s. BART), which should result from the joint learning of generation and strategy prediction.

- *Successful guidance requires accurate teaching strategies.* By comparing results in “With Golden TS” and “Need TS Predict”, we can find that most of the models perform worse when they need to predict strategies first, especially for the baselines with poor strategy prediction performance (refer to results of BiLSTM and Transformer in Table 1). This shows that guidance from inappropriate strategies might even hurt performance, which raises the need for accurate prediction in real-world applications and our proposed method can alleviate the gap significantly.

Models	CIMA		TSCC		TalkMoves	
	Acc	F1	Acc	F1	Acc	F1
BART	64.3	31.5	59.1	11.6	55.2	31.1
BART [†]	82.3	57.1	64.4	18.9	75.9	50.5
Frequency	62.7	15.4	58.4	4.1	52.5	11.5
BiLSTM	57.3	30.1	56.5	11.2	50.1	25.6
Transformer	63.3	33.9	57.2	16.2	53.6	30.7
Our Model(BART)	69.7	39.2	<u>60.6</u>	<u>17.4</u>	57.8	35.5
Our Model(mBART)	<u>70.4</u>	<u>39.8</u>	60.4	17.0	<u>59.6</u>	<u>37.6</u>

Table 1: Teaching strategy prediction results (in %). † indicates the prediction is based on the target tutor response. The best and second-best results in each column are in **bold** and underlined respectively.

Models	CIMA		TSCC		TalkMoves		
	BLEU	BERT	BLEU	BERT	BLEU	BERT	
W/O TS	BiLSTM	9.08	72.6	1.04	69.0	0.43	73.2
	Transformer	10.1	72.2	1.53	70.4	0.74	74.9
	BART	6.77	71.9	1.27	71.2	0.85	78.0
	mBART	10.6	70.9	1.96	68.6	2.95	78.1
With Golden TS	BiLSTM	8.61	71.8	1.32	69.1	1.42	75.8
	Transformer	11.2	72.8	1.99	69.9	2.35	77.4
	BART	9.17	70.8	1.47	68.6	2.93	78.0
	mBART	11.1	72.3	1.57	69.5	3.38	75.7
	Our Model(BART)	10.8	71.4	2.02	70.6	3.18	78.0
	Our Model(mBART)	12.1	73.8	2.93	72.6	5.47	79.7
Need TS Predict	BiLSTM	7.65	69.8	0.68	68.2	0.48	74.7
	Transformer	8.04	68.6	0.79	69.3	2.05	76.8
	BART	7.64	69.5	1.13	69.4	1.49	73.8
	mBART	7.77	70.2	1.57	69.7	2.44	77.1
	Our Model(BART)	8.67	70.8	2.83	70.0	2.22	77.5
	Our Model(mBART)	11.9	73.0	2.98	71.9	4.51	78.6

Table 2: Generation results (in %). The best results in each setting are in **bold**. Our full model achieves significantly better performance than the baselines with the same architecture in the same settings (paired t-test $p < 0.05$).

Student: how to say under in Italian?		
CTS:	[Hint]	“Is under the” is “e sotto il”. Do you know how to say box?
CTS:	[Question]	“Is under the” is “e sotto il”.
Student: La pianta e accanto al congilio giallo.		
CTS:	[Correction]	You’re very close. but remember that adjective follows the noun.
CTS:	[Confirmation]	Yes, that’s right!

Figure 3: Our CTS generates different responses when giving different teaching strategies (in red).

5.3 Effects of Teaching Strategy

We explore how teaching strategy affects the generation in Fig. 3. We feed our conversational tutoring system (CTS) with different teaching strategies and find that CTS generates totally different responses

regarding the same context input. This also validates that teaching strategy is important for a CTS and strategizing before teaching is also essential.

6 Conclusion

In this work, we benchmarked neural models on various conversational tutoring datasets and proposed a self-distillation based model that jointly trains a teaching strategy prediction model and a response generation model. Experiments on three conversational tutoring datasets show that our model outperforms various standard baselines by a significant margin. Finally, we ended with an interesting case study to demonstrate the importance of strategizing before teaching.

Limitations

There are only three publicly available datasets (CIMA, TSCC and TalkMoves) for conversational tutoring task and they are quite small (less than 10K instances). There are significant data imbalance problems in these datasets – some teaching strategies occur much more frequently than others. These small and imbalanced datasets bring a lot of challenges to this task, but we did not discuss these issues in our paper due to the space limit. Besides, there are no standard teaching strategy annotation schemes, which prevents us from combining these three datasets together for more interesting experimental analyses. Another limitation of our work is that we only evaluate our approaches on automatic generation metrics. In the future, it would be interesting to also evaluate the model on learning related evaluations.

References

- Quincy Brown. 2009. Mobile intelligent tutoring system: moving intelligent tutoring systems off the desktop.
- Andrew Caines, Helen Yannakoudakis, Helena Edmondson, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. 2020. The teacher-student chat-room corpus. In *Proceedings of the 9th Workshop on NLP for Computer Assisted Language Learning*, pages 10–20.
- Jaime R Carbonell. 1970. Ai in cai: An artificial-intelligence approach to computer-assisted instruction. *IEEE transactions on man-machine systems*, 11(4):190–202.
- Elizabeth Emily Carter. 2014. An intelligent debugging tutor for novice computer science students.
- Robert Costello. 2012. *Adaptive intelligent personalised learning (AIPL) environment*. Ph.D. thesis, University of Hull.
- Beate Grawemeyer, Manolis Mavrikis, Wayne Holmes, Sergio Gutierrez-Santos, Michael Wiedmann, and Nikol Rummel. 2016. Affecting off-task behaviour: how affect-aware feedback can improve student learning. In *Proceedings of the sixth international conference on learning analytics & knowledge*, pages 104–113.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).
- Sanghyun S Jeon and Stanley YW Su. 2010. Adaptive e-learning using ecpaa rules, bayesian models, and group profile and performance data. *International Journal of Learning Technology*, 5(4):415–434.
- Aytürk Keleş, Rahim Ocak, Ali Keleş, and Aslan Gülcü. 2009. Zosmat: Web-based intelligent tutoring system for teaching–learning process. *Expert Systems with Applications*, 36(2):1229–1239.
- Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.
- Utku Kose and Ahmet Arslan. 2017. Optimization of self-learning in computer engineering courses: An intelligent software system supported by artificial neural network and vortex optimization algorithm. *Computer Applications in Engineering Education*, 25(1):142–156.
- Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. *Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures*. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. *BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Jakub Macina, Nico Daheim, Lingzhi Wang, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. Opportunities and challenges in neural dialog tutoring. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayan-deh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

- Martha C Polson and J Jeffrey Richardson. 2013. *Foundations of intelligent tutoring systems*. Psychology Press.
- Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.
- Jeffrey Ralph James Richardson. 1988. *Foundations of intelligent tutoring systems*. Psychology Press.
- Timo Schick and Hinrich Schütze. 2021. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402.
- Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. [Building end-to-end dialogue systems using generative hierarchical neural network models](#). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3776–3784. AAAI Press.
- Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. [A neural network approach to context-sensitive generation of conversational responses](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.
- Katherine Stasaski, Kimberly Kao, and Marti A. Hearst. 2020. [CIMA: A large open access dialogue dataset for tutoring](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64, Seattle, WA, USA. Online. Association for Computational Linguistics.
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H. Martin, and Tamara Sumner. 2022. [The TalkMoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves](#). pages 4654–4662.
- Abhijit Suresh, Tamara Sumner, Jennifer Jacobs, Bill Foland, and Wayne Ward. 2019. Automating analysis and feedback to improve mathematics teachers’ classroom discourse. In *Proceedings of the the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, volume 33, pages 9721–9728.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Dinesha Weragama and Jim Reye. 2014. Analysing student programs in the php intelligent tutoring system. *International Journal of Artificial Intelligence in Education*, 24(2):162–188.
- Chen Xing, Wei Wu, Yu Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. [Topic aware neural response generation](#). In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3351–3357. AAAI Press.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.
- Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskenazi. 2017. [Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 27–36, Saarbrücken, Germany. Association for Computational Linguistics.

ICA-Proto: Iterative Cross Alignment Prototypical Network for Incremental Few-Shot Relation Classification

Wangjie Jiang^{1,*}, Zhihao Ye^{2,*}, Bang Liu³, Ruihui Zhao²

Jianguang Zheng², Mengyao Li⁴, Zhiyong Li⁴, Yujiu Yang^{1,†}, Yefeng Zheng^{2,†}

¹Tsinghua Shenzhen International Graduate School, Tsinghua University

²Tencent Jarvis Lab, ³Université de Montréal Mila & CIFAR, ⁴Hunan University

jwj20@mails.tsinghua.edu.cn, evanzhye@tencent.com

yang.yujiu@sz.tsinghua.edu.cn, yefengzheng@tencent.com

Abstract

In the task of incremental few-shot relation classification, model performance is always limited by the incompatibility between the base feature embedding space and the novel feature embedding space. To tackle the issue, we propose a novel model named ICA-Proto: Iterative Cross Alignment prototypical network. Specifically, we incorporate the query representation into the encoding of novel prototypes and utilize the query-aware prototypes to update the query representation at the same time. Further, we implement the above process iteratively to achieve more interaction. In addition, a novel prototype quadruplet loss is designed to regulate the spatial distributions of embedding space, so as to make it easier for the relation classification. Experimental results on two benchmark datasets demonstrate that ICA-Proto significantly outperforms the state-of-the-art baseline model.

1 Introduction

Relation classification (RC) is an important sub-task of relation extraction (RE), aims at classifying the relation between two marked entities in a given sentence. For example, the instance “[Newton]_{e1} served as the president of [the Royal Society]_{e2}” expresses the relation *member_of* between the two entities *Newton* and *the Royal Society*. Some conventional methods (Zeng et al., 2014; Gormley et al., 2015; Soares et al., 2019) for relation classification adopt supervised training and usually suffer from the scarcity of manually annotated data. To alleviate this problem, distant supervision (DS) is adopted to automatically label abundant training instances by heuristically aligning knowledge graphs (KGs) with texts (Mintz et al., 2009). However, existing DS-based methods fail to deal with the problem of long-tail relations in KGs and still suffer from data deficiency (Han et al., 2018).

* Equal contribution.

† Corresponding authors.

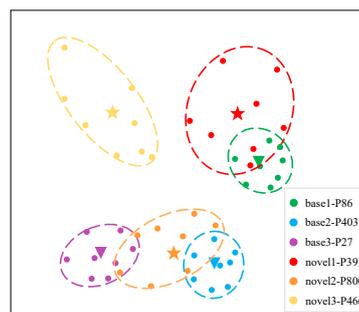


Figure 1: Visualization of the representations of the query instances and prototypes of BERT-IncreProto. We randomly sampled three base relations and three novel relations from the real-world dataset FewRel 1.0, each relation with its corresponding prototype (triangles for base relations and stars for novel relations) and eight query instances (points).

To address the above long-tail problem, few-shot RC was proposed, which formulates RC in a few-shot learning scenario. This task requires the models trained with base relations to generalize well to novel relations with only few labeled instances. Base relations are those relations that contain adequate instances and can be utilized effectively in the training phase to mimic the test phase on novel relations with few samples. Fine-tuning pre-trained models (Bengio, 2012; Gao et al., 2020) is straightforward while suffering from the overfitting problem. Therefore, metric-based methods (Ravi and Larochelle, 2017; Dong et al., 2020; Geng et al., 2020; Liu et al., 2020b) were proposed to grasp the fast-learning ability from previous experiences and then quickly generalize to new concept. These methods have been experimentally proven to be effective.

Taking a step further, incremental few-shot RC (Ren et al., 2020) considers a more realistic scenario, where the model is required to dynamically recognize the novel relations with a few samples, without reducing the base relation identification

capability learned on the large-scale data of base relations. Hence in the test phase, the query set consists of instances of not only base relations but also novel relations, which is more challenging. Several related works (Liu et al., 2020a; Chen and Lee, 2020; Kukleva et al., 2021) have been proposed in the field of computer vision, focusing on image classification task. As for the task of incremental few-shot RC, InceProtoNet (Ren et al., 2020) is the first work, which proposes a two-phase prototypical network model.

Specifically, InceProtoNet contains two separate prototypical networks (Snell et al., 2017). One is pre-trained in the first phase to acquire the base prototypes and base feature extractor, and the other obtains the novel prototypes and novel feature encoder with few-shot episode training in the second phase. However, InceProtoNet suffers from insufficient interaction between the class prototypes and the query instances. Therefore, in the embedding space, the novel relations often overlap significantly with the base relations, and the query representations are scattered, as shown in Figure 1. In addition, the triplet loss used by InceProtoNet may be affected by noise samples, and its effectiveness decreases on tasks with domain shift. As a result, a low accuracy in the recognition of novel relationships has been observed.

To alleviate the above problem, we propose a novel model named ICA-Proto that contains a specially-designed ICA module. ICA module consists of two sub-modules, i.e., *Cross Alignment* (CA) and *Iterative Alignment* (IA). Specifically, CA is built to dynamically and interactively encode the novel prototypes and query instances. On the one hand, the obtaining of novel prototypes is query-aware, namely that the query-related support instances contribute more to the final prototypes. On the other hand, the encoding of query instances is prototype-aware, since the query-related prototypes have more influence on the query representations. Furthermore, to achieve sufficient interaction and alignment, we construct IA, which is to implement the above CA iteratively. In addition, *Prototype Quadruplet* (PQ) loss is proposed to enlarge the distance between different types of prototypes, while making the distance between query and prototype of the same class as close as possible.

The contributions of this paper can be summarized below:

- We propose a novel incremental few-shot clas-

sification model ICA-Proto, which is able to dynamically recognize the novel relations with a few support instances.

- We design a novel and effective ICA module which learns the representations of the query instances and the novel prototypes interactively and iteratively. Besides, a novel prototype quadruplet loss is presented to regulate the feature space distribution.
- Experiments on FewRel 1.0 and 2.0 datasets demonstrate that our method significantly outperforms the state-of-the-art method.

2 Task Formulation

In the task of incremental few-shot RC, first we are given a large dataset containing N_{base} base relations: $D_{base} = \cup_{b=1}^{N_{base}} \{I_{b,i} = (x_{b,i}, h_{b,i}, t_{b,i}, r_b)\}_{i=1}^{K_b}$, in which K_b is the number of instances of relation r_b , and $I_{b,i}$ represents its i -th instance consisting of the sentence $x_{b,i}$ and the mentioned entity pair $(h_{b,i}, t_{b,i})$. Then we are given a support set

$S = \cup_{n=1}^{N_{novel}} \{I'_{n,i}\}_{i=1}^{K'_n}$ of N_{novel} novel relations,

where K'_n is the number of support instances of novel relation r'_n and $I'_{n,i}$ is the i -th supporting instance. With D_{base} and S , the task is to recognize the relations of the instances in the query set

$Q = \cup_{q=1}^{N_{base}+N_{novel}} \{I''_{q,i}\}_{i=1}^{K''_q}$, in which K''_q is the

number of query instances of relation r''_q and $I''_{q,i}$ is its i -th query instance. Therefore, the model is required to dynamically recognize the novel relations based on a few novel support instances while keeping the base relation identification capability learned on the large base dataset.

3 Method

In this section, we elaborate on the details of our proposed ICA-Proto model for incremental few-shot RC. First, we give a brief introduction to the InceProtoNet in Section 3.1. Then, we introduce the overall framework of our model in Section 3.2. Next, we present the proposed ICA module with CA and IA sub-modules in Section 3.3. Moreover, the proposed PQ loss is discussed in Section 3.4.

3.1 Introduction to InceProtoNet

InceProtoNet (Ren et al., 2020) is the first work focusing on incremental few-shot RC. The proposed model is a two-phase prototypical network.

In the first phase, a deep prototypical network, consisting of a convolutional neural network based encoder and a prototype based classifier, is pre-trained on a large training dataset for base relations in a supervised manner to learn the feature embedding space of base relations. Therefore, the base prototypes, denoted $P_{base} = \{p_1, p_2, \dots, p_{N_{base}}\}$, can be obtained by averaging the representations of all training instances within each base class b :

$$p_b = \frac{1}{K_b} \sum_{i=1}^{K_b} x_{b,i}, \quad (1)$$

where $x_{b,i}$ is the embedding of $I_{b,i}$ through the base encoder.

In the second phase, another prototypical network, named incremental few-shot prototypical network, is proposed to learn the feature embedding space of novel relations. The support set is encoded to obtain the novel prototypes $P_{novel} = \{p'_1, p'_2, \dots, p'_{N_{novel}}\}$ as follows:

$$p'_n = \frac{1}{K'_n} \sum_{i=1}^{K'_n} x'_{n,i}, \quad (2)$$

where $x'_{n,i}$ is the embedding of $I'_{n,i}$ through the novel encoder. For a query instance q from the query set, the representation x_q is calculated as the weighted sum of the x_q^{base} from the base feature embedding space and x_q^{novel} from the novel feature embedding space:

$$x_q = \omega_b x_q^{base} + \omega_n x_q^{novel}, \quad (3)$$

where the weights ω_b and ω_n are determined by considering the similarity of the query representation with the base prototypes P_{base} and novel prototypes P_{novel} , respectively. To better show the relationships, we summarize and rewrite the query representation calculation equation (3) as:

$$x_q = f(x_q^{base}, x_q^{novel}, P_{base}, P_{novel}), \quad (4)$$

where f is a composite function and represents a series of attention operations. More details can be found in the original paper (Ren et al., 2020). Lastly, the probability of q belonging to the i -th relation r_i can be measured as:

$$p_\theta(r_i | q) = \frac{\exp(-d(\mathbf{x}_q, \mathbf{p}_i^{all}))}{\sum_{j=1}^{N_{base} + N_{novel}} \exp(-d(\mathbf{x}_q, \mathbf{p}_j^{all}))}, \quad (5)$$

where \mathbf{p}_i^{all} is the i -th prototype in $\mathbf{P}_{all} = \{P_{base}, P_{novel}\}$.

Although IncreProtoNet performs well in recognizing instances of base relations, it is still difficult for this model to deal with novel relations. The experimental results in Ren et al. (2020) show that the accuracy for novel relations is much lower than that of base relations, which is unsatisfactory. There are several reasons as follows. First, IncreProtoNet obtains the novel prototypes independent of the query instance, lacking interaction between them. Second, IncreProtoNet ignores the alignment between base relations and novel relations, which is vital in incremental learning scenarios. Third, there is no effective regulation to the feature embedding spaces of base relations and novel relations, which causes discrepancy between them.

3.2 Overall Framework of ICA-Proto

To tackle the above issues, we propose the ICA-Proto model on the basis of IncreProtoNet. Similar to IncreProtoNet, our model contains two stages, including the base pretraining stage and the few-shot episode training stage. Furthermore, we innovatively propose the ICA module and PQ loss, of which ICA module is demonstrated in the dashed boxes in Figure 2.

3.3 Iterative Cross Alignment

In the task of incremental few-shot RC, it is important to make an alignment between the base feature embedding space and the novel feature embedding space so as to flexibly encode the query instance and further make correct relation classification. This requires full interaction between base relations and novel relations. To this end, we propose the ICA module, which consists of CA and IA sub-modules.

Cross Alignment. To this end, the CA sub-module is designed to encode the novel prototypes and the query instance in an interactive manner. To be specific, we first initialize the novel prototypes P_{novel} and the query instance embedding x_q with equations (2) and (4), respectively. Then, CA updates $p'_n \in P_{novel}$, encouraging the model pay more attention to those query-related supporting instances,

$$p'_n = \sum_{i=1}^{K'_n} \gamma_{n,i} x'_{n,i}, \quad (6)$$

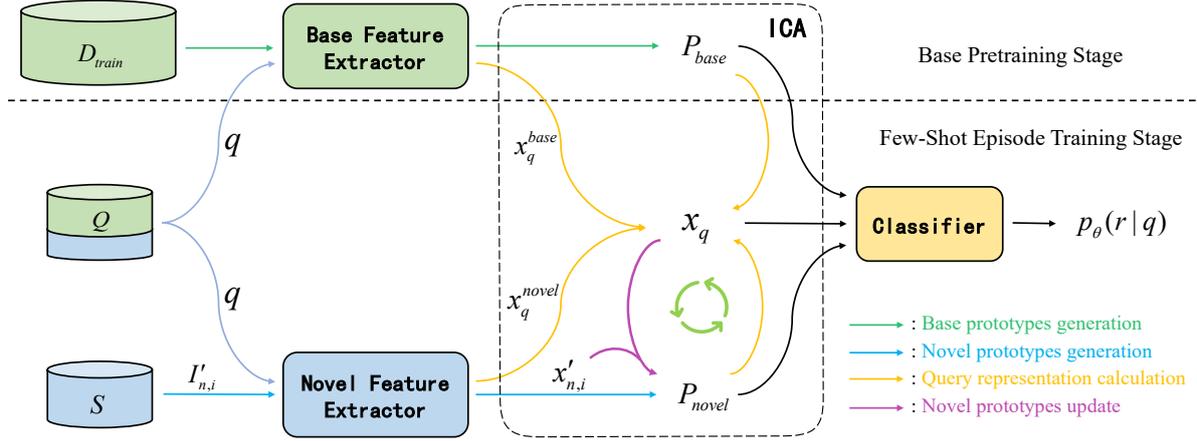


Figure 2: The framework of ICA-Proto. In the dashed box representing ICA module, the yellow arrows refer to the process of the query representation calculation, while the purple arrows means the process of the novel prototypes update. The green loop arrows represents the iterative refining of both query representation and novel prototypes.

where $\gamma_{n,i}$ is defined as:

$$\gamma_{n,i} = \frac{\exp(-d(x_q, x'_{n,i}))}{\sum_{i=1}^{K'_n} \exp(-d(x_q, x'_{n,i}))}, \quad (7)$$

where d is the euclidean distance. In short, the novel prototype embedding process can be summarized as:

$$P_{novel} = g(x_q, \cup_{n=1}^{N_{novel}} \{I'_{n,i}\}_{i=1}^{K'_n}). \quad (8)$$

Correspondingly, the query instance representation x_q is further updated with equation (4), which requires the model to pay more attention to the query-related base prototypes and novel prototypes. Since most of the query instances belong to base relations, CA actually enhances the interaction between instances of base relations and novel relations, achieving better alignment between the two feature embedding spaces.

Iterative Alignment. The aligned query representation can help group the different support samples from the same novel class together to optimize the novel prototype. Meanwhile, the optimized novel prototype can further help align query representations from different encoders. Inspired by traditional iterative cross-optimization algorithms, such as the EM (McLachlan and Krishnan, 2007) or k -means (Hartigan and Wong, 1979) algorithms, we further propose to carry out the above CA in an iterative way, namely Iterative Alignment (IA). The implementation is straightforward, since we just need to iteratively update P_{novel} and x_q with equations (6) and (4), respectively, until the predefined

Algorithm 1 Iterative Cross Alignment

Input: Base prototypes P_{base} , support set S , query instance q and predefined maximum iteration number N .

Parameter: Base encoder Θ_1 and novel encoder Θ_2 .

Output: Novel prototypes P_{novel} , query instance representation x_q and probability distribution for relation of q : $p_{\theta}(r | q)$.

- 1: Initialize novel prototypes P_{novel} with equation (1).
 - 2: Initialize query instance representation x_q with equation (2).
 - 3: **for** $t = 1 \rightarrow N$ **do**
 - 4: Update query representation x_q^t :
 $x_q^t = f(x_q^{base}, x_q^{novel}, P_{base}, P_{novel}^{t-1})$,
 - 5: Update novel prototypes P_{novel}^{t+1} :
 $P_{novel}^{t+1} = g(x_q^t, \cup_{n=1}^{N_{novel}} \{I'_{n,i}\}_{i=1}^{K'_n})$.
 - 6: **end for**
 - 7: **return** P_{novel} , x_q and $p_{\theta}(r | q)$.
-

maximum number of steps is reached. Finally, the refined novel prototypes and query instance representations are obtained. The IA expands CA from single round to multiple rounds, further promoting the interaction and alignment.

Algorithm 1 outlines the key steps of our ICA module.

ICA for Increment Few-Shot Domain Adaptation. In the real world, especially in the few-shot

scenario, the test domain (new classes) and training domain (base classes) are often different, so how to improve the ability of our model to transfer across domains is also very important. Since the test domain usually has no annotations and could differ vastly from the training domain, we first initialize novel class prototypes with average representation of support set instances and query representations with initialized novel class prototypes. Then CA cross-aligns novel support instances and query from different domains. Furthermore, in the cross-domain scenario, the initial query and the novel prototypes are more likely to be incompatible; therefore, the ICA module can significantly improve the representations of the novel prototypes and the query from different domains.

3.4 Prototype Quadruplet Loss

In our method, there are two feature embedding spaces for base and novel classes separately and the query instance is encoded by the two jointly. Therefore, it is important to measure which embedding space contributes more and further estimate which prototype is the nearest. In addition, the feature spaces of base classes and novel classes should be separated as much as possible when they are embedded into the same space. To this end, we design a novel *Prototype Quadruplet* loss (\mathcal{L}_{PQ}), denoted as follows:

$$\mathcal{L}_{PQ} = \sum_{i=1}^M \sum_{k=1}^{N_{novel}} \max(0, \delta_1 + d_1 - d_2) + \max(0, \delta_2 + d_1 - d_3), \quad (9)$$

where δ_1 and δ_2 are hyper-parameters, M is the total number of training episodes, and three distances d_1 , d_2 , d_3 are defined as follows:

$$d_1 = d\left(f\left(a_i^k\right), P_{p,i}^k\right), \quad (10)$$

$$d_2 = d\left(f\left(a_i^k\right), P_{n,i}^k\right), \quad (11)$$

$$d_3 = d\left(P_{n,novel,i}^k, P_{n,base,i}^k\right), \quad (12)$$

where $\left(a_i^k, P_{p,i}^k, P_{n,novel,i}^k, P_{n,base,i}^k\right)$ is a quadruplet consisting of the anchor instance, the positive prototype from the same novel class, the first negative prototype from another novel class and the second negative prototype from one of the base classes, $f(\cdot)$ is the feature extractor, and $P_{n,i}^k$ is

randomly selected from $P_{n,novel,i}^k$ or $P_{n,base,i}^k$. Unlike IncreProtoNet, inspired by the triplet-center loss (He et al., 2018), which can further enhance the discriminative power of the features, we also learn the center representation of each class and then require that the distances between anchors and centers from the same class are smaller than those from different classes. Note that $P_{p,i}^k$, $P_{n,novel,i}^k$, $P_{n,base,i}^k$ are all virtual instances and denote the corresponding prototypes.

In addition, to enhance the abilities of our model to transfer across domains, inspired by the quadruplet loss (Chen et al., 2017) which introduces the absolute distance between the positive and negative sample pairs, we add d_3 to better align different domains, which narrows the domain gap and further alleviates the issue of incompatible feature embedding between base classes and novel classes, so as to achieve more effective domain adaptation.

Finally, the joint loss function \mathcal{L} is a trade-off between the cross-entropy loss \mathcal{L}_{CE} and the above \mathcal{L}_{PQ} by a hyper-parameter λ :

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \cdot \mathcal{L}_{PQ}. \quad (13)$$

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. We carry out extensive experiments on two benchmark datasets. The first one is FewRel 1.0 (Han et al., 2018), which contains 80 relations and provides 700 instances for each relation. We adopt the same split as Ren et al. (2020). To be specific, 54 relations are randomly selected as the base relations each with 550 instances for base pre-training, 50 instances for episode training and 100 instances for testing. 10 other relations each with 700 instances are sampled as the novel relations for the episode training. The rest 16 relations each with 700 instances are used as the novel relations in testing. The other dataset is FewRel 2.0 (Gao et al., 2019b), which is constructed on top of the FewRel 1.0 by adding a new test set in a quite different domain (i.e., medicine), requiring the models to transfer across domains.

Evaluation Metrics. To compare our proposed method with the state-of-the-art methods, we adopt the same evaluation metrics as Ren et al. (2020), namely, three kinds of classification accuracy, including classification accuracy for instances of base relations, novel relations, and all relations. Since

Table 1: Average classification accuracy (%) on the FewRel 1.0 dataset.

Models	1-shot learning			5-shot learning		
	Base	Novel	Both	Base	Novel	Both
Proto	43.20 ± 0.12	39.86 ± 0.26	42.91 ± 0.22	66.74 ± 0.05	57.33 ± 0.15	65.94 ± 0.11
HATT-Proto	51.58 ± 0.11	45.16 ± 0.18	51.03 ± 0.15	67.77 ± 0.13	61.12 ± 0.09	67.20 ± 0.08
BERT-PAIR	76.03 ± 0.05	58.29 ± 0.13	75.30 ± 0.11	80.01 ± 0.03	64.34 ± 0.14	78.68 ± 0.12
ProtoNet (Increment)	75.63 ± 0.04	18.44 ± 0.02	70.78 ± 0.03	75.07 ± 0.03	47.11 ± 0.04	72.70 ± 0.02
Imprint	62.62 ± 0.13	16.79 ± 0.34	58.73 ± 0.27	67.72 ± 0.09	16.49 ± 0.31	63.38 ± 0.25
AttractorNet	66.48 ± 0.19	5.32 ± 0.25	61.29 ± 0.23	68.26 ± 0.22	6.45 ± 0.26	62.78 ± 0.24
GloVe-IncreProtoNet	70.96 ± 0.21	48.38 ± 0.11	69.36 ± 0.15	72.54 ± 0.16	61.57 ± 0.11	71.54 ± 0.13
GloVe-ICA-Proto	72.15 ± 0.18	54.47 ± 0.04	70.65 ± 0.08	72.70 ± 0.06	71.91 ± 0.10	72.63 ± 0.13
BERT-IncreProtoNet	82.10 ± 0.04	60.15 ± 0.11	80.65 ± 0.10	84.64 ± 0.04	65.77 ± 0.09	82.26 ± 0.08
BERT-ICA-Proto	82.56 ± 0.02	63.25 ± 0.09	80.93 ± 0.08	84.89 ± 0.05	69.49 ± 0.06	83.59 ± 0.04

the number of base relations is much larger than that of novel relations, the classification accuracy for instances of all relations depends largely on that of base relations.

4.2 Implementation Details

To systematically validate the effectiveness of the proposed ICA-Proto model, we experiment with two kinds of word embedding initialization methods, namely, GloVe (Pennington et al., 2014) and BERT (Devlin et al., 2019). Besides, the compared methods are all evaluated in both 1-shot and 5-shot learning. The hidden dimension of feature extractor is 230, as well as the prototype dimension. The stochastic gradient descent (SGD) is employed for optimization and the initial learning rate in episode training is set as 0.1, except for BERT as 0.001. For the PQ loss, the two margins δ_1 and δ_2 are set as 5.0 and 10.0 respectively, while the balance weight λ is set as 1.0.

4.3 Comparison Methods

First of all, we compare with several few-shot learning models, namely, Proto (Han et al., 2018), HATT-Proto (Gao et al., 2019a) and BERT-PAIR (Gao et al., 2019b) and the incremental few-shot learning model ProtoNet (Increment) (Snell et al., 2017). Besides, following (Ren et al., 2020), we compare with Imprint (Qi et al., 2018) and LwoF (Gidaris and Komodakis, 2018), which are the incremental few-shot learning models in the computer vision field. Finally, we take IncreProtoNet as our baseline, which is the current state of the art.

4.4 Main Results

Our model gains significant improvement in incremental few-shot learning tasks. From Table 1, we can observe that for the FewRel 1.0 dataset, our model achieves the best in both 1-shot and 5-shot tasks. Compared with the best baseline model IncreProtoNet, our model remarkably improves the novel class classification accuracy by 3-10%, while maintaining high accuracy on base class recognition. This shows that the proposed ICA module and PQ loss can greatly promote the models' recognition capabilities for novel classes. We conjecture this is because the ICA module can obtain more effective novel prototypes and better align the query representations from different encoders.

The more support set instances, the larger the improvement for novel class classification. As can be seen from Table 1, using either GloVe or BERT as the initial text encoder, the improvement on the 5-shot learning is more significant than that of 1-shot learning for novel class. This is because when there are more support set samples, the ICA module and PQ loss can help separate the base and novel classes, reduce the distance between similar classes, and make the query of novel class and corresponding prototype as close as possible.

4.5 Domain Adaptation Results

To further demonstrate the superiority of our method, we extend the few-shot domain adaptation (few-shot DA) task in FewRel 2.0 (Gao et al., 2019b) to the incremental few-shot domain adaptation (inre-few-shot DA) task in our work. Different from the original incre-few-shot RC, the novel instances in the test set are replaced by new instances from the medical domain. Since the do-

Table 2: Results (%) of incre-few-shot DA on the FewRel 2.0 dataset.

Models	1-shot learning			5-shot learning		
	Base	Novel	Both	Base	Novel	Both
GloVe-IncreProtoNet	71.37 ± 0.25	36.85 ± 0.13	68.44 ± 0.18	71.71 ± 0.22	49.15 ± 0.14	69.80 ± 0.17
GloVe-ICA-Proto	71.39 ± 0.11	37.03 ± 0.15	68.48 ± 0.14	73.11 ± 0.15	55.58 ± 0.10	71.63 ± 0.11
BERT-IncreProtoNet	86.27 ± 0.06	52.68 ± 0.20	83.42 ± 0.11	87.83 ± 0.05	56.70 ± 0.14	85.19 ± 0.09
BERT-ICA-Proto	86.72 ± 0.04	52.85 ± 0.16	83.85 ± 0.12	87.49 ± 0.16	65.27 ± 0.08	85.60 ± 0.14

Table 3: Ablation Studies. † indicates ICA-Proto without the ICA module; and ‡ indicates ICA-Proto without the PQ loss.

Models	1-shot learning			5-shot learning		
	Base	Novel	Both	Base	Novel	Both
GloVe-IncreProtoNet	70.96 ± 0.21	48.38 ± 0.11	69.36 ± 0.15	72.54 ± 0.16	61.57 ± 0.11	71.54 ± 0.13
GloVe-ICA-Proto †	72.03 ± 0.12	52.47 ± 0.05	69.42 ± 0.01	72.32 ± 0.04	67.36 ± 0.10	71.94 ± 0.08
GloVe-ICA-Proto ‡	71.15 ± 0.03	53.97 ± 0.12	69.82 ± 0.10	71.12 ± 0.06	69.14 ± 0.16	71.64 ± 0.11
GloVe-ICA-Proto	72.15 ± 0.18	54.47 ± 0.04	70.42 ± 0.08	72.70 ± 0.06	71.91 ± 0.10	72.63 ± 0.13
BERT-IncreProtoNet	82.10 ± 0.04	60.15 ± 0.11	80.65 ± 0.10	84.64 ± 0.04	65.77 ± 0.09	82.26 ± 0.08
BERT-ICA-Proto †	82.20 ± 0.13	62.72 ± 0.15	80.67 ± 0.08	84.04 ± 0.12	68.06 ± 0.28	82.15 ± 0.10
BERT-ICA-Proto ‡	82.15 ± 0.14	63.07 ± 0.09	80.92 ± 0.13	84.98 ± 0.10	69.36 ± 0.12	83.25 ± 0.15
BERT-ICA-Proto	82.56 ± 0.02	63.25 ± 0.09	81.50 ± 0.08	84.90 ± 0.05	69.50 ± 0.06	83.64 ± 0.04

main of novel instances in the test set is no longer consistent with the training set, the models are required to be able to transfer across domains, which is more challenging.

Table 2 illustrates the comparison results of Incre-ProtoNet and our model, and we have two observations: (1) Huge drops on almost all metrics have been witnessed for both IncreProtoNet and our model, which demonstrates the difficulty of incre-few-shot DA. However, the performance of our method deteriorates much slower than that of IncreProtoNet. (2) Our model outperforms Incre-ProtoNet on all metrics. Especially in 5-shot settings, the accuracy of novel relation recognition is improved by more than 7% in absolute percentage. It indicates that our proposed ICA module provides more accurate, robust and general representations for the relation prototypes and query instances.

4.6 Ablation Studies

As shown in Table 3, on the FewRel 1.0 dataset, compared with the baseline IncreProtoNet, our model can get a large improvement with either the ICA module or the PQ loss. Especially for the ICA module, benefited from the full interaction brought by it, better query representation and novel prototype representation greatly improve the model’s ability in incremental few-shot learning tasks. Fur-

thermore, these two designs are complementary to each other, and combining them together, we can achieve even larger improvement.

4.7 Visualization Analysis

We visualize different types of query representations and prototype representations. As shown in Figure 3, benefited from the ICA module and PQ loss, prototypes of different classes are pushed apart, and the representations of different queries are more accurate and fall close to the corresponding prototype of the same class.

4.8 Impact of the Iteration Number in ICA

As shown in Table 4, the ICA module with two (N=2) or three (N=3) iterations achieves better results than the single iteration (N=1). This shows that the ICA module which optimizes query representation and novel prototype representation step by step can effectively improve the accuracy of incremental few-shot learning. In addition, when N is greater than 3, the accuracy of the model decreases. The reason is probably that larger N leads to overfitting of the model. Finally, it can be seen from Table 4 that no matter how many times the model is iteratively aligned, our models are significantly better than the current best baseline IncreProtoNet.

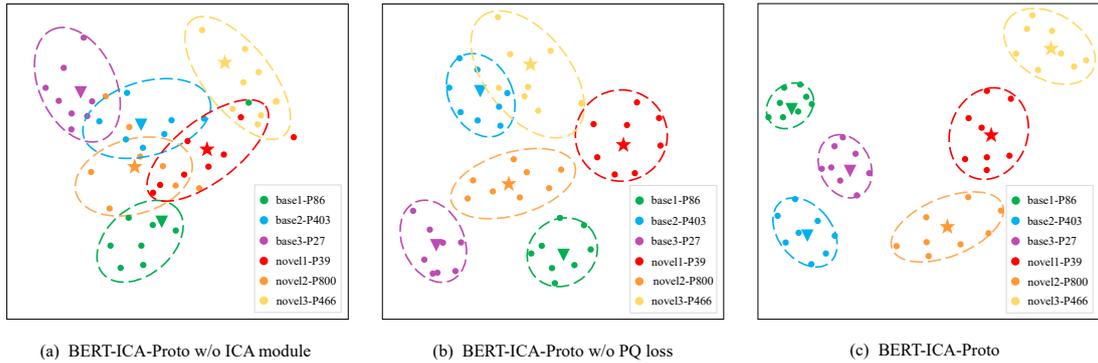


Figure 3: Visualization of the representations of the query instances and prototypes when BERT-ICA-Proto is equipped (a) without ICA module and (b) without PQ loss.

Table 4: Impact of the iteration number in ICA module.

Models	5-shot learning		
	Base	Novel	Both
GloVe-IncreProtoNet	72.43	61.57	71.54
GloVe-ICA-Proto (N=1)	72.33	69.91	72.12
GloVe-ICA-Proto (N=2)	72.55	68.91	72.24
GloVe-ICA-Proto (N=3)	72.70	71.91	72.63
GloVe-ICA-Proto (N=4)	72.77	70.01	72.53
BERT-IncreProtoNet	84.54	65.77	82.26
BERT-ICA-Proto (N=1)	84.25	67.50	82.83
BERT-ICA-Proto (N=2)	84.36	69.50	83.10
BERT-ICA-Proto (N=3)	84.89	69.49	83.58
BERT-ICA-Proto (N=4)	84.43	68.10	82.06

5 Related Work

RC is a fundamental task in natural language processing, aiming to recognize the semantic relation between two marked entities in a sentence. With the development of deep learning in recent years, many models based on neural networks have been proposed for this task and achieved great progress. For example, Zeng et al. (2014) and dos Santos et al. (2015) utilized convolutional neural networks to capture the global and local semantic information. Later, some attention-based models (Wang et al., 2016; Zhou et al., 2016; Jin et al., 2020) have been proposed to better capture the more useful semantic information. These models may suffer from the scarcity of high-quality training data. To mitigate the problem, some works (Mintz et al., 2009; Jia et al., 2019; Qin et al., 2018) adopt DS to construct large-scale datasets, while ignore the effect of long-tail relations.

Few-shot RC aims to learn high-quality features with only a small number of training samples. Early

works employed the paradigm of pretraining and fine-tuning (Bengio, 2012; Donahue et al., 2014; Gao et al., 2020), which aimed to acquire and transfer knowledge from support set containing instances of common relations. Later, metric learning methods (Vinyals et al., 2016; Snell et al., 2017) were proposed to learn different representations across relations. One representative work is prototypical networks (Snell et al., 2017), aiming to learn robust class representations and classify the query set based on the distance to the class prototypes in the feature space. A series of works (Han et al., 2018; Gao et al., 2019a,b) employed prototypical network in few-shot RC and achieved excellent performance.

Incremental learning is a setting where new information is arriving continuously while prior knowledge needs to be maintained. Combining incremental learning with few-shot RC, incremental few-shot RC constitutes a more realistic scenario, where the model is required to leverage the representations of base relations learned from large-scale training dataset meanwhile effectively learn the representations of novel relations from a few support instances. To deal with this task, Ren et al. (2020) proposed a prototypical network based model consisting of two encoders for base relations and novel relations, respectively. In this paper, we argue that the previous work (Ren et al., 2020) is sub-optimal and introduce a preferable solution.

6 Conclusion

In this paper, we presented a novel and effective approach with iterative cross alignment module and prototype quadruplet loss for the task of incremental few-shot learning. Benefit from the extensive interaction offered by the iterative cross alignment

and the feature space regulation brought by the prototype quadruplet loss, our method outperformed the state-of-the-art baseline method significantly, as verified in our extensive experiments. In future work, we aim to further improve the performance of our model under the one-shot task setting, as well as accelerate the training process.

Limitations

In this paper, we propose a novel model named ICA-Proto for the task of incremental few-shot relation classification. Experimental results have shown that our method outperforms the existing best baselines. However, there are two major limitations. First, our method iteratively calculates the representations of query instances and relation prototypes, which is more time-consuming. Second, the best iteration number in ICA module may vary with different datasets. Therefore, we should conduct extra experiments to determine the best iteration number when applying our method in a new dataset, which is not convenient enough to some degree.

Acknowledgements

This work was partly supported by the National Key Research and Development Program of China (No. 2020YFB1708200) and the Shenzhen Key Laboratory of Marine IntelliSense and Computation under Contract ZDSYS20200811142605016.

References

- Yoshua Bengio. 2012. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 17–36.
- Kuilin Chen and Chi-Guhn Lee. 2020. Incremental few-shot learning via vector quantization in deep embedded space. In *International Conference on Learning Representations*.
- Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. 2017. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–412.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. DeCAF: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, pages 647–655.
- Bowen Dong, Yuan Yao, Ruobing Xie, Tianyu Gao, Xu Han, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2020. Meta-information guided meta-learning for few-shot relation classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1594–1605.
- Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.
- Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019a. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6407–6414.
- Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019b. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6251–6256.
- Xiaoqing Geng, Xiwen Chen, Kenny Q Zhu, Libin Shen, and Yingong Zhao. 2020. MICK: A meta-learning framework for few-shot relation classification with small training data. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 415–424.
- Spyros Gidaris and Nikos Komodakis. 2018. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375.
- Matthew R Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1774–1784.
- Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification

- dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.
- John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.
- Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. 2018. Triplet-center loss for multi-view 3D object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1945–1954.
- Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. ARNOR: Attention regularization based noise reduction for distant supervision relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1399–1408.
- Yanliang Jin, Dijia Wu, and Weisi Guo. 2020. Attention-based LSTM with filter mechanism for entity relation classification. *Symmetry*, 12(10):1729.
- Anna Kukleva, Hilde Kuehne, and Bernt Schiele. 2021. Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9020–9029.
- Qing Liu, Orchid Majumder, Alessandro Achille, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. 2020a. Incremental few-shot meta-learning via indirect discriminant alignment. In *European Conference on Computer Vision*, pages 685–701. Springer.
- Xiaoqian Liu, Fengyu Zhou, Jin Liu, and Lianjie Jiang. 2020b. Meta-learning based prototype-relation network for few-shot classification. *Neurocomputing*, 383:224–234.
- Geoffrey J McLachlan and Thriyambakam Krishnan. 2007. *The EM algorithm and extensions*, volume 382. John Wiley & Sons.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.
- Hang Qi, Matthew Brown, and David G Lowe. 2018. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5822–5830.
- Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2137–2147.
- Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *Proceedings of the 5th International Conference on Learning Representations*, pages 224–234.
- Haopeng Ren, Yi Cai, Xiaofeng Chen, Guohua Wang, and Qing Li. 2020. A two-phase prototypical network model for incremental few-shot relation classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1618–1629.
- Jake Snell, Kevin Swersky, and Richard S Zemel. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*.
- Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.
- Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29:3630–3638.
- Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307.
- Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.
- Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.

A Large-Scale Multilingual Study of Visual Constraints on Linguistic Selection of Descriptions

Uri Berger^{1,2}, Lea Frermann², Gabriel Stanovsky¹, and Omri Abend¹

¹School of Computer Science and Engineering, The Hebrew University of Jerusalem

²School of Computing and Information Systems, University of Melbourne
{uri.berger2, gabriel.stanovsky, omri.abend}@mail.huji.ac.il
lea.frermann@unimelb.edu.au

Abstract

We present a large, multilingual study into how vision constrains linguistic choice, covering four languages and five linguistic properties, such as verb transitivity or use of numerals. We propose a novel method that leverages existing corpora of images with captions written by native speakers, and apply it to nine corpora, comprising 600k images and 3M captions. We study the relation between visual input and linguistic choices by training classifiers to predict the probability of expressing a property from raw images, and find evidence supporting the claim that linguistic properties are constrained by visual context across languages. We complement this investigation with a corpus study, taking the test case of numerals. Specifically, we use existing annotations (number or type of objects) to investigate the effect of different visual conditions on the use of numeral expressions in captions, and show that similar patterns emerge across languages. Our methods and findings both confirm and extend existing research in the cognitive literature. We additionally discuss possible applications for language generation. We make our codebase publicly available.¹

1 Introduction

In recent years, vision and language models have been shown to outperform models trained on a single modality in a variety of domains, such as language modeling (Ororbia et al., 2019), document quality assessment (Shen et al., 2020), and visual classification and segmentation (Frome et al., 2013; Radford et al., 2021; Berger et al., 2022).

While empirical evidence exists, including from the studies cited above, that each modality can benefit the other for task performance, less attention has been devoted to the broader question of *how* the two modalities influence and constrain one another. In this study, we focus on one aspect of this question: *How does vision constrain language?*

¹github.com/SLAB-NLP/visual_constraints_on_descriptions



Figure 1: A demonstration of how visual cues in images may constrain linguistic choices in their captions. The image on the left in which the agent is visible is described using an active voice “A man... is *throwing* a child... in the air”, while in the right image the agent is not visible and the annotator chose a passive construction: “A little boy... is *thrown* in the air”. Images and captions taken from Flickr30k.

We study the relation between the semantic content of an image and the language used to describe it. As an example, consider Figure 1. Captions of the right image, which crops the agent, use passive voice more frequently than those of the left (taken from Flickr30k, Young et al., 2014). We aim to study such trends by examining the influence of visual features of the image on the linguistic choices taken when describing it.

A number of psycholinguistic studies have aimed to answer this question by systematically varying visual conditions (such as image cropping) and analyzing verbal descriptions of the scenes by human participants for recurring differences (e.g., Chesney and Gelman, 2015; Rissman et al., 2019). Although such controlled studies allow for precise measurement, the visual stimuli are synthetic (rather than depicting natural scenes), the manual annotation of descriptions limits the size of the dataset, and typically only one linguistic property is investigated in a single language.

We address this gap by proposing a scalable methodology that uses existing image-caption corpora in multiple languages. We measure the correlation of visual features and linguistic properties of

the caption by training visual classifiers to predict, for a given raw image, whether a linguistic property is expressed in its captions. We compare the impact of different training sets (single vs. multiple languages) and different types of pre-training (none vs. object categories vs. visual vs. textual pre-training objectives). We use 9 large-scale image-caption datasets (overall 2.9M captions for 604K images), covering four languages (English, German, Chinese, Japanese), to study lexical properties (use of numerals and negation words) and structural properties (use of passive voice, transitivity of the main verb, choice of verbal vs. nominal constructions), which we automatically annotate in the captions. To the best of our knowledge, this is the first large-scale, multilingual study of the impact of visual input on linguistic choice. We find evidence showing that the visual input imposes constraints on linguistic properties, and that such trends are detectable using the proposed methodology.

In a complementary corpus study, we link the prevalence of linguistic properties to existing, high-level visual annotations (number and type of objects), and find that these properties can be linked to the use of numeral expressions in similar patterns across languages, and in accordance with small-scale, highly-controlled psycholinguistic studies.

Our findings have both cognitive and computational implications. On the cognitive side, this study confirms findings from small-scale cognitive studies at scale: for naturalistic scenes, typologically diverse languages, and descriptions from thousands of native speakers. The magnitude of the data that can be studied with our method also allows the derivation of new insights, which can motivate additional controlled studies, making the proposed practice an effective exploration method.

On the computational side captioning models have been shown to generalize better when first predicting the syntactic structure of the generated caption (Bugliarello and Elliott, 2021). This research direction may benefit from the signal provided by our classifiers of linguistic properties.

2 Background

The notion that grammatical and lexical phenomena can be characterized semantically, at least in their prototypical instances, has a long tradition in linguistics (Dixon, 1979; Goldberg, 1995; Croft, 2012, among many others). For example, transitive clauses are often characterized as corresponding to

actions instigated by volitional agents over passive objects, that are affected by the action. However, formally defining these semantic features in a non-linguistic way and showing empirically that the presence of these features indeed entails the presence of the corresponding linguistic feature, has proven to be methodologically challenging. One possible direction to address this is the use of images that implicitly define a type of non-linguistic semantics. This section briefly reviews different approaches for studying visual constraints on our set of five phenomena from a cognitive and computational perspective (omitting phenomena that have not been previously covered).

2.1 Cognitive Studies

Numerals. *Subitizable* numbers are numbers that are rapidly and accurately visually counted by humans. Studies have shown that the threshold for subitizability is 4 (Kaufman et al., 1949; Mandler and Shebo, 1982), with Barr et al. (2013) showing that humans tend to describe non-subitizable numbers using quantifiers (e.g., *many*). In Section 5.2 we confirm this result at scale. Chesney and Gelman (2015) asked participants to count objects in a given image, and found that participants were less likely to include objects located in frames (windows, mirrors or picture frames) in their count, suggesting that visual cues influence linguistic choices.

Negation. Several studies have challenged the traditional view that images cannot express negation (Worth, 1981; Khemlani et al., 2012). Giora et al. (2009) use visual negation markers (e.g., red cross road signs) to study neural processing of visual negation. Oversteegen and Schilperoord (2014) ask Dutch native speakers to describe images of objects missing integral parts (e.g., a woman without a mouth) and show that the descriptions are likely to contain a negation word.

Passive voice. Myachykov et al. (2012) show that English native speakers have a stronger preference for using passive-voice when describing transitive events with visual cueing of their attention toward the agent (compared to the control condition).

Transitivity. Rissman et al. (2019) show that participants had a preference for intransitive descriptions of visual events (a person acting on an inanimate object) when the person was removed by cropping the image (whereas transitive descriptions were preferred in the base condition).

2.2 Computational Studies

Computational studies on how vision constrains language are rare. However, several studies examined various aspects of the linguistic properties studied in this work, typically focusing on individual properties and/or languages.

Negation. A series of studies (van Miltenburg et al., 2016, 2017), investigated negation in Flickr30k image descriptions using a smaller set of negation words compared to our study, comparing the use of negation in English, German, and Dutch, and finding no significant differences. Dobrev and Keller (2021) show that the performance of vision and language models decreases when the text contains negation, but did not show that this decrease is caused by negation-related visual features. Text-only models also have difficulty processing negations (e.g., Ettinger (2020)), and the drop in performance could be due to the text encoder alone.

The line of work most similar to this study train models to predict whether images from comics (Sato et al., 2021) or real life (Sato and Mineshima, 2021) express negation, achieving chance-level results. In contrast to the current study, they used a single dataset, a single language (Japanese), and a single linguistic property (negation).

Transitivity. Nikolaus et al. (2019) show that captioning models generalize better to unseen action – object pairs when the action is transitive, hypothesizing that this improvement is due to the additional arguments (e.g. cake) that images describing transitive events (e.g. eating) contain.

Verbal vs. nominal constructions. Su et al. (2021) study syntactic parsing and compare the Part-Of-Speech (POS) tag of the root of predicted and gold dependency trees of MSCOCO English captions, showing that the gold distribution is approximately 60:40 in favour of nouns, while models tend to never produce trees with a verb root.

3 Approach

We draw inspiration from the cognitive studies presented in Section 2.1. These studies carefully design visual scenes that differ only in terms of the visual feature of interest (e.g., visibility of the agent), ask participants to describe the scenes, and compare the linguistic properties of the descriptions across different conditions. If a linguistic property is significantly more prevalent in one condition, it is assumed that this visual feature constrains that linguistic property.

While such setups allow careful control over the experimental design, they are also less ecologically valid in that they impose (1) synthetic visual stimuli and (2) limitations on the number and diversity of participants, phenomena and languages to include. We address both shortcomings by (1) using large existing image caption datasets as a corpus of diverse language descriptions of naturalistic scenes, and (2) annotating the captions automatically, yet accurately, with linguistic properties.

Using a large amount of data instead of a controlled experiment raises an issue. Unlike in controlled cognitive studies, the sets of images we use are not arranged into ‘minimal pairs’, which are identical except for a visual feature of interest. To overcome this limitation, we exploit the large amount of data available via the automatic annotation of linguistic properties. We train visual classifiers to predict if a linguistic property is expressed in image captions when only the image is provided. If the classifiers achieve high accuracy on a held out test set, it is an indication that the visual features are informative enough to predict the linguistic property.² Figure 2 gives a high-level depiction of our approach.

To complement our analysis, we also conduct a corpus study. First, we use semantic annotations (object classes and bounding boxes) already available in existing datasets to group images by high-level properties and analyze the prevalence of linguistic properties in each group. Second, we compare the linguistic properties of captions for the same image in different languages. If a property is salient in the captions of all languages for a given image, it is likely that its visual content constrains descriptions that use that property. We present a corpus analysis using both approaches in Section 5.2.

4 Experimental Setup

In this section we describe the languages (4.1), linguistic properties (4.2) and datasets (4.3) used in our experiments.

4.1 Languages

We study English (**En**), German (**De**), Chinese (**Zh**), and Japanese (**Ja**) for three main reasons. First, multiple language families are required to

²However, if the accuracy is low, we cannot determine the cause; our modeling or data annotation assumptions may have led to this result, rather than the absence of a statistical relation.

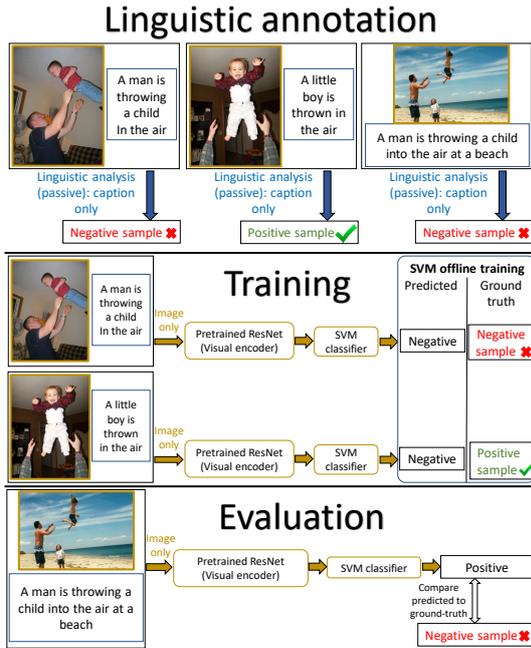


Figure 2: High-level depiction of our approach, demonstrating one linguistic property (passive voice). Top: Samples are annotated as expressing (positive) or not expressing (negative) the property. Middle: The SVM classifier is trained to predict whether an image expresses the property. Bottom: The classifier is evaluated; high classification accuracy indicates a strong relation between visual features and the linguistic property.

study language-agnostic constraints imposed by images. Second, all of these have large image-caption datasets with non-translated captions. Third, all of these have publicly available tools for annotation of some of the linguistic properties we study.

4.2 Annotation of Linguistic Properties

Below, we describe the automatic annotation of occurrences of linguistic properties in captions. All annotation methods were validated by asking in-house native speakers to verify a random sample of 100 (50 positive and 50 negative) instances per property and language. Across all languages and properties, accuracy exceeded 92%, confirming that our automatic annotations are of high quality.

For Japanese we only study the use of numerals since we were not able to achieve accurate annotation for the other properties.

Numerals (Num). We use Microsoft’s Recognize-Text package³ to identify the use of numerals in all languages. We ignore numerals with value of 1 for the following reasons: (1) In German and Chinese, the same word can refer to the

number *one* or the determiner *a*; (2) In Japanese, several non-numeral words contain the character for 1 (一), confusing the recognizing algorithm.

Negation words (Neg). We use the list of English negation words composed by Dobрева and Keller (2021), and add the word *nope*. We translate all words in the English list into the other languages, and verify the resulting lists with a native speaker.⁴

Verbal vs. nominal descriptions (Verb). We label captions with the root part-of-speech tag of their dependency tree, identified using Stanza’s dependency parser (Qi et al., 2020). We only consider captions with a single root which is a verb or a noun, filtering 0.8% of the captions. Note that we consider sentences where the root corresponds to the English verb *to be* (*sein* in German, 有 in Chinese) as noun roots, as no activity is described.

Transitivity of main verb (Tran). We use Stanza’s dependency parser and filter all captions with at least one of the following: (1) a non-verb root, (2) more or less than a single root, (3) the verb *be* (or its equivalents in languages other than English) as a root, filtering 47% of the captions. After filtering, a caption is labeled as transitive if its root verb has a child labeled as a direct object, and intransitive otherwise.⁵

Passive voice (Pass). We use the passive voice identifier tool for English and German (Ramm et al., 2017). For Chinese we search for the passive indicator 被, filtering cases where it is part of another word.⁶

4.3 Datasets

We use the following datasets: Pascal (Rashtchian et al., 2010), MSCOCO (Lin et al., 2014), Flickr30k (Young et al., 2014), Multi30k (Elliott et al., 2016), Flickr8kcn (Li et al., 2016), AIC-ICC (Wu et al., 2017), COCO-CN (Li et al., 2019), YJCaptions (Miyazaki and Shimizu, 2016), STAIR-captions (Yoshikawa et al., 2017). Table 1 presents additional information. We only use datasets with original captions generated by native speakers and avoid using datasets with captions translated from English.⁷ In addition to captions, MSCOCO and Flickr30k contain object classes and bounding box annotations. A description of the data collection

⁴All negation words are listed in Appendix A.1.

⁵In German and Chinese we automatically identify edge cases missed by the parser, see Appendix A.1.

⁶Words containing 被 are listed in Appendix A.1.

⁷See Appendix D for a comparison of original and translated captions.

³github.com/microsoft/Recognizers-Text

Lan	Name	Based on	Size (im,cap)
En	Pascal		1k, 5k
	MSCOCO		123k, 616k
	Flickr30k		31k, 158k
	Overall		156k, 779k
De	Multi30k	Flickr30k	31k, 155k
	Overall		31k, 155k
Zh	Flickr8kCN	Flickr8k	8k, 40k
	AIC-ICC		240k, 1.2M
	COCO-CN	MSCOCO	20k, 22k
	Overall		268k, 1.262M
Jp	YJCaptions	MSCOCO	26k, 131k
	STAIR-captions	MSCOCO	123k, 616k
	Overall		149k, 747k

Table 1: The list of datasets used in this study. For non-English datasets based on images from an English dataset, the *Based on* column indicates the name of the original dataset. The *Size* column indicates the number of images and the number of captions in the dataset.

process for each dataset is provided in Appendix B.

Property expression probability. Given an image I , a set of captions c_1, \dots, c_{N_I} , and an annotation function $f_p(c_i)$ mapping caption c_i to 1 if it expresses property p , and to 0 otherwise, we define the probability that image I expresses p as

$$P_p(I) = \frac{\sum_{i=1}^{N_I} f_p(c_i)}{N_I}$$

Given a language \mathcal{L} we denote with $P_{p,\mathcal{L}}$ the same computation with the set of captions filtered to include only captions in \mathcal{L} . Given a set of images S we denote the expected probability of expressing property p across all images in S as

$$\mathbb{E}_{I \in S}[P_p(I)] = \frac{\sum_{I \in S} P_p(I)}{|S|}$$

Table 2 presents the expected probability of each property p occurring in captions in language \mathcal{L} , $\mathbb{E}_{I \in S_{\mathcal{L}}}[P_{p,\mathcal{L}}(I)]$, where $S_{\mathcal{L}}$ is the set of all images in all datasets of language \mathcal{L} .

5 Experiments

We now describe our experiments and analyses. In Section 5.1 we train visual classifiers to predict linguistic properties, Section 5.2 presents a complementary corpus analysis, and Section 5.3 presents additional insights that may lead to future research.

5.1 Predicting Properties from Images

We study the task of predicting, given an image, whether human annotators will use a particular linguistic property when describing it. The input is

	Num	Neg	Pass	Tran	Verb
En	0.13 (156k)	0.0046 (156k)	0.076 (148k)	0.35 (137k)	0.41 (156k)
De	0.19 (31k)	0.0024 (31k)	0.012 (31k)	0.25 (31k)	0.78 (31k)
Zh	0.34 (268k)	0.0002 (268k)	0.002 (268k)	0.48 (245k)	0.60 (268k)
Jp	0.13 (123k)	–	–	–	–

Table 2: Expected probability of images expressing each property, in each of the 4 languages. Number of images are in parentheses.

a raw image and the output is binary, indicating whether the descriptions express the property.

Models. Our model consists of a visual encoder (ResNet50, He et al., 2016) to embed the raw image, followed by a set of binary SVM classifiers, one per linguistic property.⁸ We investigate four different pre-training methods with varying levels of supervision from different modalities.

First, we randomly initialize the visual encoder (no pre-training; **None**), avoiding unwanted bias through pre-training with human annotated information. Using a random encoder renders the task for the classifier more difficult, and the classifier might perform poorly even if linguistic properties are highly correlated with visual features, so we consider **None** as a lower bound.

To equip our model with some prior visual knowledge, we use MoCo (He et al., 2020), a self-supervised pre-training method based only on visual signals (**MoCo**). MoCo creates multiple manipulated versions of an image and trains the encoder to predict if two manipulated images correspond to the same original.

We also include ImageNet (Deng et al., 2009) pre-training (**IN**). The visual encoder is first trained to classify images in the ImageNet dataset, and then the classification head is discarded. Although semantic information is provided in ImageNet pre-training through class-labels, no textual input is provided which *describes* the visual scene.

Finally, we use CLIP (Radford et al., 2021) pre-training. CLIP is a multimodal self-supervised model, trained to project images and corresponding captions to similar vectors in a joint space. We use CLIP’s visual encoder, discarding the text encoder. This method is pre-trained with explicit textual

⁸We also experimented with neural classifiers, but SVM performed significantly better: see Appendix A.2.2 for details.

	En	De	Zh	Jp	Mul
Numerals	88k	21k	224k	86k	337k
Passive	47k	2k	2k	–	50k
Negation	6k	0.7k	0.3k	–	7k
Transitivity	128k	29k	223k	–	339k
Verb Root	150k	20k	198k	–	333k

Table 3: Number of images used in the experiments, for all properties and languages (Mul: Multilingual).

input, and hence its predictions will be skewed by the prior probability of linguistic properties in general language, obscuring the correlation with image features. In terms of raw performance, we consider CLIP as an upper bound.

We study two settings: monolingual (all images from datasets in a single language) and multilingual (all images from all datasets). In each setting, for each linguistic property p , we compute the probability of all relevant images to express p and binarize the data by using the median probability value as a threshold above which the image is considered to express p . Finally, we create a balanced dataset⁹ of images that express p and those that do not. We evaluate our models using 5-fold cross-validation. Table 3 shows the statistics of the generated datasets (note that the size of the datasets is smaller than in Table 2 because the data was balanced using down sampling). Implementation details are in Appendix A.2.

Multilingual results. Results are presented in Table 4. First, we observe that except for the model without pre-training, all models predict all properties above chance levels, supporting the hypothesis that linguistic properties are constrained by visual context. Second, results for the two non-textual pre-training methods (MoCo, IN) were significantly higher than the lower bound (None) and lower than the upper bound (CLIP) in all properties. Finally, numerals seem easiest to predict, which concurs with our corpus analysis where we find that mentions of numerals were easiest to link to visual properties (Section 5.2).

Monolingual results. We applied MoCo, the best performing method without human annotated pre-training, individually to each language (Table 5). Note that model performance does not always correlate with training data size (Table 3): in English,

⁹Although balancing the test set is usually considered a bad practice, in this study we only study image-text correlation and our classifiers would not be used for classifying new samples in the future.

	Num	Pass	Neg	Tran	Verb
None	60.5 ±0.9	52.7 ±0.3	51.0 ±1.6	54.3 ±0.5	54.2 ±0.3
MoCo	76.4 ±0.2	66.2 ±0.4	62.6 ±1.2	64.7 ±0.3	63.1 ±0.2
IN	74.6 ±0.4	65.9 ±0.5	62.4 ±1.9	64.5 ±0.1	62.6 ±0.2
CLIP	81.4 ±0.2	68.2 ±0.2	65.3 ±1.5	68.7 ±0.3	65.4 ±0.1

Table 4: Multilingual classification 5-fold cross-validation accuracy on all linguistic properties and pre-training methods. In all configurations, chance level is 50. IN: ImageNet. Numerals is the highest scoring property (in bold).

	Num	Pass	Neg	Tran	Verb
En	68.3 ±0.3	66.8 ±0.7	62.5 ±0.8	64.6 ±0.3	58.8 ±0.1
De	69.5 ±0.6	58.5 ±3.1	51.5 ±4.3	62.0 ±0.7	57.8 ±0.8
Zh	80.6 ±0.2	70.9 ±2.6	55.4 ±4.3	65.8 ±0.3	67.3 ±0.2
Jp	67.4 ±0.3	–	–	–	–

Table 5: Monolingual classification 5-fold cross-validation accuracy on all linguistic properties and languages, using the MoCo pre-training method. In all configurations, chance level is 50. In all languages, the use of numerals was predicted most accurately (in bold).

the verb root dataset was the largest but the classifier achieved the lowest accuracy; and prediction accuracy was high for passive voice in Chinese despite a small dataset. Across all languages, use of numerals was predicted most reliably.

5.2 Corpus Analysis

In this section we show that large image captioning corpora not only allow us to build predictive models to test hypotheses about the constraints of visual properties on language, but also support large-scale corpus studies. Our goal is to correlate image properties (e.g., the type or number of objects in an image) with linguistic choice (e.g., the use of numerals). The ground truth image properties are typically unavailable, but we can use additional information in MSCOCO and Flickr30k as proxies. In particular, we use the fact that the corpora are multilingually aligned (each image contains captions in different languages, all generated by native speakers) and they contain additional annotations (class labels and bounding boxes).

We take the expression of numerals as a test case, since it was the one most accurately predicted in Section 5.1. We emphasize, however, that the approach generalizes to other properties as well.

Although both MSCOCO and Flickr30k contain object classes and bounding box annotations, MSCOCO’s granularity is much higher (80 classes compared to 10 classes), so we only use

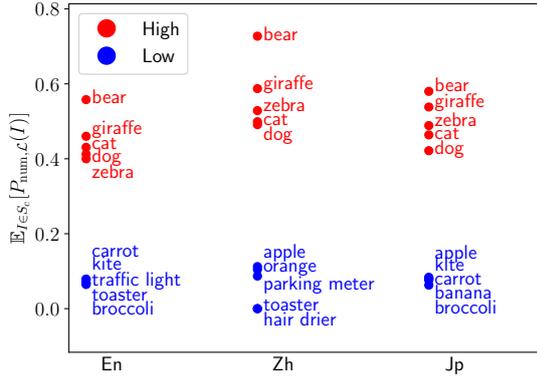


Figure 3: MSCOCO classes with highest and lowest expected numeral expression probability $\mathbb{E}_{I \in S_c}[P_{\text{num}, \mathcal{L}}(I)]$, for $\mathcal{L} \in \{\text{En}, \text{Zh}, \text{Jp}\}$. The probability of classes of animals is high in all languages.

MSCOCO’s class and bounding box annotations in our analysis. German is excluded from the class and bounding box analysis as there is no German version of MSCOCO with original captions.

Images containing animals are most likely to be described using numerals across languages.

For each MSCOCO class c , we find the set S_c of all images instantiating that class and compute $\mathbb{E}_{I \in S_c}[P_{\text{num}}(I)]$. We note that the expected $P_{\text{num}}(I)$ of some classes might be lower simply because they are more likely to occur in singles, and avoid this bias by filtering out images with a single instantiation of c from S_c .¹⁰

Figure 3 shows the 5 classes with the highest and lowest $\mathbb{E}_{I \in S_c}[P_{\text{num}}(I)]$ for each language. In all languages, images depicting animals are most likely to be described with numerals.

Our findings corroborate cognitive findings, placing the human subitizability threshold at 4.

We use MSCOCO bounding boxes annotation to investigate whether the use of numerals in image descriptions reflects the subitizability threshold (see Section 2.1). For each integer k , we find the set S_k of all images with k labeled bounding boxes, and compute $\mathbb{E}_{I \in S_k}[P_{\text{num}}(I)]$. We also label captions with quantifiers (e.g., *some*, *a few*¹¹) and compute $\mathbb{E}_{I \in S_k}[P_{\text{quant}}(I)]$. Figure 4 shows the results, for all k where $|S_k| \geq 100$. In all languages, $\mathbb{E}_{I \in S_k}[P_{\text{num}}(I)]$ initially increases with

¹⁰No classes were completely filtered out; only two classes (*toaster*, *hair-drier*) were left with less than 80 images.

¹¹The full lists are in Appendix A.1.

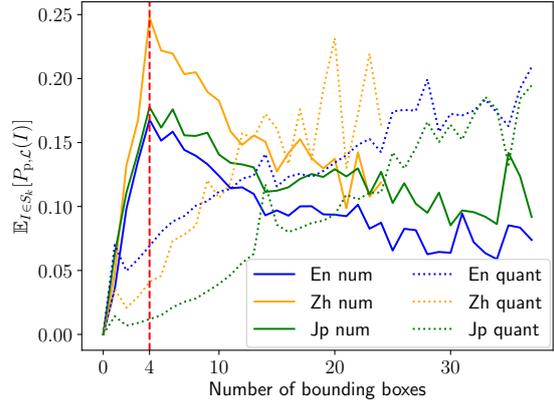


Figure 4: Expected probability of expressing the use of numerals and quantifiers $\mathbb{E}_{I \in S_k}[P_{p, \mathcal{L}}(I)]$ as a function of the number of bounding boxes in MSCOCO, for $\mathcal{L} \in \{\text{En}, \text{Zh}, \text{Jp}\}$ and $p \in \{\text{num}, \text{quant}\}$. All k with $|S_k| < 100$ were removed from the plot. Red line: subitizability threshold. In all languages, the probability increases up to 4 objects (consistent with cognitive studies) and then decreases. Quantifiers expression probability increases steadily.

Flickr30k (7k)			MSCOCO (13k)		
Zh/De	Zh/En	De/En	Zh/En	Zh/Jp	Jp/En
0.75	0.72	0.87	0.59	0.52	0.68

Table 6: Pearson correlation of the probability of use of numerals across pairs of languages in Flickr30 and MSCOCO. Number of images in parentheses.

a clear peak at 4, while quantifiers expression steadily increases.

Captions of the same image in different languages tend to agree on numerals usage.

We use the multilingual datasets Flickr30k (En, De, Zh) and MSCOCO (En, Zh, Jp), identify a list of images with captions in all respective languages $\{I_k\}_{k=1}^N$, and compute the list of probabilities of numerals expression for each image $L_{\mathcal{L}} = \{P_{\text{num}, \mathcal{L}}(I_k)\}_{k=1}^N$, in each language \mathcal{L} . Next, we compute the Pearson correlation coefficient of $L_{\mathcal{L}_1}, L_{\mathcal{L}_2}$ for each pair of languages $\mathcal{L}_1, \mathcal{L}_2$. The results are shown in Table 6. The correlation is *high* (> 0.5 ; Cohen (2013)) across all languages and datasets.

5.3 Additional Insights

The proposed methodology can also be used as an exploration method for further cognitive research. In this section, we present findings obtained by manually investigating extreme cases of property



Figure 5: Top: images described using numerals in all languages. Bottom: images described without numerals. Images taken from Flickr30k.

expression. This is an exploratory analysis, presenting preliminary findings that may lead to future research in a more controlled setting.

Use of numeral expressions. We manually inspect all images that use numerals in all captions across all languages in Flickr30k (N=105). The top images in Figure 5 are representative examples. All images depict multiple participants taking similar roles and positioned in a regular pattern (e.g., all the children in the upper right image in Figure 5 are swinging and facing the camera). The bottom of Figure 5 shows comparable images, which were never described using numerals. Here, participants appear in different poses and roles. We hypothesize that people count more easily and accurately when objects are arranged in a regular pattern, compared to a random formation (Burgess and Barlow, 1983).

We also present differences in the use of numerals across languages. We analyze images for which at least two captions use numerals with the same numeral value in each language, but different values across languages (N=46). We find two main reasons for cross-language inconsistencies: First, different languages tend to either count all participants in a single group or split them into smaller groups based on gender, role, or age.¹² These differences may be due to different annotation guidelines or different cultural backgrounds of the annotators.

Second, the multilingual datasets were originally created for English captioning, making the selected images highly related to English and especially North American culture.¹³ For example, in the sports domain, the datasets contain mainly images of Basketball and Baseball, popular sports in the United States. While English annotators use a de-

¹²Examples for all partition types are in Appendix C.

¹³This is a well known problem in multimodal datasets, previously discussed by Liu et al. (2021).

En: Basketball player wearing a white, number 23 jersey jumps up with the ball while guarded by number 13 on the opposite team
De: Zwei Männer spielen Basketball
Zh: 有两个男人正在打篮球



Figure 6: An image of a basketball game. The English captions are highly detailed, while both the German and Chinese caption translates to *Two men are playing basketball*. Image taken from Flickr30k, captions taken from Flickr30k (En), Mutli30k (De), Flickr8kcn (Zh).

tailed description, commonly mentioning the players' shirt number, German and Chinese descriptions are mostly short and count the number of players in the image (Figure 6).

Passive. We notice that in images with high probability for using passive voice, the patient is commonly located at the center of the scene either by the pose of the camera or the borders of the image. We hypothesize that this visual feature is correlated with the use of passive voice. The right image of Figure 1 shows one example. More examples are in Appendix C.

5.4 Discussion

Our experiments suggest that various linguistic properties are predictable from visual context, most notably in the case of the use of numeral expressions. Our classifiers were able to predict the presence of numerals in captions with high accuracy. Correspondingly, our corpus analysis provides evidence that the type and number of objects in the image constrain the use of numerals. Both results hold across different languages, and present high agreement between languages in the selection of images that are described with numerals. This lends support to the hypothesis that visual context constrains the use of numerals across a variety of languages from different families, and that such trends can be studied using the proposed methodology.

A surprising result is that without pretraining of the visual encoder (**None**), above chance-level performance can be obtained, most notably for the numerals property. A randomly initialized visual encoder applies a random dimensionality reduction to the input image, and the fact that the SVM classifier was able to learn to predict the presence of numerals in the captions of images at above chance level following this random transformation supports the hypothesis that this property correlates with visual features.

6 Conclusion

The synergistic relation between vision and language has been shown in the cognitive literature and leveraged in computational models, but *how* the two modalities inform each other has not been sufficiently studied at scale. We present a large scale study of the correlation of visual properties with linguistic phenomena, using naturalistic images described by a large crowd of native speakers of four languages.

In addition to confirming results of previous cognitive studies, we present new findings, e.g., the effect of object type on the use of numerals in visual scene description and the cross-lingual correlation of the use of numerals. Considering the effort needed to execute a controlled study, our proposed method can be used as an effective exploration technique for finding hypotheses for future controlled studies. In addition, our framework is general, and extends naturally to more languages and properties.

Beyond the cognitive contribution, our work can inform NLP models. Recent work suggests that in captioning models, training the model to predict a structured representation of the caption (e.g., based on POS prediction) before the text improves compositional generalization (Bugliarello and Elliott, 2021). In future work, we will study the utility of predicting our proposed linguistic properties for improving captioning models.

Limitations

We acknowledge several limitations of our suggested methodology. First, confounding factors may have affected our results, e.g., the difference in wording of the annotation guidelines for the original image-caption dataset could have a significant impact on the linguistic properties of the descriptions. In cognitive research, there is a well-known trade-off and ongoing discussion on the merits of highly-controlled, yet often oversimplified settings and the larger-scale, yet typically confounded, studies. The “reproducibility crisis” has highlighted that controlled studies are often difficult to reproduce, and initiated a discussion about the (complementary) utility of large-scale experiments which are typically more realistic. We propose such a method in the context of language/vision research, which can complement small-scale cognitive studies by considering natural scenes, while covering several languages and linguistic phenomena. We present empirical results that support the validity

of the methodology, in the sense that it often accords with established findings from the literature, as well as small scale qualitative analysis, that suggests trends for future work. We emphasize the importance of both paradigms, which should coexist and complement one another.

Second, with the exception of AIC-ICC, all image collections for all languages are based on original English image-caption datasets and hence are Anglocentric in their selection of concepts. The impact of such bias on NLP research has recently been discussed (Liu et al., 2021). We hope to extend the analysis with additional culture- and language-specific datasets in the future.

Finally, we do not distinguish between differences in linguistic properties that are due to annotators’ focus choices (i.e., the selection of what details in the image to describe) and those that are due to linguistic choices. The prevalence of linguistic properties could be influenced by the content that the annotator chose to describe (e.g., some annotators describe the background in addition to the main object(s), and others do not). This is a challenging and important line of future work, but outside the scope of this study.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments and feedback. We would also like to thank Rotem Dror, Sharon Goldwater and Grzegorz Chrupała for consulting, and the native speakers that consulted and validated our annotation tool: Assaf Porat, Kozue Watanabe, Arie Cattan, and Yilin Geng. This work was supported in part by the Israel Science Foundation (grant no. 2424/21), the Israeli Ministry of Science and Technology (grant no. 2336), and by the HUJI-UoM joint PhD program.

Ethics Statement

We use publicly available resources in our experiments, in accordance with their license agreements. The datasets are fully anonymized and do not contain personal information about the caption annotators or any information that could reveal the identity of the photographed subjects.

References

Dale Barr, Kees van Deemter, and Raquel Fernández. 2013. [Generation of quantified referring expressions](#):

- Evidence from experimental data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 157–161, Sofia, Bulgaria. Association for Computational Linguistics.
- Uri Berger, Gabriel Stanovsky, Omri Abend, and Lea Frermann. 2022. A computational acquisition model for multimodal word categorization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*.
- Emanuele Bugliarello and Desmond Elliott. 2021. The role of syntactic planning in compositional image captioning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 593–607, Online. Association for Computational Linguistics.
- A Burgess and HB Barlow. 1983. The precision of numerosity discrimination in arrays of random dots. *Vision Research*, 23(8):811–820.
- Dana L Chesney and Rochel Gelman. 2015. What counts? visual and verbal cues interact to influence what is considered a countable thing. *Memory & cognition*, 43(5):798–810.
- Jacob Cohen. 2013. *Statistical power analysis for the behavioral sciences*. Routledge.
- William Croft. 2012. *Verbs: Aspect and causal structure*. OUP Oxford.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society.
- R. M. W. Dixon. 1979. Ergativity. *Language*, 55:59–138.
- Radina Dobрева and Frank Keller. 2021. Investigating negation in pre-trained vision-and-language models. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 350–362, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Desmond Elliott, Stella Frank, Khalil Sima’an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.
- Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.
- Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, et al. 2008. The pascal visual object classes challenge 2007 (voc2007) results.
- Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc’Aurelio Ranzato, and Tomáš Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2121–2129.
- Rachel Giora, Vered Heruti, Nili Metuki, and Ofer Fein. 2009. “when we say no we mean no”: Interpreting negation in vision and language. *Journal of Pragmatics*, 41(11):2222–2239.
- Adele E. Goldberg. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press.
- Deepak Gupta, Pabitra Lenka, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A unified framework for multilingual and code-mixed visual question answering. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 900–913, Suzhou, China. Association for Computational Linguistics.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.
- Julian Hitschler, Shigehiko Schamoni, and Stefan Riezler. 2016. Multimodal pivots for image caption translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2399–2409, Berlin, Germany. Association for Computational Linguistics.
- Edna L Kaufman, Miles W Lord, Thomas Whelan Reese, and John Volkman. 1949. The discrimination of visual number. *The American journal of psychology*, 62(4):498–525.
- Sangeet Khemlani, Isabel Orenes, and Philip N Johnson-Laird. 2012. Negation: A theory of its meaning, representation, and use. *Journal of Cognitive Psychology*, 24(5):541–559.

- Xirong Li, Weiyu Lan, Jianfeng Dong, and Hailong Liu. 2016. Adding chinese captions to images. In *Proceedings of the 2016 ACM on international conference on multimedia retrieval*, pages 271–275.
- Xirong Li, Chaoxi Xu, Xiaoxu Wang, Weiyu Lan, Zhengxiong Jia, Gang Yang, and Jieping Xu. 2019. Coco-cn for cross-lingual image tagging, captioning, and retrieval. *IEEE Transactions on Multimedia*, 21(9):2347–2360.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. 2021. Visually grounded reasoning across languages and cultures. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10467–10485, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- George Mandler and Billie J Shebo. 1982. Subitizing: an analysis of its component processes. *Journal of experimental psychology: general*, 111(1):1.
- Takashi Miyazaki and Nobuyuki Shimizu. 2016. Cross-lingual image caption generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1780–1790, Berlin, Germany. Association for Computational Linguistics.
- Andriy Myachykov, Simon Garrod, and Christoph Scheepers. 2012. Determinants of structural choice in visually situated sentence production. *Acta psychologica*, 141(3):304–315.
- Mitja Nikolaus, Mostafa Abdou, Matthew Lamm, Rahul Aralikatte, and Desmond Elliott. 2019. Compositional generalization in image captioning. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 87–98, Hong Kong, China. Association for Computational Linguistics.
- Alexander Ororbia, Ankur Mali, Matthew Kelly, and David Reitter. 2019. Like a baby: Visually situated neural language acquisition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5127–5136, Florence, Italy. Association for Computational Linguistics.
- Eleonore Oversteegen and Joost Schilperoord. 2014. Can pictures say no or not? negation and denial in the visual mode. *Journal of pragmatics*, 67:89–106.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Anita Ramm, Sharid Loáiciga, Annemarie Friedrich, and Alexander Fraser. 2017. Annotating tense, mood and voice for English, French and German. In *Proceedings of ACL 2017, System Demonstrations*, pages 1–6, Vancouver, Canada. Association for Computational Linguistics.
- Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting image annotations using Amazon’s Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 139–147, Los Angeles. Association for Computational Linguistics.
- Lilia Rissman, Amanda Woodward, and Susan Goldin-Meadow. 2019. Occluding the face diminishes the conceptual accessibility of an animate agent. *Language, cognition and neuroscience*, 34(3):273–288.
- Yuri Sato and Koji Mineshima. 2021. Can humans and machines classify photographs as depicting negation? In *International Conference on Theory and Application of Diagrams*, pages 348–352. Springer.
- Yuri Sato, Koji Mineshima, and Kazuhiro Ueda. 2021. Visual representation of negation: Real world data analysis on comic image design. *ArXiv preprint*, abs/2105.10131.
- Aili Shen, Bahar Salehi, Jianzhong Qi, and Timothy Baldwin. 2020. A general approach to multimodal document quality assessment. *Journal of Artificial Intelligence Research*, 68:607–632.
- Ruisi Su, Shruti Rijhwani, Hao Zhu, Junxian He, Xinyu Wang, Yonatan Bisk, and Graham Neubig. 2021. Dependency induction through the lens of visual perception. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 17–26, Online. Association for Computational Linguistics.
- Emiel van Miltenburg, Desmond Elliott, and Piek Vossen. 2017. Cross-linguistic differences and similarities in image descriptions. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 21–30, Santiago de Compostela, Spain. Association for Computational Linguistics.

Emiel van Miltenburg, Roser Morante, and Desmond Elliott. 2016. **Pragmatic factors in image description: The case of negations**. In *Proceedings of the 5th Workshop on Vision and Language*, pages 54–59, Berlin, Germany. Association for Computational Linguistics.

Sol Worth. 1981. Studying visual communication. In *Studying Visual Communication*. University of Pennsylvania Press.

Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, et al. 2017. **Ai challenger: A large-scale dataset for going deeper in image understanding**. *ArXiv preprint*, abs/1711.06475.

Yuya Yoshikawa, Yutaro Shigeto, and Akikazu Takeuchi. 2017. **STAIR captions: Constructing a large-scale Japanese image caption dataset**. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 417–421, Vancouver, Canada. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. **From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions**. *Transactions of the Association for Computational Linguistics*, 2:67–78.

A Implementation Details

A.1 Linguistic Properties Annotation

Use of numerals In the bounding boxes experiment in section 5.2 we search for quantifiers. Following are the lists of quantifiers we search for in each language. English: *some, a lot of, many, lots of, a few, several, a number of*. Chinese: 些, 多. Japanese: 多くの, たくさん, いくつか.

Use of negation words Following are the lists of negation words for each language.

English: *not, isn't, aren't, doesn't, don't, can't, cannot, shouldn't, wont, wouldn't, no, none, nobody, nothing, nowhere, neither, nor, never, without, nope*.

German: *nicht, kein, nie, niemals, niemand, nirgendwo, nirgendwohin, nirgends, weder, ohne, nein, nichts, nee*. We lemmatize the words in the sentence before searching in this list.

Chinese: 不, 不是, 不能, 不可以, 没, 没有, 没什么, 从不, 并不, 从没有, 并没有, 无人, 无处, 无, 别, 绝不. We use the Jieba tokenizer¹⁴. We also identify cases where one of the words above is part of a longer non-negation word and filter those cases. Following is the list of non-negation words: 别着, 不小心, 不一样.

¹⁴github.com/fxsjy/jieba

Use of passive verbs In Chinese we search for the passive indicator 被, filtering cases where it is part of the 被子 word (meaning quilt), a common word in the AIC-ICC dataset.

Transitivity In German and Chinese we identify several important edge cases in which the Stanza parser is consistently incorrect, which we fix manually. All edge cases were verified by native speakers.

In German we identify sentences containing a node which is a child of the root and labeled with the *PTKVZ* POS tag, and label these as intransitive.

In Chinese we identify sentences where (1) the lemma of the root word ends with the preposition token 在; (2) the lemma of the word following the root word is 在; or (3) the lemma of the word following the root word starts with the preposition token 向, and label these as intransitive.

A.2 Model Details

A.2.1 SVM Classifier

We use the SVC model from the sklearn Python package with the RBF kernel and default hyperparameters.

A.2.2 Neural Classifier

We use a feed-forward neural network with 1 or 2 hidden layers, with different activation functions (ReLU, Sigmoid, Tanh). In all configurations, the SVM classifier performed better.

A.2.3 Pre-trained Backbone Models

For MoCo and CLIP we use the models provided in the officially published code. For ImageNet pre-training we use the pre-trained model provided by the PyTorch package. In all cases, model contains 25.6M parameters.

A.3 Training

Training with the largest training set (the transitivity multilingual setting, see table 3) took 30 hours on a single GM204GL GPU.

B Dataset Collection Details

Following is a brief description of the process of data collection for each of the datasets.

Pascal Sentences (Rashtchian et al., 2010) contains the set of images from the PASCAL Visual Object Classes Challenge (Everingham et al., 2008) with captions generated by Amazon's Mechanical Turk workers. The annotators were instructed to (1)

describe the image in a single sentence including the main characters, the setting or the relation of the objects; (2) If possible, include adjectives such as colors, spacing, emotion, or quantity; (3) pay attention to grammar and spelling.

Flickr30k (Young et al., 2014) is a large English image-caption dataset. The objects in each image are segmented using bounding boxes and classified into one of 10 classes. Annotators were crowdsource workers and were asked to “write sentences that describe the depicted scenes, situations, events and entities (people, animals, other objects)”.

Multi30k (Elliott et al., 2016) is a German version of Flickr30k. It contains both original and translated captions. Translations are generated by professional translators, original captions were generated by crowdworkers via the Crowdfunder platform. Instructions were translated from the English instructions of Flickr30k.

Flickr8kcn (Li et al., 2016) is a Chinese version of the smaller Flickr8k dataset on which the Flickr30k dataset was based. Descriptions were generated by crowdworkers that were asked to “write sentences describing salient objects and scenes in every image, from their own point of views”.

MSCOCO (Lin et al., 2014) is another large English image-caption dataset with additional annotations (object classes and bounding boxes). The captions were generated using human subjects on Amazon’s Mechanical Turk. The annotators were given the following instructions:

- Describe all the important parts of the scene.
- Do not start the sentences with “There is”.
- Do not describe unimportant details.
- Do not describe things that might have happened in the future or past.
- Do not describe what a person might say.
- Do not give people proper names.
- The sentences should contain at least 8 words.

COCO-CN (Li et al., 2019) is a Chinese version of MSCOCO, annotated by a group of volunteers and paid undergraduate students. Annotators were instructed that the caption shall cover the main objects, actions and scene in a given image, and were provided with suggested captions retrieved in

the following process: all the captions in the original MSCOCO dataset were machine-translated to Chinese, and the 5 most relevant suggestions for each image were chosen by a model. However, they were asked to provide their own descriptions, and only draw inspiration from the suggestions. In addition, they manually translated 5000 captions.

YJCaptions (Miyazaki and Shimizu, 2016) is a Japanese version of MSCOCO. Captions were generated using Yahoo! crowdsourcing, where signing up requires a Japanese proficiency, leading the authors to assume that participants were fluent in Japanese. Annotation guidelines can be translated to English as “Please explain the image using 16 or more Japanese characters. Write a single sentence as if you were writing an example sentence to be included in a textbook for learning Japanese. Describe all the important parts of the scene; do not describe unimportant details. Use correct punctuation. Write a single sentence, not multiple sentences or a phrase”.

STAIR-captions (Yoshikawa et al., 2017) is another Japanese version of MSCOCO. Annotation guidelines can be translated to English as “(1) A caption must contain more than 15 letters. (2) A caption must follow the da/dearu style (one of the writing styles in Japanese). (3) A caption must describe only what is happening in an image and the things displayed therein. (4) A caption must be a single sentence. (5) A caption must not include emotions or opinions about the image”.

AIC-ICC (Wu et al., 2017) is a large Chinese image-caption dataset. The annotators were instructed to (1) include key objects/attributes, locations and human actions; (2) generate fluent captions; (3) use Chinese idioms or descriptive adjectives.

C Additional Visual Examples

Numerals disagreement Further to the numerals disagreement analysis in Section 5.3, we present examples of images that were described by captions in multiple languages with numeral value disagreement caused by differences in partition of the participants. For each of these images, the captions in one language do not partition the participants while the captions in the other is partitioning based on gender (Figure 7), role (Figure 8) or age (Figure 9).

Passive voice Figure 10 shows three images with high probability for the use of passive voice. In the upper right image the passive participant is



Figure 7: An image taken from Flickr30k. The English caption splits participants based on gender: “A man in a beret and thin mustache gestures to two women in conversation”. The Chinese caption does not split participants at all: “三个人正在谈话” (Three people are talking).



Figure 8: An image taken from Flickr30k. The English caption splits participants based on role: “One dog is chasing another dog that is carrying something in its mouth along the beach”. The German caption does not split participants at all: “Zwei weiß-braune Hunde, die am Strand laufe” (Two white and brown dogs running on the beach).

centered by the pose of the camera, while in the other two images the borders of the image locates the passive participant in the center.

D Original vs. Translated Captions

When studying multimodal tasks in non-English languages (e.g., multimodal machine translation (Hitschler et al., 2016), visual question answering (Gupta et al., 2020)), it is common to translate an existing English image-caption corpus into the target language using crowd sourcing or translation APIs. We show that captions generated in this setting are not representative of the target language. We use the Multi30k dataset (De) and the COCO-CN dataset (Zh), both of which contain



Figure 9: An image taken from Flickr30k. The English caption splits participants based on age: “A man and two children in life jackets in a boat on a lake”. The Chinese caption does not split participants at all: “坐在船上出海的三个人” (Three people on a boat going out to the sea).

original as well as translated captions in the target language. We use the statistical method described in Section 5.2 in the Cross-lingual analysis paragraph to compute the agreement of English and translated captions, and compare it with the agreement of original and translated captions. As shown in Figure 11, in 9/10 cases the English-Translated agreement is higher than Original-Translated agreement, suggesting that translated captions are not representative of the target language. The effect is most pronounced with negation.



Figure 10: images with high probability for the use of passive voice. In all images, the passive participant is centered by the pose of the camera or the borders of the image.

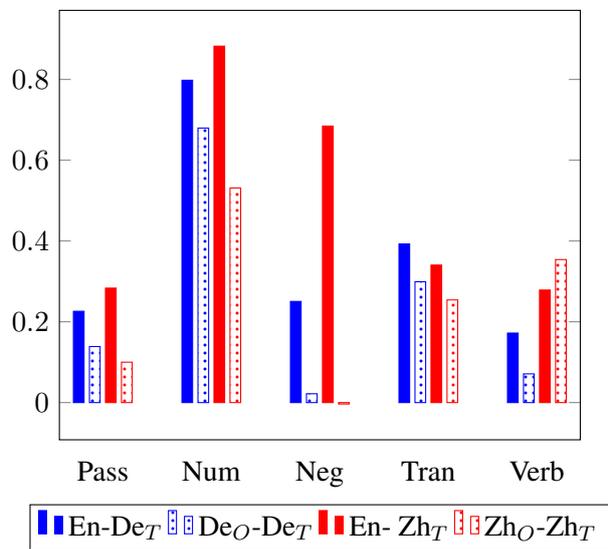


Figure 11: English – Translated agreement (En-De_T and En-Zh_T) and Original – Translated agreement (De_O-De_T and Zh_O-Zh_T) for German and Chinese, in different linguistic properties.

How Much Syntactic Supervision is “Good Enough”?

Hiroshi Noji^{†*}

Artificial Intelligence Research Center
AIST
hiroshi.noji@aist.go.jp

Yohei Oseki^{*}

Graduate School of Arts and Sciences
University of Tokyo
oseki@g.ecc.u-tokyo.ac.jp

Abstract

In this paper, we explore how much syntactic supervision is “good enough” to make language models (LMs) more human-like. Specifically, we propose the new method called *syntactic ablation*, where syntactic LMs, namely Recurrent Neural Network Grammars (RNNGs), are gradually ablated from full syntactic supervision to zero syntactic supervision (\approx unidirectional LSTM) by preserving NP, VP, PP, SBAR non-terminal symbols and the combinations thereof. The 17 ablated grammars are then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark. The results of our syntactic ablation demonstrated that (i) the RNNG with zero syntactic supervision underperformed the RNNGs with some syntactic supervision, (ii) the RNNG with full syntactic supervision underperformed the RNNGs with less syntactic supervision, and (iii) the RNNG with mild syntactic supervision achieved the best performance comparable to the state-of-the-art GPT-2-XL. Those results may suggest that the “good enough” approach to language processing seems to make LMs more human-like.

1 Introduction

In the literature on targeted syntactic evaluation (Linzen et al., 2016; Marvin and Linzen, 2018), recurrent neural networks (RNNs) such as LSTMs have been demonstrated to implicitly learn syntactic structures of natural language (e.g., subject-verb agreement), despite the lack of explicit syntactic supervision (cf. Hewitt and Manning, 2019). Moreover, those RNNs also turned out to benefit from explicit syntactic supervision. RNNs integrated with explicit syntactic supervision, namely Recurrent Neural Network Grammars (RNNGs; Dyer et al. 2016), have received considerable attention for their cognitive plausibility and outperformed

RNNs in not only targeted syntactic evaluation (Kuncoro et al., 2018; Wilcox et al., 2019) but also psychometric predictive power (Hale et al., 2018; Wilcox et al., 2020; Yoshida et al., 2021).

However, despite the previous debate over the dichotomy between the presence and absence of syntactic supervision, how much syntactic supervision is necessary and sufficient remains to be investigated. Especially, there are two potential reasons to believe that full syntactic supervision is suboptimal. Theoretically, full syntactic supervision may override lexical heuristics implicitly learned with RNNs, where information on terminal symbols vanishes via recursive composition operations (cf. Kuncoro et al., 2017). Empirically, full syntactic supervision seems to destroy the performance of long-distance dependencies, especially (pseudo-)cleft constructions, where both acceptable (e.g., *What he **did** was prepare the meal.*) and unacceptable (e.g., **What he **ate** was prepare the meal.*) sentences share the exactly same syntactic structure (Figure 1) and should be distinguished via lexical heuristics alone (cf. Noji and Oseki, 2021). Therefore, it is reasonable to hypothesize that optimal syntactic supervision lies somewhere between full and zero syntactic supervision in order to balance syntactic structures and lexical heuristics. Intuitively speaking, if we teach too much syntax to language models, those models will forget lexicon.

In this paper, we explore how much syntactic supervision is “good enough” to make language models more human-like. Specifically, we propose the new method called *syntactic ablation*, where RNNGs are gradually ablated from full syntactic supervision to zero syntactic supervision (\approx unidirectional LSTM) by preserving NP, VP, PP, SBAR nonterminal symbols and the combinations thereof. The 17 ablated grammars are then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark (Gauthier et al., 2020). The results demonstrate that the RNNG with mild syntactic

[†]Currently affiliated with LeapMind Inc.: noji@leapmind.io.

^{*}Denotes equal contribution.

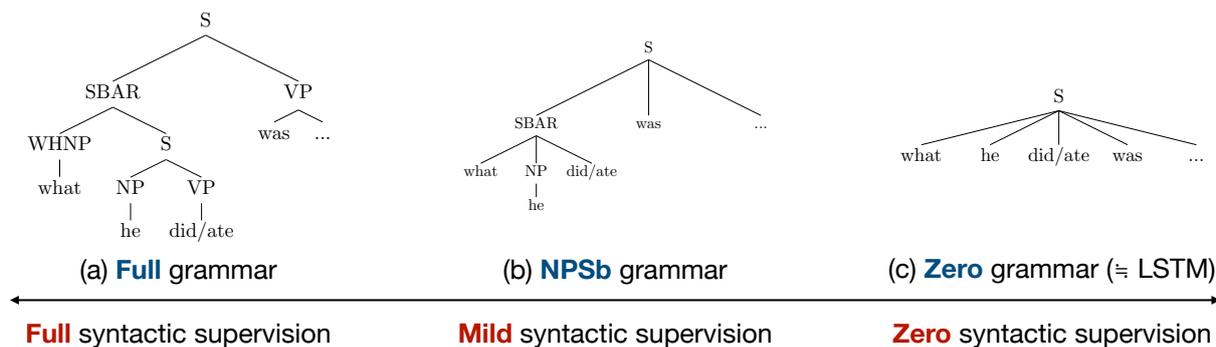


Figure 1: Our proposed method of syntactic ablation. RNNGs are gradually ablated from (a) full syntactic supervision, through (b) mild syntactic supervision, to (c) zero syntactic supervision (\approx unidirectional LSTM) by preserving NP, VP, PP, SBAR nonterminal symbols and the combinations thereof, hence the 17 ablated grammars.

supervision achieved the best performance comparable to the state-of-the-art GPT-2-XL, which are then discussed in the broader context of the computational psycholinguistic literature (Ferreira et al., 2002; Ferreira and Patson, 2007).

2 Methods

2.1 Recurrent Neural Network Grammars

Recurrent Neural Network Grammars (RNNGs; Dyer et al. 2016) are deep generative models of sentences and structures. RNNGs employ the stack LSTM (Dyer et al., 2015) to compute probability distributions over 3 parsing actions below:

- NT: Open nonterminal symbols.
- GEN: Generate terminal symbols.
- REDUCE: Close nonterminal symbols.

For the REDUCE action, RNNGs adopt the bidirectional LSTM to encode terminal and nonterminal symbols both left-to-right and right-to-left into phrasal representations. For inference, RNNGs utilize word-synchronous beam search (Stern et al., 2017) implemented in Noji and Oseki (2021).¹

2.2 Syntactic ablation

Our proposed method of syntactic ablation is summarized in Figure 1. RNNGs are gradually ablated from full syntactic supervision to zero syntactic supervision by preserving NP, VP, PP, SBAR nonterminal symbols and the combinations thereof, hence 17 ablated grammars below:

- Zero: Zero grammar.
- N: NP nonterminal symbol only.

- V: VP nonterminal symbol only.
- P: PP nonterminal symbol only.
- Sb: SBAR nonterminal symbol only.
- NV: NP and VP nonterminal symbols.
- NP: NP and PP nonterminal symbols.
- NSb: NP and SBAR nonterminal symbols.
- VP: VP and PP nonterminal symbols.
- VSb: VP and SBAR nonterminal symbols.
- PSb: PP and SBAR nonterminal symbols.
- NVP: NP, VP, and PP nonterminals.
- NVSb: NP, VP, and SBAR nonterminals.
- NPSb: NP, PP, and SBAR nonterminals.
- VPSb: VP, PP, and SBAR nonterminals.
- NVPSb: NP, VP, PP, and SBAR nonterminals.
- Full: Full grammar.

RNNGs are trained on the parsed sentences. We created the training data for each grammar, which only provides designated nonterminal symbols. Our original dataset is the same as the XL dataset of Hu et al. (2020), which is about 42M tokens from BLLIP corpus (Charniak et al., 2000) and re-parsed by Berkeley neural parser (Kitaev et al., 2019), from which we only kept the ablated nonterminals to create the dataset. For each grammar, we trained an RNNG with three different random seeds. For the other training settings, we follow Noji and Oseki (2021)’s 100M token experiment.

2.3 Targeted syntactic evaluation

Those ablated grammars were then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark (Gauthier et al., 2020) which includes 6

¹<https://github.com/aistairc/rnng-pytorch>

syntactic *circuits*: Agreement, Garden-Path Effects, Licensing, Center Embedding, Gross-Syntactic State, and Long-Distance Dependencies.

We adopted the “perfect match” evaluation metric proposed in Hu et al. (2020), not the “partial match” evaluation metric utilized in the SyntaxGym leaderboard, which seems to overestimate the accuracies of syntactic generalization.

3 Results

3.1 Overall accuracies

Overall accuracies of our syntactic ablation experiments are summarized in Figure 2. Accuracies of SyntaxGym (the vertical axis) are plotted against grammars with different amounts of syntactic supervision (the horizontal axis), together with the accuracies of RNNG and GPT-2-XL reported in Hu et al. (2020). Zero (leftmost) and Full (rightmost, except RNNG and GPT-2-XL) represent zero and full grammars, respectively, the former of which is equivalent to the unidirectional LSTM.² N, V, P, and Sb indicate grammars with NP, VP, PP, and SBAR nonterminal symbols preserved, respectively. Therefore, NP represents the grammar with NP and PP nonterminal symbols preserved, not to be confused with the grammar with the NP nonterminal symbol preserved.

²They are practically equivalent because the REDUCE action does not occur except the end of the sentence, where the only difference affecting each word probability is the existence of “(ROOT)” symbol at the beginning of the sentence.

There are three key observations here. First, the Zero grammar, which is equivalent to the unidirectional LSTM, underperformed the grammars with some syntactic supervision, suggesting that syntactic supervision plays an important role for human-like syntactic generalization. Second, the Full grammar also underperformed the grammars with less syntactic supervision and GPT-2-XL in Hu et al. (2020), meaning that full syntactic supervision does not always make LMs human-like. Finally, and most importantly, the NPSb grammar achieved the best performance (84.585417) comparable to (or even numerically larger than) the state-of-the-art GPT-2-XL (84.241459).

3.2 Circuit accuracies

Circuit accuracies of our syntactic ablation experiments are summarized in Figure 3. Accuracies of 6 circuits on SyntaxGym (the vertical axis) are plotted against 4 grammars with different amounts of syntactic supervision (the horizontal axis).

Interestingly, the NPSb grammar outperformed the Full grammar for 5 among 6 syntactic circuits (Agreement, Center Embedding, Garden-Path Effects, Licensing, Long-Distance Dependencies). Notice that the performance advantage of the NPSb grammar is significantly larger in Long-Distance Dependencies, especially (pseudo-)cleft constructions, corroborating the hypothesis that optimal syntactic supervision lies somewhere between full and zero syntactic supervision in order to balance syntactic structures and lexical heuristics.

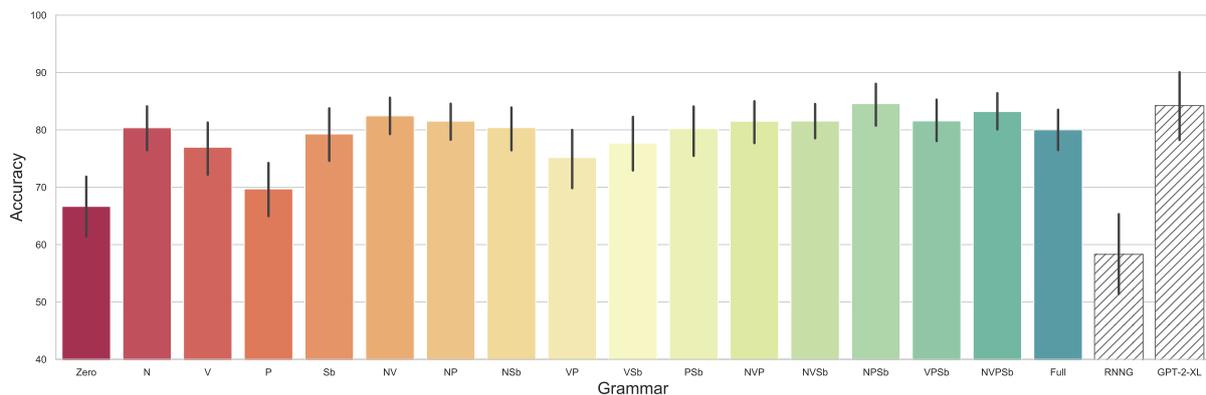


Figure 2: Overall accuracies of our syntactic ablation experiments. Accuracies averaged over 6 circuits on SyntaxGym and random seeds (the vertical axis) are plotted against grammars with different amounts of syntactic supervision (the horizontal axis), together with the accuracies of RNNG and GPT-2-XL reported in Hu et al. (2020). Error bars denote bootstrapped 95% confidence intervals. Zero (leftmost) and Full (rightmost, besides RNNG and GPT-2-XL) represent zero and full grammars, respectively. N, V, P, and Sb indicate the grammars with NP, VP, PP, and SBAR nonterminal symbols preserved, respectively. Therefore, NP represents the grammar with NP and PP nonterminal symbols preserved, not to be confused with the grammar with the NP nonterminal symbol preserved.

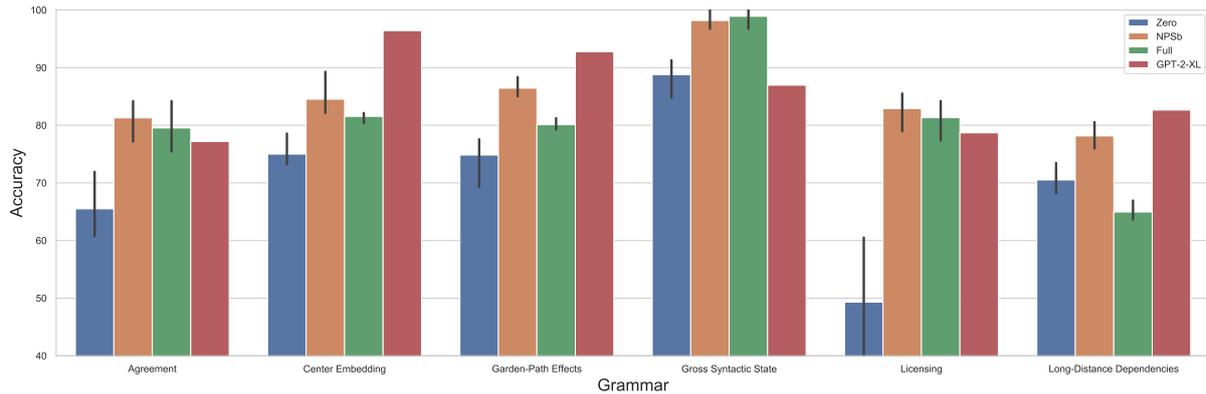


Figure 3: Circuit accuracies of our syntactic ablation experiments. Accuracies of 6 circuits on SyntaxGym (the vertical axis) are plotted against 4 grammars with different amounts of syntactic supervision (the horizontal axis).

4 Discussion

In summary, we performed the syntactic ablation experiments where RNNs were gradually ablated from full syntactic supervision to zero syntactic supervision (\approx unidirectional LSTM), and then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark. In this section, the results of our syntactic ablation experiments will be discussed in the broader context of the computational psycholinguistic literature.

4.1 The “good enough” language processing

The overall accuracies reported in Section 3.1 demonstrated that the RNN with mild syntactic supervision, especially the NPSb grammar, outperformed the RNNs with zero and full syntactic supervision, as well as GPT-2 XL in Hu et al. (2020). Those results are consistent with the “good enough” approach to language processing (Ferreira et al., 2002; Ferreira and Patson, 2007), where human language processing does not always generate deep syntactic structures, but rather employs shallow syntactic structures and frugal lexical heuristics. Here, we suggest that the RNN with mild syntactic supervision serves as the mechanistic model of the “good enough” approach to language processing, in that neither deep/hierarchical syntax is necessary nor shallow/flat syntax is sufficient; rather, some syntax in between is “good enough”.

4.2 Long-Distance Dependencies

The circuit accuracies reported in Section 3.2 revealed that the NPSb grammar outperformed the Full grammar for 5 syntactic circuits such as Agreement, Center Embedding, Garden-Path Effects, Licensing, Long-Distance Dependencies.

Upon closer inspection (cf. Hu et al., 2020), those 5 syntactic circuits share the isomorphic syntactic structure with long-distance dependencies between dependents inside and outside “heavy” subjects (where the *dependents* are italicized):³

- **Agreement:** [_{NP} The *farmer* [_{PP} near the clerks]] *knows* many people.
- **Center Embedding:** [_{NP} The *painting* [_{SBAR} that the artist painted]] *deteriorated*.
- **Garden-Path Effects:** [_{NP} The *child* [_{SBAR} kicked in the chaos]] *found* her way back home.
- **Licensing:** [_{NP} *No* managers [_{SBAR} that respected the guard]] have had *any* luck.
- **Long-Distance Dependencies:** [_{SBAR} What he *did*] was *prepare* the meal.

Importantly, NP, PP, and SBAR representations effectively make linearly distant dependents hierarchically close, while VP representations have no designated *raison d’être* and, moreover, may override lexical heuristics of verbs (e.g., *knows*, *deteriorated*) via recursive composition operations (cf. Kuncoro et al., 2017; Noji and Oseki, 2021). Thus, at least for those 5 syntactic circuits, the NPSb grammar is the optimal syntactic supervision that balances syntactic structures and lexical heuristics.

³While those 5 syntactic circuits are not named long-distance dependencies (except the Long-Distance Dependencies circuit which includes filler-gap dependencies and cleft constructions), they all involve long-distance dependencies.

5 Conclusion

In this paper, we explored how much syntactic supervision is “good enough” to make language models more human-like. Specifically, we performed the syntactic ablation experiments where RNNs were gradually ablated from full syntactic supervision to zero syntactic supervision (\approx unidirectional LSTM), and then evaluated via targeted syntactic evaluation on the SyntaxGym benchmark. The results demonstrated that the RNN with mild syntactic supervision achieved the best performance comparable to the state-of-the-art GPT-2-XL. We hope that the “good enough” approach to language processing (Ferreira et al., 2002; Ferreira and Patson, 2007) provides the promising direction for future research.

Limitations

There are several limitations with this paper. First, the evaluated models are limited; the syntactic ablation was applied to only one model (i.e. RNN) and remains to be generalized to other models (cf. Sartran et al., 2022). Second, the evaluation datasets are also limited; our ablated RNNs were evaluated against only one dataset (i.e. SyntaxGym) and remain to be extended to other datasets (cf. Warstadt et al., 2020). In addition, from engineering perspectives, our ablated RNNs, though lightweight, still require some syntactic supervision, which may induce the scalability bottleneck.

Acknowledgements

This paper is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This work was also supported by JSPS KAKENHI Grant Numbers 20K19877 and 19H04990, and JST PRESTO Grant Number JPMJPR21C2.

References

Eugene Charniak, Don Blaheta, Niyu Ge, Keith Hall, John Hale, and Mark Johnson. 2000. [BLLIP 1987-89 WSJ Corpus Release 1 LDC2000T43](#). Linguistic Data Consortium.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. [Transition-based dependency parsing with stack long short-term memory](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*

and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 334–343, Beijing, China. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. [Recurrent neural network grammars](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Fernanda Ferreira, Karl G.D. Bailey, and Vittoria Ferraro. 2002. [Good-enough representations in language comprehension](#). *Current Directions in Psychological Science*, 11(1):11–15.

Fernanda Ferreira and Nikole D. Patson. 2007. [The ‘good enough’ approach to language comprehension](#). *Language and Linguistics Compass*, 1(1-2):71–83.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. [SyntaxGym: An online platform for targeted evaluation of language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. [Finding syntax in human encephalography with beam search](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2727–2736, Melbourne, Australia. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. [A structural probe for finding syntax in word representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. [A systematic assessment of syntactic generalization in neural language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. [Multi-lingual constituency parsing with self-attention and pre-training](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2017. [What do recurrent neural network grammars learn about syntax?](#) In *Proceedings of the 15th*

- Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1249–1258, Valencia, Spain. Association for Computational Linguistics.
- Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. [LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. [Assessing the ability of LSTMs to learn syntax-sensitive dependencies](#). *Transactions of the Association for Computational Linguistics*, 4:521–535.
- Rebecca Marvin and Tal Linzen. 2018. [Targeted syntactic evaluation of language models](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.
- Hiroshi Noji and Yohei Oseki. 2021. [Effective batching for recurrent neural network grammars](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4340–4352, Online. Association for Computational Linguistics.
- Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. [Transformer grammars: Augmenting transformer language models with syntactic inductive biases at scale](#).
- Mitchell Stern, Daniel Fried, and Dan Klein. 2017. [Effective inference for generative neural parsing](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700, Copenhagen, Denmark. Association for Computational Linguistics.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. [BLiMP: A benchmark of linguistic minimal pairs for English](#). In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.
- Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019. [Structural supervision improves learning of non-local grammatical dependencies](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3302–3312, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger P. Levy. 2020. On the predictive power of neural language models for human real-time comprehension behavior. *ArXiv*, abs/2006.01912.
- Ryo Yoshida, Hiroshi Noji, and Yohei Oseki. 2021. [Modeling human sentence processing with left-corner recurrent neural network grammars](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2964–2973, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Are the Best Multilingual Document Embeddings simply Based on Sentence Embeddings?

Sonal Sannigrahi,¹ Josef van Genabith,^{1,2} Cristina España-Bonet²

¹Saarland University, Saarland Informatics Campus, Germany

²German Research Center for Artificial Intelligence (DFKI)

sosa00001@stud.uni-saarland.de

{cristinae, Josef.Van_Genabith}@dfki.de

Abstract

Dense vector representations for textual data are crucial in modern NLP. Word embeddings and sentence embeddings estimated from raw texts are key in achieving state-of-the-art results in various tasks requiring semantic understanding. However, obtaining embeddings at the document level is challenging due to computational requirements and lack of appropriate data. Instead, most approaches fall back on computing document embeddings based on sentence representations. Although there exist architectures and models to encode documents fully, they are in general limited to English and few other high-resourced languages. In this work, we provide a systematic comparison of methods to produce document-level representations from sentences based on LASER, LaBSE, and Sentence BERT pre-trained multilingual models. We compare input token number truncation, sentence averaging as well as some simple windowing and in some cases new augmented and learnable approaches, on 3 multi- and cross-lingual tasks in 8 languages belonging to 3 different language families. Our task-based extrinsic evaluations show that, independently of the language, a clever combination of sentence embeddings is usually better than encoding the full document as a single unit, even when this is possible. We demonstrate that while a simple sentence average results in a strong baseline for classification tasks, more complex combinations are necessary for semantic tasks. Our code is publicly available.¹

1 Introduction

Semantic representations, especially embeddings, are crucial for natural language processing (NLP). In fact, the field has exploded since the success of dense word embeddings (Mikolov et al., 2013). For some tasks like finding semantic or syntactic relations among words, high quality word embeddings

are enough. Other tasks, like question classification or paraphrase detection, benefit from sentence embeddings. Finally, lots of tasks deal with documents: summarisation, document classification, question answering, etc. Document representations are difficult to be learned, especially multilingually, given the amount of available training data and the length of each training instance.

For these reasons, document embeddings usually resort to sentence embeddings. Since some of the state-of-the-art techniques for language modelling and sentence embeddings are based on self-attention architectures such as BERT (Devlin et al., 2019), and self-attention scales quadratically with the input length, one cannot afford arbitrarily long inputs. Training is usually constrained to input fragments up to 512 tokens (subunits). This limit goes well beyond an average sentence length and can cover several paragraphs. However, full documents can be significantly longer. The average length of a Wikipedia article in English is 647 words (not subunits) for example,² and the average for two of the tasks that we consider in this work, document alignment and ICD code classification, is around 800 words, with documents up to 40k words.

In order to be able to process long inputs, more efficient architectures such as Linformer (Wang et al., 2020), Big Bird (Zaheer et al., 2020) or Longformer (Beltagy et al., 2020) implement sparse attention mechanisms that scale linearly instead of quadratically. These architectures accept at least 4096 input tokens. With this length, one can embed most Wikipedia articles, news articles, medical records, etc. These architectures are available as pre-trained models in English³ and can be fine-tuned for NLP tasks such as document classification, question answering or summarisation. How-

¹https://github.com/sonalsannigrahi/Document_Embeddings

²https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia
Consulted on Feb. 2023.

³<https://huggingface.co>

ever, multilingual or non-English versions are rare. For most languages, it is not just a matter of training a model from scratch, but the amount of documents is just not enough to train high quality models.

LASER (Artetxe and Schwenk, 2019; Heffernan et al., 2022), Sentence BERT (Reimers and Gurevych, 2019, 2020) and LaBSE (Feng et al., 2022) are representative and state-of-the-art models which largely adapt language models to be used as task-independent sentence representations. These models are available as pre-trained models and, contrary to the long sequence models introduced before, they are multilingual. LASER, which is not transformer-based, allows longer inputs.

These observations explain why the two main approaches to obtain multilingual (or non-English) document embeddings are simply (i) truncating the input to 512 tokens and feeding it into a sentence-level encoder or (ii) splitting the document in shorter fragments and then combine their embeddings. There are few works that do a systematic comparison among methods. Park et al. (2022) perform a systematic study for document classification in English and found that the most sophisticated models such as Longformer do not always improve on a baseline that truncates the input to fit it into a fine-tuned BERT. The results mostly depend on how the information is distributed along a document and therefore varies from dataset to dataset.

In this work we explore multilingual document-level embeddings in three tasks in detail: *document alignment*, a bilingual semantic task; *ICD code (multi-label) classification* in 2 languages; and *cross-lingual document classification* in 8 languages. We compare input token number truncation, sentence averaging as well as some simple windowing and in some cases new augmented and learnable approaches. Our results show that a simple sentence average is a very strong baseline, even better than considering the whole document as a single unit, but that positional information is needed when the distribution of information across a document is not uniform.

2 Related Work

Word embeddings have been exceptionally successful in many NLP applications (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017). Subsequent works developed methods to learn continuous vector representations for longer sequences such as sentences or even documents. Skip-thought

embeddings (Kiros et al., 2015) train an encoder-decoder architecture to predict surrounding sentences. Conneau et al. (2017) showed that the task on which sentence representations are learnt significantly impacts their quality. InerSent (Conneau et al., 2017), a Siamese BiLSTM network with max pooling, and Universal Sentence Encoder (Cer et al., 2018), a transformer-based network, are trained over the SNLI dataset which is suitable for learning semantic representations (Bowman et al., 2015).

These methods primarily work on a single language but as multilingual representations have attracted more interest, sentence-level embeddings have been extended to obtain a wider language coverage. Artetxe and Schwenk (2019) (LASER) learn joint multilingual sentence representations for 93 languages based on a single BiLSTM encoder with a shared BPE vocabulary trained on publicly available parallel corpora. However, this architecture was shown to underperform in high-resource scenarios (Feng et al., 2022). LASER is especially interesting for our work as, being LSTM-based, it does not have the 512-length constraint. Li and Mak (2020) introduce T-LASER, which is a version of LASER that uses a transformer encoder in place of the original bidirectional LSTM. However, this model was tested only on the Multilingual Document Classification (MLDoc) corpus (Schwenk and Li, 2018), which does not have significantly long documents. Similarly, Reimers and Gurevych (2019) (sBERT in the following) extended a transformer-encoder architecture, BERT, by using a Siamese network with cosine similarity for contrastive learning in order to derive semantically meaningful sentence representations. More recently, Feng et al. (2022) (LaBSE) explored cross-lingual sentence embeddings with BERT by introducing a pre-trained multilingual language model component and show that on several benchmarks, their method outperforms many state-of-the-art embeddings such as LASER.

While sentence-level representations have been widely explored in literature, document-level representations are less well-explored. The earliest approaches in learning document-level vector representations included an extension of the Word2Vec algorithm named Doc2Vec (Le and Mikolov, 2014) with two variants proposed, a bag-of-words and a skip-gram based model. However, while these methods worked well at the word-level,

the document-level counterpart led to issues in scaling due to large vocabulary sizes (Lau and Baldwin, 2016). Due to these limitations, further works have attempted to improve the computational bottlenecks involved with training on long sequences such as documents. Linformer (Wang et al., 2020) is a transformer-based architecture with linear complexity due to a sparse self-attention mechanism making it significantly more memory- and time-efficient in comparison with the original transformer (Vaswani et al., 2017). Works such as Big Bird (Zaheer et al., 2020) and Longformer (Beltagy et al., 2020) introduced a sparse attention mechanism and localised global attention respectively. BigBird is able to handle sequences of up to 4,096 tokens and Longformer scales linearly with the sequence length, with experiments on sequences of length upto 32,256. To the best of our knowledge, to date not much has been done to extend them beyond English. Shen (2021) and Romero (2022) made available Chinese and Spanish Longformer models, respectively, while Sagen (2021) trained a multilingual version starting from a RoBERTa checkpoint and not from scratch. We use Longformer as a comparison system in our experiments but we do not consider the multilingual model given that multilinguality was achieved by fine-tuning on question answering data and we do not explore this task.

3 Sentence Embeddings

We use three multilingual sentence-level embedding models that cover different languages, architectures and learning objectives:

LASER (Schwenk and Douze, 2017; Artetxe and Schwenk, 2019) uses max-pooling over the output of a stacked BiLSTM-encoder. The encoder is extracted from an encoder-decoder machine translation setup trained on parallel corpora over 93 languages. Since it is not based on transformers but on LSTMs, the maximum number of input tokens can in principle be arbitrary and is set to 12,000.

LaBSE Feng et al. (2022) train a multilingual BERT-like model with a masked LM and translation LM objective functions. A dual-encoder transformer is initialised with the model and fine-tuned on a translation ranking task. The final model covers 109 languages. The maximum number of input tokens is 512.

sBERT Reimers and Gurevych (2019) use the output of BERT-base with mean pooling to create a fixed-size sentence representation. A Siamese-BERT architecture trained on NLI is used to obtain the final sentence-embedding model. The maximum number of input tokens is 512, with a default value of 128. We use the multilingual version (Reimers and Gurevych, 2020).

4 Document Embeddings

We divide our approaches to build document embeddings into three families: in (i) *Document Excerpts*, we feed token sequences as they are directly into LASER, LaBSE and sBERT to obtain a document-level representation, in (ii) *Sentence Weighting Schemes*, we divide documents into sentences represented using base sentence embeddings and then explore different combination and weight strategies to obtain document embeddings, in (iii) *Windowing Approaches*, we study different distributions to learn document-level positional and semantic information.

(i) Document Excerpts

All Tokens: The full document is fed into the system (no truncation). We explore this option only with LASER which does not have the 510-token-length restriction⁴ and when possible (English, Spanish and Chinese) with Longformer.

Top-N Tokens: The document is truncated to the first $n = 510$ tokens.

Bottom-N Tokens: The last $n = 510$ tokens are fed into the system.

Top-N + Bottom-M Tokens: We select $N = 128$ and $M = 382$ to use the first N and last M tokens of the documents. These values are based on empirical explorations by Sun et al. (2019).

(ii) Sentence Weighting Schemes

Sentence Average: Each base sentence embedding (obtained with LASER, LaBSE or SBERT) is given a uniform weight. This computes the vanilla average embedding vector of all sentences in the document.

⁴That is the maximum length of tokens accepted by transformer-style embedding models, 512 without the [CLS] and [SEP] tokens.

Top/Bottom-Half Average: Only the top (bottom) half of the sentences in the document are considered for averaging.

TF-IDF Weights: We compute TF-IDF scores for all terms in a document, and average their values at sentence level. The base sentence embeddings (LASER, LaBSE, SBERT) are then weighted by the normalised value of the TF-IDF averages. Following [Buck and Koehn \(2016b\)](#), we use different TF-IDF computations based on variations of term frequency tf and inverse document frequency idf definitions. For words w in a document d belonging to a collection D we report results using:

$$tf_2(w, d) = \text{freq}(w, d) \quad (1)$$

$$tf_4(w, d) = 0.4 + 0.6 \frac{\text{freq}(w, d)}{\max_{\tilde{w}} \text{freq}(\tilde{w}, d)} \quad (2)$$

$$idf_4(w, d) = \log\left(1 + \frac{|D|}{df(w, |D|)}\right), \quad (3)$$

with $df(w, D) = |\{d \in D | w \in d\}|$, and

$$tf_i idf_j(S_k) = \frac{\sum_{w \in S_k} tf_i(w, d) idf_j(w, d)}{\#w_k}, \quad (4)$$

where S_k is a sentence in a given document d , and $\#w_k$ is the number of words in sentence S_k .

The weights of these models are fixed for the static tasks and used as initialisation when training a classifier.

(iii) Windowing Approaches

TK-PERT: [Thompson and Koehn \(2020\)](#) introduced a windowing approach that weights the contribution of each sentence according to the modified PERT function ([Vose, 2008](#)) and a down-weighting function for boilerplate text. The latter was introduced to deal with webpages but it can be ignored for other types of documents. The smoothed overlapping windowing functions based on a cache of the PERT distribution (PERT-cache) encode fine-grained positional information into the resultant document vector.

A document with N sentences $S_{i|i \in \{0, \dots, N-1\}}$ is split uniformly into J parts and the final representation D for a document is given by a concatenation of normalised position-weighted (via PERT) sub-vectors where each sub-vector D_j is

$$D_j = \sum_{n=0}^{N-1} \text{emb}(S_n) P_j(n) B(S_n), \quad (5)$$

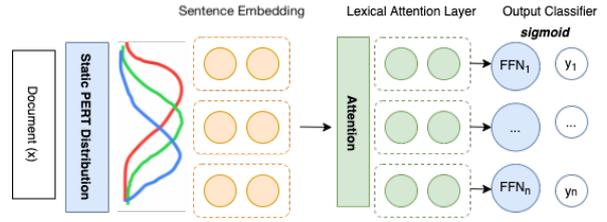


Figure 1: ATT-PERT model for classification. A static modified PERT distribution is used to extend the sentence embeddings to documents. Afterwards, an attention-weighted classifier is learnt.

emb is the (LASER, LaBSE, SBERT) embedding of sentence n , P is the modified PERT function for part j and B is a boilerplate function if there is one. In cases when no boilerplate text is present, we set it to 1.

Following [Thompson and Koehn \(2020\)](#) setting for the modified PERT distribution, we use $J = 16$ and set its shape parameter to $\gamma = 20$.

TF-PERT: is a new extension of TK-PERT to further incorporate semantics. PERT focuses on positional information encoded in the document while TF-IDF focuses on the semantic information, therefore a combined metric would likely be able to consider both features. We combine the two contributions with a multiplication at sentence level:

$$D_j = \sum_{n=0}^{N-1} \text{emb}(S_n) P_j(n) B(S_n) tfidf(S_n), \quad (6)$$

where we use the same notation as in Eqs. 4 and 5.

ATT-PERT: is a new extension of TK-PERT to further incorporate a global learnable attention. Figure 1 illustrates the basic architecture. The PERT distribution encodes global positional information of the document. By adding an attention layer over it, we introduce a *global attention* that weights the different parts of the document and that is combined with the standard *local attention* at word level performed by the sentence encoder. Mathematically,

$$D_j = \sum_{n=0}^{N-1} \text{emb}(S_n) P_j(n) a_j(n), \quad (7)$$

where S_n refers to the sentence embedding that has been trained for a classification task and $a_j(n)$ is the respective global attention weight.

In TK-PERT, the static PERT distribution is multiplied by the fine-tuned sentence embeddings. In

	# Documents		Length	
	Train	Test	Avg.	Max.
<i>Document Alignment, WMT2016</i>				
English	349k	682k	737	43.3k
French	225k	522k	842	45.2k
Web Domains	49	203	-	-
Gold Pairs	1624	2402	-	-
<i>Multi-label Classification, ICD Code Classification</i>				
Spanish	1001	1600	792	4352
German	8385	407	876	2249
<i>Document Classification, MLDoc</i>				
English	10k	4000	275	576
German	10k	4000	342	675
French	10k	4000	445	782
Italian	10k	4000	376	765
Spanish	9458	4000	354	778
Japanese	10k	4000	327	897
Russian	5216	4000	235	967
Chinese	10k	4000	562	983

Table 1: Number of documents and average tokenised document length in sentencepiece units (prior to boilerplate downweighting for Document Alignment) for the three tasks used in the experiments.

contrast, in ATT-PERT, the distribution is multiplied with the embeddings prior to training a classifier without freezing the embedding layer, as this allows the positional weights in the PERT distribution to be trained for the specific task.

ATT-TF-PERT: is a new extension of TF-PERT to further incorporate a global learnable attention as in ATT-PERT. In this configuration, we learn combined TF-IDF-PERT weighted embeddings whose attention weights are further updated while training the classifier. We use the same *global attention* $a_j(n)$ as in ATT-PERT, however here it is multiplied with both the TF-IDF weight of the sentence $tfidf_j(w, S_n)$ as computed in the TF-IDF set up and the PERT distribution $P_j(n)$ as in TK-PERT:

$$D_j = \sum_{n=0}^{N-1} \text{emb}(S_n) P_j(n) a_j(n) \text{tfidf}(S_n). \quad (8)$$

5 Evaluation Tasks

We apply the different configurations discussed above across the following tasks:

Bilingual Document Alignment aims at aligning documents from two collections in language L1 and language L2 according to whether they are

parallel or comparable. In our experiments, we use the data given for the WMT 2016 Shared Task on Bilingual Document Alignment to align French web pages to English web pages for a given crawled webdomain (Buck and Koehn, 2016a). In these experiments we do not perform any learning using the training data, but just estimate document-level semantic similarity between the pairs of documents in the test set. To compute this, we find the top $K=32$ candidate translations using approximate nearest neighbor search via FAISS⁵ as in (Buck and Koehn, 2016a). We use cosine similarity to quantify semantic similarity on the document embeddings.

Multi-label ICD Code Classification aims at assigning one or more ICD-10 codes to medical-domain texts (electronic health records). Here there can be an arbitrary number of ICD-10 codes assigned to the input text. In particular, out of all the possible ICD-10 Codes, 4 account for more than 90% of the documents, making this an imbalanced classification task and leading to the ‘tail end problem’ (Chapman and Neumann, 2020). We use the CLEF eHealth 2019 task for German non-technical summaries (Neves et al., 2019) and CANTEMIST-CODING (Miranda-Escalada et al., 2020) for Spanish electronic health records. Here, we learn a weighted-attention classifier layer (Lee et al., 2022) on top of the base document embeddings consisting of a feed-forward neural network with a single hidden layer of 10 units.

Cross-lingual Document Classification aims at classifying documents in a set of predefined categories in a language (usually English) and then transfer the model to unseen languages. We use the MLDoc dataset for this purpose (Schwenk and Li, 2018). The corpus contains 1,000 development documents and 4,000 test documents in eight languages (English, German, French, Italian, Spanish, Japanese, Russian and Chinese), divided in four different genres with uniform class priors. For zero-shot transfer, we train a classifier on top of the multilingual document representations estimated as described in Section 4 by using only the English training data and the hyperparameters optimised in Artetxe and Schwenk (2019). Similar to the previous classification task, we use a feed-forward neural network with one hidden layer with 10 units.

⁵<https://github.com/facebookresearch/faiss>

	LASER	LaBSE	sBERT
All tokens	81.2 ^{+0.3} _{-0.4}	—	—
Top-510 tokens	70.8 ^{+0.2} _{-0.3}	71.2 ^{+0.5} _{-0.4}	72.3 ^{+0.2} _{-0.4}
Bottom-510 tokens	65.8 ^{+0.5} _{-0.3}	66.3 ^{+0.7} _{-0.8}	67.1 ^{+0.6} _{-0.7}
Top-128 + Bot-312	75.3 ^{+0.5} _{-0.5}	76.1 ^{+0.3} _{-0.5}	74.2 ^{+0.3} _{-0.3}
Sentence Average	81.8 ^{+0.7} _{-0.5}	83.4 ^{+0.6} _{-0.6}	82.3 ^{+0.4} _{-0.6}
Top-Half Avg.	82.2 ^{+0.3} _{-0.5}	81.3 ^{+0.6} _{-0.8}	81.7 ^{+0.7} _{-0.6}
Bottom-Half Avg.	67.8 ^{+0.8} _{-0.7}	66.5 ^{+0.4} _{-0.3}	65.3 ^{+0.5} _{-0.4}
TF-IDF Weighted			
$tf_2 - idf_4$	80.2 ^{+0.7} _{-0.4}	80.5 ^{+0.7} _{-0.6}	79.3 ^{+0.2} _{-0.4}
$tf_4 - idf_4$	86.3 ^{+0.5} _{-0.4}	87.2 ^{+0.3} _{-0.4}	85.4 ^{+0.6} _{-0.5}
TK-PERT (Euclidean)	93.2 ^{+0.7} _{-0.8}	93.5 ^{+0.6} _{-0.5}	92.8 ^{+0.5} _{-0.4}
TK-PERT (cosine)	96.4 ^{+0.6} _{-0.5}	94.2 ^{+0.5} _{-0.4}	95.3 ^{+0.8} _{-0.9}
TF-PERT (cosine)	93.4 ^{+0.5} _{-0.3}	92.5 ^{+0.3} _{-0.4}	93.1 ^{+0.4} _{-0.4}

Table 2: Document recall on WMT-16 Shared Task on English–French document alignment. Best score for each family is in bold.

We use this classifier on top on the multilingual embeddings to evaluate the system on the remaining languages.

Table 1 shows the statistics for the datasets used in the three tasks as well as an average length of training instances in terms of sentencepiece tokens.⁶ The average document length in the document alignment and ICD code classification tasks is larger than 512 tokens, making the usage of sentence embeddings alone insufficient. This is not the case for document classification, but we still consider it in order to compare the different approaches and add a highly multilingual setting.

6 Results and Discussion

Thompson and Koehn (2020) empirically obtained the best trade-off between accuracy and inference time when using PCA-reduced sentence embeddings of 128 dimensions in the bilingual document alignment task. We performed equivalent experiments with 128 and 256 dimensions for selected configurations in the three tasks and confirmed the trend. As we obtained no major gains in using more dimensions, we report all the results for the three tasks with 128-dimensional sentence embeddings.

We report confidence intervals at 95% confidence level using bootstrap resampling with 1000 samples for document alignment, 500 samples for ICD code classification and 1000 samples for document classification.

⁶<https://github.com/google/sentencepiece>

Bilingual Document Alignment quality ranges from 65% to 96% recall depending on the document embedding method. Table 2 shows the results obtained for all the configurations considered. A simple sentence average achieves a recall around 82% (depending on the sentence embedding used). When using LASER, the only method that allows the comparison, the recall with sentence average is larger but not statistically significantly over embedding the full document as a single unit (81.8% vs 81.2%). Taking a token-based excerpt of the document is 10 percentage points below sentence-averaging the same excerpt. The information in webpages seems to be more densely distributed towards the top of the page. Looking at the top-half versus the bottom-half of the sentences of the webpages, there is a 17% reduction in the scores obtained. In these unweighted and average configurations in both the token and sentence-based methods, we do not encode any positional information: sentence order and semantic relevance is not considered in the final document embeddings. However, intuitively, these factors are indicative of each sentence’s contribution to the larger document embedding. In order to incorporate semantic relevance into our final embeddings, we consider the weighted average using TF-IDF. We explore several TF-IDF forms and obtain a difference of 7% on average among them. Table 2 shows the 2 most promising ones. With the best option ($tf_4 - idf_4$), TF-IDF weighting improves between 3 and 5 percentage points with respect to the sentence averaging which uses uniform weights. We use $tf_4 - idf_4$ for the next experiments when required as these formulae empirically performed the best. To include sentence order, we use the PERT-window based approach. TK-PERT outperforms all other methods by a margin of 11.7%. This result attests the relevance of contextual information, sentence order, and positional importance. Although we find improvements over the baseline models by introducing TF-IDF weights and the PERT distribution, a combination of the two in TF-PERT does not lead to further improvements.

The other dimension of the study, the particulars of sentence embeddings, is less important to the recall. LASER, LaBSE and sBERT achieve similar results. As we are working with French and English documents, both languages being high-resource, all base sentence embeddings are high-quality and therefore they do not impact the final

	LASER		LaBSE		sBERT	
	de	es	de	es	de	es
All tokens	73.1 ^{+0.5} _{-0.6}	<i>18.4</i> ^{+0.4} _{-0.3}	—	—	—	—
Top-510 tokens	65.6 ^{+0.8} _{-0.7}	16.5 ^{+0.5} _{-0.8}	68.2 ^{+0.5} _{-0.5}	19.2 ^{+0.6} _{-0.4}	63.2 ^{+0.7} _{-0.8}	18.3 ^{+0.5} _{-0.6}
Bottom-510 tokens	67.8 ^{+0.4} _{-0.9}	17.5 ^{+0.4} _{-0.9}	66.7 ^{+0.8} _{-0.6}	17.4 ^{+0.7} _{-0.6}	61.5 ^{+0.8} _{-0.6}	16.8 ^{+0.5} _{-0.7}
Top-128 + Bot-312	66.4 ^{+0.8} _{-0.6}	17.2 ^{+0.8} _{-0.7}	<i>69.1</i> ^{+0.7} _{-0.9}	18.7 ^{+0.7} _{-0.8}	<i>64.8</i> ^{+0.7} _{-0.5}	17.5 ^{+0.6} _{-0.8}
Sentence Average	72.1 ^{+0.9} _{-0.8}	17.0 ^{+0.7} _{-0.6}	74.5 ^{+0.8} _{-0.9}	24.2 ^{+0.8} _{-0.6}	68.9 ^{+0.7} _{-0.6}	20.3 ^{+0.8} _{-0.4}
Top-Half Avg.	68.4 ^{+0.7} _{-0.9}	16.5 ^{+0.5} _{-0.6}	68.3 ^{+0.4} _{-0.8}	18.9 ^{+0.5} _{-0.5}	61.5 ^{+0.7} _{-0.6}	16.4 ^{+0.8} _{-0.6}
Bottom-Half Avg.	63.1 ^{+0.7} _{-0.6}	15.8 ^{+0.8} _{-0.7}	67.4 ^{+0.8} _{-0.9}	15.2 ^{+0.6} _{-0.7}	58.6 ^{+0.9} _{-0.8}	17.9 ^{+0.7} _{-0.6}
TF-IDF Weighted	65.3 ^{+0.5} _{-0.4}	17.2 ^{+0.7} _{-0.8}	68.2 ^{+0.9} _{-1.0}	19.2 ^{+0.9} _{-0.7}	63.2 ^{+0.6} _{-0.8}	18.3 ^{+0.7} _{-0.6}
TK-PERT	68.2 ^{+0.8} _{-0.6}	22.1 ^{+0.7} _{-0.4}	70.1 ^{+0.8} _{-0.7}	20.1 ^{+0.7} _{-0.4}	65.2 ^{+0.8} _{-0.6}	19.5 ^{+0.7} _{-0.8}
TF-PERT	68.5 ^{+0.4} _{-0.3}	23.4 ^{+0.6} _{-0.6}	68.6 ^{+0.3} _{-0.7}	21.3 ^{+0.5} _{-0.4}	65.4 ^{+0.6} _{-0.7}	18.7 ^{+0.4} _{-0.3}
ATT-PERT	<i>70.7</i> ^{+0.7} _{-0.9}	32.2 ^{+0.7} _{-0.4}	72.1 ^{+0.8} _{-0.6}	30.1 ^{+0.7} _{-0.3}	66.3 ^{+1.3} _{-1.3}	27.4 ^{+0.8} _{-0.7}
ATT-TF-PERT	70.3 ^{+0.5} _{-0.4}	31.4 ^{+0.8} _{-0.7}	73.2 ^{+0.4} _{-0.8}	29.7 ^{+0.6} _{-0.5}	66.1 ^{+0.9} _{-0.8}	27.1 ^{+0.5} _{-0.6}

Table 3: F1 scores for the Multi-label ICD code classification task for German (de) and Spanish (es) documents. Best scores are in bold, and best scores per family are in italics.

model strongly in a consistent way.

Multi-label ICD Code Classification shows the same trend with respect to different sentence embeddings as above for German and Spanish, with a slight preference towards LaBSE embeddings. Table 3 shows the results for this task. There is a large discrepancy between the scores for the German and the Spanish datasets, as already noticed by the evaluations in the original corresponding shared tasks. The classification in Spanish achieves much lower results probably because of a very small training corpus. Our results indicate that the information is spread throughout documents in this case. The difference between only using the top of the document and only using the bottom part is small, and using the whole document either by sentence averaging or considering it a single unit is always better than any of its parts at a 95% significance level. Semantic (TF-IDF) and positional (TK-PERT) information is less relevant. For the German task, either considering the full document as a whole (*All tokens*) or averaging all the sentences gives the highest performance. For the Spanish task, even with a very low overall quality, learning specific weights for different parts of the document (ATT-PERT) boosts the quality. Comparing ATT-PERT with TK-PERT, we find that the trainable alternative performs better for all languages and base embeddings considered, however, the improvements are not statistically significant for all base embeddings in the case of German. In general, the windowing approaches that combine semantics with position (TF-PERT and ATT-TF-PERT) do not perform significantly better than the pure positional methods

(TK-PERT and ATT-PERT). This can be explained by looking a concrete example. Figure 2 shows the distribution of weights across a document from the CANTEMIST health record corpus for 8 configurations based on LASER embeddings. The example shows that the effect of the *tfidf* component in ATT-TF-PERT (configuration 7) is equivalent to move weight mass from ATT-PERT (configuration 6) into TF-IDF (configuration 3). When this happens, the result is a score in the middle of the way between ATT-PERT and TF-IDF. In this document, a medical diagnostic evaluation is detailed and includes patient information, past diagnoses, family medical history, as well potential evolution of the disease. We observe that while the ‘Sentence average’ configuration places largely equivalent weights on all the sentences, the TF-IDF weights place more emphasis on the beginning and end of the document which stores information about the patient and the evolution of the disease respectively. This behaviour is similar to the one exhibited by the PERT family of methods: the weight pattern observed for configurations 3-7 remain quite consistent but vary in their intensity.

Cross-lingual Document Classification data allows us to test the embedding methods on 8 languages (Table 4). The languages belong to three families, Indo-European (Germanic, Romance and Slavic), Japonic and Sino-Tibetan. All languages are high-resourced and included in our pre-trained sentence representation models. MLDoc documents are shorter than 1,000 tokens with an average length of 275 tokens for English and 562 for Chinese; the other languages stay in the middle.

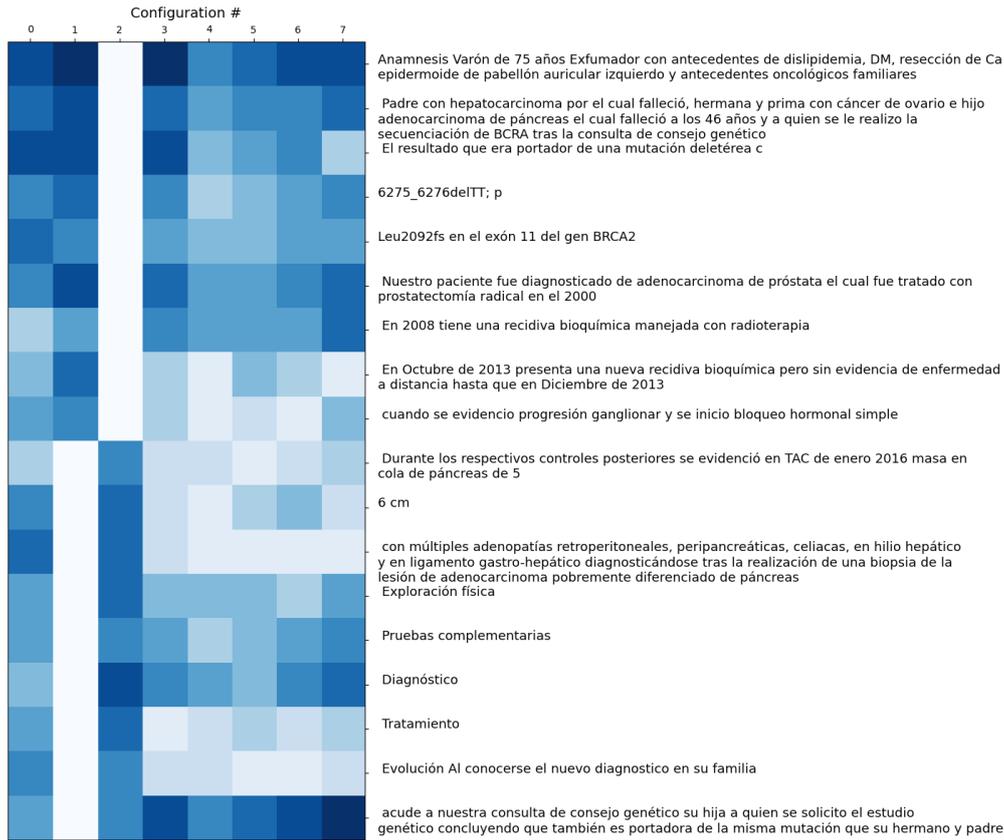


Figure 2: Sentence weights for an example document with LASER embeddings and the configurations: 0-Sentence average, 1-Top-Half, 2-Bottom-Half, 3-TF-IDF weighted, 4-TK-PERT, 5-TF-PERT, 6-ATT-PERT, 7-ATT-TF-PERT.

Given that length, the methods that use different 510-sized excerpts of the documents do not differ much as all the excerpts are—for most of the documents—the same.

Accuracies in Table 4 show that the documents convey slightly more meaning at the top part than at the bottom (*Top-Half Avg.* vs *Bottom-Half Avg.*). The sentence average is a very strong baseline and, for half of the languages (English, German, Russian and Chinese), this is statistically significantly better at 95% confidence level than treating the document as a single unit with LASER. The TF-IDF version is worse than the simple sentence average except for Japanese. Japanese has the lowest accuracy for all the languages and a high difference between the information at the top and the bottom of its documents. In general, position (TK-PERT) is more important than semantics (TF-IDF) and learning task-specific weights (ATT-PERT) further increases accuracy. Additional experiments with TF-PERT and ATT-TF-PERT do not show statistically significant improvements over their counterparts TK-PERT and ATT-PERT, similarly to the trend observed in the previous tasks. For English, Chinese

and Spanish, we are further able to compare the performance of pre-trained large-input transformers. Longformer achieves 92.3% of accuracy for English, which is 4.1% better than the 88.7% that LASER achieves in the *All tokens* configuration and about 2% better than the best performing architecture, the sentence average of LaBSE embeddings (90.9%). However, the latter is not statistically significant at 95% confidence level. The result is different for Chinese and Spanish. In both cases, considering all tokens with LASER and sentence average are better than Longformer, although the difference is not statistically significant for Spanish. This indicates that smaller amounts of training data can prevent native full document-level embeddings to be extended to languages other than English.

7 Summary and Conclusions

In this work, we studied effective methods for developing multilingual document-level representations. We used state-of-the-art sentence-level embeddings as basic units and systematically compare different pooling methods to evaluate these representations at the document level. We performed

		en → xx							
		en	de	es	fr	it	ja	ru	zh
Longformer		92.3 ^{+0.7} _{-0.8}	–	76.9 ^{+0.6} _{-0.7}	–	–	–	–	68.5 ^{+0.4} _{-0.5}
LASER	All tokens	88.7 ^{+1.1} _{-0.8}	83.6 ^{+0.5} _{-0.4}	77.4 ^{+0.9} _{-0.8}	78.1 ^{+0.7} _{-0.8}	65.1 ^{+0.6} _{-0.7}	61.8 ^{+0.6} _{-0.4}	66.6 ^{+0.5} _{-0.6}	70.1 ^{+0.9} _{-0.8}
	Sentence Average	89.9 ^{+0.9} _{-0.8}	84.8 ^{+0.7} _{-0.6}	77.3 ^{+0.9} _{-0.7}	77.9 ^{+0.5} _{-0.9}	<i>64.9</i> ^{+0.4} _{-0.8}	60.3 ^{+0.8} _{-0.7}	67.8 ^{+0.8} _{-0.9}	71.9 ^{+0.8} _{-0.7}
	Top-Half Avg.	86.4 ^{+0.3} _{-0.9}	83.5 ^{+0.4} _{-0.5}	75.8 ^{+0.9} _{-0.6}	76.2 ^{+0.8} _{-0.5}	63.2 ^{+0.7} _{-0.9}	56.5 ^{+0.7} _{-0.6}	64.1 ^{+0.7} _{-0.8}	67.5 ^{+0.8} _{-0.7}
	Bottom-Half Avg.	83.2 ^{+0.4} _{-0.6}	81.4 ^{+0.7} _{-0.8}	71.2 ^{+0.7} _{-0.6}	70.5 ^{+0.8} _{-0.9}	59.2 ^{+0.5} _{-0.4}	50.4 ^{+0.6} _{-0.7}	56.2 ^{+0.6} _{-0.4}	60.3 ^{+0.6} _{-0.7}
	TF-IDF Weighted	86.3 ^{+0.8} _{-0.8}	85.1 ^{+0.5} _{-0.8}	75.3 ^{+0.7} _{-0.4}	74.1 ^{+0.7} _{-0.8}	56.4 ^{+0.7} _{-0.7}	61.4 ^{+0.6} _{-0.8}	60.2 ^{+0.7} _{-0.6}	71.5 ^{+0.4} _{-0.5}
	TK-PERT	89.1 ^{+0.4} _{-0.7}	85.2 ^{+0.6} _{-0.6}	75.6 ^{+0.8} _{-0.7}	78.2 ^{+0.8} _{-1.1}	63.6 ^{+0.9} _{-0.7}	62.3 ^{+0.8} _{-0.4}	67.8 ^{+0.6} _{-0.7}	71.1 ^{+0.4} _{-0.6}
	TF-PERT	88.7 ^{+0.6} _{-0.5}	84.8 ^{+0.8} _{-0.6}	75.4 ^{+0.5} _{-0.4}	77.9 ^{+0.6} _{-0.4}	61.2 ^{+0.9} _{-0.8}	61.8 ^{+0.3} _{-0.4}	67.2 ^{+0.5} _{-0.5}	70.8 ^{+0.6} _{-0.5}
	ATT-PERT	89.2 ^{+0.7} _{-0.8}	86.2 ^{+0.6} _{-0.5}	77.5 ^{+0.8} _{-0.7}	79.1 ^{+1.0} _{-0.8}	64.0 ^{+0.3} _{-0.9}	62.5 ^{+0.6} _{-0.4}	66.2 ^{+0.8} _{-0.9}	71.3 ^{+0.7} _{-0.6}
	ATT-TF-PERT	88.5 ^{+0.6} _{-0.5}	86.0 ^{+0.4} _{-0.3}	76.7 ^{+0.4} _{-0.5}	78.9 ^{+0.5} _{-0.5}	63.8 ^{+0.5} _{-0.6}	62.8 ^{+0.5} _{-0.4}	66.5 ^{+0.4} _{-0.7}	70.5 ^{+0.3} _{-0.5}
LaBSE	Sentence Average	90.9 ^{+0.6} _{-0.7}	85.2 ^{+0.8} _{-0.7}	75.6 ^{+0.5} _{-0.5}	79.9 ^{+0.5} _{-0.3}	66.9 ^{+0.9} _{-0.6}	58.3 ^{+0.7} _{-0.6}	65.4 ^{+0.5} _{-0.5}	70.1 ^{+0.5} _{-0.6}
	Top-Half Avg.	86.1 ^{+0.2} _{-0.8}	80.5 ^{+0.5} _{-0.9}	73.2 ^{+0.7} _{-0.8}	76.5 ^{+0.3} _{-0.7}	62.5 ^{+0.6} _{-0.8}	56.1 ^{+0.5} _{-0.6}	61.8 ^{+1.0} _{-0.9}	67.3 ^{+0.7} _{-0.8}
	Bottom-Half Avg.	85.4 ^{+1.2} _{-1.1}	78.7 ^{+0.5} _{-0.6}	71.4 ^{+0.6} _{-0.7}	73.3 ^{+0.8} _{-0.6}	59.6 ^{+0.4} _{-0.7}	50.7 ^{+0.3} _{-0.8}	58.9 ^{+0.7} _{-0.6}	61.4 ^{+0.9} _{-1.1}
	TF-IDF Weighted	86.2 ^{+0.2} _{-0.6}	84.1 ^{+0.5} _{-0.4}	73.9 ^{+0.6} _{-0.3}	77.1 ^{+0.3} _{-0.4}	62.6 ^{+0.2} _{-0.5}	59.3 ^{+0.3} _{-0.6}	65.4 ^{+0.5} _{-0.4}	68.1 ^{+0.8} _{-0.7}
	TK-PERT	87.1 ^{+0.5} _{-0.9}	83.6 ^{+0.8} _{-0.6}	75.8 ^{+0.5} _{-0.4}	79.1 ^{+0.7} _{-0.8}	62.5 ^{+0.3} _{-0.8}	60.0 ^{+0.6} _{-0.7}	64.9 ^{+0.6} _{-0.4}	70.6 ^{+0.7} _{-0.6}
	TF-PERT	86.2 ^{+0.5} _{-0.4}	84.7 ^{+0.4} _{-0.7}	77.3 ^{+0.7} _{-0.6}	76.3 ^{+0.6} _{-0.5}	62.8 ^{+0.5} _{-0.5}	61.2 ^{+0.7} _{-0.6}	64.5 ^{+0.6} _{-0.5}	69.2 ^{+0.5} _{-0.6}
	ATT-PERT	88.9 ^{+0.8} _{-0.6}	84.3 ^{+0.8} _{-0.9}	77.3 ^{+0.5} _{-0.5}	79.4 ^{+0.7} _{-0.9}	63.8 ^{+0.6} _{-0.7}	62.2 ^{+0.8} _{-0.5}	65.9 ^{+0.7} _{-0.9}	71.2 ^{+0.8} _{-0.7}
	ATT-TF-PERT	88.4 ^{+0.4} _{-0.3}	85.4 ^{+0.9} _{-0.6}	77.2 ^{+0.4} _{-0.4}	78.2 ^{+0.4} _{-0.5}	65.7 ^{+0.5} _{-0.3}	61.3 ^{+0.7} _{-0.5}	65.3 ^{+0.7} _{-0.8}	67.4 ^{+0.6} _{-0.8}
sBERT	Sentence Average	85.1 ^{+0.6} _{-0.7}	85.2 ^{+0.6} _{-0.7}	75.7 ^{+0.6} _{-0.8}	78.2 ^{+0.6} _{-0.7}	64.5 ^{+0.7} _{-0.5}	60.4 ^{+0.8} _{-0.6}	66.4 ^{+0.8} _{-0.7}	69.5 ^{+0.8} _{-0.7}
	Top-Half Avg.	83.2 ^{+0.8} _{-0.6}	84.1 ^{+0.7} _{-0.6}	71.3 ^{+0.5} _{-0.6}	76.5 ^{+0.8} _{-0.6}	60.8 ^{+0.7} _{-0.9}	60.4 ^{+0.9} _{-1.2}	62.8 ^{+0.8} _{-0.7}	63.5 ^{+0.9} _{-0.8}
	Bottom-Half Avg.	80.6 ^{+0.7} _{-0.6}	81.3 ^{+0.5} _{-0.8}	66.5 ^{+0.4} _{-0.4}	70.1 ^{+0.6} _{-0.4}	56.5 ^{+0.4} _{-0.8}	58.7 ^{+0.5} _{-0.6}	56.1 ^{+0.7} _{-0.6}	60.5 ^{+0.5} _{-0.4}
	TF-IDF Weighted	84.2 ^{+0.4} _{-0.5}	82.8 ^{+0.5} _{-0.4}	75.1 ^{+0.6} _{-0.7}	74.3 ^{+0.4} _{-0.6}	63.2 ^{+0.2} _{-0.2}	61.2 ^{+0.4} _{-0.7}	63.4 ^{+0.5} _{-0.3}	65.8 ^{+0.7} _{-0.6}
	TK-PERT	86.2 ^{+0.6} _{-0.7}	84.1 ^{+0.8} _{-0.7}	73.9 ^{+0.6} _{-0.6}	77.1 ^{+0.8} _{-0.6}	62.6 ^{+0.6} _{-0.8}	59.3 ^{+0.5} _{-0.5}	65.4 ^{+0.8} _{-0.6}	68.1 ^{+0.6} _{-0.7}
	TF-PERT	85.8 ^{+0.5} _{-0.4}	83.7 ^{+0.2} _{-0.4}	72.7 ^{+0.6} _{-0.5}	76.5 ^{+0.4} _{-0.3}	62.0 ^{+0.6} _{-0.5}	60.4 ^{+0.3} _{-0.6}	64.3 ^{+0.4} _{-0.8}	68.2 ^{+0.7} _{-0.8}
	ATT-PERT	88.5 ^{+0.7} _{-0.6}	85.8 ^{+0.5} _{-0.5}	76.2 ^{+0.8} _{-0.4}	77.4 ^{+0.5} _{-0.6}	62.1 ^{+0.6} _{-0.7}	60.8 ^{+0.3} _{-0.6}	66.1 ^{+0.7} _{-0.4}	69.5 ^{+0.8} _{-0.6}
	ATT-TF-PERT	85.6 ^{+0.5} _{-0.6}	84.3 ^{+0.3} _{-0.4}	75.1 ^{+0.6} _{-0.6}	76.8 ^{+0.5} _{-0.6}	61.3 ^{+0.8} _{-0.5}	62.7 ^{+0.4} _{-0.5}	65.8 ^{+0.5} _{-0.3}	66.4 ^{+0.6} _{-0.6}

Table 4: Accuracy for MLDoc classification on the zero-shot transfer task. Best results per language are shown in bold and per family in italics.

exhaustive evaluations across three sentence embedding models, three tasks and eight languages.

Our experiments show that specific **base sentence embedding models** (LASER, LaBSE, sBERT) do not impact the performance of the document-level embeddings much. We observe similar performance amongst them across all experiments. However, it is to be noted that we experiment with languages that while being morphologically distinct, are well resourced and covered by the three base sentence-embedding models. It would be interesting to explore how models behave when embeddings have a lower quality. For this, one would need to create evaluation datasets at the document level for low-resourced languages but this is out of the scope of this work.

We observed that a simple sentence average is a very strong **pooling strategy**, specially for classification tasks. Positional and contextual information is more important than semantic information for the final performance as exemplified by the fact that PERT-based weightings perform better than TF-IDF’s in all the tasks. When combining both,

positional and semantic information, we do not observe statistically significant improvements with respect to only including positional information. For the classification tasks which include a learnable layer, we extend TK-PERT to ATT-PERT (and the semantic counterparts) and include global trainable attention on the positional information. This global attention is beneficial in all the cases.

The **type of document** is also relevant to chose the best method. Long documents might have the most crucial information stored in different parts. For instance, webpages have a majority of their information in the first half of the document as we observed in the document alignment task. In this case, the positional information significantly outperforms any model that does not take it into account.

Limitations

One of the main focal points of this work is multilinguality. In the presented approaches, the multilinguality of the resultant document embeddings depends solely on the language coverage and cross-

lingual transfer ability of the pre-trained sentence embeddings used as basic units. Document-level representations are as robust to new languages and scripts as the base sentence embeddings are. Cross-lingual transfer is a perpendicular dimension not studied in this work.

We introduce ATT-PERT, a new learnable approach for the combination of sentence embeddings. This model is therefore of use for tasks with a learning/fine-tuning phase but it is not intended for ready-to-use multilingual document-level embeddings in contrast to the existing pre-trained sentence-level counterparts.

Acknowledgements

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under funding code 01IW20010 (CORA4NLP).

References

- Mikel Artetxe and Holger Schwenk. 2019. [Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond](#). *Transactions of the Association for Computational Linguistics*, 7:597–610.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#). *CoRR*, abs/2004.05150.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching word vectors with subword information](#). *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Christian Buck and Philipp Koehn. 2016a. [Findings of the WMT 2016 bilingual document alignment shared task](#). In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 554–563, Berlin, Germany. Association for Computational Linguistics.
- Christian Buck and Philipp Koehn. 2016b. [Quick and reliable document alignment via TF/IDF-weighted cosine distance](#). In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 672–678, Berlin, Germany. Association for Computational Linguistics.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. [Universal sentence encoder for English](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.
- Kathryn Annette Chapman and Günter Neumann. 2020. [Automatic ICD Code Classification with Label Description Attention Mechanism](#). In *IberLEF@ SE-PLN*, volume 2664 of *CEUR Workshop Proceedings*, pages 477–488. CEUR-WS.org.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. [Supervised learning of universal sentence representations from natural language inference data](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. [Language-agnostic BERT sentence embedding](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.
- Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. [Bitext mining using distilled sentence representations for low-resource languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2101–2112, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. [Skip-thought vectors](#). *Advances in neural information processing systems*, 28.
- Jey Han Lau and Timothy Baldwin. 2016. [An empirical evaluation of doc2vec with practical insights into document embedding generation](#). In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 78–86, Berlin, Germany. Association for Computational Linguistics.
- Quoc Le and Tomas Mikolov. 2014. [Distributed Representations of Sentences and Documents](#). In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Ma-*

- chine Learning Research, pages 1188–1196, Beijing, China. PMLR.
- Min Seok Lee, Seok Woo Yang, and Hong Joo Lee. 2022. Weight attention layer-based document classification incorporating information gain. *Expert Systems*, 39(1):e12833.
- Wei Li and Brian Kan-Wing Mak. 2020. Transformer based multilingual document embedding model. *ArXiv*, abs/2008.08567.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26:3111–3119.
- Antonio Miranda-Escalada, Eulàlia Farré, and Martin Krallinger. 2020. Named entity recognition, concept normalization and clinical coding: Overview of the cantemist track for cancer text mining in spanish, corpus, guidelines, methods and results. In *IberLEF@SEPLN*, volume 2664 of *CEUR Workshop Proceedings*, pages 303–323. CEUR-WS.org.
- Mariana Neves, Daniel Butzke, Antje Dörendahl, Nora Leich, Barbara Grune, and Gilbert Schönfelder. 2019. [Non-technical Summaries \(NTS\) of Animal Experiments Indexed with ICD-10 Codes \(Version 1.0\)](#).
- Hyunji Park, Yogarshi Vyas, and Kashif Shah. 2022. [Efficient classification of long documents using transformers](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 702–709, Dublin, Ireland. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.
- Manuel Romero. 2022. Spanish LongFormer. <https://huggingface.co/mrm8488/longformer-base-4096-spanish>.
- Markus Sagen. 2021. Large-context question answering with cross-lingual transfer. Master’s thesis, Uppsala University, Department of Information Technology.
- Holger Schwenk and Matthijs Douze. 2017. [Learning joint multilingual sentence representations with neural machine translation](#). In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.
- Holger Schwenk and Xian Li. 2018. [A corpus for multilingual document classification in eight languages](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Liu Shen. 2021. Chinese LongFormer. <https://huggingface.co/schen/longformer-chinese-base-4096>.
- Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.
- Brian Thompson and Philipp Koehn. 2020. [Exploiting sentence order in document alignment](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5997–6007, Online. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- David Vose. 2008. *Risk analysis: a quantitative guide*. John Wiley & Sons.
- Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. [Linformer: Self-attention with linear complexity](#). *CoRR*, abs/2006.04768.
- Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. [Big bird: Transformers for longer sequences](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Improving User Controlled Table-To-Text Generation Robustness

Hanxu Hu¹, Yunqing Liu², Zhongyi Yu¹ and Laura Perez-Beltrachini¹

¹ School of Informatics, University of Edinburgh, United Kingdom

² The Hong Kong Polytechnic University, HongKong

{huhhanxu1233, lyq6175215241, zhongyics}@gmail.com

lperez@exseed.ed.ac.uk

Abstract

In this work we study user controlled table-to-text generation where users explore the content in a table by selecting cells and reading a natural language description thereof automatically produce by a natural language generator. Such generation models usually learn from carefully selected cell combinations (clean cell selections); however, in practice users may select unexpected, redundant, or incoherent cell combinations (noisy cell selections). In experiments, we find that models perform well on test sets coming from the same distribution as the train data but their performance drops when evaluated on realistic noisy user inputs. We propose a fine-tuning regime with additional user-simulated noisy cell selections. Models fine-tuned with the proposed regime gain 4.85 BLEU points on user noisy test cases and 1.4 on clean test cases; and achieve comparable state-of-the-art performance on the ToTTo dataset.¹

1 Introduction

The goal of table-to-text generation is to provide the user with a description of the most relevant content in a given table (Lebret et al., 2016; Wiseman et al., 2018; Perez-Beltrachini and Lapata, 2018; Puduppully et al., 2019). Recently, Parikh et al. (2020) proposed a controlled table-to-text generation task where the goal is to automatically create a description for a determined subset of the table, namely the highlighted table cells. The main focus on Parikh et al.’s 2020 work is to assess the performance of neural text generators in a more controlled setting, i.e., when given an input table with explicit instructions (i.e., highlights) on what should be expressed in the output description. In this work, we view this task in the context of a natural language interface, as a *user controlled table-to-text* generation task, where users provide those

¹Our code is available at <https://github.com/hanxuhu/controllT2Trobus>

Table Title: Robert Craig (American football)
Section Title: National Football League statistics
Table Description:None

YEAR	TEAM	RUSHING					RECEIVING				
		ATT	YDS	AVG	LNG	TD	NO.	YDS	AVG	LNG	TD
1983	SF	176	725	4.1	71	8	48	427	8.9	23	4
1984	SF	155	649	4.2	28	4	71	675	9.5	64	3
1985	SF	214	1050	4.9	62	9	92	1016	11	73	6
1986	SF	204	830	4.1	25	7	81	624	7.7	48	0
1987	SF	215	815	3.8	25	3	66	492	7.5	35	1
1988	SF	310	1502	4.8	46	9	76	534	7.0	22	1
1989	SF	271	1054	3.9	27	6	49	473	9.7	44	1
1990	SF	141	439	3.1	26	1	25	201	8.0	31	0
1991	RAI	162	590	3.6	15	1	17	136	8.0	20	0
1992	MIN	105	416	4.0	21	4	22	164	7.5	22	0
1993	MIN	38	119	3.1	11	1	19	169	8.9	31	1
Totals	-	1991	8189	4.1	71	56	566	4911	8.7	73	17

Target Text: Craig finished his eleven NFL seasons with 8,189 rushing yards and 566 receptions for 4,911 receiving yards.

Figure 1: An example in the ToTTo dataset. The figure is retrieved from (Parikh et al., 2020). The cells coloured in yellow are the highlight cells.

highlights interactively by exploring the content of a given table and study these user interactions. Figure 1 illustrates the case where a user selects some cells (highlighted in yellow) and the generator provides a description thereof (shown below the table).

A crucial aspect of usability assessment for a generator in this interactive table-to-text task is robustness. In a recent study by (Mille et al., 2021), it has been shown that neural generation models fail to maintain their in distribution performance when confronted with realistic scenarios at test time such as typos in the input text. In the case of user controlled table-to-text generation, users may introduce noise when exploring the table content and select cell combinations that turn out to be unexpected, redundant, or incoherent. For example, in Figure 1, when the user wants to express "eleven seasons", they might miss one year or highlight the header cell. They may also select unrelated headers, for instance adding the header "LNG" ¹ to the current selection. Existing controlled table-to-text generation models (Parikh et al., 2020; Su et al., 2021; Kale and Rastogi, 2020) are trained on carefully selected cell combinations (**clean** cell highlights) from the ToTTo dataset (Parikh et al., 2020). We argue that these models will not generalize well in practice with user **noisy** highlights. No previous work has study model robustness under this practical set up.

We carry out a usability study to observe how users highlight cells in a table. Based on the imperfect cell selections that users produce, we automatically create additional data examples by corrupting examples from the original ToTTo dataset. We then fine-tune state-of-the-art table-to-text neural generation models with this additional data. We compare the performance of models fine-tuned only with clean cell highlights versus those trained with additional noisy cell highlights, both on a test set with clean and noisy highlights. Experimental results show that models fine-tuned with clean cell highlights only perform well on clean test cases (i.e., performance drops dramatically when evaluated on noisy cell highlights). That is, these models do not generalise well in practice with user noisy cell selections. In contrast, the proposed training scheme with additional noisy cell highlights not only makes user controlled table-to-text models achieve better performance in practical scenarios, but it also boosts performance on perfect inputs. Experimental results show that models fine-tuned with our proposed training regime gain 4.85 BLEU points on noisy and 1.4 BLEU points on clean highlights; and achieve comparable state-of-the-art performance on the ToTTo dataset.²

2 Methodology

We describe the process for creating user noisy cell highlights from examples in ToTTo (Parikh et al., 2020) (§2.1 and §2.2). Then, we evaluate models optimized with the standard training scheme (i.e., only on clean cell highlights) on the created noisy test cases. Results show that these models perform poorly. To improve model robustness, we propose a new learning regime described in §2.3. To further improve performance, we fine-tune with Reinforcement Learning (RL) based optimisation (§2.4). Finally, §2.5 summarises the learning schemes and objective functions we propose for robust user controlled table-to-text generation.

2.1 How Do Users Select Cells?

To understand how users proceed when exploring a table and selecting cells we carry out a human study using examples from the ToTTo dataset. Participants are given a plain table (i.e., without highlights) and asked to highlight cells according to an exploratory intention. For a more controlled setting, we give the sentence associated to the ta-

²ToTTo leaderboard.

ble as the exploratory intention. In this way, we avoid ambiguous post-selection analysis of what the user intention was. In addition, this allows us to compare user selections with reference highlights as well as differences (if any) in model generated texts given user and reference highlights.

We conduct this study on Amazon Mechanical Turk (the interface is described in Appendix C). We collect 90 user highlights (3 participants, volunteers known by the authors, and 30 examples from the validation set) and observe the following noise in their highlights. Participants apply different criteria to include (or not) table headers; select additional cells in columns/rows around cells containing relevant content; and do not select cells that contain content relevant to the intention.

2.2 Creating User Noisy Cell Selections

Given the input table T , the reference text S , and the reference highlight cells $H \in T$ relevant for generating S , we create noisy user cell selections as follows. We provide an example illustrating each noise type in Figure 2.

Noise 1: Additional Table Cells In practical scenarios, users may accidentally select random cells that are not related to their exploration intention. Thus, we randomly select k cells from the table cells in T that are not in H and add them into H to form a corrupted input H_1 . H_1 can be viewed as adding irrelevant information in the generation of the target text S .

Noise 2: Table Headers as Additional Inputs Reference highlight cells in the ToTTo dataset do not cover table headers. As we have observed, users may decide to include (or not) table headers in different cases. To simulate this, we first retrieve table headers corresponding to highlight cells in H . Then, we randomly select k unique headers and add them into H to get the corrupted input H_2 .

Noise 3: Similar Table Cells For this type of noise, we select cells that are in the same row/column as the highlight cells. The intuition, as seen in the user study, is that these cells will have similar semantics to those cells underlying the exploratory intention and users tend to select them. For H_3 , we first retrieve table cells that are in the same row/column as highlight cells. Then, we randomly select k unique cells thereof and add them into H .

Noise 4: Remove Cells from H Users also miss some of the highlight cells in H . For this type of noise, we first retrieve those cells in H that are irrelevant (i.e., their content is not expressed in) for generating S . After getting the irrelevant cells in H , we randomly choose k thereof and remove them from H to create H_4 .

2.3 Augmenting the Training Dataset

We propose to fine-tune models on the training set augmented with noisy data. We extend the original ToTTo training set $\mathcal{D} = \{(T, S, H)\}_{j=1}^{|\mathcal{D}|}$ with data instances with user noisy cell selections. Specifically, we replace data instances with clean cell selections H in \mathcal{D} with corrupted data instances with noisy cell selections H_i . This results in a training set \mathcal{D}_i consisting of noisy cell selections of noise type i . The final training set \mathcal{D}_{final} contains both clean and corrupted data instances, its size is 603,805 (5 times the size of the original training set), and it is defined as $\mathcal{D}_{final} = \mathcal{D} \cup \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3 \cup \mathcal{D}_4$. We set $k = 1$ for creating data instances of type Noise 1, Noise 2, and Noise 3. This is because the average number of highlight cells in ToTTo dataset is small (3.55). To create instances of type Noise 4, we remove all irrelevant cells found in H .

2.4 Robustness via Sequence Level Training

Inspired by PlanGen (Su et al., 2021), to further enhance the robustness of table-to-text models on clean and noisy cell selections, we further fine-tune model parameters with Reinforcement Learning (RL) (Williams, 1992). Formally, given an input data pair $\{T, S, H\} \in \mathcal{D}_{final}$ and a sampled output sequence $S' = (S'_1, \dots, S'_{|S'|})$, the RL training objective is formulated as:

$$\mathcal{L}_{RL} = -R(S, S') \sum_{i=1}^{|S'|} \log P \left(S'_i \mid S'_{<i}, E(T, H) \right) \quad (1)$$

where $E(\cdot)$ denotes the encoder module of a table-to-text generator. The reward function $R(S, S')$ measures the similarity between the reference text and the text generated by the model; it is formulated as $R(S, S') = B(S, S')$ where $B(\cdot, \cdot)$ is the BLEU score (Papineni et al., 2002). By doing this, we make the outputs of both clean and noisy cell selections to be more similar to the reference texts. This implicitly improves the similarity between outputs of clean and noisy cell selections.

Model	Clean	Noise	Noise	#Param
		Avg.	Var.	
BART-BASE (clean)	47.8	44.0	9.09	141M
BART-LARGE (clean)	48.6	43.9	14.43	408M
BART-BASE (\mathcal{D}_{final})	48.5	48.03	0.16	141M
BART-BASE + RL (\mathcal{D}_{final})	49.2	48.85	0.14	141M
BART-LARGE (\mathcal{D}_{final})	49.1	48.16	0.69	408M
BART-LARGE + RL (\mathcal{D}_{final})	49.6	48.75	0.60	408M

Table 1: BLEU scores on clean and noisy development sets. Average BLEU score across the four noisy development sets (Noise Avg.). Variance of BLEU scores across the four noisy development sets (Noise Var.). Model parameters (#Param). The attribute in parenthesis indicates the dataset used for model fine-tuning.

2.5 Table-to-Text Generation Models

Our models are based on BART (Lewis et al., 2020). We fine-tune them for user controlled table-to-text generation as follows. Given a training data pair $\{T, S, H\}$, the fine-tuning process proceeds in two stages. The first stage fine-tunes the model with a conventional conditional language modelling training objective:

$$\mathcal{L}_{LM} = - \sum_{i=1}^{|S|} \log P \left(S_i \mid S_{1:i-1}, E(T, H) \right) \quad (2)$$

where E denotes the encoder of the table-to-text generator. The second stage further adjusts model parameters by using $\mathcal{L}_{mix} = \mathcal{L}_{LM} + \mathcal{L}_{RL}$.

3 Experimental Results

Implementation details for our table-to-text generation models can be found in Appendix B. We use the same hyperparameters as the baseline in the ToTTo (Parikh et al., 2020).

As shown in Table 1 (detailed results per Noise type are given in Appendix A), when using the training scheme with clean cell highlights, the average BLEU score of **BART-BASE (clean)** drops from 47.8 to 44 when tested on noisy cell selections. Similar trend can be seen for **BART-LARGE (clean)** with a BLUE score drop from 48.6 to 43.9. In addition, the ‘‘Noise Variance’’ of **BART-BASE (clean)** and **BART-LARGE (clean)** is large, indicating that these models are not stable (or robust) to different types of noisy cell selections. All this suggests that a training scheme with carefully selected cells alone results in systems that perform poorly in practical scenarios with user interactions.

In contrast, we observe that our proposed learning scheme makes generators achieve better performance both on clean and noisy cell selections. On

Kosuke Matsuura					
Section Title: IndyCar Series					
Year	Team	14	16	Rank	Points
2004	Super Aguri Fernandez Racing	CHI Ret	TX2 Ret	14th	280
2005	Super Aguri Fernandez Racing	SNM 6	WGL 6	14th	320

Reference: In 2005, Kosuke Matsuura again drove for Super Aguri Fernandez Racing, and again finished 14th with a best place finish of 6th in the two races.

Ours: In 2005, Kosuke Matsuura drove for Super Aguri Fernandez Racing in the IndyCar Series and finished 14th in points.

Baseline: In 2005, Kosuke Matsuura drove for Super Aguri Fernandez Racing and finished 14th in the WGL 6 and 280 points.

Asian Beach Games				
Section Title: List of Asian Beach Game				
Edition	Year	City	Start Date	End Date
IV	2014	Phuket	14 November	23 November
V	2016	Da Nang	24 September	3 October
VI	2020	Sanya	24 November	5 December

Reference: The last Asian Beach Games was held in Danang, Vietnam from 24 September to 3 October 2016, while the next will be held in 2020 in Sanya, China, the first to breakaway from the 2-year cycle.

Ours: The Asian Beach Games are scheduled to be held in Da Nang, Vietnam from September 24 to October 3, 2016 and in Sanya, China in 2020.

Baseline: The Asian Beach Games were held from 2014 to 2016 in Da Nang, Vietnam and from 3 October to 3 October 2020 in Sanya, China.

List of rulers of Brittany		
Section Title: House of Montfort		
Name	Birth	Death
Peter II the Simple (Pêr II) 1450–1457	7 July 1418	22 September 1457 Nantes aged 41
Arthur III the Justicier (Arzhur III) 1457–1458	24 August 1393	26 December 1458 Nantes aged 65

Reference: At the very end of his life, Arthur III became duke of Brittany, succeeding Peter II.

Ours: Arthur III the Justicier was Duke of Brittany from 1457 until his death in 1458, succeeding Peter II the Simple.

Baseline: Arthur III (26 December 1458) was Duke of Brittany from 1450 to his death.

Iain Glen				
Section Title: Awards and nominations				
Year	Title	Award	Category	Result
1990	Silent Scream	Silver Bear	Best Actor	Won

Reference: In 1990, Glen won the Silver Bear for the Best Actor in the Silent Scream.

Ours: In 1990, Iain Glen won the Silver Bear for Best Actor for Silent Scream.

Baseline: In 1990, Iain Glen received the Silver Bear for Best Actor for Silent Scream.

Figure 2: Model outputs for synthetic noisy cell selections of type Noise 1 (left top) and Noise 2 (left bottom), and for user noisy cell selections from the human study of type Noise 3 (right top) and Noise 4 (right bottom).

	Model	FL	FA	CC
clean	BART-LARGE (clean)	0.83	0.83	0.89
	BART-LARGE + RL (\mathcal{D}_{final})	0.88	0.89	0.93
Noisy	BART-LARGE (clean)	0.80	0.81	0.87
	BART-LARGE + RL (\mathcal{D}_{final})	0.89	0.91	0.91

Table 2: Results of Human Evaluation. Percentage of outputs perceived as Fluent (FL), Faithful (FA), and better Covering selected Cells (CC).

Method	Overall		
	BLEU	PARENT	BLEURT
NCP	19.2	29.2	-0.576
Pointer Generator	41.6	51.6	0.076
Bert-to-Bert	44.0	52.6	0.121
LATTICE	48.4	58.1	0.222
T5-3B	49.5	58.4	0.230
PlanGen	49.2	58.7	0.249
Ours	49.3	58.8	0.235

Table 3: ToTTo test set results. All reported results can be found in the ToTTo leaderboard.

clean cell selections (ToTTo original development set), the model trained using the proposed learning scheme **BART-BASE** (\mathcal{D}_{final}) outperforms the model using the same pre-trained model but fine-tuned with the standard learning scheme **BART-BASE (clean)** by 0.7 BLEU scores. On noisy cell selections, **BART-BASE** (\mathcal{D}_{final}) outperforms **BART-BASE (clean)** by 4.03 BLEU points on average. In addition, **BART-BASE** (\mathcal{D}_{final}) has a

small “Noise Variance” score across four noisy and one clean development sets, suggesting that the proposed learning scheme can make controlled table-to-text generators more robust and less sensitive to various types of noisy cell selections. Fine-tuning with RL, **BART-BASE + RL** (\mathcal{D}_{final}), can further boost models’ performance.

In Appendix A we provide additional experiments on ablation results on the contribution of each Noise dataset, training with a subset of \mathcal{D}_{final} (i.e., training with one fifth of the data also improves robustness), and evaluating on cases with different amount of noise (i.e., our approach generalises better to cases with higher values of k).

To gain insights on how the improvements are perceived in generated descriptions, we conduct a human evaluation. We follow the setup described in (Parikh et al., 2020). We sample 100 development instances and have five human judges (voluntary MSc level students fluent in English) to annotate them across three criteria. **Fluency** (users select amongst *Fluent*, *Mostly Fluent*, and *Not Fluent*; we report the percentage of outputs annotated as *Fluent*); **Faithfulness** (a candidate sentence is considered to be faithful if all the information in it is supported by the highlight cells and metadata of the table; we

report the percentage of outputs that users annotate as faithful); and **Covered Cells** (the percentage of highlighted cells that the candidate sentence covers; we report average percentage of covered cells across all sampled instances). Table 2 shows that judges find outputs by the model variants fine-tuned with the proposed regime more faithful, fluent and with better cell coverage.

We choose the best performing model, **BART-LARGE + RL** (D_{final}), fine-tuned with the proposed approach and compare it with state-of-the-art models on the ToTTo test set. These are NCP (Puduppully et al., 2019), Pointer-Generator (See et al., 2017), Bert-to-Bert (Parikh et al., 2020), and T5-3B (Raffel et al., 2020), LATTICE (Wang et al., 2022), and PlanGen (Su et al., 2021). Table 3 shows overall results (detailed overlap/non-overlap results are provided in Appendix A). Our model performs in par with T5-3B and PlanGen despite the fact that the first one has more parameters and the second one possesses a dedicated planning step.

Figure 2 shows two instances of synthetic noisy cell selections of type Noise 1 (i.e., accidentally selected random cell not related to the exploration intention) and type Noise 2 (i.e., random criteria for header selection); and two instances of user noisy cell selection from the human study of type Noise 3 (i.e., highlight 2014 semantically close to cells in the exploratory intention) and Noise 4 (i.e., *won* is not highlighted). Cells in yellow indicate original highlights from the ToTTo dataset and those in orange are noisy selections. In both cases, the outputs produced by the model fine-tuned with the proposed regime are not affected by noise and show better coverage, factual accuracy, and lexicalisation. This illustrates human evaluation preferences.

4 Conclusion

We study the performance of user controlled table-to-text generation. We show that standard training schemes with only carefully selected cells causes poor robustness of generators in practice when confronted with user noisy cell selections. To address this, we introduce a training scheme with simulated user noisy cell selections. Experimental results show that generators optimized with our proposed scheme can achieve better performance on both clean and noisy cell selections. In the future, it would be interesting to investigate how to apply our approach to other data-to-text datasets to improve model generalisation.

5 Acknowledgments

We thank the anonymous reviewers for their feedback. We gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (award number 681760).

Limitations

We create synthetic data simulating real users interactions (i.e., user cell selections on a table). However, the automatic noise generation method does not cover all possible user interactions and may fail to exactly reproduce them in some cases. For example, our process for creating Noise 3 randomly highlights cells in the same row/column as a reference highlighted cell. However, the probability distribution of a user highlighting a cell around a reference highlighted cell is not always uniform, but in some cases based on some reasoning process about the concerned cells. In the future, it would be interesting to investigate how to simulate this reasoning process to predict where the user is likely to highlight cells. Nevertheless, the set of noise types that we propose in this work shows that models trained only on cleaned data are brittle.

References

- Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.
- Simon Mille, Kaustubh Dhole, Saad Mahamood, Laura Perez-Beltrachini, Varun Gangal, Mihir Kale, Emiel van Miltenburg, and Sebastian Gehrmann. 2021. Automatic Construction of Evaluation Suites for Natural Language Generation Datasets. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. (NeurIPS 2021).

- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqi, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186.
- Laura Perez-Beltrachini and Mirella Lapata. 2018. Bootstrapping generators from noisy data. In *North American Chapter of the Association for Computational Linguistics*, New Orleans, Louisiana. Association for Computational Linguistics. (NAACL 2018).
- Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6908–6915.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.
- Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 895–909.
- Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. *arXiv preprint arXiv:2205.03972*.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.
- Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, Brussels, Belgium. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing.

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

A Detailed and Ablation Results

Table 4 provides detailed results for the different model variants (**clean**) and (\mathcal{D}_{final}) evaluated on different development sets with different types of noise (cf., Table 1 in Section 3). Table 5 provides detailed results comparing our model **BART-LARGE + RL** (\mathcal{D}_{final}) with other state-of-the-art methods in ToTTo’s leaderboard (cf., Table 3 in Section 3).

We conduct an ablation study to investigate the impact of each type of noise in \mathcal{D}_{final} (see Section 2.2). Specifically, we remove one of the four noise types at a time from \mathcal{D}_{final} , then train the **BART-BASE** model using the remaining data. This study shows that all types of user noisy cell selections help to improve performance and robustness (Table 6).

We construct corrupted ToTTo development datasets with *different amount of noise* (i.e., different number k of noisy cells) added to each original input highlighted cells. In the ToTTo dataset, there are on average 3.5 highlighted cells for each table; when $k = 3$, the injected noise has roughly the same proportion as the original highlight cells. We then examine BLEU scores for **BART-BASE** trained with our approach and the baseline on these noisy development sets. As shown in Table 7, performance drops significantly as more noise is injected, from 47.8 when $k = 0$ (clean) to 34.8 when $k = 3$, for the model trained only on clean cell selections, **BART-BASE** (clean). It also indicates that the models trained with our proposed method, **BART-BASE** (\mathcal{D}_{final}) and **BART-BASE + RL** (\mathcal{D}_{final}), can reduce this performance drop.

We also combine all noise types with clean data for training in a way that the resulting dataset has the same size as the original clean dataset. Specifically, we randomly divide the original dataset into five equal parts and replace four of them each by a different type of noisy data subset; one of the parts is not replaced (i.e., one part of the original clean set is kept). We merge these five parts together and call this the mixed dataset \mathcal{D}_{mix} . Results in Table 8 indicate that training the model on a substantially smaller subset of clean and noisy data (i.e., a subset of \mathcal{D}_{final}) still yields comparable performance on clean data and significant better performance on noisy data.

Model	Clean	Noise1	Noise2	Noise3	Noise4	Noise	Noise	#Param
	Dev set	Average	Variance					
BART-BASE (clean)	47.8	40.6	45.6	42.5	47.3	44	9.087	141M
BART-LARGE (clean)	48.6	39.8	46.1	41.7	48	43.9	14.433	408M
BART-BASE (\mathcal{D}_{final})	48.5	47.7	48.6	47.9	47.9	48.025	0.156	141M
BART-BASE + RL (\mathcal{D}_{final})	49.2	48.6	49.4	48.8	48.6	48.850	0.143	141M
BART-LARGE (\mathcal{D}_{final})	49.1	46.9	48.6	47.6	48.6	48.16	0.689	408M
BART-LARGE + RL (\mathcal{D}_{final})	49.6	47.9	49.7	48.4	49.0	48.75	0.603	408M

Table 4: BLEU scores of models on clean and noisy ToTTo development set. Average BLEU score across the four noisy development sets (Noise Avg.). Variance of BLEU scores across the four noisy development sets (Noise Var.). #Param denotes the total number of parameters in the model. The attribute in parenthesis indicates the training data we use for training the model. For (clean), models are trained on clean ToTTo training set (i.e. using \mathcal{D}). For (\mathcal{D}_{final}), the noise-augmented training set described in section 2.3 is applied. For '+RL', the Reinforcement Learning algorithm described in section 2.4 is applied.

Method	Overall			Overlap			non-Overlap		
	BLEU	PARENT	BLEURT	BLEU	PARENT	BLEURT	BLEU	PARENT	BLEURT
NCP	19.2	29.2	-0.576	24.5	32.5	-0.491	13.9	25.8	-0.662
Pointer Generator	41.6	51.6	0.076	50.6	58.0	0.244	32.2	45.2	-0.092
Bert-to-Bert	44.0	52.6	0.121	52.7	58.4	0.259	35.1	46.8	-0.017
T5-3B	49.5	58.4	0.230	57.5	62.6	0.351	41.4	54.2	0.108
PlanGen	49.2	58.7	0.249	56.9	62.8	0.371	41.5	54.6	0.126
Ours	49.3	58.8	0.235	57.1	63.4	0.358	41.5	54.1	0.112

Table 5: ToTTo test set results. All reported results can be found in the ToTTo leaderboard.

Training Data	Clean Dev	Noise1 Dev	Noise2 Dev	Noise3 Dev	Noise4 Dev	Noise Avg	Noise Var
\mathcal{D}_{final}	48.5	47.7	48.6	47.9	47.9	48.025	0.156
$\mathcal{D}_{final} - \mathcal{D}_1$	48.5	47.3	48.4	47.6	47.8	47.775	0.216
$\mathcal{D}_{final} - \mathcal{D}_2$	48.6	47.6	48.5	47.9	48.1	48.025	0.143
$\mathcal{D}_{final} - \mathcal{D}_3$	48.5	47.6	48.6	47.8	47.9	47.975	0.189
$\mathcal{D}_{final} - \mathcal{D}_4$	48.3	47.7	48.6	47.9	47.4	47.900	0.260

Table 6: BLEU scores for **BART-BASE** trained on different training data and evaluated on different development sets. Noise Avg denotes the average BLEU scores on all noisy development sets. Noise Var denotes the variance of BLEU scores on noisy development sets.

Model	clean	$k = 1$	$k = 2$	$k = 3$
BART-BASE (clean)	47.8	42.7	38.1	34.8
BART-BASE (\mathcal{D}_{final})	48.5	48.1	45.9	42.3
BART-BASE + RL (\mathcal{D}_{final})	49.2	48.8	46.4	42.9

Table 7: BLEU scores on input cell highlights with different amounts of noise (development set). k denotes the amount of noise added to the original data point (higher k means more noisy cell highlights are added).

Dev/Train	Clean	\mathcal{D}_{mix}	\mathcal{D}_{final}
Clean	47.8	47.3	48.50
Noise Avg.	44.0	46.8	48.03

Table 8: BLEU scores of **BART-BASE** trained on the original dataset, the noise augmented dataset (\mathcal{D}_{final}), and a smaller dataset (\mathcal{D}_{mix}). Evaluation is on clean and Noise development sets.

B Implementation Details

The examined models are based on the Hugging-face Library (Wolf et al., 2020) with default model hyperparameters provided by the Library. We fine-tune BART (Lewis et al., 2020) using the proposed learning scheme. We use the Adam (Kingma and Ba, 2014) optimizer, with a learning rate of $2e^{-5}$ and a batch of size 32. We fine-tune with the $\mathcal{L}_{\mathcal{LM}}$ objective for 100k steps and \mathcal{L}_{mix} for 50k steps.

C Human Study Interface

Figure 3 shows the Amazon Mechanical Turk interface, instructions and annotation form, we use for the human study described in Section 2.1.

Instructions

Given a table and a sentence describing part of its content, you should highlight those table cells that you think the sentence is describing. You can "highlight" a cell by entering its header and content between square brackets: [cell0_header, cell0_content]. You can list all cells that you consider as been described in the sentence by entering one cell highlight after the other: [cell0_header, cell0_content], [cell1_header, cell1_content],

It is worth noting that you are allowed to highlight the content of both headers and content cells of the table. You should not highlight meta information (details that appear above the table); usually, these are included in the sentence but you need not select them.

It is also worth noting that there are some cases where the sentence contains some sort of aggregation or summarisation of information. For instance, in the following table, eleven seasons corresponds to highlighting the eleven year values in the column YEAR in the example table below.

Table Title: Robert Craig (American football)
Section Title: National Football League statistics
Table Description: None

YEAR	TEAM	RUSHING					RECEIVING				
		ATT	YDS	AVG	LNG	TD	NO.	YDS	AVG	LNG	TD
1983	SF	176	725	4.1	71	8	48	427	8.9	23	4
1984	SF	155	649	4.2	28	4	71	655	9.5	64	3
1985	SF	214	1050	4.9	62	9	92	1016	11	73	6
1986	SF	204	830	4.1	25	7	81	624	7.7	48	0
1987	SF	215	815	3.8	25	3	66	492	7.5	35	1
1988	SF	310	1302	4.8	46	9	76	534	7.0	22	1
1989	SF	271	1054	3.9	27	6	49	473	9.7	44	1
1990	SF	141	439	3.1	26	1	25	201	8.0	31	0
1991	RAI	162	590	3.6	15	1	17	136	8.0	20	0
1992	MIN	105	416	4.0	21	4	22	164	7.5	22	0
1993	MIN	38	119	3.1	11	1	19	169	8.9	31	1
Totals	-	1991	8189	4.1	71	56	566	4911	8.7	73	17

Target Text: Craig finished his eleven NFL seasons with 8,189 rushing yards and 566 receptions for 4,911 receiving yards.

You should always highlight all those cells that you consider give rise to the information expressed/summarised in the sentence.

In this example, if you want to highlight the header "YEAR", you can type [YEAR, YEAR]. If you want to highlight the cell with the content of "1983", you can type [YEAR, 1983].

Annotation

Brandon Starc

Section Title: International competitions

Table Section Text: None

Year	Competition	Venue	Position	Event	Notes
2012	World Junior Athletics Championships	Barcelona, Spain	6th	High jump	2.17
2013	World Championships	Moscow, Russia	25th	High jump	2.17
2014	Commonwealth Games	Glasgow, Scotland	8th	High jump	2.20 (Q) 2.21 (F)
2015	World Championships	Beijing, China	12th	High jump	2.31 (Q) 2.25 (F)
2016	Olympic Games	Rio de Janeiro, Brazil	15th	High jump	2.29 (Q) 2.20 (F)
2018	Commonwealth Games	Gold Coast, Australia	1st	High jump	2.21 (Q) 2.32 (F)
2018	Internationales Hochsprung	Eberstadt, Germany	1st	High jump	2.36
2018	IAAF Diamond League Final	Brussels, Belgium	1st	High jump	2.33

Sentence(s)

Starc qualified with 2.31 m at the World Championships.

typing the related cells by the format of [cell0_header, cell0_content], [cell1_header, cell1_content],

Submit

Figure 3: The Amazon Mechanical Turk interface, instructions and annotation form, we use for the human study described in Section 2.1.

Better Pre-Training by Reducing Representation Confusion

Haojie Zhang^{1,2,*}, Mingfei Liang^{1,*}, Ruobing Xie^{1,*}, Zhenlong Sun¹, Bo Zhang¹, Leyu Lin¹

¹WeChat Search Application Department, Tencent, China

²Peking University, China

¹{coldhjzhang, aesopliang, ruobingxie, richardsun, nevinzhang, goshawklin}@tencent.com

²zhanghaojie@stu.pku.edu.cn

Abstract

In this work, we revisit the Transformer-based pre-trained language models and identify two different types of information confusion in position encoding and model representations, respectively. Firstly, we show that in the relative position encoding, the joint modeling about relative distances and directions brings confusion between two heterogeneous information. It may make the model unable to capture the associative semantics of the same distance and the opposite directions, which in turn affects the performance of downstream tasks. Secondly, we notice the BERT with Mask Language Modeling (MLM) pre-training objective outputs similar token representations (last hidden states of different tokens) and head representations (attention weights¹ of different heads), which may make the diversity of information expressed by different tokens and heads limited. Motivated by the above investigation, we propose two novel techniques to improve pre-trained language models: Decoupled Directional Relative Position (DDRP) encoding and MTH² pre-training objective. DDRP decouples the relative distance features and the directional features in classical relative position encoding. MTH applies two novel auxiliary regularizers besides MLM to enlarge the dissimilarities between (a) last hidden states of different tokens, and (b) attention weights of different heads. These designs allow the model to capture different categories of information more clearly, as a way to alleviate information confusion in representation learning for better optimization. Extensive experiments and ablation studies on GLUE benchmark demonstrate the effectiveness of our proposed methods.

*Equal contribution.

¹"attention weights" mainly refer to the dot product between Key and Query in the self-attention module.

²MTH is the abbreviation of our proposed MLM with Token Cosine Differentiation (TCD) and Head Cosine Differentiation (HCD) pre-training task. TCD and HCD are described in detail in sec. 1(2) and sec.3.2.

1 Introduction

The paradigm of pre-training on large-scale corpus and fine-tuning on specific task datasets has swept the entire field of Natural Language Processing (NLP). BERT (Devlin et al., 2018) is the most prominent pre-trained language model, which stacks the encoder blocks of Transformer (Vaswani et al., 2017) and adopts MLM and Next Sentence Prediction (NSP) pre-training tasks, achieving the SOTA results in 2018. After that, a large number of Pre-trained Language Models (PLMs) (Liu et al., 2019; Lan et al., 2020; Raffel et al., 2019; Clark et al., 2020; He et al., 2021) that optimize the Transformer structure and pre-training objectives have emerged, which further improves the performance of the pre-trained language models on multiple downstream tasks. In this work, we identify two different types of information confusion in language pre-training, and explore two conceptually simple and empirically powerful techniques against them as follows:

(1) **Decoupled Directional Relative Position (DDRP) Encoding.** It is well known that relative position encoding is competitive and has been widely used in real PLMs (Shaw et al., 2018; Yang et al., 2019; Wei et al., 2019; Raffel et al., 2019; Su et al., 2021; He et al., 2021; Ke et al., 2021). Despite its great performance, we still notice relative position encoding methods utilizes completely separate parametric vectors to encode different relative position information, which indicates that every single parametric vector needs to learn both distance and directional features. We consider this paradigm of utilizing a single parametric vector to represent both relative distance and direction as a kind of information confusion, and question its rationality. Since relative distance features and the directional features are apparently heterogeneous information that reflects different aspects of positional information, we argue that existing methods may impose

difficult in establishing connections explicitly between parametric vectors of the same distances and the opposite directions, which in turn result in serious information losses in position encoding. Inspired by this, we propose a novel Decoupled Directional Relative Position (DDRP) encoding. In detail, DDRP decomposes the classical relative position embedding (Shaw et al., 2018) into two embeddings, one storing the relative distance features and the other storing the directional features, and then multiply the two together explicitly to derive the final decoupled relative position embedding, allowing originally confused distance and directional information to be as distinguishable as possible.

(2) **Model Representation Differentiations.** We analyze that there is non-negligible confusion in the representation of pre-trained BERT, as evidenced by the high consistency in last hidden states across different tokens and attention weights across different heads, respectively. Similar last hidden states will introduce the *anisotropic problem* (Mimno and Thompson, 2017), which will bound the token vectors to a narrow representation space and thus make it more difficult for the model to capture deep semantics. Considering attention weights contain rich linguistic knowledge (Clark et al., 2019; Jawahar et al., 2019), we argue that high consistency in attention weights also constrains the ability of the model to capture multi-aspect information. To alleviate the representation confusion between different tokens and heads caused by high information overlap, we propose two novel pre-training approaches to stimulate the potential of the pre-trained model to learn rich linguistic knowledge: Token Cosine Differentiation (TCD) objective and Head Cosine Differentiation (HCD) objective. Specifically, TCD attempts to broaden the dissimilarity between tokens by minimizing the cosine similarities between different last hidden states. In contrast, HCD attempts to broaden the dissimilarity between heads by minimizing the cosine similarities between different attention weights. We apply TCD and HCD as two auxiliary regularizers in MLM pre-training, which in turn guides the model to produce more discriminative token representations and head representations. Formally, we define our enhanced pre-training task as MLM with TCD and HCD (MTH).

Extensive experiments on the GLUE benchmark show that DDRP achieves better results than classi-

cal relative position encoding (Shaw et al., 2018) on almost all tasks without introducing the additional computational overhead and consistently outperforms prior competitive relative position encoding models (He et al., 2021; Ke et al., 2021). Moreover, our proposed MTH outperforms MLM by a 0.96 average GLUE score and achieves nearly 2x pre-training speedup on BERT_{BASE}. Both DDRP and MTH are straightforward, effective, and easy to deploy, which can be easily combined with existing pre-training objectives and various model structures. Our contributions are summarized as follows:

- We propose a novel relative position encoding named DDRP, which decouples the relative distance and directional features, giving the model a stronger prior knowledge, fewer parameters, and better results compared to conventional coupled position encodings.
- We analyze the trend of self-similarity of last hidden states and attention weights during pre-training, and propose two novel Token Cosine Differentiation and Head Cosine Differentiation objectives, motivating pre-trained Transformer to better capture semantics in PLMs.
- We experimentally verified by our proposed techniques (DDRP and MTH) that decomposing heterogeneous information and extending representation diversity can significantly improve pre-trained language models. We also analyze the characteristics of DDRP and MTH in detail.

2 Related Work

In recent years, pre-trained language models have made significant breakthroughs in the field of NLP. BERT (Devlin et al., 2018), which proposes MLM and NSP pre-training objectives, is pre-trained on large-scale unlabeled corpus and has learned bidirectional representations efficiently. After that, many different pre-trained models are produced, which further improve the effectiveness of the pre-trained models. RoBERTa (Liu et al., 2019) proposes to remove the NSP task and verifies through experiments that more training steps and larger batches can effectively improve the performance of the downstream tasks. ALBERT (Lan et al., 2020) proposes a Cross-Layer Parameter Sharing technique to lower memory consumption. XL-Net

(Yang et al., 2019) proposes Permutation Language Modeling to capture the dependencies among predicted tokens. ELECTRA (Clark et al., 2020) adopts Replaced Token Detection (RTD) objective, which considers the loss of all tokens instead of a subset. TUPE (Ke et al., 2021) performs Query-Key dot product with different parameter projections for contextual information and positional information separately and then added them up, they also add relative position biases like T5 (Raffel et al., 2019) on different heads to form the final correlation matrix. DEBERTA (He et al., 2021) separately encodes the context and position information of each token and uses the textual and positional disentangled matrices of the words to calculate the correlation matrix.

3 Method

In this section, we analyze in turn two different types of information confusion that exist in the real PLMs: (i) The paradigm of utilizing a single parametric vector of relative position embedding to represent both relative distance and direction. (ii) The high similarity and overlap in model representations. Based on above two investigations, we propose two techniques, **Decoupled Directional Relative Position (DDRP)** Encoding and **MLM with TCD and HCD (MTH)**, respectively, to help the PLMs alleviate information confusion and enhance representation clarity and diversity.

3.1 Decoupled Directional Relative Position (DDRP) Encoding

We first start to introduce DDRP by formulating multi-head attention module of BERT and BERT-R (Shaw et al., 2018). Specifically, BERT formulates multi-head attention for a specific head as follows:

$$Q = HW^Q, K = HW^K, V = HW^V, \quad (1)$$

$$A = \frac{QK^T}{\sqrt{d}}, \quad (2)$$

$$Z = \text{softmax}(A)V, \quad (3)$$

where $H \in R^{S \times D}$ represents the input hidden states; $W^Q, W^K, W^V \in R^{D \times d}$ represent the projection matrix of Query, Key, and Value respectively; $A \in R^{S \times S}$ represents attention weight; $Z \in R^{S \times d}$ represents the single-head output hidden states of self-attention module; S represents input sequence length; D represents the dimension of input hidden states; d represents the dimension of single-head hidden states. Unlike BERT, which

adds the absolute position embedding to the word embedding as the final input of the model, BERT-R first applies relative position encoding. It adds relative position embedding into K in the self-attention module of each layer to make a more interactive influence. Its formulations are as follows:

$$A_{i,j} = \frac{Q_i (K_j + K_{\sigma(i,j)}^r)^T}{\sqrt{d}}, \quad (4)$$

$$\sigma(i,j) = \text{clip}(i-j) + r_s, \quad (5)$$

where Q_i represents Query vector at the i -th position; K_j represents Key vector at the j -th position; r_s represents maximum relative position distance; $\sigma(i,j)$ represents the index of relative position embedding $K^r \in R^{2r_s \times d}$; relative position embedding for K are shared at all different heads. Note that Shaw et al. (2018) has experimentally demonstrated that adding relative position embedding to the interaction between A and V gives no further improvement in effectiveness, so the relative position embedding in V space is eliminated in all our experiments to reduce the computational overhead.

Compared with BERT, BERT-R models the correlation between words and positions more explicitly, and thus further expands the expression diversity between words. However, we notice that in BERT-R, the vectors from the same distance on both left and right sides are encoded in isolation (as shown in Figure 1(a)), which indicates that every single parametric vector from K^r is forced to maintain distance and direction, two different types of information. Since it is confirmed that directional information is crucial in language modeling (Vu et al., 2016; Fuller, 2002; Shen et al., 2018), we argue that such an approach causes unnecessary information confusion and faces several constraints: (i) Mixing relative distance and directional information for modeling makes information originally in different spaces entangled, which in turn makes the learning of parametric vectors more difficult. (ii) Dot products between word vectors and directionally confused positional vectors bring unnecessary randomness in deep bidirectional representation models.

To alleviate the confusion of distance and direction that exists in BERT-R and allow the model to perceive distances and directions more clearly, we propose a novel Decoupled Directional Relative Position (DDRP) encoding. Specifically, DDRP decouples the relative distance and directional information and maintains them with two different

embeddings. Its formula is as follows:

$$A_{i,j} = \frac{Q_i (K_j + K_{\delta(i,j)}^d)^T}{\sqrt{d}}, \quad (9)$$

$$K_{\delta(i,j)}^d = D_{\rho(i,j)} \odot K_{\delta(i,j)}^{rd}, \quad (10)$$

$$\rho(i,j) = \begin{cases} 1, & \text{if } i-j < 0 \\ 0, & \text{if } i-j = 0 \\ 2, & \text{if } i-j > 0 \end{cases}, \quad (11)$$

$$\delta(i,j) = \text{abs}(\text{clip}(i-j)), \quad (12)$$

where $\rho(i,j)$ represents the index of directional embedding $D \in R^{3 \times d}$; $\delta(i,j)$ represents the index of relative distance embedding $K^{rd} \in R^{r_s \times d}$; K^d represents the relative position matrix. Note that in terms of implementation details, the only difference between DDRP and BERT-R is that DDRP decouples K^r in BERT-R into the element-wise multiplication of D and K^{rd} . We also provide a specific comparison example in Figure 1.

Compared to previous relative position encodings, we summarize the advantage of DDRP as follows: (i) DDRP explicitly extracts the commonalities (relative distances) and differences (directions) in the positional information, leading the model to produce attention that better match the real semantic distributions, which reduces the difficulty of model learning and unlocks the potential of the model. (ii) DDRP compresses the total number of parametric vectors from $2r_s$ to $r_s + 3$.

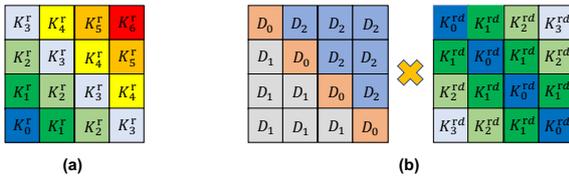


Figure 1: Fig.1(a) represents the classical relative position matrix; Fig.1(b) represents the decoupled relative position matrices we proposed. Note that the parametric vectors of the same color have the same values.

3.2 Model Representation Differentiations

Token representations. Isotropic distributions have been proved theoretically to be beneficial to token representations, which ensures that the different token vectors are directional uniform, thus maximizing the diversity of token representations. (Mimno and Thompson, 2017). In practice, Mu et al. (2017) have also empirically confirmed the effectiveness of isotropic distributions on static token representations, such as WORD2VEC (Mikolov et al., 2013) and GLOVE (Pennington et al., 2014).

Inspired by the above studies, we also wonder whether contextualized token representations (e.g., last hidden states of BERT) are isotropic. Following Mimno and Thompson (2017), we utilize the cosine similarity to evaluate the degree of isotropy in token representations. The higher similarity, the smaller isotropy; the lower similarity, the greater isotropy. For an input sequence $S = [x_1, \dots, x_n]$, we formulate the last hidden states' average self-similarity as follows:

$$f(S) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \cos(h_i, h_j), \quad (15)$$

where h_i and h_j are the last hidden states of x_i and x_j ; \cos represents cosine similarity.

Head representations. Multi-head attention, is aimed at capturing information in different heterogeneous subspaces and has been experimentally verified different heads correspond well to different linguistic notions (Clark et al., 2019). However, some studies point out that some heads contribute almost nothing to downstream tasks (Kovaleva et al., 2019; Michel et al., 2019; Voita et al., 2019; Correia et al., 2019). We are surprised by this, and speculate that the above problem may be caused by the heavy overlap of information that some heads are concerned about. To verify our point, we utilize cosine similarity to evaluate the degree of overlap in head representations, following token representations. The higher similarity, the higher overlap; the lower similarity, the lower overlap. For multiple heads $H = [H_1^1, \dots, H_m^1, \dots, H_1^L, \dots, H_m^L]$, we formulate the attention weights' average self-similarity as follows:

$$f(H) = \frac{2}{Lm(m-1)} \sum_{l=1}^L \sum_{i=1}^{m-1} \sum_{j=i+1}^m \cos(a_i^l, a_j^l), \quad (16)$$

where L represents the number of Transformer layers; a_i^l and a_j^l are the attention weights of the i -th head and j -th head of the l -th layer.

Analysis on the similarities between different tokens and heads. With curiosity about the similarity of token representations and head representations, we analyze the self-similarity trends of tokens and heads during the original MLM BERT pre-training. Specifically, we sample 5,000 sentences from the validation set and evaluate the average self-similarity of last hidden states and attention weights under multiple checkpoints during the pre-training stage as shown in Figure 2. We

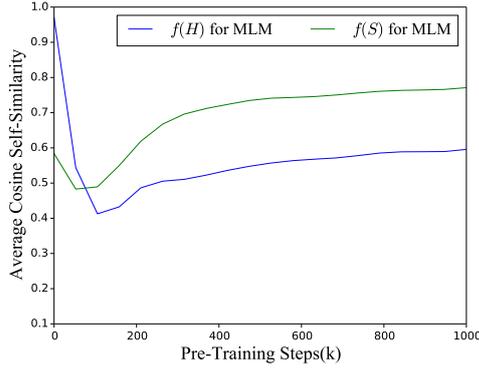


Figure 2: Average self-similarity of last hidden states and attention weights during MLM pre-training.

can notice that although $f(S)$ and $f(H)$ decrease at the beginning of pre-training, soon they start to rise gradually until the end of the training, and the similarities are always high throughout the training process. These fully demonstrates that in the MLM-based BERT, the overlap between different tokens and heads is strong and information confusion in model representations has become a problem worth to be solved.

Training objectives. To guide the model to produce more discriminative token representations and head representations, we propose a novel MTH pre-training objective, which combines original MLM with two novel Token Cosine Differentiation (TCD) objective and Head Cosine Differentiation (HCD) objective. Specifically, MTH applies the average cosine self-similarity of sampled last hidden states and attention weights as two auxiliary pre-training regularizers besides MLM³. For an input sequence $S = [x_1, \dots, x_n]$, TCD samples n' ($n' \leq n$) tokens uniformly in sequence order to obtain a subsequence $\tilde{S} = [\tilde{x}_1, \dots, \tilde{x}_{n'}]$ and calculates the average cosine self-similarity of the subsequence’s last hidden states as follows:

$$\mathcal{L}_{TCD} = \frac{2}{n'(n'-1)} \sum_{i=1}^{n'-1} \sum_{j=i+1}^{n'} \cos(\tilde{h}_i, \tilde{h}_j), \quad (17)$$

where \tilde{h}_i and \tilde{h}_j are the last hidden states of \tilde{x}_i and \tilde{x}_j . For multiple heads $H^l = [H_1^l, \dots, H_m^l]$ of a specific layer l , HCD randomly

³We have empirically verified that this below sampling strategy can greatly reduce the computational overhead with only a slight performance drop, comparing with regularizing all tokens and heads. In practice, we notice that setting $n' = 50$, $m' = 2$ is fine on both BERT and DDRP, which will compress the additional computational overhead from about 30% to 4%.

samples m' ($m' \leq m$) different heads $\tilde{H}^l = [\tilde{H}_1^l, \dots, \tilde{H}_{m'}^l]$ (Note that HCD samples by layers, so sampled heads may be different across different layers.) and then calculate the average cosine self-similarity of attention weights of sampled heads as follows:

$$\mathcal{L}_{HCD} = \frac{2}{Lm'(m'-1)} \sum_{l=1}^L \sum_{i=1}^{m'-1} \sum_{j=i+1}^{m'} \cos(\tilde{a}_i^l, \tilde{a}_j^l), \quad (18)$$

\tilde{a}_i^l and \tilde{a}_j^l are the attention weights of the i -th head and j -th head in the sampled headset of the l -th layer. Ultimately, we define the global pre-training objective MTH as follows:

$$\mathcal{L}_{MTH} = \mathcal{L}_{MLM} + \alpha_1 \mathcal{L}_{TCD} + \alpha_2 \mathcal{L}_{HCD}, \quad (19)$$

where α_1 and α_2 are hyperparameters.

4 Experiments

4.1 Pre-training Text Corpora

Follow Devlin et al. (2018), we use Wikipedia and BooksCorpus (Zhu et al., 2015), a roughly 16G uncompressed text corpus for pre-training.

4.2 Baselines

We compare DDRP with competitive pre-trained models. BERT (Devlin et al., 2018) equips Transformer (Vaswani et al., 2017) with parametric absolute position encoding. BERT-R uses the relative position encoding proposed by Shaw et al. (2018), which couples relative distance information and directional information for modeling. TUPE (Ke et al., 2021) performs Query-Key dot product with different parameter projections for contextual information and positional information separately and then adds them up, plus the relative position biases like T5 (Raffel et al., 2019). DEBERTA (He et al., 2021) uses two vectors to encode content and position and uses disentangled matrices on their contents and relative positions respectively to compute the attention weights among words.

4.3 Experimental Settings

Following the previous practice, we use a base-size model for training, which consists of 12 Transformer encoder layers, each containing 12 heads with an input dimension of 768. During pre-training, we directly use the maximum training length of 512 without taking any form of random

Models	Steps	RTE	STS-B	MRPC	CoLA	SST-2	QNLI	QQP	MNLI	Avg.
BERT (MLM)	1M	70.75	89.66	87.50	59.65	92.20	91.23	91.00	84.33	83.29
BERT-R (MLM)	1M	71.84	89.68	87.99	60.82	92.66	91.54	91.13	85.45	83.89
TUPE (MLM)	1M	68.59	89.61	86.02	<u>62.82</u>	92.66	91.26	91.04	84.88	83.36
DEBERTA (MLM)	1M	<u>73.28</u>	89.14	87.99	60.60	92.66	92.14	91.00	85.93	84.09
DDRP (MLM)	1M	72.20	<u>90.01</u>	<u>88.25</u>	<u>62.82</u>	<u>92.41</u>	92.31	91.24	<u>86.02</u>	<u>84.41</u>
DDRP (MTH)	1M	75.09	90.41	88.72	63.36	92.66	<u>92.24</u>	<u>91.22</u>	86.22	85.00

Table 1: Results on the development set of the GLUE benchmark for base-size pre-trained models. The best results are bolded, and the second results are underlined.

Approaches	Steps	RTE	STS-B	MRPC	CoLA	SST-2	QNLI	QQP	MNLI	Avg.
BERT (MLM)	500k	68.23	88.92	86.74	57.05	91.97	90.41	90.74	83.41	82.18
BERT (MTH)	500k	71.84	89.41	86.76	61.40	92.08	90.59	90.76	83.61	83.31
BERT (MLM)	1M	70.75	89.66	87.50	59.65	92.20	91.23	91.00	84.33	83.29
BERT (MTH)	1M	73.64	90.16	88.48	62.31	92.43	91.21	91.12	84.67	84.25
DDRP (MLM)	1M	72.20	90.01	88.25	62.82	92.41	92.31	91.24	86.02	84.41
DDRP (MTH)	1M	75.09	90.41	88.72	63.36	92.66	92.24	91.22	86.22	85.00

Table 2: Development scores on GLUE benchmark. BERT (MLM/MTH) represents pre-trained BERT_{BASE} with MLM/MTH pre-training objective. DDRP (MLM/MTH) represents pre-trained DDRP_{BASE} with MLM/MTH pre-training objective.

injection, and for examples less than 512 in length, we do not use the next document for padding. We remove Next Sentence Prediction (NSP) task and only keep Masked LM (MLM) as our pre-training task for all models unless noted otherwise. Considering that shorter documents may be missing semantics, we discard documents of length less than 8. We adopt the whole word masking strategy and split the whole words longer than 4 into individual subtokens. Following Devlin et al. (2018), we set the batch size to 256 sequences, the peak learning rate to $1e-4$, and the training steps to 1M. We grid search α_1 and α_2 of TCD and HCD in $\{0.01, 0.1, 1.0\}$. Eventually, we set $\alpha_1 = 1.0$ for TCD and set $\alpha_2 = 0.01$ for HCD. All the models are implemented based on the code practice of BERT⁴ in Tensorflow. We conduct all experiments on 16 Tesla-V100 GPUs (32G). All the pre-training hyperparameters are supplemented in Appendix A. To make a fair comparison, we implement BERT, BERT-R, TUPE, DEBERTA, and DDRP⁵ with the same pre-training hyperparameters and model configurations, which are consistent with vanilla BERT.

4.4 Results on GLUE Benchmark

We evaluate models on eight different English understanding tasks from General Language Under-

standing Evaluation (GLUE) benchmark (Wang et al., 2019). The datasets cover four types of tasks: natural language inference (RTE, QNLI, MNLI), paraphrase detection (MRPC, QQP), linguistic acceptability (CoLA), and sentiment classification (SST-2). For all experiments, STS-B and CoLA are reported by Pearson correlation coefficient and Matthews correlation coefficient, and other tasks are reported by Accuracy. All the fine-tuning hyperparameter configurations can be found in Appendix B. Following Ke et al. (2021), we fine-tune with five random seeds and report the median results.

4.4.1 Comparing Prior Competitive Models with DDRP

The overall comparison results are shown in Table 1. Firstly, we can notice that all the various relative position encoding models perform better than BERT, which proves that relative position encoding is a more competitive approach to encode position information. Sencodly, it is easy to find that DDRP outperforms all the strong baselines, which demonstrates modeling relative position encoding by clarifying the originally confused relative distance and directional information more clearly is more effective. Thirdly, DDRP pre-trained with MTH can consistently outperform BERT-R/DEBERTA by a 1.11/0.91 average GLUE score, which indicates that DDRP can be effectively compatible with better pre-training objectives to perform stronger.

⁴<https://github.com/google-research/bert>

⁵Following Shaw et al. (2018) and Raffel et al. (2019), we set $r_s = 64$ for all the relative position encoding models.

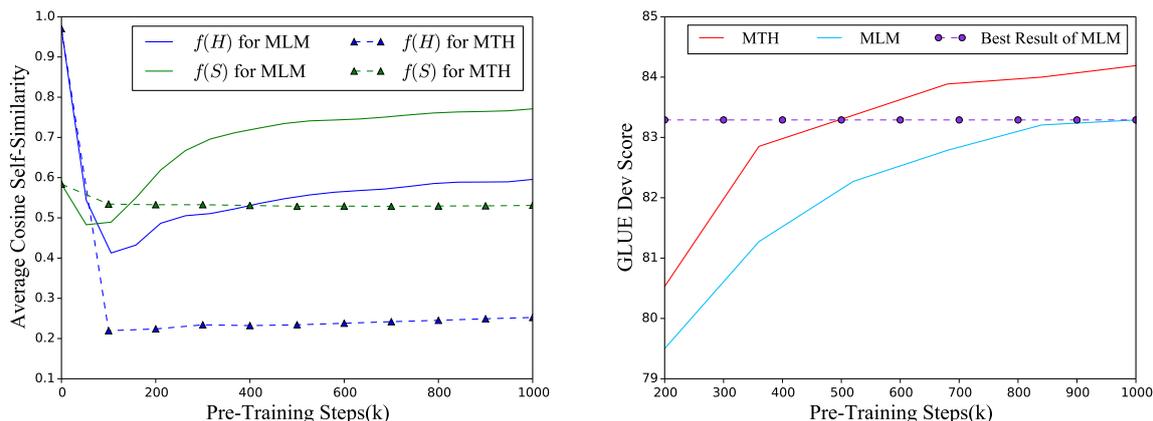


Figure 3: The left figure (a) represents the trend of average cosine self-similarity of token representations and head representations during pre-training. The right figure (b) represents the trend of GLUE average score during pre-training.

Moreover, compared to BERT-R, DDRP introduces nothing in complexity while DEBERTA increases the computational cost about 25%, we also consider DDRP is a more time-efficient alternative than the recent state-of-the-art model DEBERTA (as analyzed in Sec.5.4).

4.4.2 Comparing MTH with MLM

As illustrated in Table 2, BERT (MTH) outperforms BERT (MLM) by a 0.96 average GLUE score and is consistently better on 7 out of 8 tasks. When combining MTH with strong DDRP, it still brings an improvement by 0.59 GLUE average score. Notably, BERT (MTH) can achieve better results compared with well-trained BERT (MLM) while only using 50% training steps. Since MTH utilizes cosine similarity and sampling strategy for the penalty, only a very slight computational cost is introduced. All the above statistics can fully verify that decreasing the similarity in model representations can effectively alleviate information overlap and increase representation diversity, which in turn leads to consistent and stable improvement across different model structures.

5 Analysis and Discussion

5.1 Ablation Studies

Effect of DDRP. As shown in Table 1, BERT-R outperforms BERT by 0.6 points on average. Based on BERT-R, our proposed DDRP outperforms BERT-R by 0.52 points averagely without imposing additional computational costs. It is worth noting that compared to BERT-R, DDRP helps a great deal on low-resource tasks, such as CoLA, while further

improving the performance on high-resource tasks, such as QNLI and MNLI. These fully demonstrate utilizing separate parametric vectors to represent distances and directions, two apparently heterogeneous information, can be beneficial to the model, and further justifies the value of dissociating confusing information that is confounded in similar spaces.

Effect of TCD and HCD. MTH brings in two additional TCD and HCD regularizers besides the original MLM task. To further evaluate the relative contributions of the HCD and TCD, we develop one variation, which is BERT pre-trained with MLM and TCD. Table 3 summarizes the results on the base-size models. Firstly, it shows a 0.42 average GLUE score drop when HCD is removed from MTH, especially on MRPC, CoLA, and MNLI. Secondly, there is a 0.54 average GLUE score drop when TCD is progressively removed, especially on RTE, STS-B, and CoLA. These results indicate that both TCD and HCD regularizers play a crucial role in improving performance.

5.2 Analysis on MTH

To further understand why MTH works, we compare MLM and MTH in terms of average self-similarity of token representations and head representations and performance during pre-training in Figure 3. As shown in Figure 3.(a), it is easy to find that MTH’s average self-similarity is much lower than MLMs’. We can also clearly notice from Figure 3.(b) that the average GLUE score of MTH is always about one point higher than MLM’s during the whole pre-training process. These confirm that

Model	RTE	STS-B	MRPC	CoLA	SST-2	QNLI	QQP	MNLI	Avg.
MTH	73.64	90.16	88.48	62.31	92.43	91.21	91.12	84.67	84.25
w/o HCD	73.28	90.41	87.25	60.85	92.23	91.37	91.00	84.22	83.83
w/o TCD&HCD	70.75	89.66	87.50	59.65	92.20	91.23	91.00	84.33	83.29

Table 3: Ablation study for MTH. Note that MTH (w/o TCD&HCD) equals simply using MLM in pre-training.

(i) the differentiation of tokens and heads is important for model optimization; (ii) MTH can help to produce more discriminative token representations and head representations, extend the representation space of tokens and heads, and thus improve the performance.

5.3 Analysis on DDRP

In this subsection, we intend to analyze the attention maps of DDRP as a way to investigate why making only slight modifications on BERT-R can bring great gains. To better examine and explain the ability of DDRP to capture information in both left and right directions, we divide multiple heads into two groups evenly, where group 1 consists of the heads that focus most on the right side, and group 2 consists of the heads that focus most on the left side. As shown in Figure 4, it is easy to observe a distinct upper triangle effect in group 1 and a distinct lower triangle effect in group 2, which indicates that DDRP may allow the model to be more precise in the perception of direction, a piece of information that is crucial to understanding semantics. To further confirm our point, we sample 5,000 sentences from the validation set and count the percentage of sentences with upper and lower triangular effects according to Algorithm 1 (more details can be seen in Appendix C). It is observed that 92.11% of sentences have an up-down triangle effect. We also count the percentage for BERT-R with the same process and observe only 78.94% of the sentences have an up-down triangle effect. All the phenomena and statistics fully reveal that DDRP can make different heads focus on the token information interaction in different directions and reduce confusion between heads, thus improving the effectiveness and rationality of the model.

5.4 Complexity Analyses

DDRP. Compared with BERT, DDRP introduces additional parameters: $D \in R^{3 \times d}$ and $K^{rd} \in R^{r_s \times d}$. The total increase in parameters is $3 \times d + r_s \times d$. For base-size model ($D = 768, L = 12, S = 512, N = 12, d = 64$)⁶, the total increase

⁶N is the number of head.

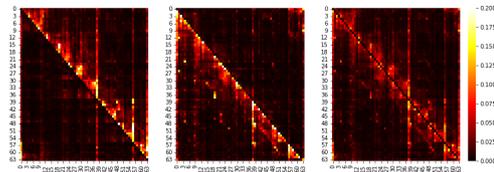


Figure 4: Attention visualization for a sampled batch of sentences. From left to right is the attention visualization for group 1, group 2, and global, respectively.

amounts to 0.0043M, which is negligible. Compared with BERT, the additional computational complexity for both BERT-R and DDRP is $O(SD)$. Since DEBERTA equips different heads with unshared K^r s, the additional computational complexity for DEBERTA is $O(NSD)$. Overall, BERT-R and DDRP increase the computational cost about 5%, and DEBERTA increases the computational cost about 30%. Although DDRP introduces a slight computational cost compared to BERT, it is more time-efficient than DEBERTA and outperforms all the above models.

MTH. Since the two regularizers of TCD and HCD are based on cosine similarity and sampling strategy, they do not introduce too much computational cost. Compared with MLM, MTH only increases a slight computational cost of about 4% while bringing excellent improvement.

6 Conclusion

In this work, we analyze and identify potential information confusion in the relative position encoding and model representations, respectively, and design two novel techniques to address these problems: DDRP (Decoupled Directional Relative Position) encoding and MTH (MLM with TCD and HCD) pre-training objectives. Specifically, DDRP decouples relative distance features and directional features to eliminate unnecessary randomness in the self-attention module. MTH utilize TCD and HCD as two regularizers to supervise the model to always maintain a certain level of critical thinking. The experimental results show that DDRP achieves better performance compared with various relative

position encoding models and MTH outperforms MLM by a large margin. We believe that reducing information confusion in representation learning may have broader application scenarios, and leave this area of exploration for future work.

7 Limitations

Our limitations lie in inducing additional computational costs. Compared with BERT, the additional computational complexity for DDRP is $O(SD)^7$, which is reflected in the 5% increase in computational cost. Compared with MLM, MTH with sampling strategy increases the computational cost by about 4%.

References

- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert’s attention. In *BlackBoxNLP*.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*.
- Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively sparse transformers. In *EMNLP-IJCNLP*.
- Graham Neubig Devendra Singh Sachan. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *WMT*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.
- Gillian Fuller. 2002. The arrow-directional semiotics: Wayfinding in transit. *Social semiotics*.
- J Pino X Li H Gong, Y Tang. 2021. Pay better attention to attention: Head selection in multilingual and multi-domain sequence modeling. In *NeurIPS*.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *ICLR*.
- Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL*.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *TACL*, 8:64–77.
- Guolin Ke, Di He, and Tie-Yan Liu. 2021. Rethinking positional encoding in language pre-training. In *ICLR*.
- Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. In *EMNLP*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *NeurIPS*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *EMNLP*.
- Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2017. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *NAACL*.
- Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *AAAI*.
- Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

⁷Here, S is the input sequence length and D is the dimension of token representations.

- Elena Voita, David Talbot, Fedor Moiseev, Rico Senrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *ACL*.
- Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze. 2016. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *ICASSP*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.
- Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*.

A Appendix A. Hyperparameters for Pre-Training

As shown in Table 4, we list the pre-training hyperparameter configurations. To make a fair comparison, all models’ pre-training hyperparameter configurations in our experiments are identical to vanilla BERT (Devlin et al., 2018).

Hyperparameter	
Vocab size	3,0522
Hidden size	768
Attention heads	12
Layers	12
Training steps	1M
Warmup ratio	0.01
Batch size	256
Learning rate	1e-4
Adam ϵ	1e-6
Adam (β_1, β_2)	(0.9, 0.999)
Learning rate schedule	linear
Weight decay	0.01
Clip norm	1.0
Dropout	0.1

Table 4: Hyperparameters used for pre-trained models.

B Appendix B. Hyperparameters for Fine-Tuning

As shown in Table 5, we enumerate the hyperparameter configurations to fine-tune the tasks on the GLUE benchmark (Wang et al., 2019). We grid search these fine-tuning hyperparameter configurations for all models. Following the BERT, we do not show results on the WNLI GLUE task for the Dev set results.

Hyperparameter	GLUE
Batch size	{16, 32}
Learning rate	{ $1e-5$, $2e-5$, $3e-5$ }
Epoch	{4, 6}
Adam ϵ	1e-6
Adam (β_1, β_2)	(0.9, 0.999)
Learning rate schedule	linear
Weight decay	0.01
Clip norm	1.0
Dropout	0.1
Warmup ratio	0.1

Table 5: Hyperparameters used for fine-tuning on the GLUE benchmark.

C Appendix C. Details for Up-Down Triangle Effect

Here we provide more details for the up-down triangle effect (in Sec. 5.3). It is rather difficult and non-intuitive to analyze the directional information in 12 different attention heads. Since previous studies have considered to group multiple heads in the self-attention module (Devendra Singh Sachan, 2018; H Gong, 2021), we thereby attempt to divide the heads into two groups evenly. Specifically, we divide the heads that focus more on the right side information into group 1 and the heads that focus more on the left side information into group 2, wishing to reveal the directional information encoded in attention weights more explicitly. We experimentally verified that this grouping approach could provide a better presentation of the directional information. Therefore, we combined this approach to conduct a comparative analysis of DDRP and BERT-R to demonstrate the more powerful directional perception of DDRP.

As illustrated in Figure 4, group 1 is more focused on the right information (greater attention values in the upper triangle region) and group 2 is more focused on the left information (greater attention values in the lower triangle region). To further analyze the universality of this phenomenon, we design the Algorithm 1 to quantitatively analyze the ability of DDRP to capture information on the both left and right sides. To make a fair comparison, we also conduct the same process for BERT-R.

Algorithm 1 Count up-down triangle percentage

Require: N : total number of sentences; n : total number of sentences that have been processed; ms : maximum sentence length; mn : total number of sentences that match the upper and lower triangle; t : the threshold value that satisfies the up-down triangular effects; amp : attention map obtained by averaging attention maps in specific group.

```
1: Initialize  $ms \leftarrow 64, n \leftarrow 1, mn \leftarrow 0, t \leftarrow 0.7$ 
2: while  $n \leq N$  do
3:   Divide heads in equal with greater attention values in
   the (upper/lower) triangle region into (group 1/group 2)
   and obtain ( $amp1/amp2$ ).
4:   // prepare for the upper and lower triangle
5:   Sum the values in upper and lower triangles of  $amp1$ 
   respectively and obtain  $amp1_{up}, amp1_{down}$ .
6:   Sum the values in upper and lower triangles of  $amp2$ 
   respectively and obtain  $amp2_{up}, amp2_{down}$ .
7:    $amp1_{up} \leftarrow amp1_{up}/ms, amp1_{down} \leftarrow$ 
    $amp1_{down}/ms$ 
8:    $amp2_{up} \leftarrow amp2_{up}/ms, amp2_{down} \leftarrow$ 
    $amp2_{down}/ms$ 
9:   // compute for the upper and lower triangle
10:  if  $amp1_{up} \geq t$  and  $amp2_{down} \geq t$  then
11:     $mn \leftarrow mn + 1$ 
12:  else
13:    Continue
14:   $n \leftarrow n + 1$ 
15: return  $float(mn/N)$ 
```

MAFiD: Moving Average Equipped Fusion-in-Decoder for Question Answering over Tabular and Textual Data

Sung-Min Lee[†], Eunhwan Park[†], Daeryong Seo[‡], Donghyeon Jeon[‡],
Inho Kang[‡], Seung-Hoon Na^{†*}

[†]Jeonbuk National University [‡]NAVER Corporation

{cap1232, judepark, nash}@jbnu.ac.kr, {daeryong.seo, donghyeon.jeon, once.ihkang}@navercorp.com

Abstract

Transformer-based models for question answering (QA) over tables and texts confront a “long” hybrid sequence over tabular and textual elements, causing long-range reasoning problems. To handle long-range reasoning, we extensively employ a fusion-in-decoder (FiD) and exponential moving average (EMA), proposing a Moving Average Equipped Fusion-in-Decoder (MAFiD). With FiD as the backbone architecture, MAFiD combines various levels of reasoning: *independent encoding* of homogeneous data and *single-row* and *multi-row heterogeneous reasoning*, using a *gated cross attention layer* to effectively aggregate the three types of representations resulting from various reasonings. Experimental results on HybridQA indicate that MAFiD achieves state-of-the-art performance by increasing exact matching (EM) and F1 by 1.1 and 1.7, respectively, on the blind test set.

1 Introduction

While most studies have focused on text question answering (QA), where unimodal textual passages are provided as a source of evidence for an answer (Joshi et al., 2017; Yang et al., 2018; Rajpurkar et al., 2018; Welbl et al., 2018; Dua et al., 2019; Karpukhin et al., 2020; Zhu et al., 2021b; Pang et al., 2022), realistic questions often need to refer to “heterogeneous” evidences based on both tabular and textual contents to generate an answer, motivating researchers to address *table-and-text* QA (Chen et al., 2020; Wenhua Chen, 2021; Talmor et al., 2021; Zhu et al., 2021a; Nakamura et al., 2022).

Among the various tasks for table-and-text QA, we address the *multi-hop* table-and-text QA described in HybridQA (Chen et al., 2020), which is a large-scale table-and-text QA dataset focusing on the multi-hop reasoning across tabular and textual contents to extract an answer.

*Corresponding author

However, a table usually contains a nontrivial number of rows and relevant passages; thus linearization of all relevant heterogeneous contents easily exceeds the maximum length limit for transformers, thereby causing *long range reasoning* problems.

To address long range reasoning, we present a novel encoder-decoder model that deploys fusion-in-decoder (FiD) (Izacard and Grave, 2021) and exponential moving average (EMA) (Ma et al., 2022), the Moving Average Equipped Fusion-in-Decoder (MAFiD). Armed with FiD as the backbone architecture, MAFiD combines various levels of reasoning:

- **Independent encoding of homogeneous data**, which independently encodes tabular and textual contents separately for each row, without being fused in the encoder step. Inherited from FiD, the resulting encoded representations are jointly fused in the decoder, which significantly reduces the computational time required for self-attention, thereby allowing us to use a longer sequence as an input for the encoder.
- **Single-row heterogeneous reasoning** (also referred to as *single-row reasoning*), which performs in-depth interaction between tabular and textual contents per row; it first concatenates the tabular and textual representations for each row and then applies the “self-attention” layer over the concatenated sequence. Thus, single-row heterogeneous reasoning is performed in a restricted manner only on heterogeneous contents within a specific row.
- **Multi-row heterogeneous reasoning** (also referred to as *multi-row reasoning*), which performs light interaction across tabular and textual contents of “multiple” rows based on the EMA layer; it concatenates the heteroge-

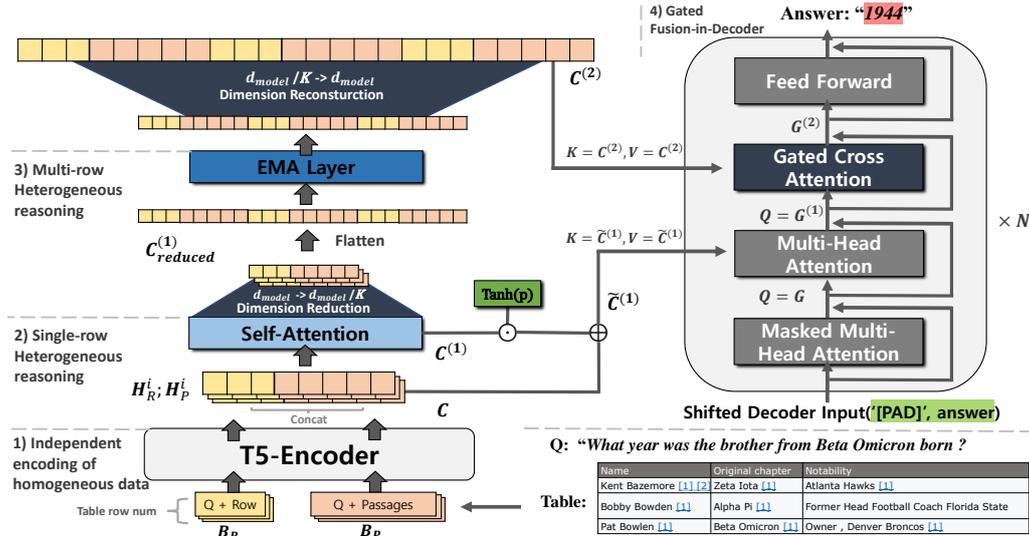


Figure 1: The overall neural architecture of the proposed MAFiD: 1) Independent encoding applies T5’s encoder on the tabular and textual blocks in i -th row, separately (i.e., $b_R^i \in \mathcal{B}_R$ and $b_P^i \in \mathcal{B}_P$) and the resulting contextual representations are concatenated to obtain the i -th row’s heterogeneous representation, C_i (Eq. (1)). 2) Single-row reasoning performs the row-specific cross-modal interaction by applying the single-head attention over C_i to generate $C_i^{(1)}$ (Eq. (2)). 3) Multi-row reasoning preforms the between-row cross-modal interaction by applying the low-dimensional EMA over the long hybrid sequence $C^{(1)}$ (Eq. (3)) to produce the $C_i^{(2)}$ (Eq. (4)). 4) The gated FiD aggregates all types of representations of C (Eq. (4)), $C^{(1)}$, and $C^{(2)}$ using the gated cross-attention layer to finally yield the decoder’s contextual representation $G^{(2)}$ (Eq. (5)) which is fed to generate an output token.

neous contents of all rows in a table to obtain a “long” hybrid sequence and then applies the EMA layer over the resulting long sequence to produce aggregated representation. To process a long sequence more efficiently, we further propose a *low-dimensional* EMA, which additionally performs a *dimensionality reduction* and *reconstruction*.

In the decoder, we further propose the use of a *gated cross-attention* layer to effectively aggregates the aforementioned three representations resulting from various reasoning, motivated by the work of (Alayrac et al., 2022).

Our contributions are summarized as follows: 1) We propose MAFiD, which augments FiD with EMA and the gated cross-attention layer, thus effectively combining various types of reasoning. 2) We propose a low-dimensional EMA to efficiently process long sequences for table-and-text QA. 3) The proposed MAFiD achieves state-of-the-art performance on the HybridQA dataset.

2 Related Works

Recently, many datasets such as HybridQA (Chen et al., 2020), OTT-QA (Wenhu Chen, 2021), MultiModalQA (Talmor et al., 2021), HybriDialogue (Nakamura et al., 2022), and TAT-QA (Zhu et al.,

2021a) have been presented for table-and-text QA. Various works on table-and-text QA have enhanced “pretraining” to strengthen the cross-modal matching and numerical reasoning, by learning on tables and texts jointly (Herzig et al., 2020; Yin et al., 2020) and exclusively on tables (Iida et al., 2021).

To handle the long range reasoning on table-and-text QA, early works employed “efficient” transformers based on *sparse attention* with selective attention masks, such as the LongFormer (Beltagy et al., 2020) in the work of (Huang et al., 2022) and ETC (Ainslie et al., 2020) in the work of (Wenhu Chen, 2021). MATE (Eisenschlos et al., 2021b) uses *structure-based* sparse attention that attends to either rows or columns in a given table. Recently, *truncation-based* approaches have been employed in MITQA (Kumar et al., 2021) where the *passage filter* module is additionally applied such that only the filtered passages are used as textual contents of a table’s row.

Compared to these existing approaches, which rather limitedly reduce the computational cost in the encoder part, MAFiD significantly lightens the encoder part by minimizing the interaction between different rows and instead fuses the encoded representations in the decoder part under the framework of FiD. Equipped with the low-dimensional EMA,

MAFiD only performs the “shallow” interaction across rows, thus mostly maintaining the efficiency of the interaction-less encoder.

3 Moving Average Equipped Fusion-in-Decoder

Figure 1 shows the overall neural architecture of the proposed MAFiD model, which combines three types of representation. Here, we present the details of the MAFiD components.

3.1 Problem Definition

Suppose that \mathcal{B}_R and \mathcal{B}_P are a set of tabular and textual blocks in a given table, where $b_R^i \in \mathcal{B}_R$ indicates the tabular block for the i -th row (i.e., a list of its cells), $b_P^i \in \mathcal{B}_P$ indicates the textual block for the i -th row (i.e., a set of its linked passages), and $L = |\mathcal{B}_R| = |\mathcal{B}_P|$ is the number of rows in a table. Given question q , the goal is to generate a correct answer by considering \mathcal{B}_R and \mathcal{B}_P as heterogeneous evidence.

3.2 Independent Encoding of Homogeneous Data: the Basic Encoder for FiD

Following the independent encoding in FiD (Izacard and Grave, 2021), independent encoding linearizes tabular and textual blocks into a sequence independently and concatenates each of them with q as follows:

$$\text{row}^i = [q; [\text{SEP}]; b_R^i], \text{psg}^i = [q; [\text{SEP}]; b_P^i]$$

where $;$ is the concatenation operator. The tabular and textual sequences are then fed into the encoder of T5 independently and concatenated as follows:

$$\begin{aligned} H_R^i &= \text{T5-enc}(\text{row}^i) \in \mathbb{R}^{|\text{row}^i| \times d_{\text{model}}} \\ H_P^i &= \text{T5-enc}(\text{psg}^i) \in \mathbb{R}^{|\text{psg}^i| \times d_{\text{model}}} \\ C_i &= [H_R^i; H_P^i] \in \mathbb{R}^{(|\text{row}^i| + |\text{psg}^i|) \times d_{\text{model}}} \end{aligned} \quad (1)$$

where $|\times|$ is the length of sequence \times and d_{model} is the dimensionality of the encoder of T5.

3.2.1 Single-row Heterogeneous Reasoning

In single-row reasoning, we perform an in-depth interaction between tabular and textual blocks for each row, b_R^i and b_P^i , using self-attention as follows:

$$C_i^{(1)} = \text{SHA}(C_i, C_i, C_i) \quad (2)$$

where $C_i^{(1)} \in \mathbb{R}^{(|\text{row}^i| + |\text{psg}^i|) \times d_{\text{model}}}$ and SHA is the *single*-head attention defined in Eq. (6) in Appendix D.

3.3 Multi-row Heterogeneous Reasoning by the Low-dimensional EMA

In multi-row reasoning, we first concatenate the contextual representations of all tabular and textual blocks as follows:

$$C^{(1)} = [C_1^{(1)}; \dots; C_L^{(1)}] \quad (3)$$

where $C^{(1)} \in \mathbb{R}^{N \times d_{\text{model}}}$, provided $N = \sum_i (|\text{row}^i| + |\text{psg}^i|)$ for notational convenience.

We then adopt the low-dimensional EMA as a variant of EMA using dimensionality reduction and reconstruction based on linear layers as follows:

$$\begin{aligned} C_{\text{reduced}}^{(1)} &= \text{Linear}(C^{(1)}) \\ C_{\text{reduced}}^{(2)} &= \text{EMA}(C_{\text{reduced}}^{(1)}) \\ C^{(2)} &= \text{Linear}(C_{\text{reduced}}^{(2)}) \end{aligned}$$

where $C_{\text{reduced}}^{(1)}, C_{\text{reduced}}^{(2)} \in \mathbb{R}^{N \times d_{\text{model}}/K}$, $C^{(2)} \in \mathbb{R}^{N \times d_{\text{model}}}$, Linear is a linear layer, and EMA is the damped EMA of (Ma et al., 2022) defined in Appendix E.

3.4 Gated Fusion-in-Decoder

In the decoder, we first concatenate the row-wise representations of independent encoding before feeding them to the FiD as follows:

$$C = [C_1; \dots; C_L] \quad (4)$$

In the decoder, we aggregate all representations of C (Eq. (1) and (4)), $C^{(1)}$ (Eq. (2) and (3)) $C^{(2)}$ (Eq. (4)) using a gating mechanism similar to that of (Alayrac et al., 2022) as follows:

$$\begin{aligned} \tilde{C}^{(1)} &= C + \tanh(p) \odot C^{(1)} \\ G^{(1)} &= \text{MHA}(G, \tilde{C}^{(1)}, \tilde{C}^{(1)}) \\ G^{(2)} &= G^{(1)} + \\ &\quad \tanh(q) \odot \text{MHA}(G^{(1)}, C^{(2)}, C^{(2)}) \end{aligned} \quad (5)$$

where $G \in \mathbb{R}^{|N^{(\text{dec})}| \times d_{\text{model}}}$ is the output of the masked multi-head attention in the decoder part, $|N^{(\text{dec})}|$ is the sequence length of the decoder input, $\tanh(\cdot)$ is the tanh function, p and q are learnable parameters, and $G^{(1)}, G^{(2)} \in \mathbb{R}^{|N^{(\text{dec})}| \times d_{\text{model}}}$.

4 Experiments

4.1 Experimental Setup

The details of the implementation and experiment setup is presented in Appendix A.

	Table				Passage				Total			
	Dev		Test		Dev		Test		Dev		Test	
	EM	F1										
HYBRIDER	51.5	58.6	52.1	59.3	40.5	47.9	38.1	46.3	43.7	50.9	42.5	50.2
HYBRIDER-Large	54.3	61.4	56.2	63.3	39.1	45.7	37.5	44.4	44.0	50.7	43.8	50.6
DocHopper	-	-	-	-	-	-	-	-	47.7	55.0	46.3	53.3
POINTR + TAPAS	68.1	73.9	67.8	73.2	62.9	72.0	62.0	70.9	63.3	70.8	62.7	70.0
POINTR + MATE	68.6	74.2	66.9	72.3	62.8	71.9	62.8	71.9	63.4	71.0	62.8	70.2
MITQA	68.1	73.3	68.5	74.4	66.7	75.6	64.3	73.3	65.5	72.7	64.3	71.9
Ours	69.4	75.2	68.5	74.9	66.5	75.5	65.7	75.3	66.2	74.1	65.4	73.6
Human	-	-	-	-	-	-	-	-	-	-	88.2	93.5

Table 1: Comparison results on the dev and blind test dataset in HybridQA. The best is bolded text.

	Table		Passage		Total	
	EM	F1	EM	F1	EM	F1
Ours	68.48	74.92	65.75	75.34	65.38	73.56
w/o Multi-row reasoning	67.44	73.74	65.50	75.23	64.86	73.08
w/o Multi-row, Single-row reasoning	41.97	49.46	60.20	69.42	51.46	59.86
w/o Single-row tanh gate	67.21	73.44	64.86	74.82	64.45	72.75
w/o Multi-row tanh gate	67.58	73.96	66.43	75.47	65.46	73.29
w/o Single-row, Multi-row tanh gate	66.09	72.51	64.81	75.22	64.01	72.65

Table 2: Ablation study on blind test dataset in HybridQA. “w/o Single-row tanh gate” and “w/o Multi-row tanh gate” correspond to the runs of fixing $\tanh(p) = 1$ and $\tanh(q) = 1$ in Eq. (5), respectively.

4.2 Baselines

In the experiment, we compare MAFiD and other baseline systems on HybridQA as follows:

- **HYBRIDER** (Chen et al., 2020) employs a sparse passage retriever (i.e., TF-IDF and a longest-substring matching) to find relevant cells and performs the reasoning step consisting of the ranking, the hop, and the reading comprehension models to extract an answer.
- **DocHopper** (Sun et al., 2021) uses the “iterative hierarchical attention” to retrieve short or long contents in a multi-step navigational manner.
- **POINTR + (TAPAS or MATE)** (Herzig et al., 2020; Eisenschlos et al., 2021a). POINTR extends the cell with its entity description and performs a two-stage method that consists of “cell selection” and “passage reading” steps. Either TAPAS (Herzig et al., 2020) or MATE (Kumar et al., 2021) is considered as a transformer encoder.
- **MITQA** (Kumar et al., 2021) uses the pipelined module including a retriever, a reader, and a joint row+span reranker, etc., being trained using the multi-instance distant supervision approach.

4.3 Main Results

As summarized in Table 1, MAFiD shows the state-of-the-art performance by increasing EM and F1 by 1.1 and 1.7 over MITQA (Kumar et al., 2021) on the blind test set. It is observed that MAFiD outperforms POINTR + (TAPAS or MATE) (Herzig et al., 2020; Eisenschlos et al., 2021a) that relies on the pretrained TAPAS, likely indicating that the long-range reasoning needs to be importantly handled on HybridQA, thus motivating the literature to go towards “reasoning”-enhanced pretraining in addition to the existing self-supervised tasks.

4.4 Ablation Studies

Single-row & Multi-row Heterogeneous Reasoning. To examine the effect of single-row and multi-row reasoning, we further evaluate MAFiD by removing either or both reasonings. As shown in Table 2, MAFiD without multi-row reasoning slightly decreases EM and F1 by 1.04 and 1.18, respectively. Importantly, MAFiD without both reasonings significantly deteriorates the performance of EM and F1 by 13.92 and 13.7, respectively. The results confirm that the cross-modal interaction should be performed at least within a specific row, whereas the between-row interaction is somehow effectively proceeded by the proposed EMA module, although its effect is not large.

Single-row & Multi-row Tanh Gating. We further examine the effect of using the gated flows

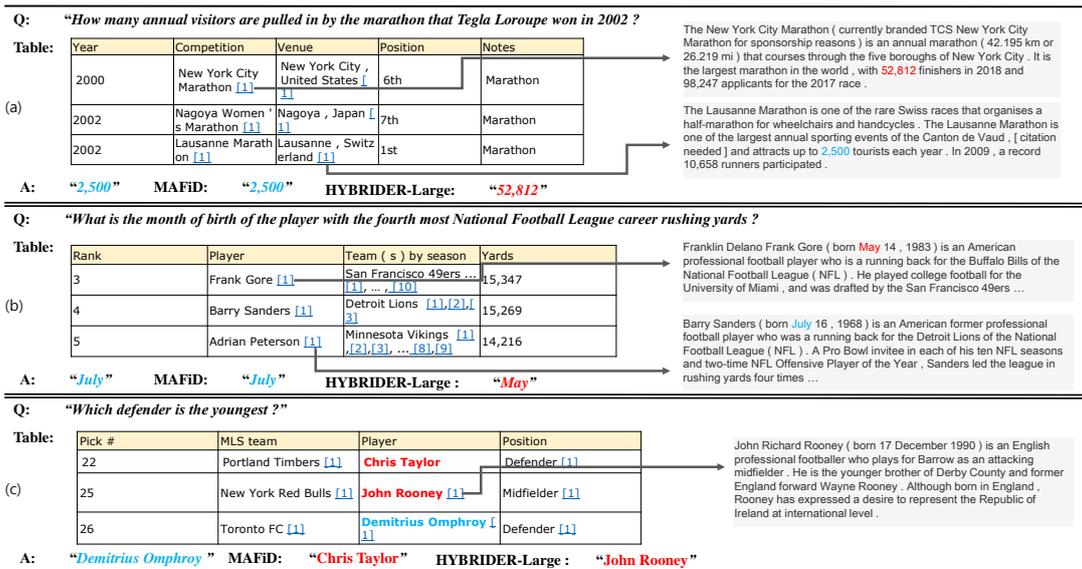


Figure 2: Illustrating examples of HYBRIDER-Large (Chen et al., 2020) and MAFiD in HybridQA.

	Total			
	Dev		Test	
	EM	F1	EM	F1
EMA	66.2	74.1	65.4	73.6
sliding window attention	65.7	73.3	65.3	73.1
Human	-	-	88.2	93.5

Table 3: Comparison results on the dev and blind test sets in HybridQA between EMA and the sliding window attention of (Beltagy et al., 2020) for long-range reasoning.

	Total			
	Dev		Test	
	EM	F1	EM	F1
original rows	66.2	74.1	65.4	73.6
permuted rows	51.5	59.4	51.1	59.2
Human	-	-	88.2	93.5

Table 4: Comparison results of MAFiD on HybridQA between the case using original rows and that with permuted rows for tabular contents.

by evaluating MAFiD by fixing $\tanh(p) = 1$ and $\tanh(q) = 1$ without being learned in Eq. (5). In particular, MAFiD without the single-row tanh gate ($\tanh(p) = 1$) slightly decreases EM and F1 by approximately 11.5, indicating that the gated FiD is helpful for further improvements.

Impact of EMA. To examine the impact of EMA for multi-row reasoning, we evaluate the sliding window attention of (Beltagy et al., 2020) as the baseline to handle long-range reasoning. As shown in Table 3, the use of EMA increases F1 and EM by 0.1 and 0.5, respectively, suggesting that EMA is more helpful for promoting the enhanced local sequence representation.

Impact of Sequential Order. To examine the impact of using the sequential order of rows in tabular contents, Table 4 further shows the results of a variant of MAFiD by randomly permuting rows in tabular contents both for training and inference, referred to as “permuted row”, comparing to the original case; the results strongly indicate that keeping original row orders is important for MAFiD.

4.5 Error Analysis

Figure 2 shows some illustrating QA examples in HybridQA dataset comparing the results of HYBRIDER-Large (Chen et al., 2020) and MAFiD; (a)-(b) require only keyword matching and numerical comparison, where HYBRIDER is failed; (c) requires sophisticated multi-hop reasoning across table rows and passages where both MAFiD and HYBRIDER are incorrect.

5 Conclusion

In this paper, we address long range-reasoning for the multi-hop table-and-text QA and propose MAFiD, which extends FiD by equipping EMA and the gated cross-attention layer for the encoder and decoder parts, respectively, to design an effective way of combining various types of encoded representations. The experimental results on HybridQA showed that the proposed MAFiD achieved state-of-the-art performances in both the development and blind test sets. In future work, we will extend MAFiD to open-domain table-and-text QA and explore a unified approach that integrates single-row and multi-row reasoning.

Limitations

This paper proposes the use of EMA under FiD to tractably perform multi-row reasoning; however, EMA simply puts strong weights on nearby contexts, thus performing a restricted type of the long-range reasoning. Thus, our EMA-based method heavily depends on the sequential order of tables and texts, so hardly performing matching between long-distance but semantically related tokens in a long hybrid sequence. In using EMA, the current limitation of our method is that we only used the “damped EMA” of MEGA (Ma et al., 2022), which is only one of the basic components in MEGA. Because MEGA additionally combines the single-head attention unit over a long sequence, using MEGA could allow us to handle long-distance semantic matching. In the future work, it will be valuable to explore such extensions of EMA, such as MEGA, to strengthen the long-range reasoning.

In MAFiD, we show that EMA can be applied over a maximally long sequence in HybridQA (Chen et al., 2020). However, when moving to OTT-QA (Wenhu Chen, 2021), EMA cannot be naively applicable over retrieved long sequences without any truncation, because the size of a retrieved set of tables and texts is significantly larger than that of HybridQA. Given that OTT-QA more closely matches the real-world situation, the EMA-based reasoning should be extended further by incorporating retrieval and selection modules. Thus, our current framework needs to be extended further to handle open-domain table-and-text QA, under the retriever-reader framework.

Acknowledgements

This work was supported by NAVER Corporation. We would like to thank all anonymous reviewers for their valuable comments and suggestions.

References

- Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. [ETC: Encoding long and structured inputs in transformers](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online. Association for Computational Linguistics.
- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. [Flamingo: a visual language model for few-shot learning](#).
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The long-document transformer](#). *arXiv:2004.05150*.
- Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. [HybridQA: A dataset of multi-hop question answering over tabular and textual data](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. [DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.
- Julian Eisenschlos, Maharshi Gor, Thomas Müller, and William Cohen. 2021a. [MATE: Multi-view attention for table transformer efficiency](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7606–7619, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Julian Martin Eisenschlos, Maharshi Gor, Thomas Müller, and William W. Cohen. 2021b. [Mate: Multi-view attention for table transformer efficiency](#).
- Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. [TaPas: Weakly supervised table parsing via pre-training](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.
- Junjie Huang, Wanjun Zhong, Qian Liu, Ming Gong, Daxin Jiang, and Nan Duan. 2022. [Mixed-modality representation learning and pre-training for joint table-and-text retrieval in openqa](#).
- Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. [TABBBIE: Pretrained representations of tabular data](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456, Online. Association for Computational Linguistics.

- Gautier Izacard and Edouard Grave. 2021. [Leveraging passage retrieval with generative models for open domain question answering](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Vishwajeet Kumar, Saneem A. Chemmengath, Yash Gupta, Jaydeep Sen, Samarth Bharadwaj, and Soumen Chakrabarti. 2021. [Multi-instance training for question answering across table and linked text](#). *CoRR*, abs/2112.07337.
- Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Zettlemoyer Luke. 2022. [Mega: Moving average equipped gated attention](#). *arXiv preprint arXiv:2209.10655*.
- Kai Nakamura, Sharon Levy, Yi-Lin Tuan, Wenhui Chen, and William Yang Wang. 2022. [HybridDialogue: An information-seeking dialogue dataset grounded on tabular and textual data](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 481–492, Dublin, Ireland. Association for Computational Linguistics.
- Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. [QuALITY: Question answering with long input texts, yes!](#) In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. [Know what you don’t know: Unanswerable questions for SQuAD](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.
- Haitian Sun, William W. Cohen, and Ruslan Salakhutdinov. 2021. [End-to-end multihop retrieval for com-](#)
- [positional question answering over long documents](#). *CoRR*, abs/2106.00200.
- Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hananeh Hajishirzi, and Jonathan Berant. 2021. [Multi-modal{qa}: complex question answering over text, tables and images](#). In *International Conference on Learning Representations*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. [Constructing datasets for multi-hop reading comprehension across documents](#). *Transactions of the Association for Computational Linguistics*, 6:287–302.
- Eva Schlinger William Wang William Cohen Wenhui Chen, Ming-wei Chang. 2021. [Open question answering over tables and text](#). *Proceedings of ICLR 2021*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. [TaBERT: Pretraining for joint understanding of textual and tabular data](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.
- Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021a. [TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.
- Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021b. [Retrieving and reading: A comprehensive survey on open-domain question answering](#). *CoRR*, abs/2101.00774.

A Implementation Details

We used the HybridQA dataset, which is a large-scale multi-hop question answering dataset over tabular and textual data. Table 5 presents detailed

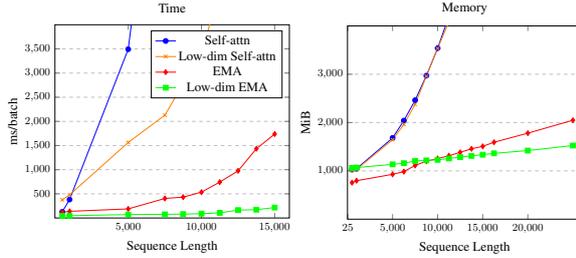


Figure 3: Comparison of memory and time complexities of self-attention, EMA, and their low-dimensional versions.

statistics of the HybridQA dataset. We used the T5-base¹ transformer encoder-decoder model as a pre-trained language model. Additional parameters were initialized from $\mathcal{N}(0, 0.2)$ and the bias was set to 0. To prepare the input of the encoder, we set the maximum sequence lengths of the tabular and textual blocks to 300 and 800, respectively. All the models were trained using the AdamW optimizer with a learning rate of $1e - 4$. All trainings were conducted for 3 epochs, and the random seed number was fixed at 42 to reproduce the results. The batch size was four with two accumulation steps. Training was conducted for 1.5 days on 4 NVIDIA Quadro RTX 8000. For answer generation, we employed a greedy decoding method. For the evaluation, we used exact matching (EM) and F1 metrics. Evaluations were conducted every 500 steps on the dev dataset and the best model with the highest EM score was chosen.

The maximum number of rows in a given table in HybridQA is 20, that is, $L \leq 20$. In our experiments, L was fixed at 20 by adding padding sequences when the number of rows was less than 20. The rate of the dimensionality reduction for the low-dimensional EMA, i.e., K , was fixed to 6.

B Dataset Statistics

Split	Train	Dev	Test	Total
In-Passage	35,215	2,025	20,45	39,285
In-Table	26,803	1,349	1,346	29,498
Missing	664	92	72	828
Total	62,682	3,466	3,463	69,611

Table 5: Hybrid QA dataset statistics. In-Passage and In-Table indicate that exact answer span is founded in a passage or table. Missing is the exact answer span not founded in given source.

¹<https://huggingface.co/t5-base>

Split	min	mean	max	Count
Table	137	763	8,298	3,466
Row	14	48	1,454	55,036
Passages per row	2	656	10,797	55,036

Table 6: Statistics of length of tokenized sequence on dev dataset. ‘Passages per row’ is the length of all concatenated passages in a row.

C An Example of Linearized Blocks

Specifically the i -th table row block is defined as follows:

$$b_R^i = [\text{TITLE}] t [\text{SECTITLE}] t_{(sec)} \\ [\text{ROW}] h^1 \text{'is'} v^{i,1} [\text{SEP}] \dots [\text{SEP}] h^N \text{'is'} v^{i,N}$$

where h and v are the head and value, t and $t_{(sec)}$ are the title and section title, respectively.

Passage block is defined as follows:

$$b_P^i = [\text{PSG}] psg_{linked}^{i,1} [\text{PSG}] \dots [\text{PSG}] psg_{linked}^{i,N}$$

where psg is a linked passage at row.

D Single-head and Multi-head Attentions

The single-head and multi-head attentions (Vaswani et al., 2017) are defined as follows:

$$\text{SHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\text{head}_1] \mathbf{W}^O, \\ \text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\text{head}_1; \dots; \text{head}_h] \mathbf{W}^O, \\ \text{head}_i = \text{Attn}(\mathbf{Q} \mathbf{W}_i^Q, \mathbf{K} \mathbf{W}_i^K, \mathbf{V} \mathbf{W}_i^V) \quad (6)$$

E EMA

EMA has widely been applied in time series and long range text modeling. Among variants of EMA presented in the work of (Ma et al., 2022), we employed the ‘‘damped EMA’’² for proceeding long range sequences. The damped EMA is based on the recursive calculation for computing the output \mathbf{Y} as follows:

$$\mathbf{y}_t = \alpha \odot \mathbf{x}_t + (1 - \alpha \odot \delta) \odot \mathbf{y}_{t-1} \quad (7)$$

where \odot is the element-wise multiplication operator, $\alpha \in (0, 1)^d$ is a decaying factor for making exponentially decreasing effects from older tokens, $\delta \in (0, 1)^d$ is the damping factor, and α and δ are learnable weight parameters. This recursive computation of EMA can be efficiently implemented as the convolution and the fast Fourier transforms.

²This is different from the further extended multi-dimensional damped EMA (Ma et al., 2022)

Transformer-based Models for Long-Form Document Matching: Challenges and Empirical Analysis

Akshita Jha

Virginia Tech, Arlington, VA
akshitajha@vt.edu

Adithya Samavedhi

Virginia Tech, Arlington, VA
adithyas@vt.edu

Vineeth Rakesh

InterDigital, CA
vineethrakesh@gmail.com

Jaideep Chandrashekar

InterDigital, CA
jaideep.chandrashekar@interdigital.com

Chandan K. Reddy

Virginia Tech, Arlington, VA
reddy@cs.vt.edu

Abstract

Recent advances in the area of long document matching have primarily focused on using transformer-based models for long document encoding and matching. There are two primary challenges associated with these models. Firstly, the performance gain provided by transformer-based models comes at a steep cost – both in terms of the required training time and the resource (memory and energy) consumption. The second major limitation is their inability to handle more than a pre-defined input token length at a time. In this work, we empirically demonstrate the effectiveness of simple neural models (such as feed-forward networks, and CNNs) and simple embeddings (like GloVe, and Paragraph Vector) over transformer-based models on the task of document matching. We show that simple models outperform the more complex BERT-based models while taking significantly less training time, energy, and memory. The simple models are also more robust to variations in document length and text perturbations.

1 Introduction

Matching long documents (*e.g.*: research papers, Wikipedia articles, patents, etc.) is an important task that can help understand the (dis)similarity between documents for downstream tasks like long document search. The first step towards better document matching is obtaining meaningful long document representations. Recent advances in this area have primarily focused on using transformer-based models for long document encoding and matching (Beltagy et al., 2020; Jha et al., 2022; Yang et al., 2020b; Zaheer et al., 2020). We use the term transformers to mean pre-trained transformers. Despite promising results, there are two primary challenges associated with such models. First, the performance gain provided by the huge transformer-based language models (LMs), like BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019), and

Longformer (Beltagy et al., 2020) come at a steep cost – both in terms of the required training time, and the resource (memory and energy) consumption. For example, the smaller $BERT_{BASE}$ model has 110 million parameters, whereas the bigger $BERT_{LARGE}$ model has a total of 340 million parameters and fine-tuning a single $BERT_{BASE}$ model on GPU can take hours. The second major limitation of transformer-based models is their inability to handle more than a pre-defined input token length at a time (512 tokens for BERT, and 4096 tokens for Longformer). This is a big drawback as they cannot handle long documents like research papers, patents, long articles, etc., without using aggregation techniques (Reimers and Gurevych, 2019).

In this work, we empirically demonstrate that *embeddings obtained from GloVe (Pennington et al., 2014), and Paragraph Vectors (Le and Mikolov, 2014)* along with simple neural models, such as feed-forward networks, and CNNs, *outperform several transformer-based models for the document matching task*. We define these models as simple as they take significantly less time to train and consume less memory and energy overall when compared to complex transformer-based models. Our long document matching setting is fundamentally different from long-form question answering and sentence similarity tasks. For the latter tasks the query is ‘short’, unlike the long document matching task where both the query and the target text are ‘long’. We experiment with three different kinds of semi-structured long document datasets in English: (i) ACL Anthology research papers, (ii) English Wikipedia articles, and (iii) Patents from US Patent and Trademark Office (USPTO). Our primary contributions are summarized as follows:

- We provide insights into the challenges of using transformer-based models for the task of long document matching. For this task, simple neural models are as effective and take a

fraction of the training time and resources to outperform transformer-based models.

- We provide insights into the best input embeddings for the simpler models in this task.
- We demonstrate that simple models are also more robust to changes in document length and text perturbation.
- We create benchmark long document datasets (by pre-processing ACL Anthology 2014 papers and Wikipedia articles) that will be made publicly available.

2 Related Work

Early work on long document matching focused on clustering techniques (Friburger et al., 2002; Huang et al., 2008; Strehl et al., 2000). Recently, Guo et al. (2016) proposed a deep learning based architecture for ad-hoc retrieval when comparing documents. Some works have also used convolutional networks (Hu et al., 2014; Pang et al., 2016; Yu et al., 2018), with weighting mechanism (Yang et al., 2016a) to generate a final query-document score. Mitra et al. (2017) propose a combination model that uses weighted sum representation-based and interaction-based results. Yang et al. (2016b) propose HAN, a hierarchical attention network for document matching. Jiang et al. (2019) propose SMASH, a multi-depth attention based hierarchical recurrent neural network for long-document matching. However, Yang et al. (2020b) pre-train SMITH, a transformer based hierarchical model for text matching that outperforms SMASH across multiple datasets. Jha et al. (2022) use supervised contrastive learning for interpretable long document matching. We compare several of these models on the required training time and resources.

A growing body of literature has used transformer based model for long document encoding (Child et al., 2019; Ho et al., 2019; Kitaev et al., 2019; Liu and Lapata, 2019; Qiu et al., 2020; Yang et al., 2020a). Longformer (Beltagy et al., 2020) adapts transformers to use an attention mechanism that scales linearly with sequence length. Big bird (Zaheer et al., 2020) uses a sparse attention mechanism that reduces BERT’s quadratic dependency on the sequence length to linear. CogLTX (Ding et al., 2020) uses text blocks for rehearsal and decay over key sentences to overcome the insufficient long-range attentions in BERT. Transformer-XL (Dai et al., 2019) and Compressive Transformers (Rae

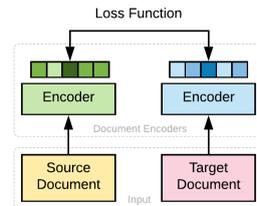


Figure 1: A schematic siamese comparison model

et al., 2019) compress the transformers to use attentive sequence over long text. Although promising, we demonstrate that transformer-based models are not considerably better than simple neural models on the task of long document matching.

3 Empirical Evaluation

Here we provide details of the simple and the transformer-based models and present an empirical comparison between them based on their overall performance, training time, resource consumption, and robustness on the document matching task.

Task Formulation We define the task of document matching as follows. Given a source document s , and a set of target documents D_T , the goal is to estimate the semantic match $\hat{y} = sim(s, t)$, where $t \in D_T$ for every document pair (s, t) . Similar target documents will have a higher similarity score. The document matching problem can be seen as a binary classification task, where the model predicts 1 for similar documents, and 0 for dissimilar documents. We use the term “matching” in the broad sense of document relevance (see Appendix A.2). The models take as input a pair of documents (source and target), and compute the cosine similarity between the encoded document representations. If the cosine similarity is greater than a similarity threshold θ , they are considered similar; otherwise they are considered dissimilar.

Models We pick a representative set of models from different categories and compare them by building their siamese versions (shown in Figure 1). The siamese network has three primary components: (i) Input (Source and Target Document), (ii) Document Encoder, and (iii) Loss Function. The source and target document encoder networks share weights. We experiment with three simple neural models: (i) DSSM: A simple feed-forward network (Huang et al., 2013), (ii) ARC-I: A CNN-based model (Hu et al., 2014), and (iii) HAN: An RNN-based Hierarchical Attention Network (Yang et al., 2016b) designed for long documents. Their performance is compared with three state-of-the-art transformer-based models: (i)

Model	AAN				WIKI				PAT			
	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc
HAN-G	0.504	0.881	0.641	0.607	0.566	0.584	0.575	0.775	0.609	0.848	0.709	0.522
DSSM-T	0.768	0.809	0.787	0.780	0.823	0.939	0.877	0.869	0.869	0.957	0.911	0.905
DSSM-G	0.550	0.541	0.545	0.549	0.966	0.986	0.975	0.975	0.992	0.998	0.995	0.995
DSSM-D	0.852	0.763	0.805	0.815	0.933	0.984	0.958	0.957	0.841	0.959	0.896	0.949
ARC-I-G	0.643	0.873	0.734	0.743	0.992	0.983	0.987	0.987	0.905	0.963	0.933	0.939
ARC-I-D	0.841	0.763	0.800	0.809	0.987	0.985	0.986	0.986	0.967	0.958	0.962	0.983
BERT	0.760	0.914	0.793	0.761	0.980	0.950	0.960	0.960	1.0	0.988	0.994	0.994
LONG	0.681	0.833	0.749	0.773	0.974	0.960	0.967	0.967	1.0	0.984	0.992	0.992
SMITH	0.726	0.565	0.635	0.676	0.949	0.982	0.965	0.963	0.892	0.939	0.865	0.939

Table 1: Performance on the document matching task up to the model’s maximum allowed input token length (512 for BERT; 4096 for Longformer, > 8000 for all other models). We experiment with Trigrams (T), GloVe (G), and Doc2Vec (D) Embeddings as input for the simple neural models. The best performance is highlighted in bold.

BERT (Devlin et al., 2019), (ii) LONG: Longformer (Beltagy et al., 2020), and (iii) SMITH: Siamese Multi-depth Transformer based Hierarchical Encoder (Yang et al., 2020b). We report the mean precision, recall, F1, and accuracy over 5 folds for the best performing hyper-parameters. The code can be found here: <https://github.com/AkshitaJha/SimpleModelsforLongDocumentMatching>.

Datasets We follow the previous literature (Yang et al., 2016b; Jiang et al., 2019; Yang et al., 2020b) and experiment with the following three standard long document datasets: (i) ACL Anthology Network Corpus (AAN)¹, (ii) Wikipedia Articles (WIKI)², and (iii) USPTO Patents (PAT)³. Each dataset consists of balanced 15,000 pairs of documents with 50% of them being similar pairs, and the remaining being dissimilar. The PAT dataset is an industry gold standard but we will publicly release the pre-processed AAN and the WIKI datasets (see Appendix A.2 for details). The dataset can be found here: <https://github.com/AkshitaJha/SimpleModelsforLongDocumentMatching>

Performance on Document Matching Task We experiment with three input representations for simple neural models: (i) char-Trigram Hashing (T) (Huang et al., 2013), (ii) GloVe Embeddings (G) (Pennington et al., 2014), and (iii) Paragraph Vector/Doc2Vec Embeddings (D) (Le and Mikolov, 2014) (See Appendix 4). Unlike most transformer-based models that take as input tokens up to a pre-defined length (512 for BERT, and 4096 for Longformer), simple models and SMITH have the ability to take the entire long document (> 8000 tokens) as input. Table 1 demonstrates the performance of different models on the task of docu-

ment matching up to their maximum allowed input document length (see Appendix A.8 for different document lengths.) We observe that despite being relatively simple and not taking into account contextual embeddings, *DSSM and ARC-I outperform the transformer based models using GloVe and Doc2Vec Embeddings* on the AAN, WIKI, and the PAT dataset.

Training Time Figure 2a shows the training time taken to reach the best performance for every model for their maximum allowed input token lengths. We only report the fine-tuning time after downloading the pre-trained models. All experiments were done on a 16GiB Tesla V100. *The simple models like DSSM, ARC-I, and HAN take 1/12 to 1/15 of the training time taken by the transformer-based models to outperform them on all three datasets* (see Appendix A.4 for the training time for different document lengths on all datasets.)

Memory and Energy consumption Memory consumption on a 16GiB Tesla V100, for a batch size of 1, for different models can be seen in Figure 2b. Compared to transformer-based models simple neural models consume significantly less memory for the same document length (12 GiB for Longformer vs. a maximum of 8 GiB for DSSM for 4000 tokens). We also compute the overall energy required for training the models to achieve their best performance (Figure 2c) by measuring the power consumption of the GPU over their training lifetime. Longformer consumes > 6MJ of energy for fine-tuning on documents with 4096 tokens, BERT consumes \approx 500kJ of energy for fine-tuning on documents with just 512 tokens, and the SMITH model consumes \approx 200kJ of energy for fine-tuning on longer documents; whereas the simple models consume < 100kJ of energy for training from

¹<https://aan.how/download/#aanNetworkCorpus>

²<https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2>

³<https://github.com/google/patents-public-data>

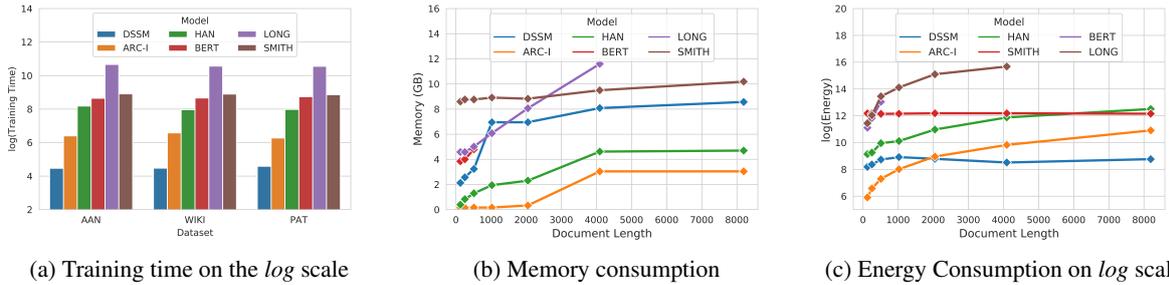


Figure 2: Comparison of simple neural models with transformer-based models based on (a) Training Time, (b) Memory Consumption, and (c) Energy Consumption.

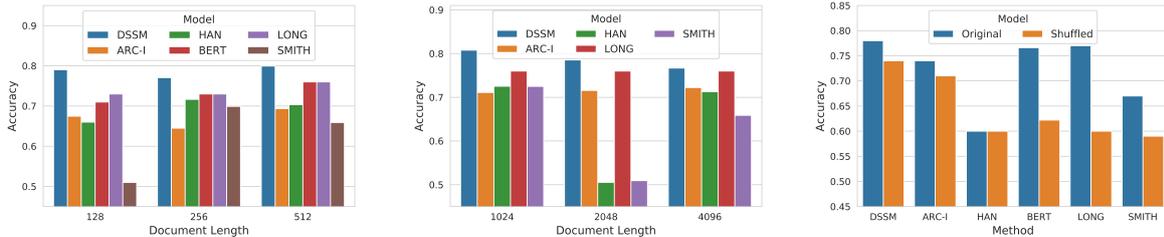


Figure 3: Comparison between the robustness of simple neural models and transformer-based models w.r.t. document length and text perturbation on the AAN dataset.

scratch for documents with > 8000 tokens.

Robustness to Document Length We limit the maximum number of tokens in each document during training and testing, and observe the final test accuracy on the document matching task. It should be noted that documents of different lengths are actually ‘truncated long documents’ without the complete contextual information needed to compute the actual similarity between two long documents. Figure 3 compares the model accuracy of simple models (with their default input embeddings) with transformer-based models upto their maximum allowed token lengths for the AAN dataset – 512 tokens for BERT (Figure 3a), and 4096 tokens for Longformer (Figure 3b). DSSM outperforms the baseline models for all documents lengths. BERT and Longformer have a consistent performance on AAN for different input lengths, unlike HAN and SMITH that are not as robust to the variations in document length, although they were designed specifically for long documents. We found similar results for WIKI and PAT dataset (see Appendix A.5). We also experiment with documents of length > 512 tokens for BERT, and > 4096 tokens for Longformer by aggregating the chunk representations upto their maximum allowed token length. We used the SUM and AVG aggregation techniques and observed an overall performance drop (see Table 5).

Robustness to Text Perturbation For text perturbation, we split documents into paragraphs of 512 tokens and randomly shuffle the order of these paragraphs before training different models to check for learned positional bias. We measure their test accuracy on the original document matching task. Figure 3c shows the model performance for all the baseline methods on AAN dataset. We observe a significant drop in the model performance for transformer-based models (BERT, Longformer, and SMITH). There is no significant change in the accuracy for the simple models – DSSM, ARC-I, and HAN. The transformer based models are more sensitive to text perturbation. The simple models, on the other hand, use non-contextual embeddings, such as GloVe, and Doc2Vec and are more robust to text perturbation (see Appendix A.6).

4 Conclusion

We empirically demonstrate the trade-off of using transformer-based models for semi-structured long English documents like research papers, Wikipedia articles, and patents. Transformer-based models have an overall strong performance and smaller variability across datasets. However, we observe that for the task of long document matching, using contextual embeddings do not provide any added advantage. **A simple feed-forward network or a CNN-based model using GloVe or Doc2Vec embedding outperforms several state-of-the-art**

pre-trained transformer-based models at a fraction of their overall training time and resources (memory and energy). These simple neural models are also more robust to changes in document length and text-perturbation.

5 Limitations

One of the limitations of our work is that we experimented only with long documents in English. Comparing simple neural models and transformer-based models in different languages would be an interesting study but is outside the scope of this short paper. We would also like to highlight that we use classification metrics instead of information-retrieval metrics due to the limitations of the dataset which has very few positive samples (2-3) for every document.

References

- Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Ming Ding, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. CogLtx: Applying bert to long texts. *Advances in Neural Information Processing Systems*, 33.
- Nathalie Friburger, Denis Maurel, and Arnaud Giacometti. 2002. Textual similarity based on proper names. In *Proc. of the workshop Mathematical/Formal Methods in Information Retrieval*, pages 155–167.
- Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international conference on information and knowledge management*, pages 55–64.
- Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. 2019. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*.
- Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. *Advances in neural information processing systems*, 27:2042–2050.
- Anna Huang et al. 2008. Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.
- Akshita Jha, Vineeth Rakesh, Jaideep Chandrashekar, Adithya Samavedhi, and Chandan K. Reddy. 2022. [Supervised contrastive learning for interpretable long-form document matching](#).
- Jyun-Yu Jiang, Mingyang Zhang, Cheng Li, Michael Bendersky, Nadav Golbandi, and Marc Najork. 2019. Semantic text matching for long-form documents. In *The World Wide Web Conference*, pages 795–806.
- Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2019. Reformer: The efficient transformer. In *International Conference on Learning Representations*.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.
- Yang Liu and Mirella Lapata. 2019. [Hierarchical transformers for multi-document summarization](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.
- Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299.
- Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

- Jiezhong Qiu, Hao Ma, Omer Levy, Wen-tau Yih, Sinong Wang, and Jie Tang. 2020. Blockwise self-attention for long document understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2555–2565.
- Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. 2013. [The acl anthology network corpus](#). *Language Resources and Evaluation*, pages 1–26.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983.
- Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. 2000. Impact of similarity measures on web-page clustering. In *Workshop on artificial intelligence for web search (AAAI 2000)*, volume 58, page 64.
- Linyi Yang, Tin Lok James Ng, Barry Smyth, and Rihui Dong. 2020a. Html: Hierarchical transformer-based multi-task learning for volatility prediction. In *Proceedings of The Web Conference 2020*, pages 441–451.
- Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. 2016a. anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 287–296.
- Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020b. Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1725–1734.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016b. [Hierarchical attention networks for document classification](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.
- Jianfei Yu, Minghui Qiu, Jing Jiang, Jun Huang, Shuangyong Song, Wei Chu, and Haiqing Chen. 2018. Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 682–690.
- Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

A Appendix

A.1 Model Description

The encoder networks and their default inputs have been described below:

A.1.1 Simple Models

- **DSSM (Huang et al., 2013)**: A *simple three-layered feed forward network* that takes as input the vectorized representation of a document.
- **ARC-I (Hu et al., 2014)**: A *CNN-based model* that takes as input a 2D-matrix representation of a document where words in the sentences are represented using *GloVe embeddings (Pennington et al., 2014)*. These are then passed through convolutional and max-pooling layers to finally obtain a document representation for both source and target documents, independently. The document representations are concatenated and passed through a multi-layer perceptron to predict if the pair of documents are similar or not.
- **Hierarchical Attention Network (HAN) (Yang et al., 2016b)**: A hierarchical *GRU-based model* with attention mechanism that aggregates *GloVe embeddings* at word level into sentence representations to arrive at the final document representation.

A.1.2 Transformer-Based Models

- **BERT (Devlin et al., 2019)**: A siamese matching model with *BERT*. For long document inputs, BERT only uses the first 512 tokens of each document. We use a pre-trained *BERT_{BASE}* model which is fine-tuned during training.
- **Longformers (LONG) (Beltagy et al., 2020)**: A siamese model with transformer-based *Longformers* for long sequences. It has an attention mechanism that scales linearly with sequence length and takes as input a maximum of 4096 tokens. We consider an attention window of size 256.
- **Siamese Multi-depth Transformer based Hierarchical Encoder (SMITH) (Yang et al., 2020b)**: A *transformer-based hierarchical encoder* which is the current SOTA model for long-form document matching task.

A.1.3 Implementation Details

For all the models presented in the paper, we use the same architecture as the original papers. We tune the hyperparameters and report the best results. The DSSM, ARC-I, HAN, and SMITH models were implemented in Tensorflow. BERT and Longformer were implemented in PyTorch. DSSM has hidden units of dimension 300 for its hidden layers and an output dimension of 128. The learning rate was 0.0075. ARC-I takes as input a 2D matrix of the size [no. of sentences x sentence length]. This is given as input to two 1D-convolutional (filter size of 200, kernel size 3) and MaxPooling layers of size 2, in order to get the final document representations. The representations of both the source and the target documents are concatenated and passed through a two-layer multi-layer perceptron. The first hidden layer of the MLP has a dimension of 64 with ReLU activation. The second layer has 1 node and sigmoid activation which predicts if the pair of documents are similar or not. The learning rate was set to 0.00075. HAN uses a bi-Directional GRU layer and applies attention mechanism to arrive at final sentence representation for the source and the target documents with a learning rate of 0.001. We use the pre-trained BERT_{BASE}⁴ and the Longformer⁵ models provided by the Huggingface library. The SMITH code was publicly available. The BERT, Longformer, and SMITH models are fine-tuned during training. All other models are trained from scratch. The learning rate is set to 5e-5 for the transformer based models. We use an Adam optimizer for all models with a weight decay of 0.01. We use binary cross entropy as the loss function for the simple models, pairwise loss for BERT and Longformer, and triplet loss for SMITH. These loss functions resulted in the best performance for the model. The three datasets are split into 80-10-10 for train, validation, and test sets, respectively. We use cosine similarity and the similarity threshold θ is set to 0.5. We perform 5 fold cross-validation and use early stopping on the validation set to prevent over-fitting. The models were trained on one 16GB Tesla V100 GPU.

A.2 Dataset

- **ACL Anthology Network Corpus (AAN)⁶**: The AAN corpus (Radev et al., 2013) consists

⁴https://huggingface.co/transformers/model_doc/bert.html

⁵<https://huggingface.co/transformers/longformer.html>

⁶<https://aan.how/download/#aanNetworkCorpus>

of 23,766 papers written by 18,862 authors in 373 venues related to NLP and forms a citation network. Each paper is represented by a node with directed edges connecting a paper (the parent node) to all its cited papers (children nodes). Papers that have been cited by the parent paper are treated as similar samples (Jiang et al., 2019). For every similar sample, an irrelevant paper is randomly chosen to create a balanced dataset. Sets of similar papers are given the same labels. To prevent leakage of information and make the task more difficult, the references and the abstract sections are removed. Papers without any content are also removed. We then randomly sample 15,000 research paper pairs for our experiment.

- **Wikipedia (WIKI)**⁷: We use the Wikipedia dump, and adopt a similar methodology proposed by Jiang et al. (Jiang et al., 2019) to process this data. From the Wikipedia dump containing 6 million articles, we randomly sampled 250,000 articles along with the articles present in their outlinks. We create a dataset of similar Wikipedia articles by assuming that similar articles have similar outgoing links. The Jaccard similarity between the outgoing links of the source and the target articles is calculated. If the Jaccard similarity > 0.5 , the documents are assumed to be similar, otherwise they are considered dissimilar. Only articles with two or more similar articles are selected. We then randomly sample 15,000 research paper pairs for our experiment.
- **Patent (PAT)**⁸: The patent dataset is an internally curated industry gold-standard. This dataset consists of patents sampled from the publicly available USPTO patents belonging to four different categories: video, wireless, image compression, and network compression. A patent document is extremely long and primarily consists of (i) Abstract, (ii) Claims, and (iii) Description sections. We only make use of the Claims and the Description sections for our experiments to prevent leakage of information from Abstracts. Three internal human annotators, with expert domain knowl-

edge, were given pairs of documents and were asked to label them as similar or dissimilar based on the technology presented in the patents. They referred to the Abstract, Claims, and CPC Codes⁹ of the patents to measure the similarity. The final document content similarity label was based on majority vote.

The dataset statistics can be found in Table 2. We would like to note that although considering only the cited papers and outgoing links for AAN and Wikipedia articles, respectively is not the most optimal approach for creating similar document pairs, we adopt it for the following two reasons: (i) We do not have annotated fine-grained similarity scores for AAN and WIKI datasets, and (ii) We follow an approach similar to the previously published work (Jiang et al., 2019; Yang et al., 2016b, 2020b). We use the term “document matching” or “document similarity” in the broad sense of “citation matching” or “document relevance” – Given a pair of documents, are the two documents relevant to each other and should they be cited by each other.

A.3 Performance on Document Matching Task

Table 3 shows the average performance of simple models when compared to transformer based models on the document matching task for AAN, WIKI, and PAT datasets. ARC-I with Doc2Vec embeddings has the best average precision and accuracy. The average F1 score is comparable to BERT which re-emphasizes the benefits of using simple models for document matching.

A.4 Training Time for Different Document Lengths

The training time for different document lengths for all three datasets can be seen in Figure 4. BERT can only take up to 512 tokens. DSSM, ARC-1, and HAN take considerably less time to train when compared to transformer-based.

A.5 Robustness to Document Length

We check the robustness of simple models and transformer-based models for different document lengths on the task of document matching. From Figure 5 and Figure 6, we observe that the simple

⁷<https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2>

⁸<https://github.com/google/patents-public-data>

⁹<https://www.uspto.gov/web/patents/classification/cpc/html/cpc.html>

Dataset	Avg #Tokens	Max #Tokens	Avg #Sentences	Max #Sentences	Vocabulary
AAN	5,381.1	54,556	215.7	2,183	515,422
WIKI	3,777.0	26,172	190.6	1,685	1,151,309
PAT	8,177.4	50,322	214.1	2,709	220,023

Table 2: Dataset statistics

Model	P	R	F1	Acc
HAN-G	0.559	0.771	0.641	0.634
DSSM-T	0.820	0.901	0.858	0.851
DSSM-G	0.836	0.841	0.838	0.839
DSSM-D	0.875	0.902	0.886	0.907
ARC-I-G	0.846	0.939	0.884	0.889
ARC-I-D	0.931	0.902	0.916	0.926
BERT	0.913	0.950	0.915	0.905
LONG	0.885	0.925	0.902	0.910
SMITH	0.855	0.828	0.821	0.860

Table 3: Average performance across AAN, WIKI, and PAT datasets on the document matching task (shown in Table 1). We experiment with Trigrams (T), GloVe (G), and Doc2Vec (D) Embeddings as input for the simple neural models. The best performance is highlighted in bold.

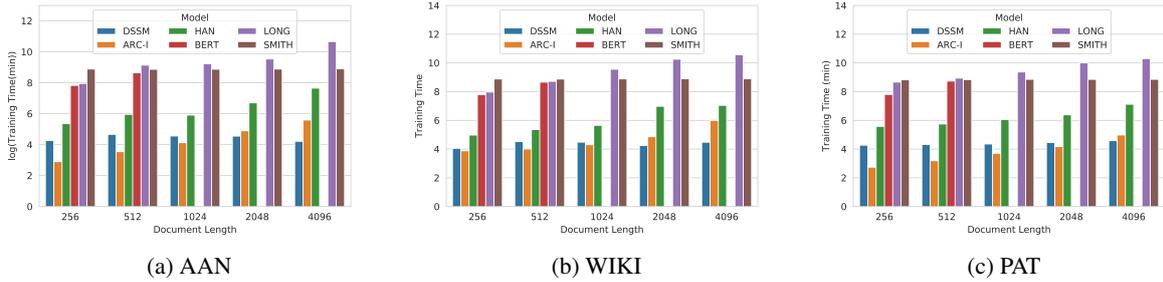


Figure 4: Doc Length vs Training Time (in *log* scale) for different document lengths.

models DSSM and ARC-I, and the transformer-based models BERT and Longformer, though not specifically designed for long documents are robust for different document lengths. BERT can only handle up to 512 tokens at a time, and Longformer can only handle up to 4096 input tokens. HAN and SMITH, on the other hand, were specially designed for long documents and have a high variance in their performance on the document matching tasks for different document lengths.

We also experimented with longer documents (> 512 tokens for BERT, and > 4096 tokens for Longformer). We obtained the final document representation by dividing the document into chunks of their maximum allowed token length. We then aggregated these chunk representations. We experimented with the SUM and AVG aggregation techniques by taking the representations of the ‘[CLS]’ token and ‘the pooler output’ for these models. We

observed an overall performance drop because of aggregation. The results were the same for both SUM and AVG aggregation techniques (Table 5).

A.6 Robustness to Text Perturbation

We randomly shuffle the documents before training different models and measure their test accuracy on the original document matching task (Figure 7) for all three datasets. The first few paragraphs in Wikipedia articles, research papers, and patents are highly informative. We wanted to verify if the models give too much importance to the position of the initial text. In the context of long documents, just re-ordering the paragraphs of a document spanning pages should not have an effect on the downstream tasks of document matching. (Note: We use the term document matching broadly to refer to citation matching or document relevance. Given a document pair, we would like to verify if the two

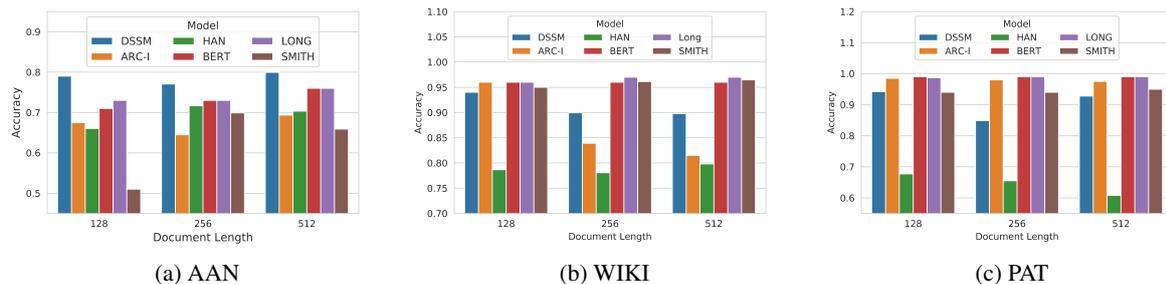


Figure 5: Document Length vs Accuracy upto 512 tokens

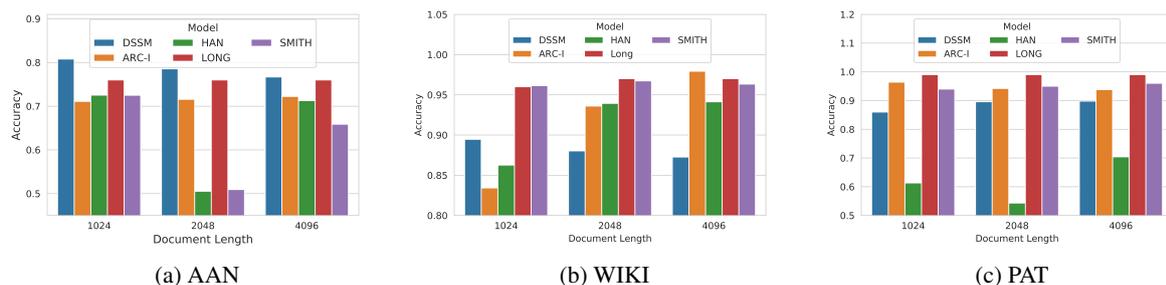


Figure 6: Document Length vs Accuracy upto 4096 tokens

documents are relevant to each other.) In order to verify this assumption, we shuffle the paragraphs to distribute the important texts randomly and check the performance of all models on this downstream task. Although, we do observe a small drop after paragraph shuffling because the simple models do take into account a shallow context of the input text, the simple neural models overall prove more robust to text perturbation when compared to transformer-based models that take into account deep contextual information.

A.7 Best Input Embeddings for Simple Models

For simple models, we evaluate if the input vector representations play a role in the final results. We use the following input representations.

- **Tri-Gram Hashing (T):** Bag-of-charTrigrams is a technique for word hashing (Huang et al., 2013) where each word is broken down into character trigrams (charTrigrams). Since, the number of possible charTrigrams are fixed and limited, this is a scalable solution for long documents. The charTrigrams are obtained for every token in the input text after appending the symbol ‘#’ before and after every token. For example, the word ‘good’ [#good#]

is split into [#go, goo, ood, od#] and then mapped to a 30,621 dimensional hash table. This vector representation for the document is then given as input to the models. For DSSM, each document is represented as a bag-of-charTrigrams and given as input to the model. For ARC-I and HAN, we split each document into n chunks which are represented in the form of a trigram hash. We construct a matrix of size $n \times$ trigram hash for the entire document which is given input to ARC-I and HAN.

- **GloVe Embeddings (G):** GloVe (Pennington et al., 2014) is an unsupervised learning algorithm for obtaining vector representations for words. We download the pre-trained GloVe embeddings and get the vector representations for words in a long document. These vector representations are given as input to different models. For DSSM, we divide the document into chunks of a specified maximum length. We then take GloVe embedding representation of tokens for each chunk upto a maximum length and average them to get a document representation. For ARC-I and HAN, each document is represented as a matrix of size [embedding dimension \times max length] in each document.

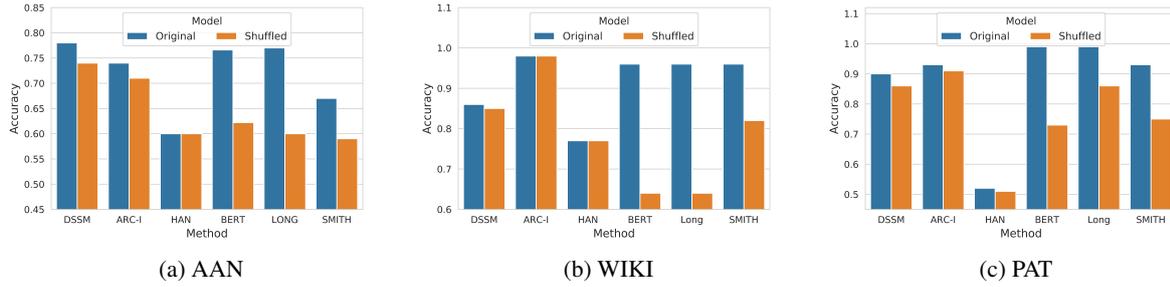


Figure 7: Original vs Shuffled Documents

Model	AAN				WIKI				PAT			
	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc
DSSM-T	0.768	0.809	0.787	0.780	0.823	0.939	0.877	0.869	0.869	0.957	0.911	0.905
ARC-I-T	0.641	0.606	0.622	0.634	0.969	0.944	0.956	0.957	0.536	0.754	0.626	0.793
HAN-T	0.665	0.885	0.759	0.720	0.911	0.929	0.920	0.920	0.477	0.857	0.618	0.751
DSSM-G	0.550	0.541	0.545	0.549	0.966	0.986	0.975	0.975	0.992	0.998	0.995	0.995
ARC-I-G	0.643	0.872	0.734	0.676	0.992	0.983	0.987	0.987	0.905	0.963	0.933	0.929
HAN-G	0.504	0.881	0.641	0.507	0.935	0.984	0.959	0.958	0.609	0.848	0.709	0.522
DSSM-D	0.852	0.763	0.805	0.815	0.933	0.984	0.958	0.957	0.841	0.959	0.896	0.949
ARC-I-D	0.841	0.763	0.800	0.809	0.987	0.985	0.986	0.986	0.967	0.958	0.962	0.983
HAN-D	0.709	0.919	0.801	0.771	0.875	0.859	0.866	0.873	0.946	0.996	0.970	0.975

Table 4: Comparison of different input aggregation techniques: (i) charTrigrams (T), (ii) GloVe Embeddings (G), and (iii) Doc2Vec embeddings (D), for simple models.

Model	AAN				WIKI				PAT			
	P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc
BERT-CLS	0.992	0.579	0.732	0.637	0.998	0.499	0.666	0.499	1	0.639	0.783	0.714
BERT-POOL	0.572	0.992	0.726	0.625	1	0.500	0.667	0.501	1	0.637	0.778	0.711
LONG-CLS	0.599	0.842	0.699	0.743	0.968	0.764	0.854	0.835	0.927	0.911	0.919	0.917
LONG-POOL	0.727	0.819	0.770	0.783	0.994	0.812	0.894	0.883	0.996	0.918	0.955	0.953

Table 5: Aggregation using the [CLS] token, and the pooler output [POOL] from BERT and Longformer for documents > 512 and > 4096 tokens for BERT and Longformer, respectively. The results were the same for SUM and AVG aggregation techniques.

- **Doc2Vec Embeddings (D):** Doc2Vec (Le and Mikolov, 2014) embeddings can be used to get vector representations for a document. We train Doc2Vec models from scratch on different datasets to get relevant document representations. These document representations are then given as input to different document matching models.

Table 4 shows the model performance of the simple models for the above three input representations. We observe that using GloVe (G) and Doc2Vec (D) input embeddings improve the model performance of the simple models overall.

A.8 Different Aggregation Techniques for Transformer Based Models

We experiment with different aggregation techniques (SUM and AVG) for > 512 tokens for BERT, and > 4096 tokens for Longformer. We

chunk the documents and aggregate the representations from the ‘[CLS]’ token and ‘the pooler output’. The results can be seen in Table 5 and were the same for SUM and AVG aggregation techniques. Aggregation resulted in an overall performance drop when compared to just truncating the documents up to 512 tokens and 4096 tokens for BERT and Longformers, respectively for the document matching.

Simple and Effective Multi-Token Completion from Masked Language Models

Oren Kalinsky *

Amazon

orenk@amazon.com

Alex Libov

Amazon

alibov@amazon.com

Guy Kushilevitz *

Amazon

guyk@amazon.com

Yoav Goldberg

Bar Ilan University

yoav.goldberg@gmail.com

Abstract

Pre-trained neural masked language models are often used for predicting a replacement token for a given sequence position, in a cloze-like task. However, this usage is restricted to predicting a single token, from a relatively small pre-trained vocabulary. Recent Sequence2Sequence pre-trained LMs like T5 do allow predicting multi-token completions, but are more expensive to train and run. We show that pre-trained masked language models can be adapted to produce multi-token completions, with only a modest addition to their parameter count. We propose two simple adaptation approaches, trading parameter counts for accuracy. The first method generates multi-token completions from a conditioned RNN. It has a very low parameter count and achieves competitive results. The second method is even simpler: it adds items corresponding to multi-token units to the output prediction matrix. While being higher in parameter count than the RNN method, it also surpasses current state-of-the-art multi-token completion models, including T5-3B, while being significantly more parameter efficient. We demonstrate that our approach is flexible to different vocabularies and domains and can effectively leverage existing pre-trained models available in different domains. Finally, a human evaluation further validates our results and shows that our solution regularly provides valid completions, as well as reasonable correctness for factual-sentence completions.

1 Introduction

Multi-Token-Completion (MTC) is the task of filling masked sentences with a sequence of tokens such that the completed sentence is probable and coherent. e.g., for the masked sentence "*The 46th president of the US, [MASK], was elected in 2020*"

a good completion would be any of "*Biden*", "*Joe Biden*", "*president Biden*" and more.

While Masked Language Models (MLMs) such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) successfully deal with a simpler variation of this task (single-token completion) these models were pre-trained on a limited vocabulary, not containing multi-word phrases. It is technically possible to complete numerous tokens simultaneously with MLMs by introducing a sequence of [MASK] tokens, but there are no clear methods of conditioning the tokens on each other, and, more importantly, the length of the completion needs to be pre-determined. Expanding MLMs' effective completion vocabulary – the actual vocabulary they are capable of completing well – will help a recent line of work using MLMs to extract knowledge/information from corpora (Jiang et al., 2020; Petroni et al., 2019; Kushilevitz et al., 2020).

Most existing works using MLMs to complete sentences either avoid the problem by limiting the completions to single tokens or use sub-optimal heuristics for MTC (e.g. presetting the length of the sequence and filling one token at a time). A trivial solution is to increase the number of tokens used in the pre-processing tokenization step, and also include multi-word phrases as tokens. However, apart from the longer tokenization time¹, the main problem with this approach is that changing the input level of the model requires adapting the weights of the full MLM to the new input.² A recent family of seq2seq pre-trained LMs, the prevalent of which is T5 (Raffel et al., 2020), are trained directly on the MTC task, and indeed perform quite well on it, especially with larger model sizes. However, the seq2seq objective, coupled with very high parameter counts, make such models expensive to train

¹Expanding BERT's tokenizer vocabulary from 30K to 100K phrases leads to a degradation of $\times 30$ in tokenization time. Further details are in Appendix A.

²This requires either full pre-training of the MLM or long, end-to-end fine-tuning using a considerable amount of data.

* Authors contributed equally.

and to run inference on, compared to MLMs.

In this work, we demonstrate how pre-trained MLMs can be *adapted* to produce multi-token completions from (large) fixed vocabularies, using a self-supervised training objective and only a modest parameter count. Specifically, we demonstrate an effective adaptation of pre-trained MLMs to predict, on top of their pre-trained vocabularies, additional $\sim 100K$ noun-phrases (NP-chunks) and entities ranging in length from 1 to 10 tokens. The only requirement is a textual corpus in which each of the desired phrases appears over k times (we used $k = 50$ in this work). The adapted model surpasses the accuracy of T5-3B predictions, while using a fraction of the parameter count and being significantly more efficient to run. Apart from evaluating with automatic measures, we also use human-annotators to explore different aspects of the proposed completions.

After experimenting on the prediction of general-purpose phrases, we also show our methods can be used to predict phrases in a specific domain. This is a significant benefit of the adaptation approach since the completion vocabularies can be tailored to a specific project's needs, using domain-specific pre-trained MLMs, even in domains where huge T5-like models are not available³.

We use any pre-trained MLM for MTC by extracting the informative representations the MLM uses for completing a single token, and feeding them into a small and simple model that chooses appropriate multi-token completions. We offer two different completion models. Depending on the scenario, both are useful; the first solution, which expands the MLM's decoder matrix, achieves SOTA accuracy. The second solution, using a small iterative generation model, is well suited for large completion-vocabularies while achieving competitive results.

The core reason for the success of our methods, despite being trained on less data and for less time than other solutions, is the fact that the MLM's pre-training is well suited for the MTC task. We provide two types of evidence to support this claim: In section 2 we present an experiment suggesting that MLMs incorporate information regarding

³For example, an e-commerce company could adapt a pre-trained general purpose MLM to complete multi-token product names, while a biomedical researcher could adapt a pre-trained biomedical MLM such as SciBERT (Beltagy et al., 2019) or BioBERT (Lee et al., 2019) to complete multi-token drug or disease names.

multi token phrases. In section 6, we show that pre-training on in-domain data helps our methods perform better. Both of these signals indicate that, as expected, the MLM pre-training captures information required for MTC. Therefore, what is left for our MTC adaptation models is only to extract this information from the MLM, an easier task that does not require a long training with a vast amount of data.

Our main contributions are MTC datasets (general purpose, PubMed-based, and a $3K$ dataset with human labeling), demonstrating that MLMs learn the meaning of Multi Token phrases, and two methods for adapting any pre-trained MLMs for MTC. We publish⁴ our datasets, models and code⁵.

2 MLMs Learn Multi-Token Phrases

Our work relies on the assumption that MLMs are capable of holding information about multi-token phrases. Despite being a common belief and an essential property for the MLMs to have in order to be as successful as they are, to the best of our knowledge this was not shown explicitly. How do we show that a token-completion MLM is encoding the semantics of multi-token phrases? Looking at completions will fail, because we cannot directly probe for MTC. Instead, we came up with the following experimental design, which measures the encoding of multi-token phrases indirectly by looking at masked positions influenced by them.

We use multi-token phrases that have single token synonyms. We construct a dataset containing quadruples, each quadruple consisting of a multi-token phrase (e.g. "*New York city*"), a single-token synonym (e.g. "*NYC*"), a phrase similar in meaning to the synonyms (e.g. "*Chicago*") and a random phrase (e.g. "*Dog*"). We collect sentences containing the multi-token phrases, and in each sentence mask out an NP-chunk different from the multi-token phrase (e.g. "*[MASK] is in New York city*"). Next, we replace each multi-token phrase with each of its other corresponding quadruple phrases, forming four similar masked sentences, each containing a different quadruple phrase (e.g. "*[MASK] is in NYC*").

Finally, we ask an MLM to complete the missing token in each masked sentence, and compare the suggested completions. If the MLM is aware that

⁴<https://registry.opendata.aws/multi-token-completion/>

⁵<https://github.com/amzn/amazon-multi-token-completion/>

the multi-token phrase is semantically similar to the single-token synonym, we expect it to suggest similar completions regardless of which of the two it sees in the sentence. Indeed, the model treats the multi-token phrase similar to the single-token synonym; We use a similarity measure comparing the MLM’s suggested completions for the multi-token phrase sentences and each of the other types of sentences. We find that the average similarity between multi-token phrase sentences and single-token synonym sentences is 0.76, while for similar-phrase sentences it is 0.71 and for random-phrase sentences it is 0.63. Our main conclusion from this experiment is that, as expected, MLMs learn semantic meaning of multi-token phrases. Full experiment details (including the similarity measure used) are in Appendix B.

3 Masked Language Modeling Data

In absence of a known dataset for the MTC task, we curate one. We follow (Devlin et al., 2018) and use data from two sources: Wikipedia articles and the Books corpus (Zhu et al., 2015).

Completion Vocabulary. We start by building a vocabulary of phrases. As we aim for general-purpose MTC, and following (Trask et al., 2015) by considering phrases as NP-chunks or entities, we simply focus on phrases that appear frequently in the corpus. We use spacy (Honnibal and Montani, 2017) to extract 64M unique NP-chunks and entities. We keep phrases appearing 500 times or more in the corpus, leaving us with $\sim 93K$ phrases. Only 10.2% are single tokens using BERT’s cased tokenizer, indicating the importance of MTC. 52.9% of the phrases consist of 2 tokens, 36.9% consist of 3 or more. 53% of the phrases are single-words, 47% span two words or more. Further statistics regarding the number of words and the number of tokens assembling the phrases selected are reported in Table 1.

Masked sentences. We split the corpus using spacy’s sentencer. We sample 50 unique sentences containing each vocabulary-phrase. We eliminate recurring sentences and mask out the vocabulary phrase to form masked sentences, using the masked span as the label. We randomly split the data into train (90%), validation (5%) and test (5%) sets.

4 Adapting MLMs for MTC

We propose two simple yet effective solutions, capable of integrating with any pretrained MLM and

#Tokens	%Phrases	#Words	%Phrases
1	$\sim 10.2\%$	1	$\sim 53.3\%$
2	$\sim 52.9\%$	2	$\sim 38.3\%$
3	$\sim 23\%$	3	$\sim 5.4\%$
4	$\sim 9.5\%$	4	$\sim 2.1\%$
5	$\sim 2.8\%$	5	$\sim 0.5\%$
> 5	$\sim 1.6\%$	> 5	$\sim 0.4\%$

Table 1: **Expanded completion vocabulary.** Number of tokens and words assembling the phrases collected.

extending its completion vocabulary to perform MTC. Both solutions utilize the MLM by using the contextual embedding vector of the masked place, which was originally pre-trained to be relevant for completion tasks. Instead of using it to choose the appropriate single token replacement (see Figure 1 (1)) like the MLM, we use it as an input to extension models tuned for completing out of an expanded vocabulary. We train only the relatively small extensions and use the pre-trained MLMs as is, allowing short and effective training.

4.1 Extended-Matrix (EMAT) decoder

Our first extension model is based on extending the decoder matrix of the MLM to include the new multi-token phrases in the completion vocabulary. We assign each new multi-token phrase to an embedding vector, which is added only to the output token-prediction matrix. This is in contrast with simply adding it to the base-model’s vocabulary, which results in longer training due to the full model being affected and in increased tokenization time (see Appendix A for tokenization time evaluation). An illustration of the architecture can be found in Figure 1 (2), where we use the MLM to compute the contextual embedding of the masked token and feed it to the extended-matrix decoder. Even though each multi-token phrase is assigned with its own embedding vector, this solution allows us to adapt MLMs trained with a smaller vocabulary (with all mentioned advantages this comes with) to complete phrases that are multi-token phrases in their *original* vocabulary.

During training, we extract the contextual embedding from the masked sentence and only train the extended-matrix decoder yielding a quick training phase. We train the decoder over the train set described in Section 3, expanding the completion vocabulary to $\sim 93K$ phrases. When BERT-base is the base MLM, this approach adds 138M parameters, considerably less than models comparable in performance such as T5-3B (3B parameters).

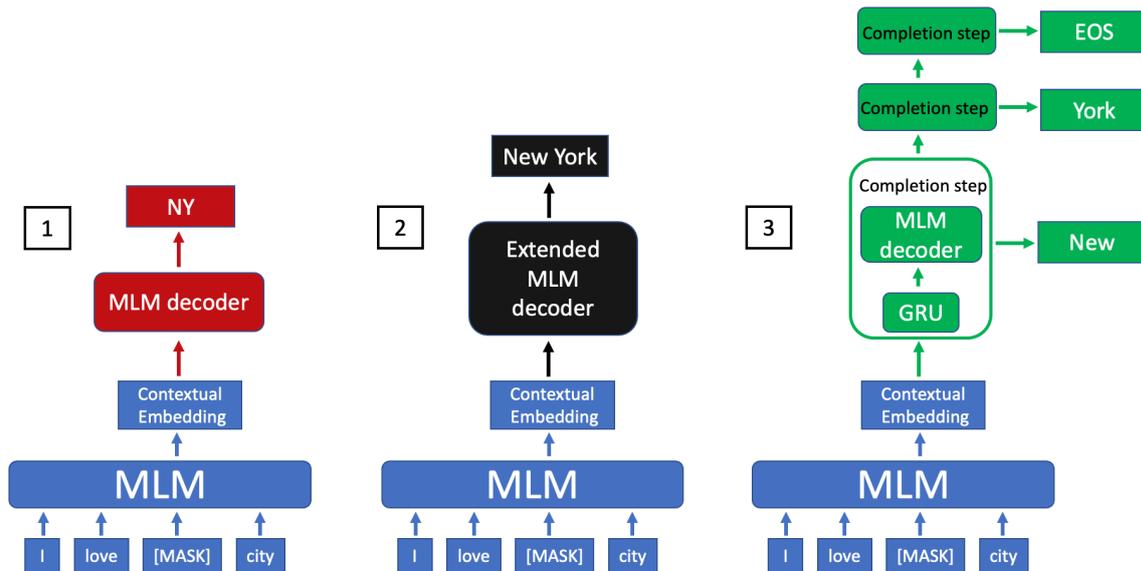


Figure 1: **Architectures.** The MLM body is shared across architectures. It outputs a contextual embedding for the [MASK] token, used by all decoding solutions. The MLM (1) predicts a single token from its vocabulary. EMAT (2) uses an extended decoder to predict a phrase from the extended vocabulary. The Generative extension (3) completes tokens until reaching [EOS]. Some arrows and layers are deducted for readability.

4.2 RNN decoder

Our second model is based on an RNN decoder, specifically a GRU (Cho et al., 2014). We use it to replace the MLM’s matrix decoder.

To train our GRU decoder for MTC, similarly to our first solution, we utilize the pretrained MLM by extracting the embedding vector corresponding with the masked token for each sentence. For the GRU-based solution we use this vector as the first hidden vector fed into the GRU. As the first input to the GRU, we wish to provide the context of the multi-token phrase to be generated and thus feed the static embedding of the token *preceding* the [MASK] token. For later steps, the input is the previous token completed by the generative model. The inputs to the GRU are depicted in Figure 1 (3), where the contextual embedding of [MASK] is fed as the first hidden state and the static embedding of ‘love’ is the first input of the GRU.

Next, at each step we concatenate: a) the output of the GRU; b) the previous vector embedding; and c) the masked token contextual embedding vector from the pretrained MLM, and feed them to a feed forward (FF) layer in order to reduce the dimension. We find that providing these three contexts to the decoder improves its accuracy. The output of the FF is then fed as input to the MLM decoder (For BERT, this is a FF layer followed by an embedding layer and a SoftMax layer) to obtain the

token predicted for this step of the autoregressive generation. We train the model to complete the next token of the missing multi-token phrase using a standard back propagation approach with Cross Entropy loss. We utilize the dev set to tune the batch size, number of GRU layers, teacher forcing rate, and learning rate (used parameters can be found in appendix C).

We find that initializing the GRU’s decoder embeddings with the same parameters used by the MLM we are plugging into, improves the performance. Also, it helps to pre-train the generative model on a vanilla Language Modeling task before starting MTC training. The idea is to give the fresh extension model some sense of the distribution of the language before the training on the MTC task. An ablation study demonstrating the benefit of these components is shown in Appendix C.

Our generative model consists of only 32M parameters when using BERT as a base model, which are the only trained parameters used for our solution (i.e. the pretrained base MLM is not fine-tuned) and independent of the vocabulary size. During inference, we simply use the MLM to extract the embedding vector corresponding with the masked token and feed it, along with the static embedding of the preceding token, as input to the trained GRU model. Next, we use beam-search to generate sequences until reaching an EOS token.

Model	Coverage	Size	Data size	$a@1$	$a@3$	$a@5$	$a@10$	$a@50$
BERT-cased	13.8%	110M	16G	13.5%/1.8%	21.8%/3.0%	25.9%/3.5%	31.2%/4.3%	43.1%/5.9%
RoBERTa	15.6%	110M	160G	19.3% /3.0%	30.6% /4.7%	36.1% /5.6%	43.2% /6.7%	59.4% /9.2%
Naive-MTC-BERT	66.1%	220M	16G	6.6%/ 4.3%	11.9%/ 7.8%	14.9%/ 9.8%	19.5%/ 12.8%	32.3%/ 21.3%

Table 2: **MLMs performing limited completion.** Coverage is the % of sentences from the test set that the model is capable of completing. MLMs are limited to single token completions. For computational reasons, we allow the Naive-MTC-BERT to generate only single tokens or two-token phrases. Accuracy is reported as x/y ; x is the accuracy out of the specific model’s limited coverage and y is the accuracy out of the full test set ($coverage * x$).

5 Results

Setup. Measuring success in completion tasks is not trivial since a masked-sentence can have many suitable replacements. In many cases returning the expected phrase in a top-k place, and not necessarily first, is acceptable. Hence, we measure $accuracy@k$ (sklearn): the percentage of masked sentences where the label is among the top-k predictions. To better ground our performance expectation from MTC, we first evaluate the accuracy of an easier completion task; Single Token Completion. We go on to evaluate Naive MTC using MLMs. As expected, results are not sufficient. Finally, we evaluate our methods, adapting MLMs for MTC.

5.1 Single Token Completion

Single Token Completion is an easier⁶ task than MTC, and MLMs like BERT and RoBERTa are trained directly to perform it. Therefore, results of these MLMs on the single token completion task can be considered as an upper limit for the performance of solutions adapting these models for MTC. We test BERT and RoBERTa on the single token completion task, results of these models on the test set are reported in Table 2.

5.2 Naive MTC with MLMs

A naive way to utilize MLMs for MTC is using a two-step method: given a masked sentence, first predict the number of missing tokens and then duplicate the masked token to the predicted number (e.g. "The US state [MASK]" becomes "The US state [MASK] [MASK]") and complete them using the MLM. We train a BERT-based classification model predicting the number of missing tokens by assigning a label to the data: for each masked sentence, the label assigned is the number of tokens the masked span splits into using BERT’s

⁶For example, BERT’s search-space size is $\sim 30K$ while some MTC solutions (e.g. our generation plugin and T5) have an infinite search space.

tokenizer. Due to computational limits of the generation phase, we are only interested in 3 classes: a single token missing, two tokens missing, and 3 or more tokens missing. The class distribution is (10%, 53%, 37%), respectively. We use BERT’s default hyper-parameters and train a classification model. This is a task with ambiguous labeling⁷, thus we do not expect high accuracy results. The model reaches 64.7% accuracy on the test set.

During inference, we use the model’s predictions with a SoftMax function to acquire a probability for a single token mask-replacement and a double token mask-replacement. We utilize these probabilities as follows. For each specific single token, we compute a replacement probability by simply multiplying the probability for any single token replacement by the probability BERT assigns the specific token to replace the mask in the masked sentence. For a double token term we multiply the probability of having a double token replacement by the probability BERT predicts for the specific term, estimated with a standard generating heuristic. After duplicating the mask, we complete the first missing token using the MLM. Then, we replace the first mask with each of the top-100 predicted tokens and complete the second mask. Finally, the probability for each double token term is the product of the probabilities the two tokens assembling it to replace the two missing [MASK] tokens.

Results on the test set are reported in Table 2. Even when considering only sentences where the missing phrase consists of one or two tokens, results are not sufficient and call for a different way to use MLMs for MTC. This is likely due to the fact that the first token replacement is computed in an unconditional manner to the second one.

5.3 Multi Token Completion

We use BERT, RoBERTa and SpanBERT as MLMs adapted for MTC with our methods described in section 4. MTC models are capable of completing

⁷e.g. for "The US state [MASK]" "New York" and "Texas" are adequate replacements resulting in different labels.

Model	Size	Data Size	Inf-time	$a@1$	$a@3$	$a@5$	$a@10$	$a@50$
ILM (GPT-2)	124M	1G/41G	120ms	1.3%	2.7%	3.7%	5.3%	10.8%
T5-base	220M	700G	224ms	5.2%	8.3%	9.7%	11.4%	15.6%
T5-3B	3B	700G	802ms	11.6%	18.5%	21.5%	25.4%	NA
BERT-Adapted-RNN	32M/142M	1G/17G	44ms	9.24%	14.64%	17.39%	21.32%	31.28%
BERT-Adapted-EMAT	138M/248M	1G/17G	15ms	12.64%	20.48%	24.63%	30.65%	45.85%
ROBERTA-Adapted-RNN	48M/158M	1G/161G	59ms	8.14%	12.99%	15.51%	19.07%	28.40%
ROBERTA-Adapted-EMAT	161M/271M	160G/161G	16ms	11.59%	19.06%	23.18%	29.27%	45.79%
SPAN-BERT-Adapted-RNN	32M/142M	1G/17G	46ms	8.29%	12.82%	15.07%	18.34%	26.334%
SPAN-BERT-Adapted-EMAT	138M/248M	1G/17G	15ms	6.91%	11.6%	14.12%	17.93%	27.45%

Table 3: **MTC results.** Models are described in Section 5.3. Adapted models are trained with our solutions as described in Section 4. Size is the number of parameters; Data size is the amount of data seen during training. For adapted models we report both of these numbers excluding/including the base MLM. Inf-time is the average inference time of a single sentence, measured using an Nvidia T4 GPU. T5-3B is too large for a beam of size > 10 .

a phrase of any length hence have a coverage of 100% on the test set. Apart from our methods, we also report on several baselines.

ILM (Donahue et al., 2020) is a GPT-2 based framework for infilling, a task similar to MTC. The LM is shown a sentence with missing places and is trained to generate text probable of replacing the missing parts. A crucial observation is that GPT-2 is an LM and not an MLM. Therefore, ILM’s pretraining seems less appropriate for completing relatively short phrases (a task similar to the MLM objective), and more appropriate for longer generation tasks (a task more similar to the LM objective). We fine-tune ILM on our dataset for a single epoch, taking ~ 3.5 days using an Nvidia T4 GPU.

T5(Raffel et al., 2020) is a transformer model pretrained on the MTC task. It is a strong baseline reaching SOTA on many NLP tasks. We report on two versions, T5-base and T5-3B, both trained on a dataset 1-2 order of magnitudes larger than ours.

Results and Discussion. Results are reported in Table 3. Our extended matrix (EMAT) solution performs best, even when compared to the huge T5-3B. The RNN plugin is slightly inferior, but still performs better than existing solutions comparable in size. Note that even though the size of the EMAT plugin is $\sim \times 4$ of the size of the RNN plugin, inference time for the tested vocabulary is shorter because of GPU optimizations and the fact that the RNN model may run several times (~ 3 in average on the test set) for each completion. However, increasing the completion vocabulary size will not affect the GRU size but will increase the size of the matrix extension, making the RNN plugin a better fit for large completion vocabularies⁸.

⁸For a completion vocabulary with 1M phrases and BERT as the MLM, the size of the RNN plugin remains 32M, while the size of the EMAT plugin grows to 769M.

6 Domain Specific MTC

We investigate MTC in a specific domain.

Datasets. We chose the Biology Domain, due to the availability of data and models. Using PubMed (pubmed) abstracts as a corpus, we extract two MTC datasets. Similarly to section 3, we first construct the completion vocabularies.

Key-phrase vocabulary: we use $\sim 20K$ MeSH vocabulary⁹ phrases appearing frequently as key-phrases in pubmed papers. This assures us phrases from the Biology domain. We discard phrases appearing less than 50 times as NP-chunks or entities in the corpus, leaving us with $\sim 12.5K$ phrases.

Frequent-phrases vocabulary: Similarly to section 3, we extract a vocabulary of phrases appearing more than 500 times as NP-chunks or entities in the corpus ($\sim 144K$ phrases). To make sure we are considering mostly phrases from the Biology domain, we discard phrases appearing in our general purpose vocabulary (described in section 3), leaving us with $\sim 118K$ phrases.

Finally, for both vocabularies we extract 50 sentences containing each of the vocabulary phrases, mask the phrase out in each sentence and split the data into train, development and test sets.

Base Models. To evaluate the impact that the domain-specific pre-training has on the performance of our methods, we test our adaptation methods on top of models pretrained on different datasets. *SciBERT* (Beltagy et al., 2019) is a BERT-like model trained on the Semantic Scholar data - data from the scientific domain, closer to the biology domain than the general purpose BERT. *BioBERT* (Lee et al., 2019) is a BERT-like model trained on PubMed abstracts.

⁹MeSH (Medical Subject Headings) is NLM’s controlled vocabulary of biomedical terms used to describe the subject of each journal article in MEDLINE.

Model	Dataset	$a@1$	$a@3$	$a@5$	$a@10$	$a@50$
T5-base	<i>key-phrases</i>	3.6%	5.2%	5.8%	6.7%	8.4%
T5-3B	<i>key-phrases</i>	10.3%	15.1%	16.9%	19.5%	NA
BERT-Adapted-EMAT	<i>key-phrases</i>	12.9%	20.7%	24.9%	31.1%	48.8%
SciBERT-Adapted-EMAT	<i>key-phrases</i>	13.4%	21.2%	25.2%	30.8%	44.9%
BioBERT-Adapted-EMAT	<i>key-phrases</i>	16.3%	26.9%	32.1%	39.5%	57.2%
T5-base	<i>NP-chunks and Entities</i>	3.9%	5.7%	6.6%	7.7%	10.5%
T5-3B	<i>NP-chunks and Entities</i>	8.2%	12.8%	15.0%	18.1%	NA
BERT-Adapted-EMAT	<i>NP-chunks and Entities</i>	7.1%	11.8%	14.4%	18.5%	30.5%
SciBERT-Adapted-EMAT	<i>NP-chunks and Entities</i>	8.2%	14.2%	17.5%	22.5%	35.4%
BioBERT-Adapted-EMAT	<i>NP-chunks and Entities</i>	8.0%	13.5%	16.4%	20.7%	31.77%

Table 4: MTC on the pubmed test sets. T5-3B is too large for a beam of size > 10 .

Results are reported in table 4. For brevity, we compare our Extended Matrix method, tuned on three base models (BERT, SciBERT, BioBERT), to the two primary baselines: T5-base and T5-3B. We report on the two PubMed datasets we curated.

Conclusions. First, our method outperforms the strong T5-3B baseline in the domain-specific scenario as well, showing MLMs can be effectively adapted for domain-specific MTC. Second, the success of *BioBERT* and *SciBERT* suggests that the pre-training of the MLM is in fact utilized by our methods (i.e. there is no catastrophic forgetting). Last, the *key-phrases* dataset is easier than the frequent *NP-chunks and Entities* one. This is likely due to the *key-phrases* dataset mostly containing phrases that are more frequent and significant.

7 Human Evaluation

Completion tasks are difficult to evaluate since a sentence can have many different valid completion options including, but not limited to, the original masked span. We use $accuracy@k$ to deal with this issue, but in some use-cases only the first completion is important. Thus, we define a manual task to more accurately evaluate sentence completions.

Human-annotators are presented with the masked sentence, the original masked span and the *first* completion suggested by each of the methods we evaluate (full annotation task is in Appendix D). Annotators were not aware which model provided each completion, and the completions were presented in a random order for each sentence. The sentences are divided between three expert English-speaking annotators. Prior to performing the annotations, the annotators met for a calibration session, each annotating the same 50 sentences and discussing results until reaching high agreement. We sample 750 sentences from the test set described in section 3. Each expert annotator is assigned with

250 sentences and the completions of each of the four methods, amounting to 1000 samples each.

First, we ask annotators whether the completions are grammatically correct and make sense in the context of the sentence, *regardless of whether they are factually correct* (denoted as a "Valid" completions). For example, any year or location is a valid completion for "*Jules Verne was born in [MASK]*".

In some cases, valid completions can be general and uninformative. For example, in the sentence "*On January 1, 1998, [MASK] was released publicly online as SGI freeware.*" the original span is "*Blender*". While any SGI freeware name would make sense, the word "*it*" (suggested by T5-3B) is also a valid completion. This happens with general words like "*he*", "*she*", or "*person*" but can also occur with entities, e.g. in the sentence "*Jules Verne was born in [MASK]*", the original masked span is "*Nantes*" and a suggested completion might be "*Europe*". We ask annotators to flag these cases, where the suggested completion is more general (less specific) than the original span.

Factual Correctness of completions is also important, since some works use completion methods for knowledge extraction (Petroni et al., 2019; Jiang et al., 2020). To evaluate this, we first ask annotators to mark whether the masked span is a part of a specific fact. For example, "*MLK, (born [MASK])*" should be marked as a fact, while "*he died aged [MASK]*" shouldn't. We find that 50.93% of the masked sentences are part of a fact¹⁰. For the factual sentences, we ask the annotators to label whether the proposed completion is factually correct. If they do not know, annotators are instructed to use the web to verify the correctness.

Results are reported in Table 5. As expected, the actual valid-completion percentage of the

¹⁰Recall that the evaluation data originates from both Wikipedia and books.

Model	Valid %	Valid Specific%	Correct %	Correct Specific %
ILM	52.2%	47.8%	9.4%	7.5%
T5-base	66.0%	49.0%	29.8%	18.5%
T5-3B	81.8%	66.8%	40.8%*	30.1%
BERT-Adapted-EMAT	84.1%	80.4%*	32.9%	31.6%

Table 5: **Human evaluation results.** Valid completions are ones that are grammatically correct and make sense. Specific completions are completions that were not annotated to be more general than the original masked span. Correct is the percentage of sentences labeled as facts that are valid completions and are also factually correct. * marks that the difference between best and second best is statistically significant.

first completions is much higher than the *accuracy@1* measured. An important observation is that while checking for valid completions using human-evaluation is obviously a more accurate measure than *accuracy@k*, the two are correlated. Models that perform well in one measure, do so also in the other. This means *accuracy@k* is a good proxy for the actual valid-completion measure. Specifically, our method performs impressively when annotating for validness as well, slightly better than the much larger T5-3B. Finally, it seems general purpose models like T5 tend to complete general phrases more than our method. This is likely due to the fact that during our MTC training, the model sees the same amount of examples for each phrase. Other methods see more examples of general phrases, since these are more common in the language.

As for factual correctness; while T5-3B completes more facts correctly than our methods, when eliminating cases where the completion is general¹¹ our method performs better. This is important since general completions are not as interesting and can be more easily acquired. It can be especially crucial for knowledge extraction methods using these models. For example, a method trying to extract presidents with the sentence “*US presidents such as [MASK]*” would benefit from completions like “*Obama*” or “*Trump*” and not “*the president*” or “*this person*” which are correct, but uninformative.

8 Related Work

Transformer-based LMs and MLMs (Peters et al., 2018; Devlin et al., 2018) have revolutionized NLP in the past couple of years. While most of the impact has been achieved using these pretrained

¹¹For example, for “Garson accepted the role, winning [MASK].” original span is “the Academy Award for Best Actress”. T5 completes “an academy award”- factually correct, but more general than the original span. Our method tries to pinpoint a specific award - a harder task. It completes “the Academy Award for Best Picture” which is factually wrong.

models as a source of meaningful contextual embeddings, recent works are using these models for the task they were pretrained for: Masked Language Modeling (Petroni et al., 2019; Kushilevitz et al., 2020; Lazar et al., 2021; Shani et al., 2021; Jiang et al., 2020).

While most MLMs are capable of completing multiple missing tokens simultaneously, they do so in an unconditional manner, yielding unsatisfying results. Therefore, some works using MLMs for Masked Language Modeling simply restrict themselves to single token completions (Petroni et al., 2019) while others use heuristics in order to generate multi token completions (Lazar et al., 2021; Jiang et al., 2020).

Some attempts towards making MLMs predict multi token units better by changing the masking technique during training include masking spans instead of tokens (Joshi et al., 2020) and masking whole words (Cui et al., 2019), but these methods still complete in an unconditioned manner during inference. XL-net (Yang et al., 2019) avoids interdependence between masked tokens using a Permutation Language Modeling pretraining objective.

Similarly to our solution, extending the decoder matrix with n-gram vocabulary items is done also by Xiao et al. (2020), but for a different purpose: they use them as an auxiliary signal during training, in order to improve BERT’s single-token pretraining. They mask n-grams, and predict a specific n-gram assigned id *together with the single tokens assembling the n-gram*, as to improve the single-token embeddings. The BERT model is pretrained end to end and the n-gram embeddings are discarded after pretraining and are not part of their final model used during fine-tuning and inference. While we share their extension of the decoder matrix, the MTC task adds additional requirements: to allow for easy MTC vocabulary support, we aim for a short MTC training and inference, building on the preexisting pretrained MLMs and avoiding

the need to pretrain from scratch. In addition, our solution does not distinguish between multi-token and single token-phrases during training since both are part of the completion vocabulary.

In Donahue et al. (2020), the ILM framework is introduced. This is a framework for infilling, a task similar to MTC. ILM is a GPT-2 (Radford et al., 2018) based solution, meaning the pre-trained model used is a LM and not an MLM, requiring it to adapt to a different task. Finally, T5 (Raffel et al., 2020) was pre-trained for the MTC task using a vast amount of data. It performs well but requires a lot of training data, time and memory, does not utilize existing MLMs and is not available in many domains and languages.

9 Conclusions

We show MLMs can be adapted for multi token completion even though they were trained for single token completion. We presented two simple but effective solutions that leverage the pretrained MLMs and offer quick adaptations to new vocabularies. The two solutions are trading-off performance and size: 1) an extended matrix decoder offering SOTA accuracy but size-dependent on the completion vocabulary; 2) an RNN decoder with slightly lower accuracy but size independent of the vocabulary size. We also demonstrate the flexibility of our approach to different vocabularies and domains by evaluating it on the PubMed dataset and showing that leveraging domain-pretrained MLMs offers significant accuracy improvement. Finally, we validate our results by conducting a human evaluation to account for valid completions that are not measured using our automatic metric. It shows that our extended matrix solution provides valid completions in 84% of the times, and can also correctly handle facts in 30% of the times, comparatively to the much larger T5-3B.

10 Limitations

The main limitation of our work is that it requires a fixed and pre-determined completion vocabulary. We acknowledge that this is a burden, and in some cases such a vocabulary might not be available. We believe a solution for adapting MLMs for MTC without such a prerequisite is feasible, and this is a goal for future work.

11 Ethical Considerations

Human Evaluation. In terms of fair-pay, the payment to the expert annotators was above the minimum wage in the US. Consent was given from the annotators to use their annotations and release them as part of this research. The annotators were not aware which solution generated each completion and the completions were presented in a random order as to avoid bias based on the order. To achieve high quality results, the annotators had a calibration session as to better understand the guidelines and requirements described in Appendix D.

Environmental. Compared to the massively large language models such as T5-3B, our models are lightweight and can be run on smaller more energy efficient hardware such as a CPU. In addition, this becomes more apparent when considering the superior inference-latency of our solutions. Since energy is composed of both time and power-consumption, our lightweight models should waste significantly less energy during inference.

Biases and Exclusion. The proposed models depend on a fixed and pre-determined vocabulary of potential multi-token completions, and the choice of this set in itself may result in omissions, exclusions under-representation of some groups or concepts, and over-representation of others. Care should be taken to select a set that alleviate such biases to the extent possible. Also after the selection of the set, the algorithm does not guarantee balanced, fair or unbiased selections of candidate completions. Users should be aware of this when designing algorithms whose predictions may influence certain groups.

12 Acknowledgements

The work of Yoav Goldberg was supported by funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme, grant agreement No. 802774 (iEXTRACT).

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. *Scibert: A pretrained language model for scientific text*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3613–3618. Association for Computational Linguistics.

- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. [On the properties of neural machine translation: Encoder-decoder approaches](#). In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111. Association for Computational Linguistics.
- Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. [Pre-training with whole word masking for chinese BERT](#). *CoRR*, abs/1906.08101.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805.
- Lee Raymond Dice. 1945. [Measures of the amount of ecologic association between species](#). *Ecology*, 26(3):297–302.
- Chris Donahue, Mina Lee, and Percy Liang. 2020. [Enabling language models to fill in the blanks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2492–2501. Association for Computational Linguistics.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020. [X-FACTR: multilingual factual knowledge retrieval from pre-trained language models](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5943–5959. Association for Computational Linguistics.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. [Spanbert: Improving pre-training by representing and predicting spans](#). *Trans. Assoc. Comput. Linguistics*, 8:64–77.
- Guy Kushilevitz, Shaul Markovitch, and Yoav Goldberg. 2020. [A two-stage masked LM method for term set expansion](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6829–6835. Association for Computational Linguistics.
- Koren Lazar, Benny Saret, Asaf Yehudai, Wayne Horowitz, Nathan Wasserman, and Gabriel Stanovsky. 2021. [Filling the gaps in ancient akkadian texts: A masked language modelling approach](#). *CoRR*, abs/2109.04513.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. [Biobert: a pre-trained biomedical language representation model for biomedical text mining](#). *CoRR*, abs/1901.08746.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- George A. Miller. 1995. [Wordnet: A lexical database for english](#). *Commun. ACM*, 38(11):39–41.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. [Language models as knowledge bases?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.
- pubmed. [pubmed](#).
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. [Language models are unsupervised multitask learners](#). *none*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Chen Shani, Nadav Borenstein, and Dafna Shahaf. 2021. [How did this get funded?! automatically identifying quirky scientific achievements](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 14–28. Association for Computational Linguistics.
- sklearn. [accuracy at k](#).
- T. Sørensen. 1948. [A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons](#). Biologiske skrifter. I kommission hos E. Munksgaard.

- Andrew Trask, Phil Michalak, and John Liu. 2015. [sense2vec - A fast and accurate method for word sense disambiguation in neural word embeddings](#). *CoRR*, abs/1511.06388.
- Dongling Xiao, Yu-Kun Li, Han Zhang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. [Ernie-gram: Pre-training with explicitly n-gram masked language modeling for natural language understanding](#). *CoRR*, abs/2010.12148.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. [Aligning books and movies: Towards story-like visual explanations by watching movies and reading books](#). In *The IEEE International Conference on Computer Vision (ICCV)*.

Multi	Single	Similar	Random
new york city	nyc	chicago	disco
fish tank	aquarium	zoo	lena
every week	weekly	monthly	fruit
extort	blackmail	scam	cat
barman	bartender	waitress	mckenzie

Table 6: Experiment data sample. **Multi** is the multi-token phrase, **Single** is the single-token synonym, **Similar** is a term close in the sense2vec space and **Random** is a random single-token word.

A Tokenizing an expanded vocabulary

We experiment to see how tokenizing time is affected by larger vocabularies. We find that including multi word phrases as tokens actually has a worse effect than just adding more tokens, since the tokenizer cannot assume a word is the maximal span for each token. We sample 10K random sentences from Wikipedia and use huggingface’s¹² implementation for BERT’s standard cased tokenizer. When using the original vocabulary (of size $\sim 30K$) the tokenization takes 4.23 seconds. When expanding the vocabulary to our collected vocabulary (of size $\sim 100K$, including multi word phrases) the tokenization time jumps to 119.26 seconds using the same machine.

B MLMs Learn Multi-Token Phrases

In section 2, we report an experiment illustrating that MLMs are capable of treating multi-token phrases properly. The purpose and outline of the experiment are described in 2, further details are provided in this Appendix. BERT-base is the MLM used.

B.1 Data collection

We curate a dataset of multi-token phrases that have single-token synonyms. We start from Wordnet (Miller, 1995) synonyms, keep only synonyms where one phrase is a single-token and the other is a multi-token phrase using BERT’s tokenizer¹³, and manually select synonyms which are interchangeable. We collect 100 such synonyms. To each pair of synonyms we add a similar phrase chosen as the phrase closest to the single-token synonym

¹²https://huggingface.co/transformers/model_doc/bert.html#berttokenizer. We used the transformers package version 4.12.3.

¹³Multi-token phrases are not necessarily multi-word phrases. Uncommon words are also split to multiple tokens.

in the sense2vec¹⁴ vector-space which is manually verified as not a synonym of the pair, and not slang or an inappropriate phrase¹⁵. Finally, to each triplet we add a random single-token word forming a quadruple. A sample of the dataset is shown in Table 6.

B.2 Experimental setup

For each quadruple in our dataset, we conduct the following experiment: We collect k sentences containing the multi-token phrase from Wikipedia. For example, given the quadruple ("*fish tank*", "*aquarium*", "*zoo*", "*lena*"), the multi-token phrase is "*fish tank*" and one such sentence is "*nemo is placed in a fish tank in a dentist’s office.*". We use each of the collected sentences to compute a similarity score between the phrases as follows.

Masked sentence per quadruple term. We first form a masked sentence for the multi-token synonym, by masking out a random NP-chunk which is not the multi-token phrase (e.g. "*[MASK] is placed in a fish tank in a dentist’s office.*"). Then, we form a masked sentence for each of the other quadruple terms by replacing the multi-token phrase with it (e.g. for the phrase "*aquarium*" we form the sentence "*[MASK] is placed in a aquarium in a dentist’s office.*")

Similarity between masked sentences. We use the MLM to compute similarity between masked sentences. Following (Kushilevitz et al., 2020), for two masked sentences (m_i, m_j) we query the MLM to complete the mask in each and compare the predicted completions using the Sørensen-Dice coefficient (Dice, 1945; Sørensen, 1948):

$$\text{sim}(m_i, m_j) = \frac{|top_q(MLM(m_i)) \cap top_q(MLM(m_j))|}{q}$$

Where $MLM(m)$ is the list of tokens proposed by the MLM to complete the mask in the masked-sentence m , ranked by their probability. $top_q(MLM(m_i))$ is the top q tokens in the list (q being a parameter). An example for the similarity measure process for a single sentence is shown in Table 7.

¹⁴Sense2vec (Trask et al., 2015) is a twist on the word2vec algorithm.

¹⁵Sense2vec is trained on data from Reddit, which yields phrases that are slang or inappropriate. These will probably not be significantly found in BERT’s training data (Wikipedia and Books) and are therefore filtered out for this experiment.

Phrase type	Phrase	Masked sentence	MLM-suggestions	Intersection	Similarity
Multi-token synonym	fish tank	"a glass fish tank is sufficient for keeping [MASK]."	1.'fish', 2.'water', 3.'ducks', 4.'trout', ..., 8.'salmon', ... , 13.'eels', ..., 17.'turtles', ...	50	1
Single-token synonym	aquarium	"a glass aquarium is sufficient for keeping [MASK]."	1.'animals', 2.'fish', 3.specimens', ..., 8.'turtles', 9.'ducks', ..., 15.'water', ..., 31.'eels', ...	33	0.66
Similar phrase	zoo	"a glass zoo is sufficient for keeping [MASK]."	1.'animals', 2.'birds', 3.cats', ..., 8.'ducks', 9.'pigeons', ..., 30.'goats', ..., 37.'lions', ...	20	0.4
Random token	lena	"a glass lena is sufficient for keeping [MASK]."	1.'warm', 2.'dry', 3.'balance', 4.'water', ..., 7.'it', ..., 20.'safe', ..., 46.'records', ...	17	0.34

Table 7: **Similarity measure example.** Original sentence found in the corpus is "a glass fish tank is sufficient for keeping tarantulas.". MLM-suggestions shown are a sample of the top 50 (we use $q = 50$) suggestions for mask completions using BERT. The bold suggestions are ones that appear in the top-50 suggestions for the multi-token synonym sentence. Intersection is the size of the intersection between the top-50 tokens suggested for the sentence and the top 50 tokens suggested for the multi-token synonym sentence. Similarity is $intersection/q$.

hyperparameter	RNN-based method	EMAT method
Batch size	128	128
Learning rate	$1e^{-3}$	$1e^{-3}$
Epochs	10	2
GRU layers	2	-
Teacher forcing	0.5	-
Dropout	0.2	-

Table 8: **Hyperparameters.**

Similarity between terms. We define similarity between two terms as the average similarity across all pairs of masked sentences containing them.

B.3 Experiment results

For each quadruple, we compute the similarity between the multi-token phrase and the other quadruple terms. We use $k = 100$ (number of sentences for each quadruple) and $q = 50$ (number of top tokens considered for the similarity measure). For 82 out of the 100 quadruples in the dataset, the single-token synonym is the most similar to the multi-token phrase. 15 times the most similar is the similar phrase and only 3 times the most similar is the random term, showing the effectiveness of the similarity measure we use. The multi-token phrase is most similar to the single-token synonym: the average similarity measured between them is 0.766 while for the similar phrase it's 0.715 and for the random phrase it's 0.632. Results show the MLM treats multi-token phrases similarly to their single-token synonyms. This corroborates the assumption that the MLM is capable of storing information about multi-token phrases.

Version	$a@1$	$a@5$	$a@50$
lm_pretrain_unshared	7.79%	15.14%	28.10%
no_pretrain_shared	8.99%	16.67%	29.11%
lm_pretrain_shared	9.31%	17.15%	30.27%

Table 9: Ablation study. The results shown use BERT as the Base MLM.

C Model Details

In Section 4, we briefly describe our adaptation solutions. Herein, we describe the exact hyperparameters used and an ablation study of the RNN-based method. Table 8 presents the hyperparameters that were selected after tuning on the dev set. The hyperparameters had a larger impact on the RNN-based solution than the extended matrix (extended-matrix) solution.

In addition, we test to check the effectiveness of two components in our RNN-based solution. The First is pre-training the GRU on a vanilla Language Modeling task, we do this in order to give the model some sense of understanding of the language distribution before training it on the MTC task. The second component is sharing the embedding layers between the original MLM and the added completion GRU. As seen in Table 9 both of these are shown to help.

D Manual Annotation Instructions

In this task, you are given a sentence with a missing phrase (marked as `___`), a correct completion (which is not necessarily the only possible correct completion) and a proposed completion. The completed sentence is the sentence that is formed by planting the proposed completion in the location

of the missing phrase. Please answer the following questions:

1. Is the completed sentence grammatically correct and does it make sense? In this question, ignore the factual correctness of the completion. For example, in the sentence "*Barack Obama was born in ____*", any place or date is a valid completion.
2. Is the proposed completion more general than the given correct completion? For example, the word "*he*" is more general than a specific name. The phrase "*North America*" is more general than the phrase "*New York*". The phrase "*Los Angeles*" is not more general than the phrase "*New York*".
3. Is the missing phrase in the context of the sentence a part of a specific fact (geographical, physical, mathematical, etc.)? For example, in the sentence "*Barack Obama was born in ____*" the missing phrase is a part of a fact. In the sentence "*He was born in ____*" the missing phrase is not a part of a specific fact because "*He*" may refer to many different people.
4. If the answer to question 3 is yes, is the completed sentence factually correct? If you are not sure, please use the web to verify.

A Survey on Dynamic Neural Networks for Natural Language Processing

Canwen Xu, Julian McAuley
University of California, San Diego
{cxu, jmcauley}@ucsd.edu

Abstract

Effectively scaling large Transformer models is a main driver of recent advances in natural language processing. Dynamic neural networks, as an emerging research direction, are capable of scaling up neural networks with sub-linear increases in computation and time by dynamically adjusting their computational path based on the input. Dynamic neural networks could be a promising solution to the growing parameter numbers of pretrained language models, allowing both model pretraining with trillions of parameters and faster inference on mobile devices. In this survey, we summarize the progress of three types of dynamic neural networks in NLP: *skimming*, *mixture of experts*, and *early exit*. We also highlight current challenges in dynamic neural networks and directions for future research.

1 Introduction

Scaling up model capacity is an obvious yet effective approach for better performance in natural language processing (NLP) tasks (Brown et al., 2020; Kaplan et al., 2020; Ghorbani et al., 2021; Zhou et al., 2020b). However, the resulting increase in computational complexity and memory consumption becomes a bottleneck for scaling, making these models hard to train and use. On the other hand, it is not necessary to allocate the same amount of computation to all instances. For example, categorizing “I love you” as a positive sentence does not require a model containing dozens of Transformer layers. To resolve the aforementioned problems, *dynamic neural networks* have been a significant thrust of recent research in NLP. Dynamic networks can adjust their computational path based on the input for better efficiency, making it possible to train models with trillions of parameters and accelerate models in a low-resource setting.

In this survey, we review the latest state of research on three types of dynamic neural networks

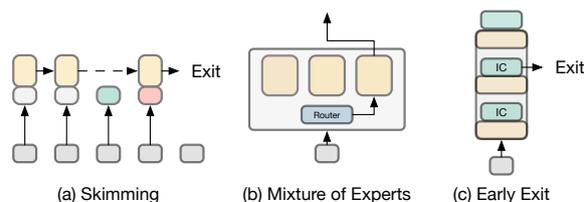


Figure 1: The three types of dynamic neural networks summarized in this paper. They dynamically adjust computation *timewise*, *widthwise* and *depthwise*, respectively.

that have been adopted in NLP: *skimming*, *mixtures of experts* (MoE), and *early exit*, as illustrated in Figure 1. These three types of techniques share a common idea of dynamically adjusting computation with respect to input, to save computation through bypassing unnecessary modules in a large neural network. However, they implement the goal via different approaches. **Skimming** was well-researched in the era of recurrent neural networks (RNN). Skimming models save computation *timewise* by dynamically allocating computation to different time steps, based on the input tokens. Since RNN models process the input sequence recurrently, it allows skimming models to achieve a substantial acceleration, especially when the sequence is long (Li et al., 2019). Different from RNN, recent works on Transformers skip tokens between layers instead of time steps.

For Transformer models (Vaswani et al., 2017; Devlin et al., 2019; Lan et al., 2020; Brown et al., 2020), the input tokens are fed into the model in parallel, while models have dozens of Transformer layers. This motivates the development of MoE and early exit. **MoE** horizontally extends a feedforward neural network (FFNN) with multiple sub-networks. During inference, only one or a few of these sub-networks will be activated for computation, thus can save *widthwise* computation. **Early exit**, on the other hand, terminates inference

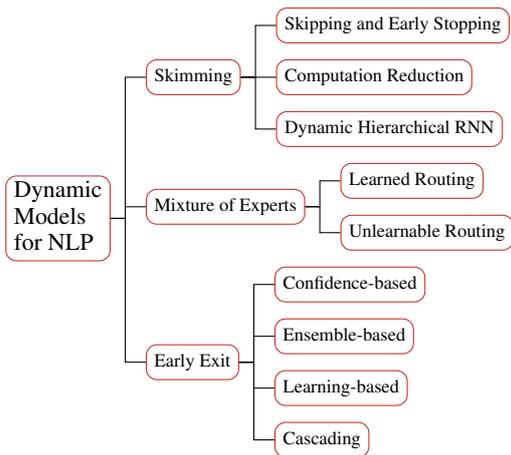


Figure 2: Taxonomy of dynamic neural networks for NLP.

at an early layer, without exhausting full computational capacity, thus saves *depthwise* computation. Early exit techniques often insert a series of lightweight classifiers which help decide when to exit, based on an exit strategy.

Note that this stream of works is distinct from static model acceleration, which is often referred to as *model compression*, including knowledge distillation, weight sharing, pruning and quantization (Sanh et al., 2019; Xu et al., 2020; Lan et al., 2020; Zafrir et al., 2019; Xu et al., 2021) (etc., see another survey (Xu and McAuley, 2022)). The major difference is that the computational path in a statically compressed model does not condition on the input and is invariable for all examples in inference. These two streams of research are in fact orthogonal and recent works Schwartz et al. (2020), Liu et al. (2020) and Zhu (2021) have shown that static and dynamic approaches can be combined for even faster inference and better performance.

To summarize, our contribution is two-fold: (1) We review the latest studies on the topic of dynamic neural networks for NLP by providing a comprehensive comparison and organize them with a new taxonomy, as shown in Figure 1. (2) We analyze current challenges in dynamic neural networks and point out directions for future research.

2 Skimming

Skimming techniques, as summarized in Table 1, skip some time steps or allocate different computation on them. Intuitively, skimming matches how human beings efficiently read text and extract information from it (Li et al., 2019). By em-

phasizing the important information within a sequence and ignoring parts with little importance, skimming helps the model achieve faster inference speed and better capture long-term dependencies. The three categories of skimming are *skipping and early stopping*, *computation reduction*, and *dynamic hierarchical RNN*, corresponding with three motivations: to skip unimportant input, to allocate less computation to unimportant input, and to increase computation to important input only.

Skipping and Early Stopping Skipping and early stopping aim to improve efficiency for a long sequence by skipping some tokens or stopping reading early. LSTM-Jump (Yu et al., 2017) is a skipping mechanism to ignore irrelevant information for natural language understanding (NLU). At each step, the current states are used to compute a “jumping softmax”, which decides how many steps to jump forward and whether to stop reading. LSTM-Jump employs policy gradient to train the model to make non-differentiable discrete jumping decisions. The reward is a binary function which rewards a correct prediction and penalizes an incorrect prediction of the label. Compared to a standard LSTM, LSTM-Jump achieves better accuracy with up to $6\times$ speed-up. Skip RNN (Campos et al., 2018) introduces a binary gate to learn whether to skip a state update. If the gate decides to skip a time step, the hidden states will be directly copied without any update.

To stop reading early as needed, ReasonNet (Shen et al., 2017) introduces a terminal state which decides whether to terminate early for machine reading comprehension on each time step at the token level. Jumper (Liu et al., 2018) first splits a paragraph to several sub-sentences and encodes them into sentence embeddings. They then apply early stopping at a sentence when the policy network decides to stop reading. Li et al. (2019) use eye-tracking devices and confirm that skipping and early stopping are common when humans read text. They propose Reading Inspired Model to mimic the behaviors of humans, which allows the model to decide whether to skip a single time step or stop reading early. Yu et al. (2018) add a re-reading operation to LSTM-Jump (Yu et al., 2017) which allows the model to stay on the current time step, allocating more computation to important information.

The aforementioned techniques can only go forward, which makes it impossible to regret if hav-

Method	Decision based on	Operation options
LSTM-Jump (Yu et al., 2017)	hidden states	skip multiple steps; stop reading
Skip RNN (Campos et al., 2018)	states of the update gate; hidden states	skip a single step
ReasonNet (Shen et al., 2017)	hidden states	stop reading
Jumper (Liu et al., 2018)	input sentence; hidden states	stop reading
RIM (Li et al., 2019)	input sentence; hidden states	skip a single step; stop reading
Yu et al. (2018)	hidden states	skip multiple steps; stop reading; re-read
LSTM-Shuttle (Fu and Ma, 2018)	hidden states	skip multiple steps; jump back multiples steps
Struc. Jump-LSTM (Hansen et al., 2019)	hidden states	stop reading; jump to next (;) or (!?)
PoWER (Goyal et al., 2020)	attention	drop tokens
TR-BERT (Ye et al., 2021)	hidden states	forward tokens
LAT (Kim and Cho, 2021)	attention	forward tokens
LTP (Kim et al., 2022)	attention	drop tokens
Transkimmer (Guan et al., 2022)	hidden states	forward tokens
VCRNN (Jernite et al., 2017)	input token; hidden states	partial update with zero-masked weights
Skim-RNN (Seo et al., 2018)	input token; hidden states	partial update with a small RNN
HM-RNN (Chung et al., 2017)	states of the gates	skip a single step; "flush"
FHRNN (Ke et al., 2018)	query; hidden states	update the upper RNN layer

Table 1: A summary of skimming techniques.

ing jumped over important information. LSTM-Shuttle (Fu and Ma, 2018) proposes a bidirectional shuttling mechanism, which can jump multiple time steps both forward and backward, allowing the model to ignore unimportant information and recover lost information if needed.

Structural information that naturally exists in sentences can also play a role in skimming. Structural Jump-LSTM (Hansen et al., 2019) can jump to the next word, next sub-sentence separator (a comma or colon), next sentence end symbols (a period, exclamation mark or question mark), or to the end of the text (i.e., stop reading).

In the era of Transformers, there have been works attempting to reduce computation by either skip tokens at higher layers or forward tokens to higher layers. The PoWER-BERT model (Goyal et al., 2020) reduces the number of tokens processed by each Transformer layer based on their attention scores. The number of tokens to be dropped, referred to as the "schedule," is optimized by combining the sparsity of a soft mask layer with the original loss function. This results in an improved balance between accuracy and processing time. TR-BERT (Ye et al., 2021) uses a dynamic approach to determine which tokens to skip, using reinforcement learning to train the model with a reward system that prioritizes classifier confidence while also penalizing the number of tokens retained. In contrast to PoWER-BERT, TR-BERT passes the skipped tokens to the final layer rather than discarding them. The Length-Adaptive Transformer (LAT, Kim and Cho, 2021)

utilizes LengthDrop to randomly skip tokens during pretraining, aiming to close the gap between pretraining and fine-tuning. The schedule for LAT is found through an evolutionary search algorithm. LTP (Kim et al., 2022) trains a threshold for each Transformer layer, instead of following a predetermined schedule. It simply drops tokens with attention scores lower than the learned threshold. Transkimmer (Guan et al., 2022) incorporates a skim predictor module, consisting of a small MLP and Gumbel-Softmax reparameterization, before each layer. This module outputs a mask to determine whether a token should be dropped, and a skim loss is used to optimize the ratio of skipped tokens to total tokens, promoting sparsity.

Computation Reduction Different from skipping, computation reduction applies a reduced computational workload for some time steps instead of skipping it completely. VCRNN (Jernite et al., 2017) explores a scheduler to decide which proportion of computation to use for each time step. Upon making the decision, only the corresponding proportion of the weight matrix will be used to update the hidden states while the rest part of the weight matrix will be masked out with zero.

Instead of using part of weights to update the hidden states, Skim-RNN (Seo et al., 2018) has a big RNN and a separate small RNN. At each time step, the model decides whether to read or skim based on hidden states from the last time step and the input token. If the model decides to skim, the small RNN will update only a fraction of the hid-

den states. Otherwise, a regular full update will be conducted by the big RNN.

Dynamic Hierarchical RNN Different from the aforementioned two categories of skimming, dynamic hierarchical RNN can *increase* computation by calling the upper layer RNN when needed. HM-RNN (Chung et al., 2017) automatically discovers the hierarchical multi-scale structure in the data for a hierarchical RNN architecture. In addition to the update and copy operations as in Skip RNN (Campos et al., 2018), they add a flush operation which ejects the summarized representation of the current time step to the upper layer and re-initializes the states for the next time step.

In question answering, only a small portion of tokens are relevant and can be used to answer the question while the rest can be safely skimmed. Based on this observation, Focused Hierarchical RNN (Ke et al., 2018) aims to only pick up information that is relevant to the query for question answering. It applies a binary gate to control whether to update the upper layer of the RNN, based on the current hidden states of the lower-level RNN and the question embedding.

3 Mixture of Experts

Increasing the number of parameters in a model often leads to increased computation and memory consumption. To take the advantages of parameter scaling without a proportional increase in computation, mixture of experts (MoE) (Jacobs et al., 1991) is introduced to large language models, as summarized in Table 2. In these models, a layer typically contains multiple sub-networks (i.e., “experts”). During inference, only part of these experts will be activated on a per-example basis.

The key element of MoE methods is the routing mechanism. The routing mechanism has to be lightweight, not to significantly slower the speed of the model. We categorize MoE methods into two groups: *learned routing* and *unlearnable routing*. Learned routing often requires some load balancing mechanisms to ensure that all experts are trained with enough examples thus are useful during inference. Unlearnable routing usually slightly underperforms learned routing but does not require complicated load balancing.

MoE Layers with Learned Routing A straightforward idea to implement MoE is to learn a router

to allocate inputs to experts. Sparsely-Gated MoE layer (Shazeer et al., 2017) contains up to thousands of feed-forward sub-networks with a trainable gating network which determines a sparse combination of these experts to use for each example. There are two major challenges to address: **(1) Sparsity.** The gating network predicts a softmax weight for the experts based on the input. The gating network is trained by simple back-propagation, together with other parts of the model. Then, only the top- k experts in the layer will be activated based on the softmax prediction of the gating network. They insert one MoE layer between stacked LSTM layers and achieve improvement on language modeling and machine translation tasks. **(2) Load balancing.** Shazeer et al. (2017) observe a self-reinforcing phenomenon that the gating network tends to converge to a state where it always produces large weights for the same few experts. They resolve the problem by defining the importance of an expert relative to a batch of training examples to be batch-wise sum of the *gate values* for that expert. Then, they introduce an additional loss, the square of the coefficient of variation of the set of importance values, to encourage a more balanced update during training. Besides encouraging a balanced update, the authors also introduce a loss function with a smooth estimator that estimate the number of examples assigned to each expert for a batch of inputs, to encourage experts to receive roughly equal *numbers of training examples*.

GShard (Lepikhin et al., 2021) enables scaling up multilingual neural machine translation Transformer beyond 600 billion parameters. It adapts Sparsely-Gated MoE (Shazeer et al., 2017) to Transformer (Vaswani et al., 2017) by replacing every other feed forward layer with an MoE layer, which routes to top-2 experts. When scaling to multiple devices, the MoE layer is sharded across devices, i.e., each device has different allocated experts, while all other layers are replicated. To achieve workload balance, GShard employs a threshold, namely expert capacity, to limit the maximum number of tokens processed by one single expert. They also introduce a local group dispatching mechanism, which partitions all tokens in a training batch evenly into groups to be processed independently in parallel, to balance the overall workload. Following (Shazeer et al., 2017), they use an additional loss to enforce even

Method	Base Model	Sparsity	Load Balance
Sparsely (Shazeer et al., 2017)	LSTM	top- k	auxiliary loss
GShard (Lepikhin et al., 2021)	Transformer (NMT)	top-2	expert capacity; local group dispatching; auxiliary loss; random routing
Switch (Fedus et al., 2021)	Transformer (T5)	top-1	expert capacity; auxiliary loss
BASE (Lewis et al., 2021)	Transformer (GPT)	top-1	linear assignment
M6-T (Yang et al., 2021)	Transformer (M6)	k top-1	expert capacity
DTS (Nie et al., 2021)	Transformer (GPT)	dynamic	sparsity scheduler
Hash (Roller et al., 2021)	Transformer	hash	deterministic hash
THOR (Zuo et al., 2022)	Transformer (NMT)	random	random selection

Table 2: A summary of Mixture of Experts (MoE) methods.

allocation for experts. Additionally, they propose a random routing mechanism, which only routes to the second-best expert with probability proportional to its weight, to simplify sparse training.

Switch Transformer (Fedus et al., 2021) aims to simplify the Sparsely-Gated MoE (Shazeer et al., 2017) for efficiency and performance. They propose a Switch Layer which only routes to one expert at a time, to reduce gating computation, batch size and communication costs. Switch Transformer inherits expert capacity and an auxiliary load balancing loss from GShard (Lepikhin et al., 2021). Combined with low-precision training, compared to T5-Base and T5-Large (Raffel et al., 2020), Switch Transformer obtains up to $7\times$ increases in pretraining speed with the same computational resources. They further scale Switch Transformer to more than 1.5 trillion parameters and achieve $4\times$ speed-up over T5-XXL.

The Balanced Assignment of Sparse Experts (BASE) layer (Lewis et al., 2021) formulates token-to-expert allocation as a linear assignment problem and solves it with the auction algorithm (Bertsekas, 1992). This allows an optimal assignment in which each expert receives an equal number of tokens, improving efficiency and getting rid of the expert capacity and auxiliary loss in previous works. The experiments show that BASE layers are more efficient for training compared to Sparsely-Gated MoE layers (Shazeer et al., 2017) and Switch Layers (Fedus et al., 2021), and can successfully learn a good balanced routing without any auxiliary balancing loss.

M6 (Lin et al., 2021) is a multi-modal multitask Transformer, trained in the same way as Switch Transformer (Fedus et al., 2021), scaling up to 100B parameters. Following this, M6-T (Yang et al., 2021) splits experts into k prototypes (i.e., groups of experts). In each forward pass, each token is sent to the k prototypes, within which the top-1 routing is done lo-

cally. The experiments demonstrate this “ k top-1” strategy outperforms the top-1 routing in Switch Transformer (Fedus et al., 2021) while being more computation-efficient than “top- k ” routing. They also claim that the load balancing loss may be ineffective for improving the performance of an MoE model, although it can indeed help balance the workload. They subsequently train a 1 trillion parameter model with the finding.

Dense-to-Sparse gate (Nie et al., 2021) begins as a dense gate that routes tokens to all experts then gradually learns to become sparser and route tokens to fewer experts, demonstrating higher training efficiency in experiments. Their experiments confirm the finding in Yang et al. (2021) that an auxiliary load balancing loss does not improve the model performance.

MoE Layer with Unlearnable Routing Although learning-based routing has shown effectiveness only with the help of complicated load balancing mechanisms, recent studies have attempted to get rid of those. Hash Layer (Roller et al., 2021) simplifies routing by using a parameter-free hashing function to route tokens to specific experts. This design eliminates the need for a load balancing loss and sophisticated assignment algorithms. They also study the performance of different hashing techniques, hash sizes and input features, and conclude that balanced and random hashes focused on the most local features work best. The experiments show that a Hash Layer achieves comparable performance with a Switch Layer (Fedus et al., 2021) and BASE Layer (Lewis et al., 2021).

THOR (Zuo et al., 2022) is a special form of MoE layer, which completely discards the conditional routing mechanism and instead optimizes the consistency between a randomly selected pair of experts. During inference, one expert will be randomly selected to be activated.

Applications and Analysis GLaM (Du et al., 2021) trains a family of GPT-style language models with up to 1.2 trillion parameters using GShard (Lepikhin et al., 2021). CPM-2 (Zhang et al., 2022b) trains a large Chinese language model with 198 billion parameters with BASE layers (Lewis et al., 2021).

Artetxe et al. (2021) conduct a detailed empirical study of how autoregressive MoE language models scale compared to dense models. They find MoEs to be substantially more efficient with the exception of fine-tuning. MoE models can match the performance of dense models with 25% of computation in a low-resource setting. Although the advantage fades at scale, their largest MoE model with 1.1 trillion parameters can consistently outperform its dense counterpart with the same amount of computation. Clark et al. (2022) examine the scaling law of BASE Layer (Lewis et al., 2021), Hash Layer (Roller et al., 2021) and earlier Reinforcement Learning-based routing algorithms providing suggestions for best-practices in training MoE models.

Zhang et al. (2021) propose MoEfication to split feedforward neural networks (FFNN) in a trained large model to experts. They find that a T5-Large (Raffel et al., 2020) model with 700 million parameters only activates 5% neurons for 80% inputs on a downstream task, indicating high redundancy within large pretrained language models. To transform a pretrained language model to an MoE model, they first construct a co-activation graph for each FFNN and then divide the graph into subgraphs with strong internal connections with graph partitioning algorithm. Each subgraph forms an expert. They train a router with oracle best routing for training data. Then, they further fine-tune the resulted model for better performance.

4 Early Exit

Early exit techniques aim to terminate model inference in early layers, to save computation and sometimes improve performance by resolving the overthinking problem (Kaya et al., 2019), i.e., possible performance degradation at a later layer. It can be useful especially in the era of pretrained language models (PLM), since increasing the size of PLMs can often lead to better performance, although a smaller model can already predict most examples (i.e., “easy examples”) correctly.

The main idea of early exit is to exit inference

at an earlier layer, rather than the last layer. Early exit often involves a series of internal classifiers inserted into a large network, providing signals for early exiting. The core of early exit methods is the exit criterion. Based on their exit strategies, we categorize the early exit methods into three classes: confidence-based, ensemble-based and learning-based, as listed in Table 3.

Despite better performance, speed and adversarial robustness (Zhou et al., 2020a), an additional benefit is that the speed-accuracy trade-off can be adjusted as needed by tuning the exit threshold (θ in Table 3), without the need of retraining the model. A main drawback is that early exit is often applied on a per-instance basis, meaning that to maximize the speed-up ratio, a small batch size (often 1) has to be used.

Confidence-based Early Exit Early works for early exit in computer vision (Park et al., 2015; Teerapittayanon et al., 2016; Kaya et al., 2019) often fall into this category. They define a metric as the proxy for confidence of a model prediction. The model exits early when the confidence hits a predefined threshold. DeeBERT (Xin et al., 2020b) applies BranchyNet (Teerapittayanon et al., 2016) to BERT inference. The training for DeeBERT is two-stage: they first train BERT on downstream tasks following standard fine-tuning. Then, they freeze the parameters of the Transformer and insert a linear classifier (i.e., internal classifier) after each Transformer layer. They train the classifiers by minimizing the sum of their cross-entropy loss. For inference, the model exits early when an internal classifier outputs a prediction probability distribution that has an entropy lower than a predefined threshold. RightTool (Schwartz et al., 2020) jointly fine-tunes BERT with internal classifiers. They use the temperature-calibrated maximum class probability as confidence. FastBERT (Liu et al., 2020) first trains the BERT backbone and the final classifier. Then, they distill the final classifier layer to the internal classifiers (Hinton et al., 2015). For inference, the model exits when the entropy of a prediction is below the threshold. RomeBERT (Geng et al., 2021) provides a simple fix for learning internal classifiers efficiently. Besides self-distillation as in FastBERT, they propose gradient regularization (GR) to facilitate distillation. SkipBERT (Wang et al., 2022) caches pre-computed representation of text chunks to re-

Method	Internal classifier training	Exit criterion
DeeBERT (Xin et al., 2020b)	two-stage; sum of CE loss	entropy $< \theta$
RightTool (Schwartz et al., 2020)	joint; sum of CE loss	calibrated max class probability $> \theta$
FastBERT (Liu et al., 2020)	two-stage; self-distillation	entropy $< \theta$
RomeBERT (Geng et al., 2021)	joint; self-distillation + GR	entropy $< \theta$
SkipBERT (2022)	joint; weighted sum of CE + KD	max class probability $> \theta$
PABEE (Zhou et al., 2020a)	joint; weighted sum of CE loss	patience (#consistent prediction $> \theta$)
Voting (Sun et al., 2021)	joint; sum of CE + diversity loss	accumulated votes $> \theta$
LeeBERT (Zhu, 2021)	joint; auto-weighted sum of CE + KD loss	patience (#consistent prediction $> \theta$)
Past-Future (Liao et al., 2021)	joint; weighted sum of CE + imitation learning	entropy $< \theta$
PCEE-BERT (2022a)	joint; weighted sum of CE	patience (#consistent IC confidence $> \theta$)
BERxiT (Xin et al., 2021)	alternate; sum of CE loss	estimated confidence $> \theta$
CAT (Schuster et al., 2021)	joint; avg. of CE loss	estimated conformity $> \theta$
CascadeBERT (Li et al., 2021a)	standard model FT with confidence calibration	calibrated max class probability $> \theta$

Table 3: A summary of early exit methods. θ is a predefined threshold for exiting. This table is extended from a table in Xu and McAuley (2022).

place lower BERT layers and uses confidence-based early exit for higher layers to achieve maximum acceleration.

Ensemble-based Early Exit One drawback in confidence-based early exit is wasted computation. That is to say, if the confidence of an internal classifier does not satisfy the exit criterion, it will be disregarded. Ensemble-based early exit recycles these predictions and considers output from multiple internal classifiers to make better predictions. Based on the similarity between overfitting and overthinking, PABEE (Zhou et al., 2020a) borrows early stopping from model training. They first jointly train the internal classifiers with BERT by a weighted sum of cross-entropy losses that assigns larger weights for upper classifiers. For inference, the model exits when k consecutive internal classifiers make the same prediction. Other than improvement on performance and efficiency, they find that PABEE can improve adversarial robustness, which they attribute to the ensemble effect. Sun et al. (2021) further introduce a diversity loss that encourages internal classifiers to have a diverse predicted probability distribution. They propose a voting mechanism to ensemble the internal classifiers by exiting early when a class has accumulated more votes than the threshold. Interestingly, LeeBERT (Zhu, 2021) adopts the opposite strategy: they promote consistency across internal classifiers by distilling them to each other. However, they introduce a learnable weight for the cross-entropy loss of each classifier and the distillation loss between each pair. They optimize these weights by a cross-level optimization algorithm.

They adopt PABEE’s patience-based strategy for exiting. Liao et al. (2021) train linear transformation layers called “imitation learners”, to approximate the hidden states of future layers based on current hidden states. For inference, the prediction after each layer is calculated by mixing the past predictions and the future predictions of the imitation learners. Entropy is used as the exit criterion. PCEE-BERT (Zhang et al., 2022a) borrows from both ensemble-based exit and confidence-based methods. The inference is terminated when multiple layers are confident.

Learning-based Early Exit Another stream of research is to *learn* a criterion for early exiting. BERxiT (Xin et al., 2021) alternates between joint fine-tuning and two-stage fine-tuning by freezing parameters of Transformer and the final classifier for even-numbered iterations and unfreezing them for odd-numbered iterations. They also train a linear layer called a *learning-to-exit* (LTE) module to predict whether the current internal classifier makes the correct prediction. It takes the hidden states as input and outputs a confidence score, which is used to decide whether to exit. CAT (Schuster et al., 2021) introduces a “meta consistency classifier” to predict whether the output of an internal classifier conforms to the final classifier and exits when the consistency classifier predicts a certain level of conformity.

Cascading Cascading can be seen as a special form of early exit, performed at the model level. Li et al. (2021a) find that shallow features and internal classifiers in the first few layers of BERT utilized by early exit methods like DeeBERT (Xin

et al., 2020b) are not sufficient and reliable, underperforming a fine-tuned BERT with the same number of layers. Therefore, they propose to use a suite of complete models with different numbers of layers for cascading. CascadeBERT executes models one by one, from the smallest to the largest. It stops when a model outputs a confidence score (calibrated maximum class probability) that reaches the threshold.

Applications Although early exit is originally developed for classification, there have been works extending it to more tasks and settings. Li et al. (2021b) propose Token-Level Early-Exit that targets early exiting for sequence labeling. They use the maximum class probability as confidence on a per-token basis. Once the confidence hits the threshold, the hidden states of the corresponding tokens will be frozen and directly copied to upper layers. These exited tokens will not attend to other tokens at upper layers but can still be attended by other tokens. The model completely exits when every token exits. A similar idea is also presented in Elbayad et al. (2020) and Liu et al. (2021b) where hidden states of some positions can be frozen and directly copied to upper layers, although the former is focused on generation and the latter is for classification. Xin et al. (2020a) apply DeeBERT (Xin et al., 2020b) to document ranking and set different thresholds to the negative and positive classes for early exiting, to accommodate the imbalanced class distribution in document ranking. ELUE (Liu et al., 2021a) is a benchmark which evaluates the Pareto Front of early exit models on the FLOPs-performance plane. They provide a BERT-like baseline with jointly pretrained internal classifiers, to mitigate the gap between pretraining and fine-tuning.

5 Challenges and Future Directions

Evaluation Evaluating dynamic neural networks can be difficult since we cannot pre-define a few break points to compare different methods at the exact same amount of computation or time. ELUE score (Liu et al., 2021a) may be a promising solution to this problem by considering both computation and performance, depicting the Pareto Front. Besides, different works have different calculation for speed-up ratio. For example, some works use the ratio of layers involved in computation to estimate speed-up ratio (Zhou et al., 2020a; Sun et al., 2021; Liao et al., 2021). This can

be misleading since internal classifiers introduce extra computational costs, especially when more complicated mechanism introduced, e.g., future-layer imitation (Liao et al., 2021). Also, the reported speed of MoE models, greatly differs on different hardware and distribution settings, making it hard to compare across papers.

Data Parallelism One drawback of dynamic neural networks is their inefficiency on data parallelism. To be specific, MoE methods introduce extra communication costs for dynamic routing and could be a bottleneck for efficiency. Skimming and early exit methods often employ an “online inference” setting where the batch size is fixed to 1, to achieve maximum acceleration. However, for batched inference, the efficiency of these methods will drastically degrade, since the already-exited instances will have to wait all instances to exit, which causes a low parallelism and low utilization of GPU.

Optimized Runtime Since dynamic neural networks are an emerging type of neural networks, most hardware and libraries are not well-optimized for these models. For example, sparse matrix multiplication in MoE needs specialized hardware and software support to achieve its theoretical efficiency. Also, current dynamic neural networks are often implemented in eager execution, which prevents them from low-level optimization of graph execution. There have been works exploring optimized runtime for MoE (Shazeer et al., 2018; Jia et al., 2020; He et al., 2021; Rajbhandari et al., 2022) and early exit (Paul et al., 2019) while more to be done in the future.

Theoretical Analysis and Support While the dynamic neural networks have demonstrated empirical improvement over static counterparts, dynamic networks are not solidly backed by theoretical analysis. For example, the theoretical analysis in PABEE (Zhou et al., 2020a) is based on an assumption that internal classifiers are independent to each other, which is unrealistic. More research should be done from the perspective of optimization and effect of data distribution on dynamic neural networks.

Explainability The decision-making process of the dynamic neural networks can be important to explain the model prediction and even understand

more fundamental research questions in machine learning, including scaling law and generalization. Can we use skimming to explain sequence classification? Is it consistent with attention-based explanation (Xu et al., 2015)? What does each expert in MoE learn and what makes them different? Why does a lower internal classifier make different prediction from an upper classifier despite equally trained with the same objective? These questions warrant further exploration, from both data and model perspectives.

Limitations

A limitation of this survey is that we do not draw a direct quantitative comparison for the methods surveyed in this paper since different methods have their own accuracy-speed curves, with their own unique limitations (e.g., many early exit methods can only handle a batch size of 1). Also, we do not discuss some works in depth and in detail due to space limit.

References

- Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, et al. 2021. Efficient large scale language modeling with mixtures of experts. *arXiv preprint arXiv:2112.10684*.
- Dimitri P. Bertsekas. 1992. Auction algorithms for network flow problems: A tutorial introduction. *Comput. Optim. Appl.*, 1(1):7–66.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.
- Víctor Campos, Brendan Jou, Xavier Giró-i-Nieto, Jordi Torres, and Shih-Fu Chang. 2018. Skip RNN: learning to skip state updates in recurrent neural networks. In *ICLR*. OpenReview.net.
- Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2017. Hierarchical multiscale recurrent neural networks. In *ICLR*. OpenReview.net.
- Aidan Clark, Diego de las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. 2022. Unified scaling laws for routed language models. *arXiv preprint arXiv:2202.01169*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.
- Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2021. Glam: Efficient scaling of language models with mixture-of-experts. *arXiv preprint arXiv:2112.06905*.
- Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. 2020. Depth-adaptive transformer. In *ICLR*. OpenReview.net.
- William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.
- Tsu-Jui Fu and Wei-Yun Ma. 2018. Speed reading: Learning to read forbackward via shuttle. In *EMNLP*, pages 4439–4448. Association for Computational Linguistics.
- Shijie Geng, Peng Gao, Zuohui Fu, and Yongfeng Zhang. 2021. Romebert: Robust training of multi-exit bert. *arXiv preprint arXiv:2101.09755*.
- Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. 2021. Scaling laws for neural machine translation. *arXiv preprint arXiv:2109.07740*.
- Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, et al. 2020. Power-bert: Accelerating BERT inference via progressive word-vector elimination. In *ICML*.
- Yue Guan, Zhengyi Li, Jingwen Leng, et al. 2022. Transkimmer: Transformer learns to layer-wise skim. In *ACL*.
- Christian Hansen, Casper Hansen, Stephen Alstrup, Jakob Grue Simonsen, and Christina Lioma. 2019. Neural speed reading with structural-jump-lstm. In *ICLR*. OpenReview.net.
- Jiaao He, Jiezhong Qiu, Aohan Zeng, Zhilin Yang, Jidong Zhai, and Jie Tang. 2021. Fastmoe: A fast mixture-of-expert training system. *arXiv preprint arXiv:2103.13262*.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

- Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.
- Yacine Jernite, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Variable computation in recurrent neural networks. In *ICLR*. OpenReview.net.
- Xianyan Jia, Le Jiang, Ang Wang, Jie Zhang, Xinyuan Li, Wencong Xiao, Yong Li, Zhen Zheng, Xiaoyong Liu, Wei Lin, et al. 2020. Whale: Scaling deep learning model training to the trillions. *arXiv preprint arXiv:2011.09208*.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. 2019. Shallow-deep networks: Understanding and mitigating network overthinking. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 3301–3310. PMLR.
- Nan Rosemary Ke, Konrad Zolna, Alessandro Sordani, Zhouhan Lin, Adam Trischler, Yoshua Bengio, Joelle Pineau, Laurent Charlin, and Christopher J. Pal. 2018. Focused hierarchical rnns for conditional sequence processing. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 2559–2568. PMLR.
- Gyuwan Kim and Kyunghyun Cho. 2021. Length-adaptive transformer: Train once with length drop, use anytime with search. In *ACL-IJCNLP*.
- Sehoon Kim, Sheng Shen, David Thorsley, et al. 2022. Learned token pruning for transformers. In *KDD*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *ICLR*. OpenReview.net.
- Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *ICLR*. OpenReview.net.
- Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. 2021. BASE layers: Simplifying training of large, sparse models. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 6265–6274. PMLR.
- Lei Li, Yankai Lin, Deli Chen, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. 2021a. Cascadebert: Accelerating inference of pre-trained language models via calibrated complete models cascade. In *EMNLP (Findings)*, pages 475–486. Association for Computational Linguistics.
- Xiangsheng Li, Jiabin Mao, Chao Wang, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. Teach machine how to read: Reading behavior inspired relevance estimation. In *SIGIR*, pages 795–804. ACM.
- Xiaonan Li, Yunfan Shao, Tianxiang Sun, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2021b. Accelerating BERT inference for sequence labeling via early-exit. In *ACL-IJCNLP*, pages 189–199. Association for Computational Linguistics.
- Kaiyuan Liao, Yi Zhang, Xuancheng Ren, Qi Su, Xu Sun, and Bin He. 2021. A global past-future early exit method for accelerating inference of pre-trained language models. In *NAACL-HLT*, pages 2013–2023. Association for Computational Linguistics.
- Junyang Lin, Rui Men, An Yang, Chang Zhou, Ming Ding, Yichang Zhang, Peng Wang, Ang Wang, Le Jiang, Xianyan Jia, et al. 2021. M6: A chinese multimodal pretrainer. *arXiv preprint arXiv:2103.00823*.
- Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. Fastbert: a self-distilling BERT with adaptive inference time. In *ACL*, pages 6035–6044. Association for Computational Linguistics.
- Xianggen Liu, Lili Mou, Haotian Cui, Zhengdong Lu, and Sen Song. 2018. Jumper: Learning when to make classification decisions in reading. *arXiv preprint arXiv:1807.02314*.
- Xiangyang Liu, Tianxiang Sun, Junliang He, Lingling Wu, Xinyu Zhang, Hao Jiang, Zhao Cao, Xuanjing Huang, and Xipeng Qiu. 2021a. Towards efficient nlp: A standard evaluation and a strong baseline. *arXiv preprint arXiv:2110.07038*.
- Yijin Liu, Fandong Meng, Jie Zhou, Yufeng Chen, and Jinan Xu. 2021b. Faster depth-adaptive transformers. In *AAAI*, pages 13424–13432. AAAI Press.
- Xiaonan Nie, Shijie Cao, Xupeng Miao, Lingxiao Ma, Jilong Xue, Youshan Miao, Zichao Yang, Zhi Yang, and Bin Cui. 2021. Dense-to-sparse gate for mixture-of-experts. *arXiv preprint arXiv:2112.14397*.
- Eunhyeok Park, Dongyoung Kim, Soobeom Kim, Yong-Deok Kim, Gunhee Kim, Sungroh Yoon, and Sungjoo Yoo. 2015. Big/little deep neural network for ultra low power inference. In *CODES+ISSS*, pages 124–132. IEEE.
- Debddeep Paul, Jawar Singh, and Jimson Mathew. 2019. Hardware-software co-design approach for deep learning inference. In *ICSCC*, pages 1–5. IEEE.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

- Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. *arXiv preprint arXiv:2201.05596*.
- Stephen Roller, Sainbayar Sukhbaatar, Jason Weston, et al. 2021. Hash layers for large sparse models. In *NeurIPS*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. 2021. Consistent accelerated inference via confident adaptive transformers. *arXiv preprint arXiv:2104.08803*.
- Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. 2020. The right tool for the job: Matching model and instance complexities. In *ACL*, pages 6640–6651. Association for Computational Linguistics.
- Min Joon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Neural speed reading via skim-rnn. In *ICLR*. OpenReview.net.
- Noam Shazeer, Youlong Cheng, Niki Parmar, Dustin Tran, Ashish Vaswani, Penporn Koanantakool, Peter Hawkins, HyoukJoong Lee, Mingsheng Hong, Cliff Young, Ryan Sepassi, and Blake A. Hechtman. 2018. Mesh-tensorflow: Deep learning for supercomputers. In *NeurIPS*, pages 10435–10444.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*. OpenReview.net.
- Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In *KDD*, pages 1047–1055. ACM.
- Tianxiang Sun, Yunhua Zhou, Xiangyang Liu, Xinyu Zhang, Hao Jiang, Zhao Cao, Xuanjing Huang, and Xipeng Qiu. 2021. Early exiting with ensemble internal classifiers. *arXiv preprint arXiv:2105.13792*.
- Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *ICPR*, pages 2464–2469. IEEE.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.
- Jue Wang, Ke Chen, Gang Chen, et al. 2022. Skipbert: Efficient inference with shallow layer skipping. In *ACL*.
- Ji Xin, Rodrigo Nogueira, Yaoliang Yu, and Jimmy Lin. 2020a. Early exiting bert for efficient document ranking. In *SustainNLP*, pages 83–88.
- Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020b. Deebert: Dynamic early exiting for accelerating BERT inference. In *ACL*, pages 2246–2251. Association for Computational Linguistics.
- Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. Berxit: Early exiting for BERT with better fine-tuning and extension to regression. In *EACL*, pages 91–104. Association for Computational Linguistics.
- Canwen Xu and Julian McAuley. 2022. A survey on model compression and acceleration for pretrained language models. *arXiv preprint arXiv:2202.07105*.
- Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing BERT by progressive module replacing. In *EMNLP*, pages 7859–7869. Association for Computational Linguistics.
- Canwen Xu, Wangchunshu Zhou, Tao Ge, Ke Xu, Julian J. McAuley, and Furu Wei. 2021. Beyond preserved accuracy: Evaluating loyalty and robustness of BERT compression. In *EMNLP*, pages 10653–10659. Association for Computational Linguistics.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org.
- An Yang, Junyang Lin, Rui Men, Chang Zhou, Le Jiang, Xianyan Jia, Ang Wang, Jie Zhang, Jiamang Wang, Yong Li, et al. 2021. M6-t: Exploring sparse expert models and beyond. *arXiv preprint arXiv:2105.15082*.
- Deming Ye, Yankai Lin, Yufei Huang, and Maosong Sun. 2021. TR-BERT: dynamic token reduction for accelerating BERT inference. In *NAACL-HLT*.
- Adams Wei Yu, Hongrae Lee, and Quoc V. Le. 2017. Learning to skim text. In *ACL*, pages 1880–1890. Association for Computational Linguistics.
- Keyi Yu, Yang Liu, Alexander G. Schwing, and Jian Peng. 2018. Fast and accurate text classification: Skimming, rereading and early stopping. In *ICLR (Workshop)*. OpenReview.net.
- Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*.
- Zhen Zhang, Wei Zhu, Jinfan Zhang, et al. 2022a. PCEE-BERT: accelerating BERT inference via patient and confident early exiting. In *NAACL-HLT (Findings)*.

Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, Yanzheng Cai, et al. 2022b. Cpm-2: Large-scale cost-effective pre-trained language models. *AI Open*.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. Moefication: Conditional computation of transformer models for efficient inference. *arXiv preprint arXiv:2110.01786*.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian J. McAuley, Ke Xu, and Furu Wei. 2020a. BERT loses patience: Fast and robust inference with early exit. In *NeurIPS*.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020b. Evaluating commonsense in pre-trained language models. In *AAAI*, pages 9733–9740. AAAI Press.

Wei Zhu. 2021. LeeBERT: Learned early exit for BERT with cross-level optimization. In *ACL-IJCNLP*, pages 2968–2980. Association for Computational Linguistics.

Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Jianfeng Gao, and Tuo Zhao. 2022. Taming sparsely activated transformer with stochastic experts. In *ICLR*.

Transformers with Learnable Activation Functions

Haishuo Fang¹ Ji-Ung Lee¹ Nafise Sadat Moosavi^{1,2} Iryna Gurevych¹

¹Ubiquitous Knowledge Processing Lab (UKP Lab)
Department of Computer Science and Hessian Center for AI (hessian.AI)
Technical University of Darmstadt
www.ukp.tu-darmstadt.de

²Department of Computer Science, The University of Sheffield

Abstract

Activation functions can have a significant impact on reducing the topological complexity of input data and therefore, improving a model’s performance. However, the choice of activation functions is seldom discussed or explored in Transformer-based language models. As a common practice, commonly used activation functions like Gaussian Error Linear Unit (GELU) are chosen beforehand and then remain fixed from pre-training to fine-tuning. In this paper, we investigate the impact of activation functions on Transformer-based models by utilizing rational activation functions (RAFTs). In contrast to fixed activation functions (FAF), RAFTs are capable of learning the optimal activation functions from data. Our experiments show that the RAFT-based Transformer model (RAFT) achieves a better performance than its FAF-based counterpart (FAFT). For instance, we find that RAFT outperforms FAFT on the GLUE benchmark by 5.71 points when using only 100 training examples and by 2.05 points on SQuAD with all available data. Analyzing the shapes of the learned RAFTs further unveils that they vary across different layers and different tasks; opening a promising way to better analyze and understand large, pre-trained language models.¹

1 Introduction

Activation functions introduce non-linearity and increase neural networks’ representational capacity, and therefore, play an essential role in designing deep learning models (Nwankpa et al., 2018; Sharma et al., 2020; Dubey et al., 2022). Naitzat et al. (2020) explain the importance of activation functions by proposing to consider data as a topology with its own shape. They empirically show that activation functions accelerate the data topology transformation through different layers of a neural network to simplify its complexity and make

it linearly separable in the output space. Their experiments show that choosing the right activation function can have a significant impact on the overall performance.

While any activation function can be used with Transformers (Vaswani et al., 2017), their choice is made before pre-training and remains fixed afterwards. Hence, the inductive bias an activation function imposes on the model cannot be adjusted during pre-training or fine-tuning. As many Transformer-based models are pre-trained on a large amount of data, and changing the activation function for or during fine-tuning may negatively impact the performance². Moreover, the simple case of finding the optimal combination of k different activation functions in n different feedforward layers results in k^n possible combinations and becomes intractable; e.g., 531,441 experiments for a 12-layer BERT model and three different activation functions. As a result, most Transformer-based pre-trained models adopt the GELU activation function that has been initially used for the BERT model (Devlin et al., 2019).

To overcome the limitation of using a potentially suboptimal activation function that remains fixed during training, we propose to use a learnable activation function, namely, the rational activation function (RAF, Molina et al. 2020). The RAF is a universal function approximator that can approximate any existing activation function. The advantage of using RAFTs over fixed activation functions (FAF) such as ReLU or GELU, is that the model can learn the optimal activation function from the data during (pre)training without the need to consider the choice of activation function as an additional dimension during hyperparameter tuning.³

²In our preliminary experiments, the performance of BERT becomes worse on downstream tasks when the activation functions are changed after pre-training.

³Liu et al. (2019a) consider different activation functions during Neural Architecture Search (Zoph and Le, 2017), but this becomes quickly infeasible for compute-intensive experi-

¹Code, models, and datasplits are available on GitHub <https://github.com/UKPLab/2022-RAFT>.

To evaluate the effectiveness of RAFs, we pre-train two encoder-only Transformers using RAF and GELU respectively, within an academic budget. In our experiments, we find that:

- The RAF-based Transformer (RAFT) learns different activation functions at different layers after pre-training with shapes that differ from frequently used activation functions.
- During fine-tuning, RAFT outperforms its fixed activation function counterpart (FAFT) on the general language understanding benchmark (GLUE) and the SQuAD machine reading comprehension dataset in various settings.
- After fine-tuning, the learned RAFs of the top layers are more task-specific and change the most, which are corresponding to layer behaviors of Transformers according to prior work (Mosbach et al., 2020; Merchant et al., 2020; Zhou and Srikumar, 2022). This provides new opportunities to analyze language models with respect to their learned activation functions at different layers for different tasks.
- RAFT boosts the performance when combined with a parameter-efficient fine-tuning approach, i.e., BitFit (Ben Zaken et al., 2022), which improves the model performance by 3.08 points in full-data scenario.

2 Related Work

Activation functions. There exists various predefined activation functions such as Sigmoid, Hyperbolic Tangent (Tanh), Rectified Linear Unit (ReLU, Fukushima 1969), and Gaussian Error Linear Unit (GELU, Hendrycks and Gimpel 2016). There are also approaches that leverage automatic search to obtain optimal combinations of several base activation functions in a predefined search space (Ramachandran et al., 2018; Manessi and Rozza, 2018; Sütfeld et al., 2020; Bingham and Miikkulainen, 2022; Bingham et al., 2020). For instance, Ramachandran et al. (2018) discovered the Swish activation function by using this method. Bingham et al. (2020) show that further extending the search space using evolutionary algorithms can also lead to an improvement. Finally, several search-based works investigate how to train a combination of a set of activation functions to better adapt to specific tasks and architectures (Manessi and Rozza, 2018; Sütfeld et al., 2020; Bingham and Miikkulainen, 2022). One substantial drawback of these search-based methods is that they are computationally expensive. Especially for pre-trained language models where pre-training is costly, it is infeasible to perform a hyperparameter search for selecting the best activation function (even more so their combination). In contrast, the flexibility of rational activation functions (RAFTs) allows them to be trained along with the model parameters in an end-to-end fashion (Molina et al., 2020). Therefore, they can learn the optimized activation function from data during training. RAFs have been successfully used in deep reinforcement learning for improving plasticity (Delfosse et al., 2021), cell detection models in biology (Prangemeier et al., 2020), and adapter architectures (Moosavi et al., 2022).

2018; Sütfeld et al., 2020; Bingham and Miikkulainen, 2022). One substantial drawback of these search-based methods is that they are computationally expensive. Especially for pre-trained language models where pre-training is costly, it is infeasible to perform a hyperparameter search for selecting the best activation function (even more so their combination). In contrast, the flexibility of rational activation functions (RAFTs) allows them to be trained along with the model parameters in an end-to-end fashion (Molina et al., 2020). Therefore, they can learn the optimized activation function from data during training. RAFs have been successfully used in deep reinforcement learning for improving plasticity (Delfosse et al., 2021), cell detection models in biology (Prangemeier et al., 2020), and adapter architectures (Moosavi et al., 2022).

Model	Act. Funct.
BERT (Devlin et al., 2019)	GELU
GPT-1 (Radford et al., 2018)	GELU
RoBERTa (Liu et al., 2019b)	GELU
XLNet (Yang et al., 2019)	GELU
ALBERT (Lan et al., 2020)	GELU
GPT-2* (Radford et al., 2019)	GELU
Megatron-LM (Shoeybi et al., 2019)	GELU
ELECTRA ⁺ (Clark et al., 2020)	GELU
T5 (Raffel et al., 2020)	ReLU
T5v1.1 (Raffel et al., 2020)	GeGLU
DeBERTa ⁺ (He et al., 2021)	GELU
BART (Lewis et al., 2020)	GELU
GPT-3* (Brown et al., 2020)	GELU
Jurassic* (Lieber et al., 2021)	GELU
Gopher* (Rae et al., 2021)	GELU
Megatron-Turing NLG* (Smith et al., 2022)	GELU
Chinchilla* (Hoffmann et al., 2022)	GELU
CANINE ⁺ (Clark et al., 2022)	GELU
LaMBDA (Thoppilan et al., 2022)	GeGLU
OPT (Zhang et al., 2022)	ReLU

Table 1: Activation functions in different NLP Transformer models. Models marked by * do not explicitly state the activation function but refer to GPT-1 as the base architecture (⁺ refers to BERT respectively). GeGLU is a variant that combines GELU and GLU.

Frequently used activation functions in NLP.

Table 1 shows a list of 20 different language models that have been introduced after BERT. As we see, the vast majority of the works (80%) use the GELU activation function. Moreover, many works even do not explicitly state the used activation function (45%). There are only a few works that investigate the impact of activation functions on pre-trained Transformer models. So et al. (2021) leverage automatic search methods to identify more efficient Transformer architectures. They find that a combi-

nation of squared ReLU used in the feedforward network (FFN) layer and a convolution layer added in self-attention can lead to a substantial boost in performance. Shazeer (2020) replace the FFN in the Transformer with a gated linear unit (GLU, Dauphin et al. 2017) combined with different activation functions and find a higher performance during pre-training as well as on downstream tasks. In our work, we do not change the structure of FFNs and only replace activation functions in them.

Closest to our work is the work by Moosavi et al. (2022) who investigate the use of RAF in adapters (Houlsby et al., 2019); i.e., lightweight layers that are added on top of pre-trained Transformer layers. They propose adaptable adapters that consist of RAFs and learnable switches to select a subset of adapter layers during training. They show that using both RAFs and a fewer number of adapter layers results in considerable performance gains, especially in low-data settings. However, only using RAF instead of ReLU does not result in a considerable gain in their experiments. Furthermore, adapter layers are only added and updated during fine-tuning, as a result using RAF in adapter layers has a limited impact compared to already applying them for pre-training.

In this work, we show that using RAF in Transformer layers brings additional flexibility to the model to learn the optimized activation function for each of its layers during training, and that this additional flexibility benefits both pre-training and fine-tuning steps.

3 RAFT: RAF-based Transformers

We adopt the BERT architecture (Devlin et al., 2019) where all activation functions in feedforward layers $Activation(W_1X)W_2$ are replaced with rational activation functions (illustrated in Appendix A). The equation of rational activation function $F(x)$ is as below:

$$F(x) = \frac{P(x)}{Q(x)} = \frac{\sum_{j=0}^m a_j x^j}{1 + |\sum_{k=0}^n b_k x^k|} \quad (1)$$

Where a and b are learnable parameters, and m and n are degrees of $F(x)$, which decide the complexity and fitting ability of rational functions. Following Molina et al. (2020), we use the *safe PAU* formulation that further stabilizes training.

Selecting m and n . Similar to Taylor series, the higher the degrees m and n are, the more precise

is the approximation of rational functions. However, indefinitely increasing the degrees also means adding more complexity and increasing training time. The challenge is to find suitable degrees that leads to rational functions with a strong fitting ability while keeping their complexity as low as possible. As this is still an open question, we set the search space of m and n to $\{4, 5\}$, and evaluate their ability to approximate the GELU function in the range of $[-3, 3]$. Our results show that using $m = 5$ and $n = 4$ perfectly fits the GELU function with a low complexity and thus, are adopted in this work (cf. Figure 5, Appendix B). This matches the findings in previous work (Telgarsky, 2017; Molina et al., 2020; Delfosse et al., 2021) as well. So overall, each rational activation function adds nine parameters, resulting in a total of 108 additional parameters in a 12-layer Transformer model (less than 0.000098% of its original parameters). The weights of $F(x)$ can further be initialized to approximate any existing activation functions. In our experiments, we initialize it with weights that approximate GELU.

4 Pre-training

To evaluate the viability of RAFT, we pre-train two comparable Transformer models from scratch—one using the common fixed GELU activation function (FAFT), and another one using RAFs (RAFT).

Model architecture. For our experiments, we use a frequently considered model configuration and train 12 Transformer encoder layers with a hidden size of 768 and 12 attention heads (Devlin et al., 2019; Liu et al., 2019b; Rae et al., 2021; Zhang et al., 2022). The only difference between RAFT and FAFT is the use of RAFs instead of GELUs as activation functions.

Data. We use English Wikipedia as our pre-training data.⁴ The dataset consists of 3.8×10^9 tokens from which we select 50k sentences containing 6.4×10^6 tokens as the validation data.

Pre-training objective. Following RoBERTa (Liu et al., 2019b), we use dynamic masked language modeling (MLM) as our learning task and randomly mask tokens in the input sentences at each step before feeding them into the model. We use the same masking probabilities and mask 15% of the tokens with an 80% chance of replacing them

⁴<https://dumps.wikimedia.org>

Model	Validation loss	Validation PPL
FAFT	1.645	5.18
RAFT	1.611	5.00

Table 2: Performance of the models on the validation set after pre-training.

with the [MASK] token, a 10% chance of replacing them with a randomly selected different token, and a 10% chance of not replacing them at all.

Training parameters. As our primary goal is to validate the effectiveness of RAFs in Transformers rather than releasing a RoBERTa-like model, we focus on training two comparable models within a limited training budget. Both models are optimized using AdamW (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and a weight decay of 0.01. The learning rate lr_θ is set to $7E-4$ for both models while the learning rate lr_{RAF} for the RAF coefficients is set to $5E-3$. Both learning rates are warmed up over the first 1% steps, then lr_θ decays linearly while lr_{RAF} remains constant.⁵ The batch size is set to 4096. Tuning hyperparameters during pre-training is expensive, to conduct hyperparameter tuning of both models with limited resources, we follow up 24hour BERT (Izsak et al., 2021) to pre-train the model for 23k steps equipped with various methods to accelerate training, including mixed-precision, sparse output prediction, fused linear layer, and tied embeddings (Press and Wolf, 2017). Detailed parameters and results of hyperparameter tuning are provided in Appendix C. It takes ~ 16 hours for RAFT and ~ 12 hours for FAFT using four A100 GPUs.

Results. Table 2 shows the MLM validation losses and validation perplexity of the best performing hyperparameter configuration for RAFT and FAFT. We observe that RAFT achieves a bit lower perplexity than FAFT during pre-training. The learned RAFs vary across different layers after pre-training (cf. Figure 6, Appendix E). More analysis is conducted in Section 6.

5 Fine-tuning

We conduct experiments on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019) and SQuAD (Rajpurkar

et al., 2016) to see how well pre-trained RAFs can adapt to downstream tasks. Dataset descriptions are provided in Appendix D. We further investigate the flexibility of the pre-trained RAFs by considering different training data sizes especially in a low-data regime. We fine-tune RAFT in two different settings:

- RAFT^{full}: We fine-tune the whole model, i.e., all model parameters including the RAFs.
- RAFT^{fixed}: We fix the pre-trained RAFs and only tune the rest of the parameters.

5.1 Evaluation on the GLUE Benchmark

We evaluate pre-trained models on GLUE benchmark in different data settings: (a) the full-data scenario, and (b) two low-data scenarios when only 100 or 300 labelled examples are available.

Experimental Setup. We split 75% of the training dataset as the training set and use the remaining 25% as the development set in the full-data scenario. Following previous works, we use the provided development set as the test dataset. For our low-data scenarios, we randomly sample 100 or 300 examples with ten different random seeds and report the average and standard deviation across all runs. For the full-data scenario, we report the average and standard deviation of the results across six runs with different random seeds. We use the same evaluation metrics as proposed in the GLUE benchmark; more specifically, for MRPC, QQP, and STSB, we use the average of the two corresponding metrics as the final score.⁶

Results. Table 3 shows the performance of RAFT and FAFT on the GLUE benchmark. We observe that on average, RAFT achieves consistent improvements in all data settings. We further find that especially in the low-data scenarios, the flexible activation functions of RAFT substantially outperform their static GLUE counterparts of the FAFT model. For 100 examples, RAFT achieves better results in seven out of eight tasks, outperforming FAFT by 5.31 points (RAFT^{full}) and 5.71 points (RAFT^{fixed}) on average, respectively. While the performance gap becomes smaller as the number of examples increases, the tendency remains the same with an average performance gain of 0.98

⁵We find in our preliminary experiments that a constant rational learning rate with warm up leads to better results.

⁶Note that the full-data scenario is computationally more expensive to run, but also more stable as the training instances experience less variability.

Model	ColA	SST2	MRPC	QQP	STSB	MNLI-matched/mismatched	QNLI	RTE	Avg.
<i>low-data 100 examples¹</i>									
FAFT	1.88±2.27	71.02±5.61	74.88±0.23	55.19±5.96	57.57±8.32	32.86±1.50/32.92±1.46	53.34±3.24	53.14±1.67	48.07
RAFT ^{full}	4.38±3.2	73.28±3.95	75.89±1.39	62.65±2.86	70.30±3.44	38.31±1.87/39.06±2.35	63.58±3.74	53.0±1.91	53.38
RAFT ^{fixed}	7.25±4.77	72.04±5.04	75.76±0.65	62.15±4.09	71.39±3.56	39.3±1.60/40.4±1.73	63.13±3.05	52.6±2.99	53.78
<i>low-data 300 examples¹</i>									
FAFT	13.12±5.29	77.67±3.07	79.37±1.56	66.63±1.35	76.70±1.89	43.74±2.20/45.33±2.29	69.17±2.25	55.45±2.66	58.58
RAFT ^{full}	12.36±5.07	78.22±2.10	77.84±1.09	68.25±1.01	79.77±2.34	45.70±1.69/47.27±1.86	71.92±1.10	54.70±2.26	59.56
RAFT ^{fixed}	17.34±3.23	78.95±2.33	76.97±0.96	68.20±0.76	80.32±0.1	45.35±1.62/46.53±1.63	72.07±1.56	55.78±2.72	60.17
<i>Full data²</i>									
FAFT	43.18±1.52	89.2±0.63	86.42±1.37	88.08±0.08	87.08±0.21	80.92±0.21/81.78±0.22	89.42±0.38	62.22±1.35	78.70
RAFT ^{full}	45.84±1.47	89.85±0.45	87.21±0.54	88.27±0.10	86.96±0.29	80.88±0.22/81.85±0.23	89.32±0.20	64.44±2.49	79.40
RAFT ^{fixed}	45.66±1.55	90.06±0.70	86.36±1.03	88.21±0.06	86.64±0.24	81.10±0.22/82.06±0.21	89.36±0.34	63.90±2.85	79.28

¹ Results are averaged over ten random seeds: 5309, 202206, 20220602, 2259, 49, 2022, 1046, 622, 320, 53

² Results are averaged over six random seeds: 5309, 202206, 20220602, 2259, 49, 2022

Table 3: The performance of RAFT and FAFT on the GLUE benchmark across different data sizes. RAFT^{full} fine-tunes all model parameters including RAFs. RAFT^{fixed} instead fixes the RAFs pre-training.

points (RAFT^{full}) and 1.59 points (RAFT^{fixed}) for 300 examples. In the full data scenario, RAFT still outperforms FAFT by 0.7 (RAFT^{full}) and 0.58 (RAFT^{fixed}) points on average.

Our experiments indicate that fixing the RAFs is a better choice for the GLUE benchmark in the low-data scenarios. We conjecture that one reason for this may be that the number of instances to tune all parameters of the model are insufficient. On the contrary, we find that in the full-data scenario tuning RAFs can lead to better results. The increasing number of instances especially benefit RAFs as they can better adapt to different downstream tasks and learn better features. We provide further analysis in Section 6.

5.2 Evaluation on SQuAD

Similar to GLUE, we evaluate models on SQuAD v1.1 in different data settings: (a) the full-data scenario, and (b) four low-data scenarios with 100, 300, 500, and 1000 training examples.

Experimental Setup. We split the official training data into separate training (75%) and development sets (25%)⁷ and use the official development set as the test data. We evaluate the results by computing the F1 score over the word overlap of the predicted answer and the gold answer. The hyperparameters search space is provided in Appendix C.

Results. Table 4 shows our results of RAFT and FAFT. Compared to GLUE, that consists of sentence-level text matching tasks, SQuAD is a more complex task in which the model needs to comprehend a longer text sequence to predict an answer span. The increased task difficulty is especially reflected in the low-data scenarios, as the

⁷Again, we use the development set to identify the best performing model across all epochs.

	100 examples ¹	300 examples ¹	500 examples ¹	1000 examples ¹	full data ²
FAFT	12.72±1.54	22.11±2.46	26.46±1.42	34.58±1.68	72.33±0.23
RAFT ^{full}	11.81±0.95	19.49±2.01	26.68±1.91	36.69±1.56	74.45±0.47
RAFT ^{fixed}	12.19±1.08	19.00±2.68	26.27±1.39	35.98±1.81	74.38±0.25

¹ Results are averaged over ten random seeds: 5309, 202206, 20220602, 2259, 49, 2022, 1046, 622, 320, 53

² Results are averaged over six random seeds: 5309, 202206, 20220602, 2259, 49, 2022

Table 4: Results of RAFTs and FAFT on SQuAD.

	Validation Loss	Validation PPL.
Identity	Divergent	Divergent
RELU	1.626	5.08
GELU	1.611	5.00

Table 5: Different initializations of RAF.

performances of both models are below 25 points when only 100 or 300 annotated examples are available. As a result, when there are not enough annotated examples available to learn the task, the use of RAFs instead of GELU is not beneficial for the Transformer model. However, we again see that RAFT outperforms the FAFT model as enough training examples become available.

In addition, we observe that tuning RAFs during fine-tuning (RAFT^{full}) is more beneficial compared to fixing RAFs (RAFT^{fixed}) when the task is more complex. Considering our findings on the GLUE benchmark, we conjecture that the task difficulty may play an additional role besides the amount of available training data for the performance of RAFT^{full} vs. RAFT^{fixed}; however, this remains to be investigated in future work.

6 Analysis

Impact of RAF initialization. To investigate how initialization affects the performance of RAFT, we train RAFT models initialized with GELU, RELU, and the identity function. Other hyperparameters are the same as those in section 4. Table 5 shows the performance of different initialization

	SNLI	Trivia QA	
		verified-web	verified-wiki
FAFT	74.22±0.19	24.62±1.48	21.01±0.75
RAFT ^{full}	74.80±0.29	25.40±1.84	21.50±0.76
RAFT ^{fixed}	74.76±0.25	25.40±1.25	21.78±0.87

Table 6: Zero-shot performance of FAFT and RAFT. Models evaluated on SNLI are trained on MNLI. Results on TriviaQA are based on models trained on SQuAD.

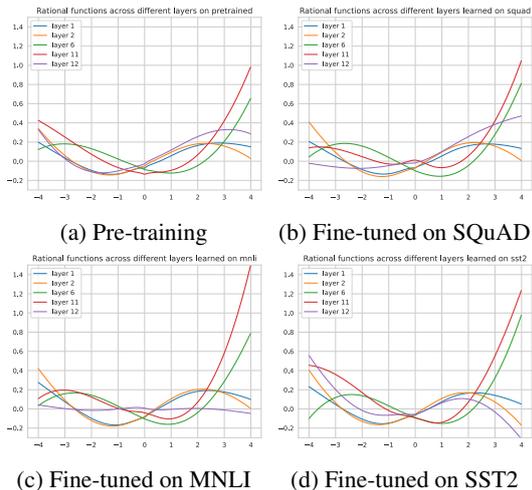


Figure 1: Rational activation functions of RAFT^{full} among different layers after pre-training and fine-tuning.

methods during pre-training. As we can see, choosing common activation functions such as ReLU or GELU leads to a similar performance while using the identity function for initialization leads to divergence.

Zero-shot generalization. To investigate if the higher performances of RAFT vs FAFT come from overfitting on the in-domain data, we conduct cross-domain zero-shot experiments. We use the models that have been fine-tuned on MNLI and SQuAD in the full-data scenario and evaluate them on the same tasks but for different data, namely, SNLI (Bowman et al., 2015) and TriviaQA (Joshi et al., 2017), respectively. MNLI and SNLI are both datasets that aim to evaluate natural language inference while SQuAD and TriviaQA contain examples for evaluating reading comprehension in different domains. Table 6 shows the results of our zero-shot evaluation. We observe that the increased flexibility and adaptivity of RAFT does not negatively impact its generalization capabilities. In fact, both variants of RAFT consistently achieve better performance than the corresponding FAFT model.

Visualizing learned RAFs. Next, we analyze how the shapes of RAFs change after pre-training and fine-tuning. First, we analyze the learned RAFs in different layers of RAFT after pre-training. As shown in Figure 1a, rational functions have different shapes across different layers, none of which are similar to GELU, or other commonly used activation functions in Transformers (cf. Table 1). This indicates that different layers may need different activation functions to achieve the optimal performance. Moreover, we see that some features like monotonicity that often are deemed to be good for predefined activation functions are not necessary, which is in line with the findings of the Swish activation function (Ramachandran et al., 2018).

Second, we analyze how the learned RAFs during pre-training change after fine-tuning in RAFT^{full}. Figures 1b–1d show learned RAFs after fine-tuning RAFT^{full} on SQuAD, MNLI and SST2 datasets. We observe that some of the learned RAFs trained on these three tasks differ from each other and the RAFs after pre-training. We further see that several RAFs between both tasks have similar shapes but different slopes across many layers.

To better understand the behavior of learned RAFs after fine-tuning in different layers on various tasks, we plot RAFs from the same layer together across all tasks. Figure 2 shows the learned RAFs in layer 1 (the bottom layer), layer 6, and layer 12 (the top layer) after pre-training and fine-tuning on different tasks. We observe that after fine-tuning, the RAFs in the top layer are more task-specific and change the most, compared to those in bottom layers. This is in line with prior work that analyzed the behavior of BERT layers during fine-tuning, which showed that higher layers exhibit more changes compared to lower layers (Mosbach et al., 2020; Merchant et al., 2020; Zhou and Sriku-mar, 2022). Our results confirm this finding from the perspective of learned activation functions. It also demonstrates that RAFs can self-adapt to different layers and tasks during fine-tuning. In addition, an interesting observation is that the output ranges of the RAFs of MNLI and QQP in the top layer are very close to zero. The output of the FFN layer $LayerNorm(FFN(x) + x)$ consists of two parts: the feedforward branch $FFN(x)$ and the skip connection branch x . The very small output of activation functions may indicate that the FFN branch of the top layer does not contribute much to the final model performance on MNLI and QQP and

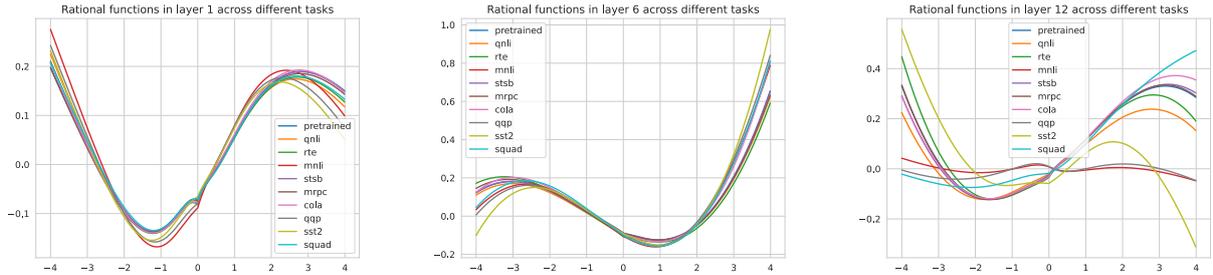


Figure 2: Learned rational activation functions of RAFT^{full} in layers 1 (bottom), 6, and 12 (top) among different tasks.

Model	CoLA	SST2	MRPC	QQP	STSB	MNLI-matched/mismatched	QNLI	RTE	Avg.
<i>low data 100 examples¹</i>									
<i>BitFit</i> _{FAFT}	1.44±2.85	63.33±9.63	68.82±1.74	55.49±3.94	46.04±24.69	32.92±1.33/32.95±1.24	51.95±3.50	52.20±2.82	45.02
<i>BitFit</i> _{full}	4.39±3.41	76.49±1.90	74.11±1.04	61.53±3.09	50.41±20.20	33.75±1.38/33.81±1.30	57.22±6.15	50.83±2.74	49.17
<i>BitFit</i> _{fixed}	6.25±3.68	75.96±1.24	74.71±0.34	61.35±3.42	49.91±26.88	33.73±1.40/ 34.04±1.71	53.19±4.02	51.63±2.26	48.97
<i>Full data¹</i>									
<i>BitFit</i> _{FAFT}	37.75±1.26	87.80±0.67	82.94±1.20	81.35±0.13	59.29±33.04	71.94±0.38/73.57±0.38	85.38±1.07	55.89±1.70	70.66
<i>BitFit</i> _{full}	38.46±1.37	88.19±0.16	86.73±1.00	81.03±0.12	85.28±0.33	70.23±0.41/72.53±0.33	80.51±10.75	60.72±1.88	73.74
<i>BitFit</i> _{fixed}	39.96±1.95	88.46±0.28	84.91±5.10	81.02±0.14	85.55±0.44	71.25±0.19/73.26±0.36	77.23±14.23	60.15±0.90	73.53

¹ Results are averaged over five random seeds: 5309, 202206, 20220602, 2259, 49

Table 7: Comparison between RAFT and FAFT combined with BitFit.

thus could be pruned. We leave this as future work.

RAFT^{fixed} vs. RAFT^{full}. In our experiments on GLUE and SQuAD (Tables 3 and 4), we observe that fixing the RAFs after fine-tuning (RAFT^{fixed}) often achieves the best or second best performance compared to the full-tuning model (RAFT^{full}) and FAFT. Fine-tuning RAFs results in higher performances when (a) more data is available, i.e., the full-data scenario in GLUE, or (b) the input task is more complex such as in SQuAD. We hypothesize that training RAFs during fine-tuning will be more effective when evaluated on more complex tasks and datasets than the ones used this work.

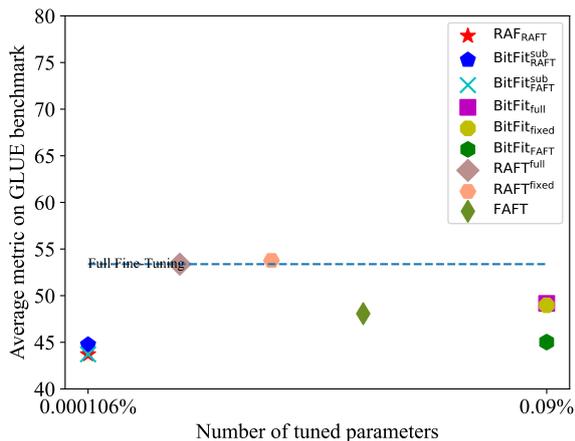
Efficiency comparison between RAFT and FAFT. In RAFT, RAFs are polynomial ratios and their coefficients are learned during training, which adds extra computation overhead. We use RAFs library with CUDA extension to accelerate. As shown in Table 8, RAFT is slower than FAFT during training since RAFs need to be updated (36.8% slower at pre-training, 14.8% slower at fine-tuning). However, RAFT is faster when doing inference due to the CUDA implementation (13.8% faster at pre-training, 3.9% faster at fine-tuning).

Parameter-efficient fine-tuning with RAFTs. In contrast to fine-tuning all parameters in a pre-trained language model, parameter-efficient tuning techniques that freeze the majority of

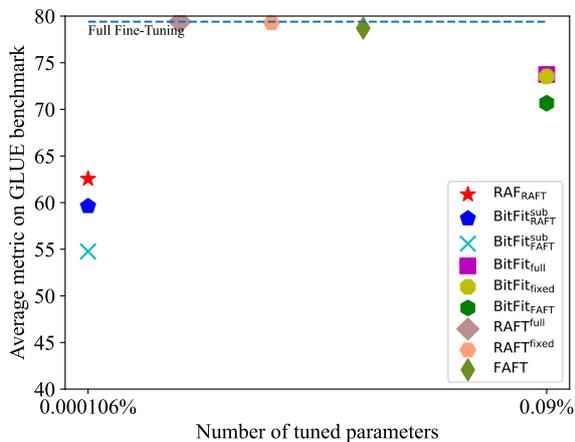
steps/second	Pre-training		Fine-tuning	
	Train	Inference	Train	Inference
RAFT	0.38	3.3	12.54	71.05
FAFT	0.52	2.9	14.4	68.38

Table 8: Number of steps per second for training and inference for RAFT and FAFT.

pre-trained parameters and only fine-tune a small set can be promising alternatives (Ding et al., 2022). One such method is BitFit (Ben Zaken et al., 2022) which only updates the bias terms in the Transformer model. To investigate the effectiveness of RAFT in a parameter-efficient fine-tuning paradigm, we fine-tune the FAFT and RAFT models with BitFit on the GLUE benchmark. We use the same settings as in our previous experiments and test RAFT and FAFT in three configurations in the low-data 100 and full-data scenario: (a) *BitFit*_{FAFT} uses BitFit with FAFT, (b) *BitFit*_{full} uses BitFit with RAFT^{full}, and (c) *BitFit*_{fixed} uses BitFit with RAFT^{fixed}. As shown in Table 7, RAFT-based BitFit achieves higher performance than the FAFT on average in both data settings: *BitFit*_{fixed} achieves 3.95 points improvements and *BitFit*_{full} gets 4.15 points improvements in the low-data scenario while *BitFit*_{fixed} performs better with a 2.87 points boost and *BitFit*_{full} performs better with a 3.08



(a) Comparison performance in low-data 100 scenario



(b) Comparison performance in full-data scenario

Figure 3: The number of parameters vs. the performance for fine-tuning of RAFT and FAFT.

points boost in the full-data scenario. It is worth noting that in some tasks, the reported results have a very large standard deviation (e.g., 33.04 for $BitFit_{FAFT}$ on STSB) due to several random seed runs not converging. In our experiments, BitFit is not as stable as fine-tuning the whole model.

How much can we achieve by only fine-tuning RAFs? To see to what extent the model can learn from different tasks by only updating RAFs, we conduct experiments to only tune RAFs on the GLUE benchmark in low- and full-data settings. We call this setup where only 117⁸ parameters of the RAFs are updated during fine-tuning, RAF_{RAFT} .

For comparison, we tune our models with the BitFit setting using the same amount of parameters,

⁸Including RAF in the pooling layer for classification

i.e., 117.⁹ $BitFit_{FAFT}^{sub}$ represents tuning the subset of BitFit of FAFT, and $BitFit_{RAFT}^{sub}$ represents tuning the subset of BitFit of RAFT. The result is presented in Appendix F (Table 13). To compare it from a broader view, we plot Figure 3 based on Table 3, Table 7 and Table 13. We observe that if only a few annotated examples are available (100 examples), $BitFit_{fixed}$ and $BitFit_{full}$ can achieve better performance than full fine-tuning of FAFT. Only fine-tuning 117 parameters ($BitFit_{FAFT}^{sub}$, $BitFit_{RAFT}^{sub}$ and RAF_{RAFT})—i.e., a negligible number of parameters compared to 110M parameters in FAFT—results in a comparable performance as fine-tuning all the parameters with only a drop of 4.21–6.68 percentage points. In the full-data scenario, the performance of BitFit ($BitFit_{full}$, $BitFit_{fixed}$ and $BitFit_{FAFT}$) lags behind full fine-tuning of both models. Only tuning RAFs or a subset of BitFit cannot achieve comparable results as well. However, RAF_{RAFT} outperforms $BitFit_{FAFT}^{sub}$ by 7.8% and performs better than $BitFit_{RAFT}^{sub}$ by 2.94% in this setting.

7 Conclusion and Future Work

In this work, we propose to utilize rational activation functions (RAF) in Transformers to directly learn optimal activation functions from data during pre-training and fine-tuning. To evaluate the effectiveness of rational activation functions, we pre-trained a Transformer-based language model, namely, RAFT. RAFT achieves a lower validation perplexity than FAFT during pre-training. Our experimental results show that RAFT performs better than FAFT in general language understanding tasks and reading comprehension tasks across different data size scenarios. We further visualize and analyze rational activation functions across different layers and tasks after pre-training and fine-tuning and find that they can substantially vary across different layers and tasks. This provides us a new way to analyze and better understand Transformer-based language models. For instance, we can investigate whether layers with similar rational activation functions encode similar linguistic properties. We further find that some layers exhibit a close to zero throughput of the rational activation function which indicates that the corresponding feedforward layer does not contribute too much to a model’s prediction. We consider these as our future work.

⁹Note that we also update the classification head in all models and experiments.

Limitations

Limited training resources. This work evaluates the effectiveness of rational activation Transformers using limited GPU resources. To provide a fair comparison, we train and release RAF- and GELU-based models for a reduced GPU budget; hence, they are not comparable to publicly available large pre-trained models such as RoBERTa-base etc. Still, a fully pre-trained RAFT could be released once more GPU resources are available. We furthermore note that we use GELU activation functions and the original FFN architecture as our baseline as it is dominantly used in existing models.

Societal impact. The main focus of this work is the evaluation of trainable activation functions. While our visualization of the learned activation functions show that they exhibit substantial differences depending on the downstream task, further analysis is necessary to better understand and interpret the shapes. Moreover, it is unclear if the additional flexibility of the models may increase their susceptibility towards capturing biases in the data. At the same time, we conjecture that especially susceptible models could also be used as good indicators to detect such biases.

Acknowledgements

We thank Quentin Delfosse for his continued support and valuable advice regarding the existing implementation of rational activation functions. We further thank our anonymous reviewers and Stella Biderman, Fengyu Cai, Nils Dycke, Haau-Sing Li, Andreas Rücklé, Martin Tutek, Kexin Wang, and Neha Warikoo for their fruitful discussions and helpful feedback. This work has been funded by the German Research Foundation (DFG) as part of the UKP-SQuARE project (grant GU 798/29-1), the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE and the hessian.AI Service Center.

References

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. [BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2:*

Short Papers), pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Garrett Bingham, William Macke, and Risto Miikkulainen. 2020. [Evolutionary optimization of deep learning activation functions](#). In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, page 289–296, New York, NY, USA. Association for Computing Machinery.

Garrett Bingham and Risto Miikkulainen. 2022. [Discovering parametric activation functions](#). *Neural Networks*, 148:48–65.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. [Canine: Pre-training an efficient tokenization-free encoder for language representation](#). *Transactions of the Association for Computational Linguistics*, 10:73–91.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [ELECTRA: pre-training text encoders as discriminators rather than generators](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. [The PASCAL recognising textual entailment challenge](#). In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First*

- PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.
- Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. [Language modeling with gated convolutional networks](#). In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 933–941. PMLR.
- Quentin Delfosse, Patrick Schramowski, Alejandro Molina, and Kristian Kersting. 2021. [Recurrent rational networks](#). *CoRR*, abs/2102.09407.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. 2022. [Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models](#). *CoRR*, abs/2203.06904.
- William B. Dolan and Chris Brockett. 2005. [Automatically constructing a corpus of sentential paraphrases](#). In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005*. Asian Federation of Natural Language Processing.
- Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. 2022. [Activation functions in deep learning: A comprehensive survey and benchmark](#). *Neurocomputing*, 503:92–108.
- Kunihiko Fukushima. 1969. [Visual feature extraction by a multilayered network of analog threshold elements](#). *IEEE Trans. Syst. Sci. Cybern.*, 5(4):322–333.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [Deberta: decoding-enhanced bert with disentangled attention](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Dan Hendrycks and Kevin Gimpel. 2016. [Bridging nonlinearities and stochastic regularizers with gaussian error linear units](#). *CoRR*, abs/1606.08415.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. [Training compute-optimal large language models](#). *CoRR*, abs/2203.15556.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for NLP](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Peter Izsak, Moshe Berchansky, and Omer Levy. 2021. [How to train BERT with an academic budget](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10644–10652, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. [TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. [ALBERT: A lite BERT for self-supervised learning of language representations](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. [The winograd schema challenge](#). In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*. AAAI Press.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. [Jurassic-1: Technical details and evaluation](#). *White Paper. AI21 Labs*.
- Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019a. [DARTS: differentiable architecture search](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis,

- Luke Zettlemoyer, and Veselin Stoyanov. 2019b. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Franco Manessi and Alessandro Rozza. 2018. [Learning combinations of activation functions](#). In *24th International Conference on Pattern Recognition, ICPR 2018, Beijing, China, August 20-24, 2018*, pages 61–66. IEEE Computer Society.
- Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. [What happens to BERT embeddings during fine-tuning?](#) In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.
- Alejandro Molina, Patrick Schramowski, and Kristian Kersting. 2020. [Padé activation units: End-to-end learning of flexible activation functions in deep networks](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Nafise Moosavi, Quentin Delfosse, Kristian Kersting, and Iryna Gurevych. 2022. [Adaptable adapters](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3742–3753, Seattle, United States. Association for Computational Linguistics.
- Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. 2020. [On the interplay between fine-tuning and sentence-level probing for linguistic knowledge in pre-trained transformers](#). In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 68–82, Online. Association for Computational Linguistics.
- Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. 2020. [Topology of deep neural networks](#). *J. Mach. Learn. Res.*, 21:184:1–184:40.
- Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. 2018. [Activation functions: Comparison of trends in practice and research for deep learning](#). *CoRR*, abs/1811.03378.
- Tim Prangemeier, Christoph Reich, and Heinz Koepl. 2020. [Attention-based transformers for instance segmentation of cells in microstructures](#). In *IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2020, Virtual Event, South Korea, December 16-19, 2020*, pages 700–707. IEEE.
- Ofir Press and Lior Wolf. 2017. [Using the output embedding to improve language models](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. [Improving language understanding by generative pre-training](#).
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. [Language models are unsupervised multitask learners](#). *OpenAI blog*, 1(8):9.
- Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. [Scaling language models: Methods, analysis & insights from training gopher](#). *CoRR*, abs/2112.11446.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2018. [Searching for activation functions](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.
- Siddharth Sharma, Simone Sharma, and Anidhya Athaiya. 2020. [Activation functions in neural networks](#). *International Journal of Engineering Applied Sciences and Technology*, 04:310–316.
- Noam Shazeer. 2020. [GLU variants improve transformer](#). *CoRR*, abs/2002.05202.
- Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. [Megatron-lm: Training multi-billion parameter language models using model parallelism](#). *CoRR*, abs/1909.08053.
- Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. [Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model](#). *CoRR*, abs/2201.11990.

David R. So, Wojciech Manke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V. Le. 2021. [Searching for efficient transformers for language modeling](#). In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 6010–6022.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIG-DAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Leon René Sütthof, Flemming Brieger, Holger Finger, Sonja Füllhase, and Gordon Pipa. 2020. [Adaptive blending units: Trainable activation functions for deep neural networks](#). In *Intelligent Computing - Proceedings of the 2020 Computing Conference, Volume 3*, volume 1230 of *Advances in Intelligent Systems and Computing*, pages 37–50. Springer.

Matus Telgarsky. 2017. [Neural networks and rational functions](#). In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3387–3393. PMLR.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. [Lamda: Language models for dialog applications](#). *CoRR*, abs/2201.08239.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. 2022. [Deepnet: Scaling transformers to 1, 000 layers](#). *CoRR*, abs/2203.00555.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. [Neural network acceptability judgments](#). *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. [Opt: Open pre-trained transformer language models](#). *CoRR*, abs/2205.01068.

Yichu Zhou and Vivek Srikumar. 2022. [A closer look at how fine-tuning changes BERT](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061, Dublin, Ireland. Association for Computational Linguistics.

Barret Zoph and Quoc V. Le. 2017. [Neural architecture search with reinforcement learning](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

A Model Architecture

Figure 4 shows the difference part of RAFT and FAFT.

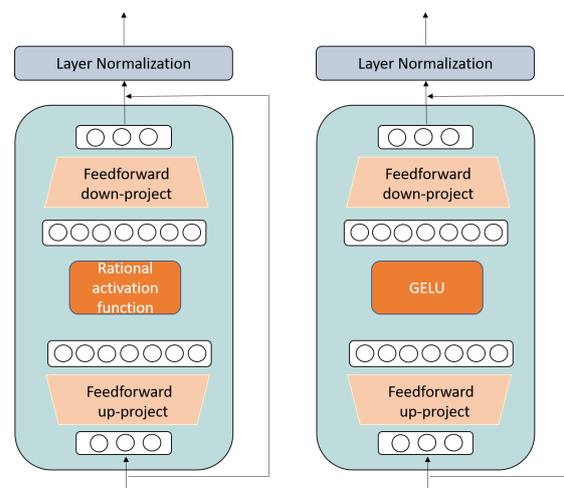


Figure 4: Rational activation function in the feed-forward layer (left) and the vanilla GELU counterpart (right).

B Fitting abilities of different degrees of Rational Functions

Figure 5 show the approximate functions of GELU using rational functions with different degrees. As we can see, when $m = 5$ and $n = 4$ or $n = 5$, rational function fit GELU very well in the same shape. Finally, it is important to note that rational functions are an universal approximator in a limited range, e.g., $[-5,5]$. Especially for out-of-bound inputs (i.e., values that are not guaranteed by rational functions), the output of rational functions may result in values very different from the approximated function (e.g., GELU). While pre-training a model from scratch with RAFs does not lead to any problem, directly replacing activation functions in pre-trained models with RAFs only for fine-tuning may lead to divergence due to out-of-bound inputs.

C Hyperparameters Tuning

C.1 Pre-training

In our preliminary experiments that some hyperparameter configurations can lead to instability during training due to diverging model updates (e.g., for $lr_\theta = 7E-4$ and batch size of 2048). To stabilize the training without having to rely on a larger warmup phase (e.g., 6% of the training steps), we instead adopt the DeepNorm (Wang et al., 2022) to initialize both models. DeepNorm stabilizes training by bounding the updates and further scaling the residual branches in Transformers. Using DeepNorm makes both models, FAFT and RAFT, achieve lower validation loss and leads to a more stable training.

We tune the learning rate lr_θ for model parameters and lr_{RAF} for RAFs, batch size, warmup steps, and learning rate scheduler as hyperparameters for both models separately. The hyperparameter search space for pre-training stage is as follows:

- Learning rate lr_θ for model parameters: 1E-4, 4E-4, 7E-4, 1E-3
- Learning rate lr_{RAF} for RAFs: 1E-3, 5E-3, 1E-2
- Batch size: 2048, 4096
- Warmup ratio: 0%, 1%, 6%

Some results of hyperparameters tuning are provided in Table 9.

Table 10 shows final hyperparameters we used for pre-training RAFT and FAFT.

	lr_θ	lr_{RAF}	Batch Size	Validation Loss
RAFT	1E-4	0.005	2048	2.217
RAFT	4E-4	0.005	2048	1.808
RAFT	7E-4	0.005	4096	1.732
RAFT	7E-4	0.005	4096	1.611
RAFT	1E-3	0.005	4096	1.638

Table 9: Part of Hyperparameters Tuning Results of RAFT

Hyperparameters	FAFT	RAFT
Peak lr_θ	7E-4	7E-4
Peak lr_{RAF}	n/a	5E-3
Learning rate decay	linear	constant
Gradient clipping	0	0
Batch size	4096	4096
Sequence length	128	128
Adam_beta1	0.9	0.9
Adam_beta2	0.98	0.98
Attention dropout	0.1	0.1
Warmup ratio	1%	1%
Training steps	23k	23k

Table 10: Hyperparameters for pre-training RAFT and FAFT

C.2 Fine-tuning

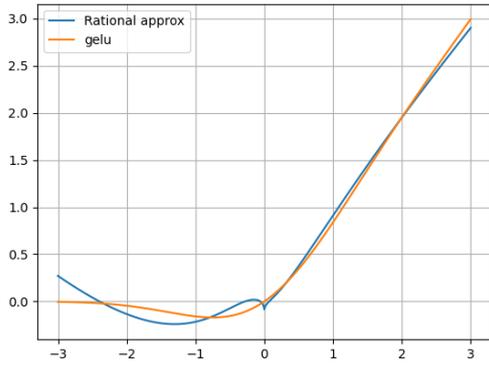
The hyperparameters search space for GLUE during fine-tuning stage is as follows:

- lr_θ : 2E-5, 5E-5
- lr_{RAF} : 1E-4, 5E-4, 1E-3, 5E-3
- Batch size: 32
- Weight decay: 0.1
- Number of epochs: 3, 10, 20

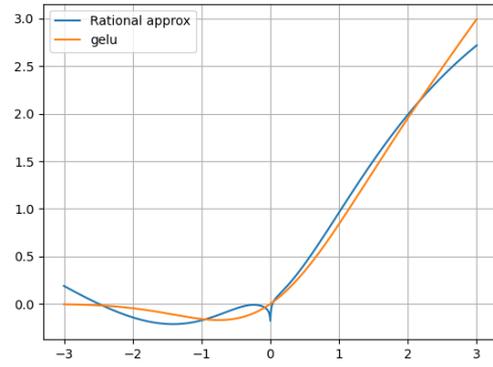
We further tune the learning rates and number of training epochs for RAFT and FAFT separately on a single random seed. For our low-data experiments we fix the number of training epochs to 20 and use early stopping with a patience of 10 epochs. For our full-data experiments, we train the large datasets (QQP, MNLI, and QNLI) for 3 epochs and the others for 10 epochs.

The hyperparameters search space for SQuAD during fine-tuning is as below:

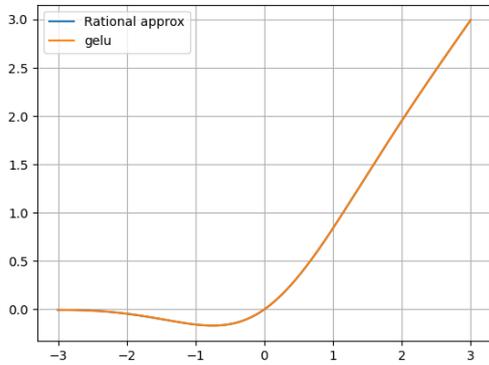
- lr_θ : 2E-5, 5E-5, 1E-4



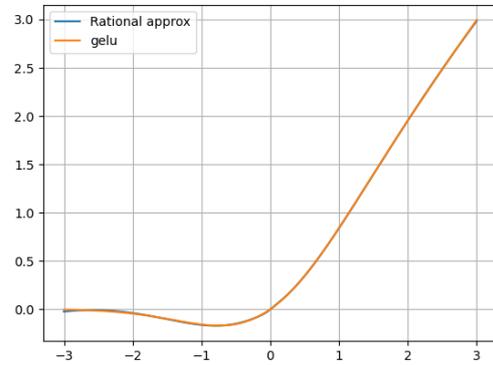
(a) Approximate function with degrees $m = 4$ and $n = 4$



(b) Approximate function with degrees $m = 4$ and $n = 5$



(c) Approximate function with degrees $m = 5$ and $n = 4$
Rational Function is overlapping with GELU



(d) Approximate function with degrees $m = 5$ and $n = 5$
Rational Function is overlapping with GELU

Figure 5: Approximate Functions of GELU using rational functions

- lr_{RAF} : 1E-4, 5E-4, 1E-3, 5E-3
- Batch size: 32
- Weight decay: 0.1
- Number of epochs: 10, 20

For our experiments, we fine-tune both models with their best performing $lr_{\theta} = 1E-4$ for 10 epochs in the full-data scenario and 20 epochs in the low-data scenario.

The hyperparameters search space for BitFit is as below:

- Learning rate lr_{θ} for model parameters: 5E-5, 1E-3, 5E-3, 1E-2
- Learning rate lr_{RAF} for RAFs: 1E-3, 5E-3, 1E-2
- Batch size: 32
- Training epochs: 3, 10, 20 epochs

We use 3 training epochs for large dataset (QQP, MNLI, QNLI), 10 epochs for other datasets and 20 epochs for low-resource scenarios. Both models can converge in the above settings.

D Data Statistics

GLUE is a collection of nine different language understanding tasks: CoLA (Warstadt et al., 2019), SST2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), QQP¹⁰, STSB (Cer et al., 2017), MNLI (Williams et al., 2018), RTE (Dagan et al., 2005), and WNLI (Levesque et al., 2012). We exclude WNLI due to the adversarial nature of its development set and the still unbeaten majority vote upper bound.¹¹

Table 11 show data statistics of GLUE benchmark.

¹⁰<https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs>

¹¹Cf. (12) in <https://gluebenchmark.com/faq>

Task	CoLA	SST2	MRPC	QQP	STSB	MNLI-matched/mismatched	QNLI	RTE
Train	8,551	67,349	3,668	363,846	5,749	392,702	104,743	2,490
Dev	1,043	872	408	40,430	1,500	9,815/9,832	5,463	277
Metric	Matthews corr.	acc.	acc./F1	acc./F1	Person/Spearman corr.	acc.	acc.	acc.

Table 11: Dataset statistics of the GLUE benchmark

SQuAD is a reading comprehension task where each example consists of a question, a context, and the respective span from the context that answers the question. Table 12 show data statistics of SQuAD.

E Learned RAFs during pre-training and after fine-tuning

Figure 6 and Figure 7 show learned RAFs in 12 layers after pre-training and fine-tuning on different tasks, respectively.

F Results of only tuning RAFs

Table 13 shows comparison results between only tuning RAFs and BitFit with the same parameters with RAFT and FAFT.

	Train	Dev	Test
SQuAD v1.1	66,236	21,530	10,789

Table 12: Statistics of SQuAD: the official training dataset is split into training and development sets, and the official development dataset is used as the test data.

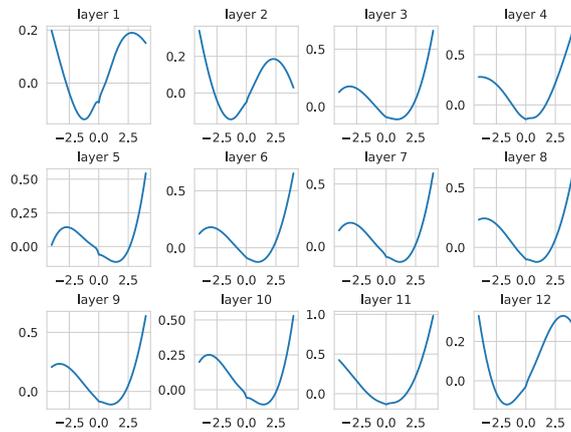


Figure 6: Learned RAFs of different layers after pre-training

Model	CoLA	SST2	MRPC	QQP	STSB	MNLI-matched/mismatched	QNLI	RTE	Avg.
<i>low data 100 examples¹</i>									
$BitFit_{FAFT}^{sub}$	1.49±1.87	62.82±7.56	74.80±0.00	52.57±3.83	14.71±7.21	32.73±1.41/32.76±1.30	49.77±0.40	50.83±1.86	41.39
$BitFit_{RAFT}^{sub}$	2.45±3.58	72.34±3.41	74.67±0.68	55.61±2.35	23.99±10.41	35.32±0.67/35.66±1.05	51.08±0.71	51.70±1.85	44.75
RAF_{RAFT}	4.33±3.02	72.91±2.82	74.47±0.88	51.92±5.03	17.27±10.60	35.24±0.61/ 35.69±0.92	51.12±0.48	50.47±1.63	43.71
<i>Full data¹</i>									
$BitFit_{FAFT}^{sub}$	6.61±7.08	79.52±0.52	71.32±0.22	70.48±0.66	37.33±5.70	53.33±1.13/55.30±0.75	64.04±2.03	54.88±1.42	54.76
$BitFit_{RAFT}^{sub}$	8.78±5.54	82.02±0.57	71.76±0.77	70.88±1.17	71.40±0.52	51.57±0.54/53.27±1.20	69.87±1.20	57.04±1.19	59.62
RAF_{RAFT}	9.71±12.04	81.70±0.12	74.81±3.09	73.57±0.48	80.79±0.60	57.34±0.19/60.69±0.51	67.89±8.64	56.53±1.83	62.56

¹ Results are averaged over five random seeds: 5309, 202206, 20220602, 2259, 49

Table 13: Comparison between fine-tuning RAFs and a subset of 117 BitFit parameters with RAFT and FAFT.

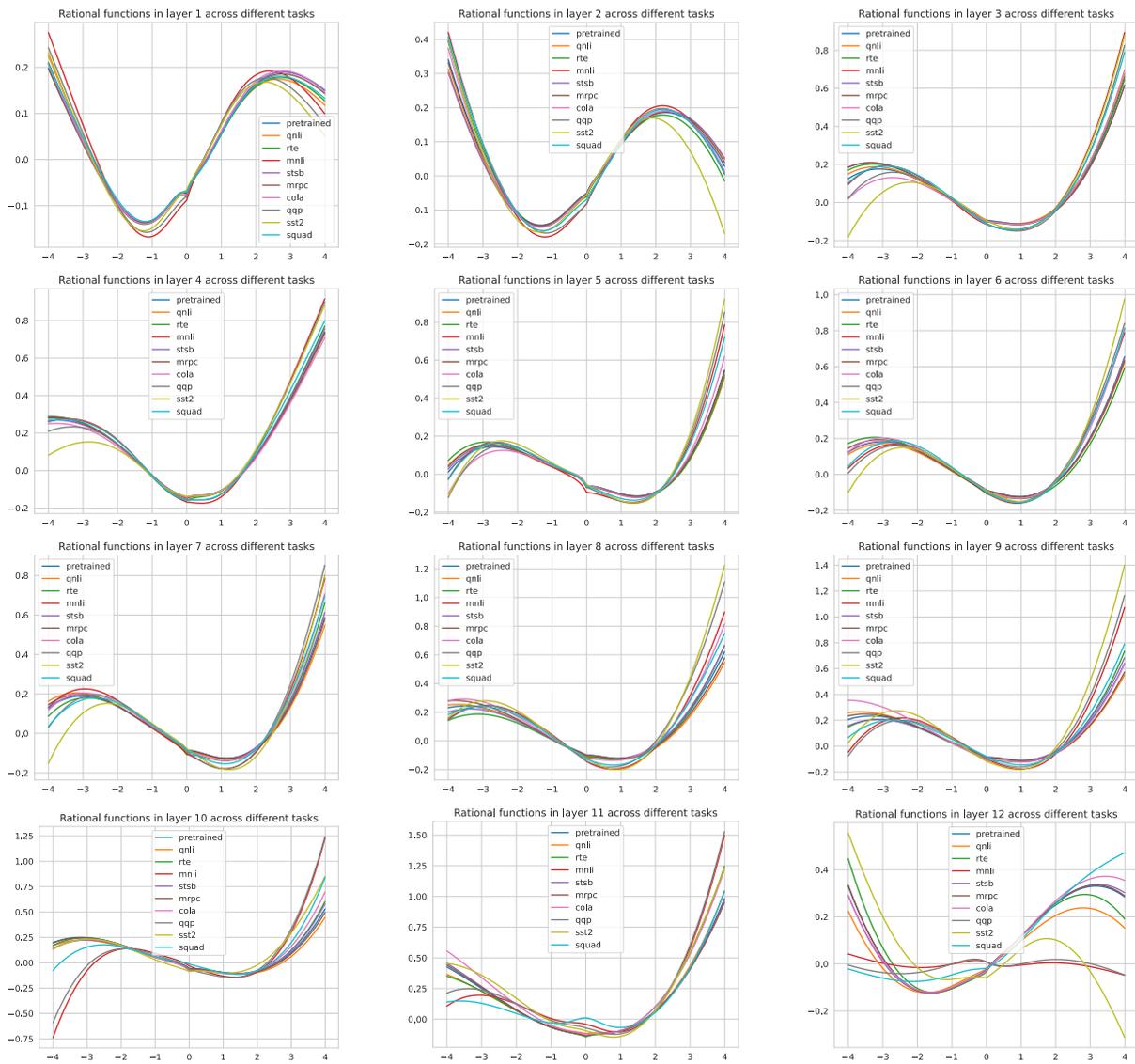


Figure 7: Learned RAFTs in 12 layers across different tasks after fine-tuning

The Solvability of Interpretability Evaluation Metrics

Yilun Zhou Julie Shah

MIT CSAIL

{yilun, julie_a_shah}@csail.mit.edu

<https://yilunzhou.github.io/solvability/>

Abstract

Feature attribution methods are popular for explaining neural network predictions, and they are often evaluated on metrics such as comprehensiveness and sufficiency. In this paper, we highlight an intriguing property of these metrics: their *solvability*. Concretely, we can define the problem of optimizing an explanation for a metric, which can be solved by beam search. This observation leads to the obvious yet unaddressed question: why do we use explainers (e.g., LIME) not based on solving the target metric, if the metric value represents explanation quality? We present a series of investigations showing strong performance of this beam search explainer and discuss its broader implication: a definition-evaluation duality of interpretability concepts. We implement the explainer and release the Python `solvex` package for models of text, image and tabular domains.

1 Introduction

For neural network models deployed in high stakes domains, the explanations for predictions are often as important as the predictions themselves. For example, a skin cancer detection model may work by detecting surgery markers (Winkler et al., 2019) and an explanation that reveals this spurious correlation is highly valuable. However, evaluating the correctness (or faithfulness) of explanations is fundamentally ill-posed: because the explanations are used to help people understand the reasoning of the model, we cannot check it against the ground truth reasoning, as the latter is not available.

As a result, correctness evaluations typically employ certain alternative metrics. For feature attribution explanations, they work under a shared principle: changing an important feature should have a large impact on the model prediction. Thus, the quality of the explanation is defined by different formulations of the model prediction change, resulting in various metrics such as comprehensiveness and

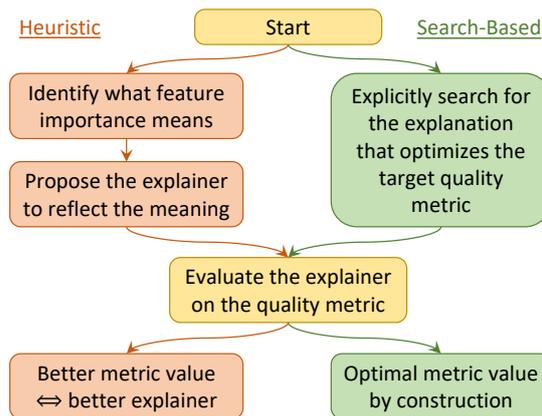


Figure 1: Left: the current process of developing new explainers. Right: the natural implication following our observation that evaluation metrics are *solvable*.

sufficiency (DeYoung et al., 2020). To develop new explanation methods (Fig. 1, left), people generally identify a specific notion of feature importance (e.g., local sensitivity), propose the corresponding explainer (e.g., gradient saliency (Simonyan et al., 2013)), evaluate it on one or more metrics, and claim its superiority based on favorable results vs. baseline explainers. We call these explainers *heuristic* as they are motivated by pre-defined notions of feature importance.

In this paper, we show that all these metrics are *solvable*, in that we can *define* an explanation as the one that optimizes a metric value and *search* for it. The obvious question is then: *if we take a specific target metric to represent correctness, why don't we just search for the metric-optimal explanation (Fig. 1, right) but take the more convoluted route of developing heuristic explanations and then evaluating them (Fig. 1, left)?*

There are several possible reasons. First, the optimization problem may be so hard that we cannot find an explanation better than the heuristic ones. The bigger concern, however, is that of Goodhart's Law. In other words, as soon as a metric is used in explicit optimization, it ceases to be a good

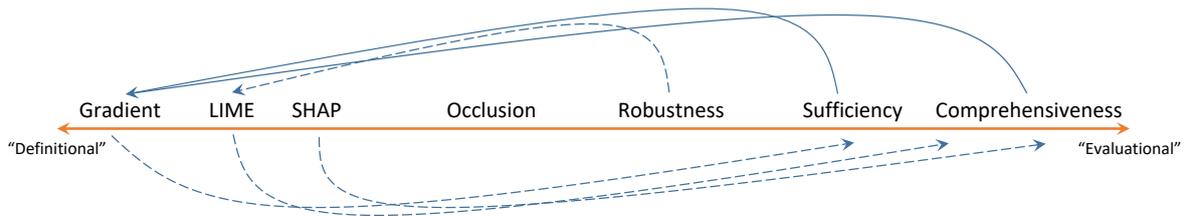


Figure 2: A definition-evaluation spectrum for various interpretability concepts currently as perceived by the community (see App. B for some justification). The proposed solvability property can move evaluational concepts towards the definitional side, for which we explore two in the paper (solid arrows). The more general definition-evaluation duality opens up new opportunities to move other concepts around (dashed arrows).

metric. Concretely, the explanation may overfit to the particular metric and perform much worse on closely related ones (Chan et al., 2022), or overfit to the model and effectively adversarially attack the model when assigning word importance (Feng et al., 2018). It may also perform poorly on evaluations not based on such metrics, such as ground truth alignment (Zhou et al., 2022a).

We assess these concerns, taking the widely used comprehensiveness and sufficiency metrics (DeYoung et al., 2020) as the optimization target. Our findings, however, largely dispel every concern. A standard beam search produces explanations that greatly outperform existing one such as LIME and SHAP on the target metric. On several other metrics, the search-based explainer also performs favorably on average. There is no strong evidence of it adversarially exploiting the model either, and it achieves competitive performances on a suite of ground truth-based evaluations.

Thus, we advocate for wider adoptions of the explainer, which is domain-general and compatible with models on image and tabular data as well. As an engineering contribution, we release the Python `solvex` package (*solvability*-based *explanation*) and demonstrate its versatility in App.A.

More broadly, the solvability phenomenon is one facet of the definition-evaluation duality, which asserts an equivalence between definitions and evaluations. Solvability recognizes that for each evaluation metric, we can define explainer that performs optimally on this metric. Conversely, for each explainer, we can also come up with an evaluation metric that ranks this explainer on top – a straightforward one would be the negative distance between the explanation under evaluation and the “reference explanation” generated by the explainer.

While the community has mostly agreed on a spectrum on which various interpretability concepts (Fig. 2) are located, duality allows every concept to be moved freely on the scale. We explored

two particular movements as represented by the solid arrows, but the more general investigation of this operation could be of both theoretical and practical interest. In addition, given that definitions and evaluations are really two sides of the same coin, we need to reflect how to best evaluate explanations. Sec. 6 argues to measure their *demonstrable utilities* in downstream tasks, and present potential ways and ideas to better align the interpretability research with such goals.

2 Background and Related Work

In this section, we give a concise but unified introduction to the popular feature attribution explainers and evaluation metrics studied in this paper.

2.1 Feature Attribution Explainers

We focus on feature attribution explanations, which explains an input $x = (x_1, \dots, x_L)$ by a vector $e = (e_1, \dots, e_L)$ where e_l represents the “contribution” of x_l to the prediction. Many different definitions for contribution have been proposed and we consider the following five.

- **Vanilla gradient (Grad)** (Simonyan et al., 2013; Li et al., 2016a) is the L2 norm of gradient of the prediction (in logit, following standard practice) with respect to the token embedding.
- **Integrated gradient (IntG)** (Sundararajan et al., 2017) is the path integral of the embedding gradient along the line segment from the zero embedding value to the actual value.
- **LIME** (Ribeiro et al., 2016) is the coefficient of a linear regression in the local neighborhood.
- **SHAP** (Lundberg and Lee, 2017) computes the Shapley value (Roth, 1988) for each word.
- **Occlusion (Occl)** (Li et al., 2016b) is the change in prediction when a word is removed from the input while all other words remain.

2.2 Feature Attribution Evaluations

Naturally, different definitions result in different explanation values. As findings (e.g., Adebayo

et al., 2018; Nie et al., 2018) suggest that some explanations are not correct (i.e., faithfully reflecting the model’s reasoning process), many evaluations are proposed to quantify the correctness of different explanations. Not having access to the ground truth model working mechanism (which is what explanations seek to reveal in the first place), they are instead guided by one principle: changing an important feature (as judged by the explanation) should have a large impact on the prediction, and the magnitude of the impact is taken as explanation quality. However, there are different ways to quantify the impact, leading to different evaluations, and we consider six in this paper.

Let $f : \mathcal{X} \rightarrow \mathbb{R}$ be a function that we want to explain, such as the probability of the target class. For an input $x = (x_1, \dots, x_L)$ of L words, according to an explanation $e = (e_1, \dots, e_L)$, we can create a sequence of $L + 1$ input deletions $\tilde{x}_e^{(0)}, \tilde{x}_e^{(1)}, \dots, \tilde{x}_e^{(L)}$ where $\tilde{x}_e^{(l)}$ is the input but with l most important features removed. Thus, we have $\tilde{x}_e^{(0)} = x$ and $\tilde{x}_e^{(L)}$ being the empty string.¹ The **comprehensiveness** κ (DeYoung et al., 2020) is defined as

$$\kappa(x, e) = \frac{1}{L + 1} \sum_{l=0}^L f(x) - f(\tilde{x}_e^{(l)}). \quad (1)$$

It measures the deviation from the original model prediction when important features (according to e) are successively removed, and therefore a larger value is desirable. It was also proposed for computer vision models as the area over perturbation curve (AoPC) by Samek et al. (2016).

Analogously, we can define the sequence of input insertions $\hat{x}_e^{(0)}, \hat{x}_e^{(1)}, \dots, \hat{x}_e^{(L)}$, where $\hat{x}_e^{(l)}$ is the input with the l most important features present. Thus, $\hat{x}_e^{(0)}$ is the empty string and $\hat{x}_e^{(L)} = x$, but otherwise the sequences of input insertions and deletions do not mirror each other. The **sufficiency** σ (DeYoung et al., 2020) is defined as

$$\sigma(x, e) = \frac{1}{L + 1} \sum_{l=0}^L f(x) - f(\hat{x}_e^{(l)}). \quad (2)$$

¹We define feature removal as the literal deletion of the word from the sentence, which is a popular practice. Other methods replace the token with [UNK], [MASK] or zero embedding, are more sophisticated such as performing BERT mask filling (Kim et al., 2020). While our current approach could lead to out-of-distribution instances, we adopt it due to its popularity. A thorough investigation for the best strategy is orthogonal to our paper and beyond its scope.

It measures the gap to the original model prediction that remains (i.e., convergence to the model prediction) when features are successively inserted from the most important to the least. Therefore, a smaller value is desirable.

Another interpretation of prediction change just considers decision flips. Let $g : \mathcal{X} \rightarrow \{0, \dots, K\}$ be the function that outputs the most likely class of an input. The **decision flip by removing the most important token** (Chrysostomou and Aletras, 2021) is defined as

$$\text{DF}_{\text{MIT}}(x, e) = \mathbb{1}_{g(\tilde{x}_e^{(1)}) \neq g(x)}, \quad (3)$$

which measures whether removing the most important token changes the decision. Across a dataset, its average value gives the overall decision flip rate, and a higher value is desirable.

The **fraction of token removals for decision flip** (Serrano and Smith, 2019) is defined as

$$\text{DF}_{\text{Frac}}(x, e) = \frac{\arg \min_l g(\tilde{x}_e^{(l)}) \neq g(x)}{L}, \quad (4)$$

and we define $\text{DF}_{\text{Frac}} = 1$ if no value of l leads to the decision flip. This metric represents the fraction of feature removals that is needed to flip the decision, and hence a lower value is desirable.

Last, two metrics evaluate correlations between model prediction and feature importance. For x and e , we define the sequence of marginal feature deletions $x_{-,e}^{(1)}, \dots, x_{-,e}^{(L)}$ such that $x_{-,e}^{(l)}$ is original input with only the l -th important feature removed. The **deletion rank correlation** (Alvarez-Melis and Jaakkola, 2018b) is defined as

$$\delta_f = [f(x) - f(x_{-,e}^{(1)}), \dots, f(x) - f(x_{-,e}^{(L)})], \quad (5)$$

$$\text{Rank}_{\text{Del}}(x, e) = \rho(\delta_f, e), \quad (6)$$

where $\rho(\cdot, \cdot)$ is the Spearman rank correlation coefficient between the two input vectors. Intuitively, this metric asserts that suppressing a more important feature should have a larger impact to the model prediction. A higher correlation is desirable.

The **insertion rank correlation** (Luss et al., 2021) is defined as

$$v = [f(\tilde{x}_e^{(L)}), \dots, f(\tilde{x}_e^{(0)})], \quad (7)$$

$$\text{Rank}_{\text{Ins}}(x, e) = \rho(v, [0, \dots, L]), \quad (8)$$

and recall that $\tilde{x}_e^{(L)}, \dots, \tilde{x}_e^{(0)}$ is the sequence of inputs with increasingly more important features inserted, starting from the empty string $\tilde{x}_e^{(L)}$ to the

full input $\tilde{x}^{(0)}$. This metric asserts that the model prediction on this sequence should increase monotonically to the original prediction. Also a higher correlation is desirable.

Related to our proposed notion of solvability is the phenomenon that some metric values seem to favor some explainers (Pham et al., 2022; Ju et al., 2022). While it is often used to argue *against* the use of certain evaluations, we take this idea to the extreme, which culminates in the solvability property, and find that metric-solving (Def. 3.1) explanations from some metrics can be high-quality.

3 The Solvability of Evaluation Metrics

Now we establish the central observation of this paper: the solvability of these evaluation metrics. Observe that each evaluation metric, e.g., comprehensiveness κ , is defined on the input x and the explanation e , and its computation only uses the model prediction function f (or g derived from f for the two decision flip metrics). In addition, the form of feature attribution explanation constrains e to be a vector of the same length as x , or $e \in \mathbb{R}^L$.

Without loss of generality, we assume that the metrics are defined such that a higher value means a better explanation (e.g., redefining the sufficiency to be the negative of its original form). We formalize the concept of solvability as follows:

Definition 3.1. For a metric m and an input x , an explanation e^* solves the metric m if $m(x, e^*) \geq m(x, e)$ for all $e \in \mathbb{R}^L$. We also call e^* the *m-solving* explanation.

Notably, there are already two explanation-solving-metric cases among the ones in Sec. 2.

Theorem 1. The occlusion explainer solves the DF_{MIT} and Rank_{Del} metrics.

The proof follows from the definition of the explainer and the two metrics. Occlusion explainer defines token importance as the prediction change when each the token is individually removed, thus the most important token is the one that induces the largest change, which makes it most likely to flip the decision under DF_{MIT} . In addition, because token importance is defined as the model prediction change, its rank correlation with the latter (i.e., Rank_{Del}) is maximal at 1.0.

Thm. 1 highlights an important question: if we take DF_{MIT} or Rank_{Del} as the metric (i.e., indicator) of explanation quality, why should we consider any other explanation, when the occlusion explanation provably achieves the optimum? A possible answer

is that the metrics themselves are problematic. For example, one can argue that the DF_{MIT} is too restrictive for overdetermined input: when redundant features (e.g., synonyms) are present, removing any individual one cannot change the prediction, such as for the sentiment classification input of “This movie is great, superb and beautiful.”

Nonetheless, the perceived quality of a metric can be loosely inferred from its adoption by the community, and the comprehensiveness and sufficiency metrics (DeYoung et al., 2020) are by far the most widely used. They overcome the issue of DF_{MIT} by also considering inputs with more than one token removed. Since a metric is scalar-valued, we combine comprehensiveness κ and sufficiency σ into comp-suff difference Δ , defined as (recall that a *lower* sufficiency value is better):

$$\Delta(x, e) = \kappa(x, e) - \sigma(x, e). \quad (9)$$

Again, we face the same question: if Δ is solvable, why should *any* heuristic explainers be used instead of the Δ -solving e^* ? In the next two sections, we seek to answer it by first proposing a beam search algorithm to (approximately) find e^* and then explore its various properties.

4 Solving Metrics with Beam Search

We first define two properties that are satisfied by some metrics: value agnosticity and additivity.

Definition 4.1. For an input $x = (x_1, \dots, x_L)$ with explanation $e = (e_1, \dots, e_L)$, we define the ranked importance as $r(x_l) = |\{e_i : e_i \leq e_l, 1 \leq i \leq L\}|$. In other word, the x_l with $r(x_l) = L$ is the most important, and that with $r(x_l) = 1$ is the least. A metric m is *value-agnostic* if for all e_1 and e_2 that induce the same ranked importance, we have

$$m(x, e_1) = m(x, e_2). \quad (10)$$

A value-agnostic metric has at most $L!$ unique values across all possible explanations for an input of length L . Thus, in theory, an exhaustive search over the $L!$ permutations of the list $[1, 2, \dots, L]$ is guaranteed to find the e^* that solves the metric.

Definition 4.2. A metric m is *additive* if it can be written in the form of

$$m(x, e) = \sum_{l=0}^L h(x, e^{(l)}), \quad (11)$$

for some function h , where $e^{(l)}$ reveals the attribution values of l most important features according to e but keeps the rest inaccessible.

Theorem 2. Comprehensiveness, sufficiency and their difference are value-agnostic and additive.

The proof is straightforward, by observing that both $\tilde{x}^{(l)}$ and $\hat{x}^{(l)}$ can be created from x and the ordering of $e^{(l)}$. In fact, all metrics in Sec. 2 are value-agnostic (but only some are additive).

A metric satisfying these two properties admits an efficient beam search algorithm to approximately solve it. As $e^{(l)}$ can be considered as a partial explanation that only specifies the top- l important features, we start with $e^{(0)}$, and try each feature as most important obtain $e^{(1)}$. With beam size B , if there are more than B features, we keep the top- B according to the partial sum. This extension procedure continues until all features are added, and top extension is then e^* . Alg. 1 documents the procedure, where $\text{ext}(e, v)$ extends e and returns a set of explanations, in which each new one has value v on one previously empty entry of e . Finally, note that e^* generated on Line 8 has entry values in $\{1, \dots, L\}$, but some features may contribute *against* the prediction (e.g., “This movie is truly innovative although slightly cursory.”). Thus, we post-process e^* by shifting all values by k such that the new values (in $\{1 - k, L - k\}$) maximally satisfy the sign of marginal contribution of each word (i.e., the sign of the occlusion saliency).

Algorithm 1: Beam search for finding e^* .

```

1 Input: beam size  $B$ , metric  $m$ , sentence  $x$ 
  of length  $L$ ;
2 Let  $e^{(0)}$  be an empty length- $L$  explanation;
3 beams  $\leftarrow \{e^{(0)}\}$ ;
4 for  $l = 1, \dots, L$  do
5   | beams  $\leftarrow \bigcup_{e \in \text{beams}} \text{ext}(e, L - l + 1)$ ;
6   | beams  $\leftarrow \text{choose\_best}(\text{beams}, B)$ ;
7 end
8  $e \leftarrow \text{choose\_best}(\text{beams}, 1)$ ;
9  $e^* \leftarrow \text{shift}(e)$ ;
10 return  $e^*$ ;

```

Without the additive property, beam search is not feasible due to the lack of partial metric values. However, Zhou et al. (2021) presented a simulated annealing algorithm (Kirkpatrick et al., 1983) to search for the optimal data acquisition order in active learning, and we can use a similar procedure to search for the optimal feature importance order. If the metric is value-sensitive, assuming differentiability with respect to the explanation value, meth-

ods such as gradient descent can be used. Since we focus on comprehensiveness and sufficiency in this paper, the development and evaluation of these approaches are left to future work.

5 Experiments

We investigate various properties of the beam search explainer vs. existing heuristic explainers, using the publicly available `textattack/roberta-base-SST-2` model on the SST dataset (Socher et al., 2013) as a case study. The sentiment value for each sentence is a number between 0 (very negative) and 1 (very positive), which we binarize into two classes of $[0, 0.4]$ and $[0.6, 1]$. Sentences with sentiment values in middle are discarded. The average sentence length is 19, making the exhaustive search impossible. We use a beam size of 100 to search for Δ -solving explanation E^* . All reported statistics are computed on the test set.

Fig. 3 presents two explanations, with additional ones in Fig. 11 of App. C. While we need more quantitative analyses (carried out below) for definitive conclusions on its various properties, E^* explanations at least looks reasonable and is likely to help people understand the model by highlighting the high importance of sentiment-laden words.

Figure 3: Two E^* explanations. The shade of background color represents feature importance.

5.1 Performance on the Target Metric

We compare E^* to heuristic explainers on the Δ metric, with results shown in Tab. 1 along with the associated κ and σ . A random explanation baseline is included for reference. We can see that E^* achieves the best Δ , often by a large margin. It also tops the ranking separately for κ and σ , which suggests that an explanation could be optimally comprehensive and sufficient at the same time.

To visually understand how the model prediction changes during feature removal and insertion, we plot in Fig. 4 the values of $f(x) - f(\tilde{x}_e^{(l)})$ and $f(x) - f(\hat{x}_e^{(l)})$ (i.e., the summands in Eq. 1 and 2), as a function of l/L . The left panel shows the curves averaged across all test set instances, and the right panel shows those for a specific instance. κ

Explainer	Comp κ \uparrow	Suff σ \downarrow	Diff Δ \uparrow
Grad	0.327	0.108	0.218
IntG	0.525	0.044	0.481
LIME	0.682	0.033	0.649
SHAP	0.612	0.034	0.578
Occl	0.509	0.040	0.469
E*	0.740	0.020	0.720
Random	0.218	0.212	0.006

Table 1: Comprehensiveness, sufficiency and their difference for various explainers.

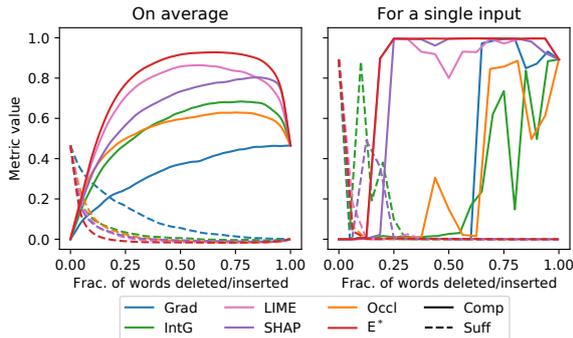


Figure 4: Comprehensiveness and sufficiency curves for the beam search optimal explainer vs. others.

and σ are thus the areas under the solid and dashed curves respectively. The curves for E* dominate the rest, and, on individual inputs, are also much smoother than those for other explanations.

One concern for beam search is its efficiency, especially compared to those that only require a single pass of the model such as the vanilla gradient. However, we note that explanations, unlike model predictions, are rarely used in real-time decision making. Instead, they are mostly used for debugging and auditing purposes, and incurring a longer generation time to obtain a higher-quality explanation is often beneficial. On a single RTX3080 GPU card without any in-depth code optimization, the metric values and time costs for various beam sizes are presented in Tab. 2, with statistics for the best explainer LIME also listed for comparison.

Expectedly, the metric values increase with increasing beam size, but the improvement is meager after 10 beams. More importantly, beam search is

B	1	5	10	20	50	100	LIME
κ	0.717	0.731	0.734	0.736	0.739	0.740	0.682
σ	0.020	0.020	0.020	0.020	0.020	0.020	0.033
Δ	0.697	0.711	0.714	0.716	0.719	0.720	0.649
T	0.38	0.77	1.15	1.72	2.85	4.37	4.75

Table 2: Effect of beam size B on κ , σ , Δ and computation time T (in seconds), compared against the statistics of the best heuristic explainer LIME.

not slow – it is still faster than LIME even with 100 beams, and the single-beam version outperforms LIME by a decent margin while being more than 10 times faster. Thus, these results establish that *if we take comprehensiveness and sufficiency as the quality metrics*, there is really no reason not to use the beam search explainer directly.

5.2 Performance on Other Metrics

Sec. 2 lists many metrics that all operationalize the same principle that changing important features should have large impact on model prediction, but in different ways. A potential argument against the explicit beam search optimization is the fulfillment of Goodhart’s Law: E* overfits to the metric by exploiting its realization (i.e., Eq. 1 and 2) of this principle and not truly reflecting its “spirit.”

To establish the legitimacy of this opposition, we evaluate all the explainers on the remaining four metrics in Sec. 2, and present the results in Tab. 3.

Explainer	DF _{MIT} \uparrow	DF _{Frac} \downarrow	Rank _{Del} \uparrow	Rank _{Ins} \uparrow
Grad	10.5%	54.5%	0.162	0.521
IntG	16.9%	39.6%	0.369	0.468
LIME	25.5%	28.1%	0.527	0.342
SHAP	23.0%	36.1%	0.369	0.458
Occl	26.4%	40.6%	1.000	0.396
E*	25.0%	25.2%	0.438	0.423
Random	3.4%	72.3%	0.004	0.599

Table 3: Performance on non-target metrics of the beam search optimal explainer vs. others.

Since the occlusion explainer solves DF_{MIT} and Rank_{Del} (Thm. 1), it ranks the best on these two metrics, as expected. Nonetheless, E* still ranks competitively on these two metrics and comes out ahead on DF_{Frac}. The only exception is Rank_{Ins}, on which the random explanation surprisingly performs the best. We carefully analyze it in App. D and identify a fundamental flaw in this metric.

Last, note that we can also incorporate any of these metrics into the objective function (which already contains two metrics: κ and σ), and search for E* that performs overall the best, if so desired. We leave this investigation to future work.

5.3 Explainer “Attacking” the Model

Another concern is that the search procedure may overfit to the model. Specifically, removing a word w in a partial sentence $\tilde{x}_e^{(l)}$ drastically changes the model prediction but does not have the same effect for most other $\tilde{x}_e^{(l')}$. This makes E* assign w an

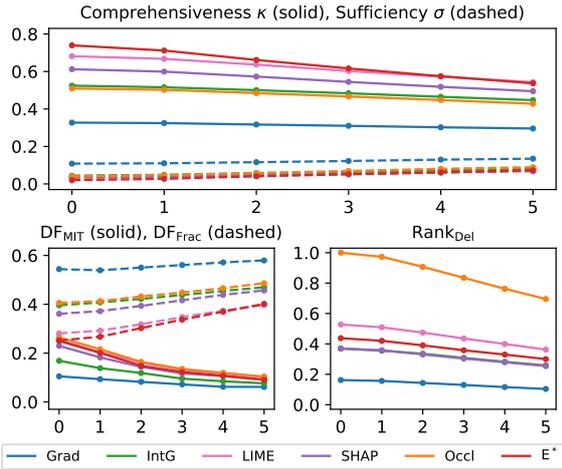


Figure 5: Metric values for explanations under different levels of perturbation represented by s on the x -axis.

overly high attribution, as w only happens to have a high impact in one particular case. By contrast, explainers like LIME and SHAP automatically avoid this issue by computing the average contribution of w on many different partial sentences.

We test this concern by locally perturbing the explanation. If E^* uses many such “adversarial attacks,” we should expect its metric values to degrade sharply under perturbation, as the high-importance words (according to E^*) will no longer be influential in different partial sentence contexts.

To perturb the explanation, we first convert each explanation e to its ranked importance version e_r using $r(\cdot)$ in Def. 4.1, which does not affect any metric as they are value-agnostic. Then we define the perturbed rank by adding to each entry of e_r an independent Gaussian noise: $e'_r = e_r + n$ with $n \sim \mathcal{N}(\mathbf{0}, s^2)$. Thus, two words x_i and x_j with $r(x_i) > r(x_j)$ have their ordering switched if $r(x_i) - r(x_j) < n(x_j) - n(x_i)$. A visualization of the switching with different s is in Fig. 12 of App. E.

Fig. 5 plots the metrics under different s values (Rank_{Ins} not shown due to its intrinsic issue discussed in App. D). Everything degrades to various extents. Although E^* degrades slightly faster than the rest on κ and DF_{Frac} (and on par on others), it still achieves best results even at $s = 4$, with many order switches (Fig. 12), and a faster degradation is reasonable anyway for metrics with better starting values (c.f. occlusion on Rank_{Del}).

The evidence suggests that there is at most a slight model overfitting phenomenon, as E^* remains comparable to other explainers under quite severe perturbation. Furthermore, we can incorporate perturbation robustness into metric solving to obtain an E^* that degrade less, similar to adver-

sarial training (Madry et al., 2018). We leave the exploration of this idea to future work.

App. F describes another assessment of model overfitting, though with a mild assumption and relying on word-level sentiment scores provided by the SST dataset. Similar conclusions are reached.

5.4 Ground Truth Recovery

For a model trained on a natural dataset, its ground truth working mechanism is rarely available – in fact, arguably the very purpose of interpretability methods is to uncover it. Thus, a series of work (e.g., Zhou et al., 2022a) proposed methods to modify the dataset such that a model trained on the new dataset has to follow a certain working mechanism to achieve high performance, which allows for evaluations against the known mechanism.

Ground Truth Definitions We construct three types of ground truths – short additions, long additions and replacements. First, we randomize the label to $\hat{y} \sim \text{Unif}\{0, 1\}$ so that the original input features are *not* predictive (Zhou et al., 2022a).

For the two addition types, a word or a sentence is inserted randomly to either the beginning or the end of the input. The inserted text is randomly chosen from the the sets in Tab. 4.

For the replacement type, each word in the input is checked against the list of replacement word sets in Tab. 5, and if the word belongs to one of the

Type	$\hat{y} = 0$	$\hat{y} = 1$
Short	terrible, awful, disaster, worst, never	excellent, great, fantastic, brilliant, enjoyable
Long	A total waste of time. Not worth the money! Is it even a real film? Overall it looks cheap.	I like this movie. This is a great movie! Such a beautiful work. Surely recommend it!

Table 4: Set of insertions for the addition type according to the new label \hat{y} . The words are comma-separated for “short”, and each line is one piece of text for “long”.

Replacement word sets	$\hat{y} = 0$	$\hat{y} = 1$
a, an, the	a	the
in, on, at	in	on
I, you	I	you
he, she	he	she
can, will, may	can	may
could, would, might	could	might
(all forms of <i>be</i>)	is	are
(all punctuation marks)	(period)	(comma)

Table 5: Replacement word sets and their target words.

Explainer	Short Addition				Long Addition				Replacement			
	Sym		Asym		Sym		Asym		Sym		Asym	
	Pr ↑	NR ↓	Pr ↑	NR ↓	Pr ↑	NR ↓	Pr ↑	NR ↓	Pr ↑	NR ↓	Pr ↑	NR ↓
Grad	0.91	0.06	0.51	0.08	0.70	0.37	0.77	0.30	0.50	0.75	0.51	0.74
IntG	0.82	0.10	0.60	0.21	0.60	0.76	0.70	0.55	0.49	0.74	0.48	0.74
LIME	1.00	0.06	1.00	0.06	0.72	0.60	0.84	0.32	0.63	0.65	0.54	0.71
SHAP	0.98	0.07	1.00	0.06	0.61	0.83	0.75	0.98	0.65	0.67	0.62	0.68
Occl	1.00	0.06	1.00	0.06	0.72	0.59	0.79	0.42	0.40	0.80	0.40	0.85
E*	1.00	0.06	1.00	0.06	0.67	0.64	0.92	0.38	0.60	0.66	0.54	0.73
Random	0.06	0.54	0.07	0.53	0.25	0.89	0.24	0.88	0.27	0.85	0.28	0.85

Table 6: Average values of precision and normalized rank of the ground truth correlated words for each explainer.

set, it is changed according to the new label \hat{y} . On average, 27% of input words are replaced.

We call these modifications symmetric since inputs corresponding to both $\hat{y} = 0$ and $\hat{y} = 1$ are modified. We also define the asymmetric modification, where only inputs with $\hat{y} = 1$ are modified, and those with $\hat{y} = 0$ are left unchanged.

Metrics We use the two metrics proposed by Bastings et al. (2022): precision and normalized rank. First, we define the ground truth correlated words. For the two addition types, they are the inserted words. In the asymmetric case, instances with $\hat{y} = 0$ does not have any words added, so we exclude them in metric value computation.² For the replacement type, they are the words that are in the replacement set (but not necessarily replaced).

Let W be the set of ground truth correlated words. Using ranked importance $r(\cdot)$ in Def. 4.1, precision and normalized rank are defined as

$$\text{Pr} = |\{w \in W : r(w) > L - |W|\}|/|W|,$$

$$\text{NR} = (L - \min\{r(w) : w \in W\} + 1)/L.$$

Precision is the fraction of ground truth words among the the top- $|W|$ ranked words, and normalized rank is the lowest rank among ground truth words, normalized by the length L of the input. Both values are in $[0, 1]$, and higher precision values and lower normalized rank values are better.

Results Tab. 6 presents the test set Pr and NR values. Many explainers including E* score perfectly on short additions, but all struggle on other types. Nonetheless, E* still ranks comparably or favorably to other methods. Its largest advantage shows on the asymmetric long addition, because this setup matches with the computation of κ and σ : E* finds the most important words to remove/add to maximally change/preserve the original prediction, and

²This also highlights an intrinsic limitation of feature attribution explanations: they cannot explain that the model predicts a class because certain features are *not* present.

those words are exactly the ground truth inserted ones. For replacement and symmetric addition, the search procedure does not “reconstruct” inputs of the other class, and E* fails to uncover the ground truth. This finding suggests a mismatch between metric computation and certain ground truth types.

Conversely, vanilla gradient performs decently on ground truth types other than short addition, yet ranks at the bottom on most quality metrics (Tab. 1 and 3), again likely due to the mismatch.

In fact, this evaluation is fundamentally different from the rest in its *non-solvability*, specifically due to its use of privileged information. To understand this point, let us first compare the evaluation of model *prediction* to that of model *explanation*, as illustrated in Fig. 6. The former runs the model on the input, receives the prediction, and compares it with the ground truth label, which is emphatically *not* available to the model under evaluation. By contrast, no such privileged information exists when computing interpretability metrics, allowing

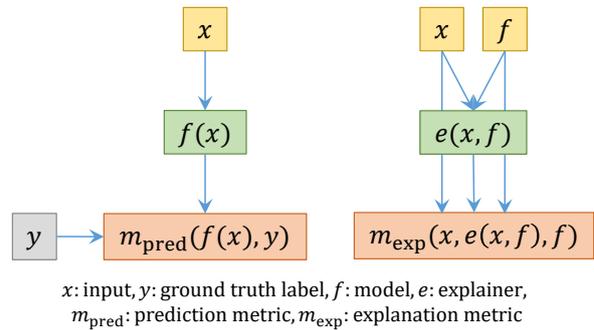


Figure 6: The complete evaluation diagrams for model predictions (left) and explanations (right). Green boxes are the model and explainer under evaluation, which have access to the information in yellow, and orange boxes are the evaluators. Notably, prediction evaluation (e.g., accuracy) uses the ground truth label y not accessible to the model, but no such privileged information is used by the interpretability evaluation.

the explainer to directly solve them. In this ground truth recovery evaluation, we employ similar privileged information (i.e., induced ground truth model working mechanism) by dataset modification and model retraining. However, as discussed by Zhou et al. (2022a), such evaluations are limited to the range of ground truths that could be induced.

6 Discussion

Definition-Evaluation Duality Our investigation demonstrates that some evaluation metrics can be used to find high-quality explanations, defined as the optimizers of the metrics. Conversely, we could also use any explanation definition d as an evaluation metric m . A very simple one would be $m(x, e) \doteq -\|e - d(x)\|$, where e is the explanation under evaluation, $d(x)$ is the “reference explanation” and $\|\cdot\|$ is a suitably chosen distance metric. It is obvious that $d(x)$ itself achieves the optimal evaluation metric value.

Therefore, in theory, there should not be a difference of using a concept as definition vs. evaluation, but in practice, we almost always see some used mainly as definitions and others as evaluations (Fig. 2). A major reason of not considering to use evaluations as definitions could be the presumed intractability of the optimization, which is experimentally refuted in this paper, as the beam search demonstrates its efficacy and efficiency.

Conversely, why do we not see more definitions (e.g., gradients and LIME) used as evaluations? Such an attempt may sound trivial yet unjustifiable at the same time: trivial because it is equivalent to claiming that the corresponding explainer definition is the best, which is in turn a seemingly unjustifiable circular logic.

More importantly, we motivate a new research direction opened up by the duality concept. Traditionally, definitions and evaluations have been considered and developed separately, but duality suggests that any interpretability concept can be used as both. Thus, we propose that we should focus on studying the *intrinsic* properties of these concepts, independent of their usage as one or another. For example, are some concepts inherently superior for model explanations than others? How can we measure the similarity between two concepts? What does the space of these concepts look like? None of them are currently answerable due to a complete lack of formalization, but research on it could lead to a much deeper understanding of local explanations.

Demonstrable Utility Given the duality, how should we evaluate explanations? Fundamentally, local explanations are used for model understanding (Zheng et al., 2022; Zhou et al., 2022b), and we advocate for evaluating *demonstrable utility*: the presence of an explanation compared to its absence, or the newly proposed explanation compared to existing ones, should lead to a measurable difference in some practically beneficial aspect.

For example, people use explanations to identify spurious correlation during development, audit fairness before deployment, and assist human decision makers during deployment. However, recent findings cast doubt on the feasibility of model explanations to support any of these use cases (Bansal et al., 2021; Jia et al., 2022; Zhou et al., 2022a).

Demonstrating such utilities would bypass discussions of solvability and directly assert their usefulness (Chen et al., 2022). The examples listed here are by no means comprehensive, and a systematic taxonomy is valuable. Furthermore, it is likely that no single explainer is a one-size-fit-all solution. More refined knowledge of the strengths and weaknesses of each method in supporting different aspects of model understanding is highly desirable.

7 Conclusion

We study the relationship between definitions and evaluations of local explanations. We identify the *solvability* property of evaluation metrics, such that for each evaluation metric, there is an explicit search procedure to find the explanation that achieves the optimal metric value. In other words, every evaluation admits a definition that *solves* it.

Compared to the current practice of defining a explainer and then evaluating it on a metric, solvability allows us to directly find the explanation that optimizes the target metric and guarantee a very favorable evaluation outcome. In this paper, we investigate the feasibility of this process. First, we propose to use beam search to find the explanation E^* that optimizes for comprehensiveness and sufficiency (DeYoung et al., 2020). Then, in a suite of evaluations, we find E^* performing comparably or favorably to existing explainers such as LIME.

Therefore, for practitioners, we recommend using the proposed explainer for computing local model explanations and provide the Python `solveX` package for easy adoption (App. A). For researchers, we propose a definition-evaluation duality inspired by solvability, which opens up many new research directions.

Limitations and Ethical Impact

The focus of our paper is to investigate the search-based explanation that explicitly optimizes a target quality metric. While the results suggest that it is comparable or favorable to existing heuristic explainers on various technical aspects, its societal properties have not been studied. For example, [Ghorbani et al. \(2019\)](#) showed that many heuristic explanations can be easily manipulated and [Slack et al. \(2020\)](#) demonstrated that discriminative models can be carefully modified such that their discrimination is hidden by heuristic explanations. It is possible that same issues exist for the search-based explanation, and thus we advise to carefully study them before deployment.

Another limitation of this approach is that E^* explainer only produces rankings of feature importance, rather than numerical values of feature importance. In other words, E^* does not distinguish whether one feature is only slightly or significantly more important than another. By comparison, almost all heuristic explainers output numerical values (e.g., magnitude of gradient). Other than the ease of search in the ranking space than the numerical value space, we give three additional reasons. First, the utility of actual values, beyond the induced rankings, has not been well studied in the literature. In addition, many popular explanation toolkits (e.g., [Wallace et al., 2019](#)) even defaults to top- k visualization. Last, popular evaluation metrics rarely consider values either, suggesting that there currently lack guiding principles and desiderata for these values. Moreover, if and when such value-aware metrics are widely adopted, we could augment our optimizer with them or incorporate them into a post-processing fix without affecting the ranking, similar to the shift operation done on Line 9 of Alg. 1.

References

- Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9505–9515.
- David Alvarez-Melis and Tommi S Jaakkola. 2018a. On the robustness of interpretability methods. In *ICML Workshop on Human Interpretability in Machine Learning*.
- David Alvarez-Melis and Tommi S Jaakkola. 2018b. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31.
- Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, et al. 2019. One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques. *arXiv:1909.03012*.
- Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv:1605.05362*.
- Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021. Does the whole exceed its parts? the effect of AI explanations on complementary team performance. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*, pages 1–16.
- Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. A protocol for evaluating the faithfulness of input saliency methods for text classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.
- Chun Sik Chan, Huanqi Kong, and Liang Guanqing. 2022. A comparative study of faithfulness metrics for model interpretability methods. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5029–5038. Association for Computational Linguistics.
- Valerie Chen, Jeffrey Li, Joon Sik Kim, Gregory Plumb, and Ameet Talwalkar. 2022. Interpretable machine learning: Moving from mythos to diagnostics. *Queue*, 19(6):28–56.
- George Chrysostomou and Nikolaos Aletras. 2021. Improving the faithfulness of attention-based explanations with task-specific information for text classification. In *Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 477–488. Association for Computational Linguistics.
- Piotr Dabkowski and Yarin Gal. 2017. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE.
- Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Annual Meeting of the Association for Computational Linguistics*

- (ACL), pages 4443–4458. Association for Computational Linguistics.
- Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3719–3728. Association for Computational Linguistics.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.
- Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 3681–3688.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE.
- Yan Jia, John McDermid, Tom Lawton, and Ibrahim Habli. 2022. The role of explainability in assuring safety of machine learning in healthcare. *IEEE Transactions on Emerging Topics in Computing (T-ETC)*, 10(4):1746–1760.
- Yiming Ju, Yuanzhe Zhang, Zhao Yang, Zhongtao Jiang, Kang Liu, and Jun Zhao. 2022. Logic traps in evaluating attribution scores. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5911–5922. Association for Computational Linguistics.
- Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3167. Association for Computational Linguistics.
- Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. 1983. Optimization by simulated annealing. *Science*, 220(4598):671–680.
- Ronny Kohavi and Barry Becker. 1996. UCI Adult data set. *UCI Machine Learning Repository*.
- Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 681–691. Association for Computational Linguistics.
- Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. Understanding neural networks through representation erasure. *arXiv:1612.08220*.
- Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4765–4774.
- Ronny Luss, Pin-Yu Chen, Amit Dhurandhar, Prasanna Sattigeri, Yunfeng Zhang, Karthikeyan Shanmugam, and Chun-Chen Tu. 2021. Leveraging latent features for local explanations. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1139–1149.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*.
- Weili Nie, Yang Zhang, and Ankit Patel. 2018. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *International Conference on Machine Learning (ICML)*, pages 3809–3818.
- Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. RISE: Randomized input sampling for explanation of black-box models. In *British Machine Vision Conference (BMVC)*.
- Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. 2022. Double trouble: How to not explain a text classifier’s decisions using counterfactuals synthesized by masked language models? In *Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 12–31. Association for Computational Linguistics.
- Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. 2018. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144.
- Alvin E Roth. 1988. *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press.
- Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 28(11):2660–2673.
- Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2931–2951. Association for Computational Linguistics.

- Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv:1312.6034*.
- Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2020. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *AAAI/ACM Conference on AI, Ethics, and Society (AIIES)*, pages 180–186. Association for Computing Machinery.
- Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. SmoothGrad: Removing noise by adding noise. *arXiv:1706.03825*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Association for Computational Linguistics.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, pages 3319–3328.
- Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP Interpret: A framework for explaining predictions of NLP models. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 7–12. Association for Computational Linguistics.
- Julia K Winkler, Christine Fink, Ferdinand Toberer, Alexander Enk, Teresa Deinlein, Rainer Hofmann-Wellenhof, Luc Thomas, Aimilios Lallas, Andreas Blum, Wilhelm Stolz, et al. 2019. Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition. *JAMA Dermatology*, 155(10):1135–1141.
- Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833. Springer.
- Hao Zhang, Jiayi Chen, Haotian Xue, and Quanshi Zhang. 2019. Towards a unified evaluation of explanation methods without ground truth. *arXiv preprint arXiv:1911.09017*.
- Yiming Zheng, Serena Booth, Julie Shah, and Yilun Zhou. 2022. The irrationality of neural rationale models. In *2nd Workshop on Trustworthy Natural Language Processing (TrustNLP)*. Association for Computational Linguistics.
- Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. 2022a. Do feature attribution methods correctly attribute features? In *AAAI Conference on Artificial Intelligence (AAAI)*.
- Yilun Zhou, Adithya Renduchintala, Xian Li, Sida Wang, Yashar Mehdad, and Asish Ghoshal. 2021. Towards understanding the behaviors of optimal deep active learning algorithms. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1486–1494.
- Yilun Zhou, Marco Tulio Ribeiro, and Julie Shah. 2022b. ExSum: From local explanations to model understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics.

A The Python `solvex` Package

We release the Python `solvex` package implementing explainer in a model-agnostic manner. The project website at <https://yilunzhou.github.io/solvability/> contains detailed tutorials and documentation. Here, we showcase three additional use cases of the explainer.

To explain long paragraphs, feature granularity at the level of sentences may be sufficient or even desired. `solvex` can use `spaCy`³ to split a paragraph into sentences and compute the sentence-level attribution explanation accordingly. As an explanation, Fig. 7 shows an explanation for the prediction on a test instance in the Yelp dataset (Asghar, 2016) made by the `albert-base-v2-yelp-polarity` model.

Contrary to other reviews, I have zero complaints about the service or the prices. I have been getting tire service here for the past 5 years now, and compared to my experience with places like Pep Boys, these guys are experienced and know what they're doing. Also, this is one place that I do not feel like I am being taken advantage of, just because of my gender. Other auto mechanics have been notorious for capitalizing on my ignorance of cars, and have sucked my bank account dry. But here, my service and road coverage has all been well explained - and let up to me to decide. And they just renovated the waiting room. It looks a lot better than it did in previous years.

Figure 7: A sentence-level explanation on a Yelp test instance. Red color indicates positive contribution.

This package can explain image predictions with superpixel features (similar to LIME (Ribeiro et al., 2016)). Fig. 8 shows the explanation for the top prediction (Class 232: Border Collie, a dog breed) by the ResNet-50 (He et al., 2016) trained on ImageNet (Deng et al., 2009).



Figure 8: An explanation for the top prediction (Class 232: Border Collie, a dog breed) on an image made by a ResNet-50 model trained on ImageNet. Red color indicates positive contribution.

Last, it can also explain models trained on tabular datasets with both categorical and numerical features. For a random forest model trained on the Adult dataset (Kohavi and Becker, 1996), Fig. 9 shows the attribution on each feature that contributes to the class 0 (i.e., income less than or equal to \$50K). Note that a more positive attribution value indicates that the feature (e.g. age or relationship) contributes more to the *low* income prediction.

³<https://spacy.io/>

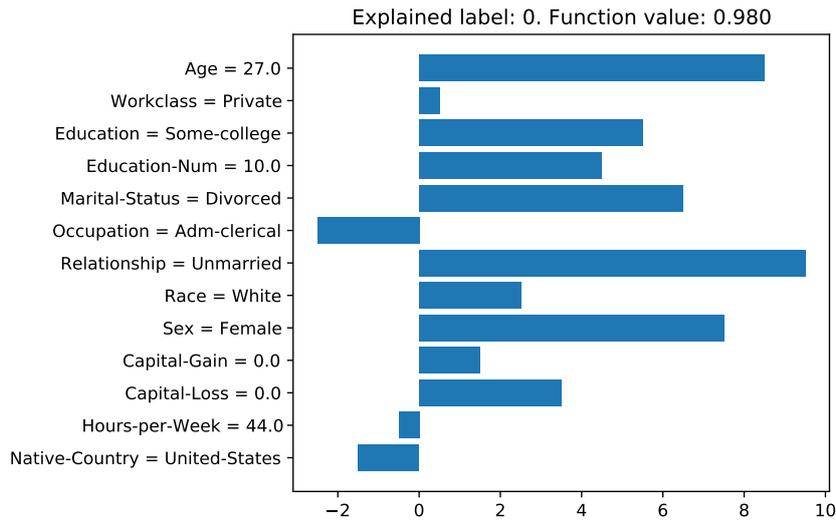


Figure 9: An explanation for the low income prediction made by a random forest model on the Adult dataset.

B The Definition-Evaluation Spectrum and Its Various Concepts

We describe the reasoning of assigning each concept to its location on the definition-evaluation spectrum (Fig. 2, reproduced as Fig. 10 below), as currently perceived by the community according to our understanding. Note that the discussion is unavoidably qualitative, but we hope that it illustrates the general idea of this spectrum.

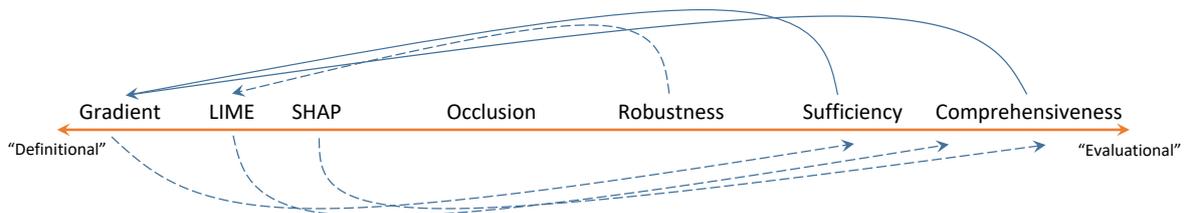


Figure 10: A definition-evaluation spectrum for various interpretability concepts, reproduced from Fig. 2.

We start on the definition side, where the gradient saliency (Simonyan et al., 2013; Li et al., 2016a) is a classic feature attribution definition but, to the best of our knowledge, has never been used in any evaluation capacity. Moving towards the evaluation side, we have LIME (Ribeiro et al., 2016), which is again used mainly to define explanations (as linear regression coefficients), but the notion of local fidelity introduced by LIME has been occasionally used to evaluate other explainers as well (Plumb et al., 2018). Similar to LIME, SHAP (Lundberg and Lee, 2017) defines explanations as those that (approximately) satisfy the Shapley axioms (Roth, 1988), which can also be used to evaluate how well a certain explanation performs with respect to these axioms (Zhang et al., 2019). Next up we have the occlusion concept, which, as seen in Sec. 2, can be used as one explainer definition, Occl (Zeiler and Fergus, 2014; Li et al., 2016b), and two (not so popular) evaluations, DF_{MIT} (Chrysostomou and Aletras, 2021) and $Rank_{Del}$ (Alvarez-Melis and Jaakkola, 2018b).

Further on the evaluation side, we now encounter concepts that are more often used for evaluations than definitions. Robustness (Ghorbani et al., 2019) evaluates the similarity between explanations among similar inputs and a higher degree of similarity is often more desirable (Alvarez-Melis and Jaakkola, 2018a). However, this robustness desideratum is incorporated explicitly into some explainers, such as via the noise aggregation in SmoothGrad (Smilkov et al., 2017). On the right-most end we have sufficiency and comprehensiveness (DeYoung et al., 2020), which evaluates whether keeping a small subset of features could lead to the original model prediction, or removing it could lead to a large drop in model prediction. They are arguably the most popular among various evaluation metrics, and have been

repeatedly proposed under different names such as the area over perturbation curve (AoPC) (Samek et al., 2016) and insertion/deletion metrics (Petsiuk et al., 2018). Using such these two ideas for definitions are relatively rare, with one notable exception of smallest sufficient/destroying regions (SSR/SDR) proposed by (Dabkowski and Gal, 2017).

Overall, it is clear that the community considers certain concepts more for definitions and others more for evaluations, which motivates the investigation for this paper and future work: can we swap the definition/evaluation roles, and if so, what are the implications?

C Additional Qualitative Examples of the E* Explanation

Fig. 11 presents more visualizations of E* explanations. These examples suggest that E* mostly focus on words that convey strong sentiments, which is a plausible working mechanism of a sentiment classifier.

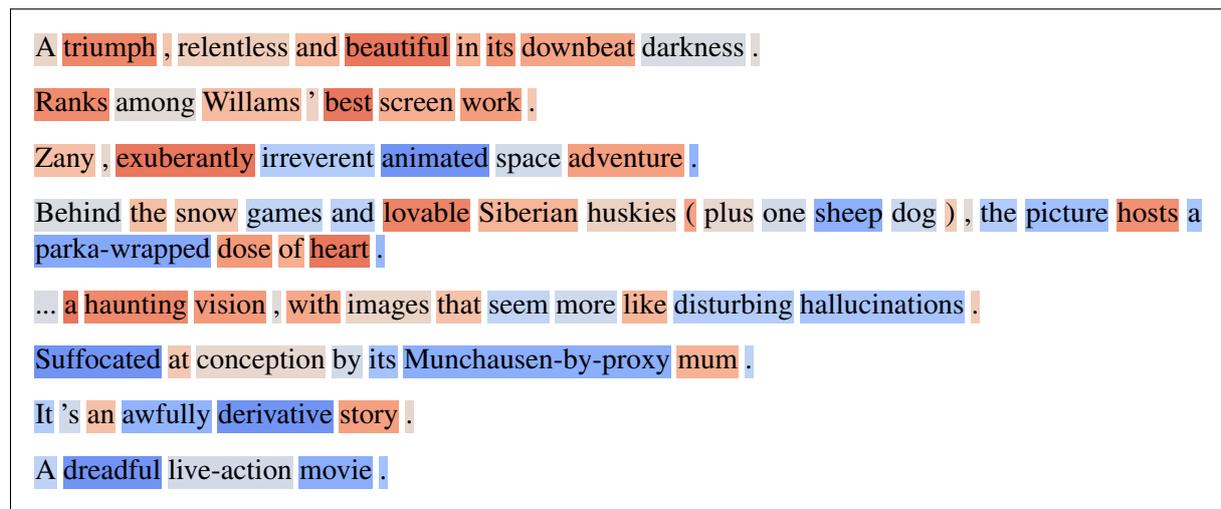


Figure 11: More E* explanations. The shade of background color represents feature importance.

D An Analysis on the Rank_{Ins} Metric

As introduced in App. 2, Rank_{Del} evaluates the monotonicity of the model prediction curve when more important features are successively inserted into an empty input. While this expectation seems reasonable, it suffers from a critical issue due to the convention in ranking features: if a feature contributes *against* the prediction, such as a word of sentiment opposite to the prediction (e.g., a positive prediction on “Other than the story plot being a bit boring, everything else is actually masterfully designed and executed.”), it should have negative attribution and the convention is to put them lower in the rank (i.e., less important) than those have zero contributions. This implementation leads to the correct interpretation of all other metric values.

However, under this convention, the first few words added to the empty input should decrease the model prediction and then increase it, leading to a U-shaped curve. In fact, it is the comprehensiveness curve shown in Fig. 4, flipped both horizontally (because features are inserted rather than removed) and vertically (because the plotted quantity is the model prediction rather than change in prediction). Thus, a deeper U-shape should be preferred, but it is less monotonic. This also explains why the random attribution baseline achieves such a high ranking correlation: as we randomly add features from the empty string to the full input, on average the curve should be a more or less monotonic interpolation between model predictions on empty and full inputs, which has better monotonicity rank correlation than the U-shape.

It is not clear how to fix the metric. Previous works that proposed (Luss et al., 2021) or used (Chan et al., 2022) this metric often ignored the issue. One work (Arya et al., 2019) filtered out all features of negative attribution values and evaluate the rank correlation only on the rest. This, however, is easily manipulatable by an adversary. Specifically, an explainer could shift all attribution values down such that only the most positive one has a non-negative value. This change results in a perfect correlation as long as

removing most positive feature induces a decrease in model prediction – an especially low requirement to satisfy. Empirically, we found that inserting features based on their (unsigned) magnitude barely affects the result either. Thus, we argue that this metric is not a good measurement of explanation quality.

E Visualization of Perturbation Effects

Fig. 12 visually presents the random perturbation, with different standard deviation s of the Gaussian noise. In each panel, the top row orders the features by their ranked importance, from least important on the left to most on the right, and the bottom row orders the features with perturbed ranked importance, with lines connecting to their original position. For example, in the top panel for $s = 1$, the perturbation swaps the relative order of the two least important features on the left.

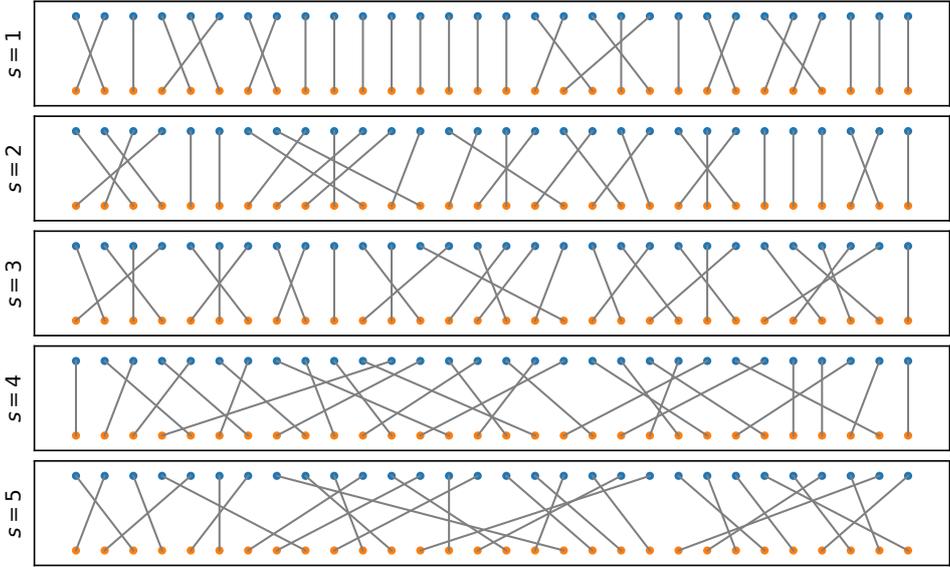


Figure 12: Visualization of rank perturbation under different values of s .

F Another Assessment on the Explainer-Attacking-Model Behavior

We describe another experiment to assess whether the explanations exploit the adversarial vulnerability of the model. While it is possible that the model could use some shortcuts (Geirhos et al., 2020), we would expect it to predominantly use sentiment-conveying words, as it achieves high accuracy and no such shortcuts are known for the dataset. In this case, we should expect an explainer that does not adversarially exploit the model to give attributions for words correlated with their sentiment values, while an explainer that attacks the model would rate words that are “adversarial bugs” to be more important.

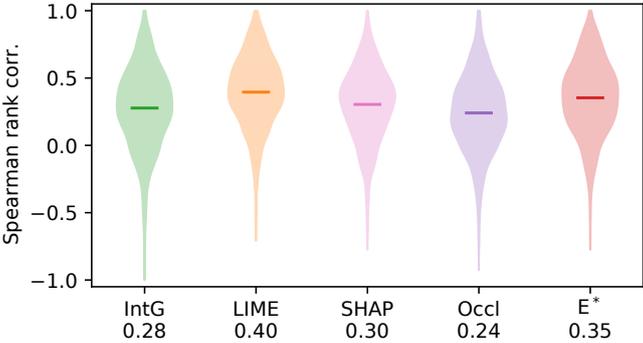


Figure 13: Spearman rank correlation coefficient between intrinsic word polarity score and attribution value.

Conveniently, the SST dataset provides human annotations of the polarity score between 0 and 1 for each word, where 0 means very negative, 1 means very positive, and 0.5 means neutral. We compute the alignment between the attribution values (for the positive class) and this score for each word. Given a sentence $x = (x_1, \dots, x_L)$ with explanation $e = (e_1, \dots, e_L)$ and word polarity score $s = (s_1, \dots, s_L)$, the alignment is defined as the Spearman rank correlation coefficient $\rho(e, s)$. Since the vanilla gradient only produces non-negative values, it is impossible to identify whether a word contributes *to* or *against* the positive class, and we exclude it from the analysis.

Fig. 13 plots the distribution of rank correlations among the test set instances, with the average shown as the bar and also annotated on the horizontal axis. Although no method achieves very high alignment, E* is the second-highest, after LIME. Thus, giving out assumption that high-polarity words are the indeed genuine signals used by the model for making predictions, we can conclude that E* does not adversarially exploit the model for its vulnerability any more severely than the heuristic explainers.

Reliable Gradient-free and Likelihood-free Prompt Tuning

Maohao Shen¹, Soumya Ghosh³, Prasanna Sattigeri³,
Subhro Das³, Yuheng Bu², Gregory Wornell¹

¹ Massachusetts Institute of Technology

² University of Florida

³ MIT-IBM Watson AI Lab, IBM Research

Abstract

Due to privacy or commercial constraints, large pre-trained language models (PLMs) are often offered as black-box APIs. Fine-tuning such models to downstream tasks is challenging because one can neither access the model’s internal representations nor propagate gradients through it. This paper addresses these challenges by developing techniques for adapting PLMs with only API access. Building on recent work on soft prompt tuning, we develop methods to tune the soft prompts without requiring gradient computation. Further, we develop extensions that in addition to not requiring gradients also do not need to access *any* internal representation of the PLM beyond the input embeddings. Moreover, instead of learning a single prompt, our methods learn a distribution over prompts allowing us to quantify predictive uncertainty. Ours is the first work to consider uncertainty in prompts when only having API access to the PLM. Finally, through extensive experiments, we carefully vet the proposed methods and find them competitive with (and sometimes even improving on) gradient-based approaches with full access to the PLM.

1 Introduction

Pre-trained language models (PLMs) are versatile learners and demonstrate impressive few-shot capabilities (Brown et al., 2020) and promising performance (Radford et al., 2018; Devlin et al., 2018; Raffel et al., 2020; Lewis et al., 2019) on various downstream tasks such as text classification (Kowsari et al., 2019), commonsense reasoning (Zellers et al., 2018), question answering (Rajpurkar et al., 2016), and machine translation (Bahdanau et al., 2014).

The conventional approach to adapting PLMs to downstream tasks involves fine-tuning the model (Peters et al., 2018; Devlin et al., 2018). Although fine-tuning is effective, it can be challenging to do in practice. First, fine-tuning large

language models are compute and memory intensive, e.g., a large model like GPT-3 (Brown et al., 2020) contains billions of parameters. Further, it is inefficient to adapt a PLM to a large number of downstream tasks since each task would require storing a copy of model parameters.

Prompt tuning alleviates these issues by providing an efficient way to adapt a PLM to a downstream task. It only learns a small number of prompt parameters while keeping the large PLM frozen but still achieves comparable performance to fine-tuning the entire PLM (Liu et al., 2021a; Shin et al., 2020; Lester et al., 2021; Liu et al., 2021c).

Although more efficient than traditional fine-tuning, prompt tuning still requires the propagation of gradients through the entire PLM. Beyond being computationally expensive, this may not be possible due to privacy risks or legal and commercial constraints. In fact, large PLMs are often only made available in the form of black-box APIs (Brown et al., 2020). Motivated by these observations, a recent line of research (Sun et al., 2022b,a) has started exploring *gradient-free* approaches to prompt tuning. BBT (Sun et al., 2022b) optimizes continuous prompt by leveraging the derivative-free optimization algorithms, and BBTv2 (Sun et al., 2022a) improves over BBT by optimizing multiple deep prompts at various intermediate layers of PLM. Although these approaches are *gradient-free*, they still assume that intermediate layers of the model being tuned are accessible.

Moreover, when deploying an NLP model in a real-world setting, it is inevitable to encounter unexpected scenarios. For example, the test data to be predicted might originate from out-of-distribution resources (Arora et al., 2021). For the model to be useful in such scenarios, it is essential that the model is able to quantify the uncertainty associated with its predictions and that these uncertainties are well-calibrated.

To this end, here we further push the limits of *gradient-free* prompt tuning in two aspects:

- First, we develop methods that add a layer of uncertainty quantification (UQ) aimed toward more reliable prompt tuning. We show that this improves calibration and UQ performance on several tasks, including selective classification and text Out-of-Distribution (OOD) detection.
- Second, we consider a much stricter notion of black-box setting, i.e., *likelihood-free* setting, where the PLM-based API does not provide probability scores or *logits* as the output, but only the discrete outcome labels. We propose a simulation-based-inference approach that yields competitive performance in the stricter setting even compared to the SOTA prior works on the relaxed black-box setting.

2 Background

Prompt Tuning Prompting, in the simplest form, involves appending manually curated words or tokens to a text input such that the language model, conditioned on such an augmented input, generates the desired output (Liu et al., 2021a). Such curated prompts were shown to be much more efficient than fine-tuning the entire PLM (Brown et al., 2020). However, curating good prompts for a new task can be difficult without deep domain expertise (Liu et al., 2021c; Zhao et al., 2021). One solution is to search the space of discrete prompts (Shin et al., 2020; Gao et al., 2020). This search in discrete space can be a hard optimization problem. Recent works instead learn continuous or soft prompts in the form of a small number of free parameters injected into certain layers of the PLM (Li and Liang, 2021). In this paper, we work with the simpler form of continuous *prompt tuning*, where the free parameters are only injected in the embedding layer (Lester et al., 2021).

Gradient-free Prompt Tuning *Gradient-free* prompt tuning aims to learn the continuous prompt without the propagating gradients through the PLM. BBT (Sun et al., 2022b) utilizes derivative-free optimization algorithms to optimize the continuous prompt. BBTv2 (Sun et al., 2022a) extends BBT by incorporating the idea of *deep prompt tuning*, which optimizes the deep prompt injected at additional intermediate layers of the PLM. Since our goal is to treat the PLM as a black-box, deep

prompt tuning is out of the scope of this work. We instead focus on the problem setting of the original BBT (Sun et al., 2022b) that learns a single prompt at the input layer.

Beyond point-estimates of prompts Many applications demand accurate quantification of uncertainty in predictions. This can be achieved in the prompt-tuning setting by not just learning a point estimate of the prompts but also inferring a distribution over the prompts for a given downstream task. In a non-black-box setting, to infer such a distribution, we can apply classical frequentist or Bayesian approaches. Although a few recent works focus on uncertainty quantification in NLP applications (Arora et al., 2021; Xiao and Wang, 2019; De-sai and Durrett, 2020; Kumar and Sarawagi, 2019), quantifying uncertainty in prompt-tuned large language models remains a severely under explored area. Our paper is the first to explore prompt uncertainty in gradient-free settings.

Simulation-based Inference Classic approaches for statistical inference mentioned above are intractable when the likelihood function is not accessible. The problem of inferring parameters of such a black-box model, called Simulation-based Inference (SBI) (Cranmer et al., 2020), is gaining popularity. Traditional SBI approaches include Approximate Bayesian Computation (ABC) (Beaumont et al., 2002; Marjoram et al., 2003; Marin et al., 2012; Beaumont et al., 2009; Bonassi and West, 2015) and synthetic likelihood (SL) (Wood, 2010; Turner and Sederberg, 2014). More recently, the neural density estimation-based approaches utilize the powerful deep neural network density estimator to directly learn the likelihood, i.e., Sequential Neural Likelihood Estimation (SNLE) (Lueckmann et al., 2017; Greenberg et al., 2019), or the likelihood ratio, i.e., Sequential Neural Ratio Estimation (SNRE) (Papamakarios et al., 2019), or the posterior, i.e., Sequential Neural Posterior Estimation (SNPE) (Hermans et al., 2020; Durkan et al., 2020).

3 Problem Formulation

In this paper, we focus on text classification and restrict ourselves to the few-shot learning setting considered in BBT (Sun et al., 2022b). Given a dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and a pre-trained language model (PLM) f , we aim to adapt f to predict the label y_* for an unseen text pas-

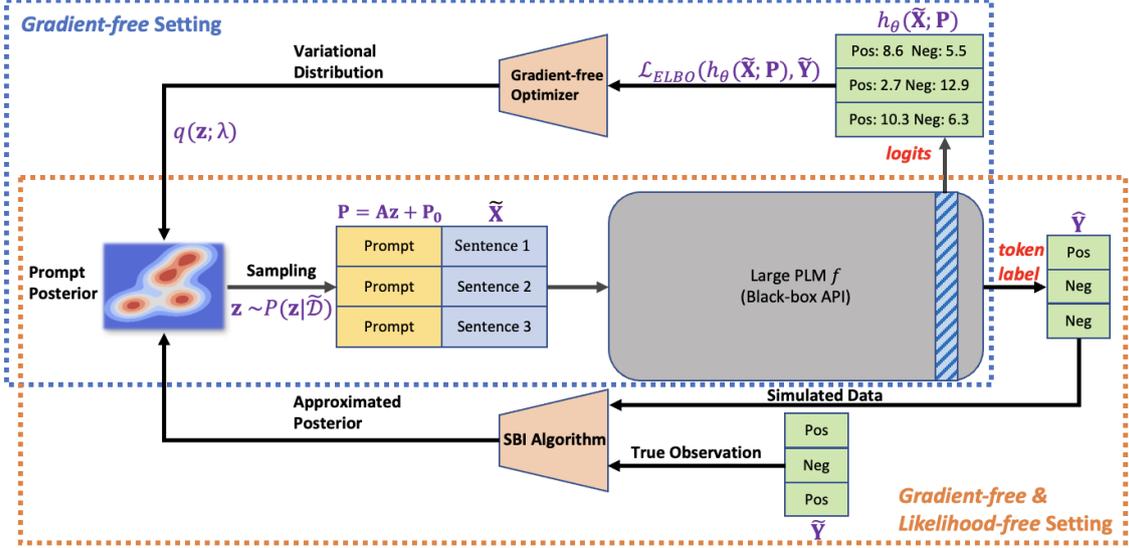


Figure 1: Our general goal is to estimate the posterior distribution of prompts. The *Gradient-free setting* uses the internal logits of PLM for optimization. Our proposed Gradient-free Variational inference approach utilizes the likelihood to compute the ELBO objective and leverage the gradient-free optimizer to optimize the variational distribution. The *Gradient-free and likelihood-free setting* can be formulated as an SBI problem, where the PLM is treated as a black-box simulator, and its output discrete outcome labels are the simulated data. The posterior samples can be efficiently approximated by the proposed ABC-SMC algorithm.

sage x_* . We formulate the classification task as a masked language modeling problem, where the input text x_i is converted into \tilde{x}_i via predefined templates, e.g., adding trigger words like “It was [MASK]”, and the labels y_i are mapped to label tokens \tilde{y}_i in the vocabulary such as “great” or “bad”. We denote this transformed dataset $\tilde{D} = (\tilde{X}, \tilde{Y})$.

We use soft prompt tuning (Lester et al., 2021) to adapt f , i.e., we construct a continuous prompt embedding $\mathbf{P} \in \mathbb{R}^D$ and feed it along with the converted input text \tilde{x}_i to the PLM f to generate a label token, $\hat{y}_i = f(\tilde{x}_i; \mathbf{P})$, where the notation $\hat{y}_i = f(\tilde{x}_i; \mathbf{P})$ is short hand for $\hat{y}_i \sim \text{Cat}(\sigma(h_\theta(\tilde{x}_i; \mathbf{P})))$. Here, Cat denotes the Categorical distribution, σ is the softmax function, and θ represents the frozen parameters of the PLM. We use h_θ to denote all but the final layer of the PLM f . Finally, we aim to learn an optimal prompt

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} - \sum_{i=1}^N \log \text{Cat}(\tilde{y}_i | \sigma(h_\theta(\tilde{x}_i; \mathbf{P}))). \quad (1)$$

This is just the standard cross-entropy loss and can be easily minimized using standard stochastic gradient based approaches provided (i) we can propagate gradients through the PLM f , and (ii) we can access the PLM’s logits, i.e., $h_\theta(\tilde{x}_i; \mathbf{P})$. The problem becomes substantially more challenging when these requirements are not satisfied.

When we are unable to propagate gradients

through f , we need to rely on *gradient-free* approaches to optimize Equation 1. Recent work (Sun et al., 2022b) has demonstrated promising gradient-free prompt tuning results by first employing a lower dimensional re-parameterization, $z \in \mathbb{R}^d$ with $d \ll D$, $\mathbf{P} = \mathbf{A}z + \mathbf{P}_0$, where $\mathbf{A} \in \mathbb{R}^{D \times d}$ is a random projection matrix and \mathbf{P}_0 is a fixed prompt embedding, and then using gradient-free evolutionary algorithms, in particular, Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen and Ostermeier, 2001; Hansen et al., 2003) to optimize,

$$z^* = \arg \min_z - \sum_{i=1}^N \log \text{Cat}(\tilde{y}_i | \sigma(h_\theta(\tilde{x}_i; \mathbf{A}z + \mathbf{P}_0))) \quad (2)$$

Going forward, we also adopt this lower dimensional parameterization, but instead of learning a point estimate z^* , we learn a distribution $p(z | \tilde{D})$ in a *gradient-free* setting. Similar to the point estimated variants, our algorithms to learn $p(z | \tilde{D})$ also rely on CMA-ES.

Next, we consider the fully black-box setting — *likelihood-free* and *gradient-free*. Here, beyond being unable to propagate gradients through f , we are further handicapped by only observing the predicted label tokens, $\hat{y}_i = f(\tilde{x}_i; \mathbf{P})$ for each training instance \tilde{x}_i , and not the corresponding logits, i.e.,

$h_\theta(\tilde{\mathbf{x}}_i; \mathbf{P})$. In this more challenging setting we found CMA-ES based approaches to be unreliable, often getting stuck in poor optima. Instead, we found it effective to pose the *likelihood-free* and *gradient-free* prompt tuning task as a simulation-based inference (SBI) (Cranmer et al., 2020) problem. We view the PLM f as a black-box simulator that given a realization of \mathbf{z} and the text $\tilde{\mathbf{x}}_i$ produces \hat{y}_i . We then use a sequential Monte-Carlo approximate Bayesian computation (SMC-ABC) approach to infer the distribution $p(\mathbf{z}|\tilde{\mathcal{D}})$.

Finally, we use the distribution $p(\mathbf{z}|\tilde{\mathcal{D}})$ to characterize the uncertainty in predictions via the predictive distribution $p(\tilde{y}|\tilde{\mathbf{x}}, \tilde{\mathcal{D}}) = \int p(\tilde{y}|\tilde{\mathbf{x}}; \mathbf{z})p(\mathbf{z}|\tilde{\mathcal{D}})d\mathbf{z}$. We form Monte-Carlo approximations to this integral. In the *gradient-free* case, this is,

$$p(\tilde{y}|\tilde{\mathbf{x}}, \tilde{\mathcal{D}}) \approx \frac{1}{S} \sum_{s=1}^S p(\tilde{y}|\tilde{\mathbf{x}}; \mathbf{z}_s),$$

where $\mathbf{z}_s \sim p(\mathbf{z}|\tilde{\mathcal{D}})$. In the *likelihood-free* and *gradient-free* case, since we only have access to the label tokens, we approximate the predictive distribution,

$$p(\tilde{y} = c|\tilde{\mathbf{x}}, \tilde{\mathcal{D}}) \approx \frac{1}{S} \sum_{s=1}^S \mathbb{1}\{\hat{y}_s = c\}, \quad (3)$$

where $\hat{y}_s = f(\tilde{\mathbf{x}}; \mathbf{A}\mathbf{z}_s + \mathbf{P}_0)$, $\mathbf{z}_s \sim p(\mathbf{z}|\tilde{\mathcal{D}})$. In Section 5 we empirically demonstrate that by characterizing the uncertainty in \mathbf{z} through $p(\mathbf{z}|\tilde{\mathcal{D}})$ we get better calibrated predictive uncertainties, improved selective classification, and out-of-distribution detection.

4 Methods

We now describe our methods in greater detail. First, we discuss two algorithms for the *gradient-free* setting in 4.1 and 4.2. After that, we focus on addressing the *gradient-free* and *likelihood-free* setting from the SBI perspective in 4.3.

4.1 Prompt Ensembles

Deep ensembles (Lakshminarayanan et al., 2017) are a simple yet effective technique for quantifying uncertainty in deep neural network predictions. They generate a uniformly-weighted ensemble by re-training the same neural network from different random initialization. Leveraging the CMA-ES algorithm (Hansen and Ostermeier, 2001; Hansen

et al., 2003), we can adapt this idea to *gradient-free* prompt tuning.

CMA-ES is an evolutionary strategy that maintains a multivariate normal distribution $\mathcal{N}(m_t, \sigma_t^2 C_t)$ over a population of solutions. Each iteration of the algorithm involves sampling a set of possible solutions and updating the normal distribution to favor low loss solutions. To build a prompt ensemble, we run S instances of CMA-ES, each initialized with a different random initialization of the mean m_t and variance σ_t^2 and record the optimized prompt embeddings produced by each instance. This collection of S prompt embeddings $\{\mathbf{z}_s\}_{s=1}^S$ form the distribution $p(\mathbf{z}|\tilde{\mathcal{D}})$ and are used to approximate the predictive distribution via Equation 2.

4.2 Gradient-free Variational Inference

An alternative way to estimate the predictive distribution is by approximating the posterior distribution of prompt embedding $p(\mathbf{z}|\tilde{\mathcal{D}})$. Since direct computation of posterior is intractable, in our setting we resort to variational inference (VI) and approximate the posterior distribution with a tractable surrogate $q(\mathbf{z}; \boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ denotes the variational parameters. VI minimizes KL-divergence between variational distribution and true posterior distribution with respect to $\boldsymbol{\lambda}$. i.e., $\boldsymbol{\lambda}^* = \arg \min_{\boldsymbol{\lambda}} \text{KL}(q(\mathbf{z}; \boldsymbol{\lambda}) \| p(\mathbf{z}|\tilde{\mathcal{D}}))$. This is equivalent to maximizing the evidence lower bound (ELBO), i.e.,

$$\boldsymbol{\lambda}^* = \arg \max_{\boldsymbol{\lambda}} \mathbb{E}_{q(\mathbf{z}; \boldsymbol{\lambda})} [\log p(\tilde{\mathcal{D}}|\mathbf{z})] - \text{KL}(q(\mathbf{z}; \boldsymbol{\lambda}) \| p(\mathbf{z})) \quad (4)$$

$$= \arg \max_{\boldsymbol{\lambda}} \sum_{i=1}^N \mathbb{E}_{q(\mathbf{z}; \boldsymbol{\lambda})} [\log \text{Cat}(\tilde{y}_i | \sigma(h_\theta(\tilde{\mathbf{x}}_i; \mathbf{P})))] - \text{KL}(q(\mathbf{z}; \boldsymbol{\lambda}) \| p(\mathbf{z})), \quad (5)$$

where $\mathbf{P} = \mathbf{A}\mathbf{z} + \mathbf{P}_0$, and $p(\mathbf{z})$ denotes the prior distribution, which is assumed to be a normal distribution with zero mean and diagonal covariance matrix, i.e., $\mathcal{N}(0, \sigma \cdot \mathbf{I})$. Optimizing the ELBO objective requires taking derivative w.r.t $\boldsymbol{\lambda}$ as well as computing the gradient of log likelihood w.r.t \mathbf{z} , i.e., $\nabla_{\mathbf{z}} \mathbb{E}_{q(\mathbf{z}; \boldsymbol{\lambda})} [\log \text{Cat}(\tilde{y}_i | \sigma(h_\theta(\tilde{\mathbf{x}}_i; \mathbf{A}\mathbf{z} + \mathbf{P}_0)))]$, which causes standard variational inference algorithms to be infeasible in the *gradient-free* setting.

Instead of back-propagation, we propose a gradient-free variational inference algorithm lever-

aging the derivative-free optimizer CMA-ES. Specifically, we consider the variational distribution as a multivariate normal distribution $q(\mathbf{z}; \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where we assume the covariance matrix is diagonal, i.e., $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\alpha}) \in \mathbb{R}^{d \times d}$. The variational parameter, as the target for optimization, is the mean and diagonal elements of the covariance matrix, i.e., $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \boldsymbol{\alpha}) \in \mathbb{R}^{2d}$. At each iteration of the optimization, the CMA-ES outputs a collection of candidate solutions $\{\boldsymbol{\lambda}_j\}_{j=1}^m = \{(\boldsymbol{\mu}_j, \boldsymbol{\alpha}_j)\}_{j=1}^m$. For each candidate variational parameter $\boldsymbol{\lambda}_j$, we evaluate the corresponding ELBO loss using the variational distribution $q(\mathbf{z}; \boldsymbol{\lambda}_j) = \mathcal{N}(\boldsymbol{\mu}_j, \text{diag}(\boldsymbol{\alpha}_j))$, where the expectations in Equation 5 is approximated by Monte-Carlo samples obtained from the variational distribution. Finally, the CMA-ES optimizer takes the current collection of variational parameter $\{\boldsymbol{\lambda}_j\}_{j=1}^m$ and their corresponding ELBO loss to conduct the next iteration of optimization. The schematic of the process is shown in Figure 1, and the overall algorithm is summarized as Algorithm 1 in Appendix A.

After we obtain the optimal variational parameter $\boldsymbol{\lambda}^*$ that maximizes the ELBO loss, the predictive label distribution can be estimated by taking Monte Carlo samples from the optimal variational distribution, i.e., $q(\mathbf{z}; \boldsymbol{\lambda}^*) = \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$.

4.3 SBI-based Algorithm for Likelihood-free Prompt Tuning

Now, we describe our proposed approach for the *gradient-free* and *likelihood-free* case. For this problem, the most naive algorithm applicable is rejection approximation Bayesian computation (ABC) (Pritchard et al., 1999) that repeatedly samples from a prior distribution $\mathbf{z} \sim p(\mathbf{z})$ and obtains the corresponding simulated observation $\hat{\mathbf{Y}}$. The algorithm only accepts the sampled prompt embedding if the simulated observation is sufficiently close to the ground truth observation $\tilde{\mathbf{Y}}$ based on a distance function ρ and tolerance ϵ , i.e., $\rho(\hat{\mathbf{Y}}, \tilde{\mathbf{Y}}) < \epsilon$. The collection of accepted samples can be used to approximate the posterior distribution. However, rejection ABC typically suffers from poor computational efficiency, especially when ϵ is small and the dimensionality of observations is large. In preliminary experiments, we found rejection ABC to not be effective for our purposes. Instead, in this work, we adapt a more advanced technique — sequential Monte Carlo ap-

proximate Bayesian computation (ABC-SMC) algorithm (McKinley et al., 2009) to enable efficient prompt posterior inference. The core idea of ABC-SMC is to use a sequential tolerance schedule, i.e., $\epsilon_1 > \epsilon_2 > \dots > \epsilon_T$ to construct a sequence of intermediate distributions, which gradually converges to the true posterior distribution.

First, we draw prompt embedding samples from the prior $p(\mathbf{z}) = \mathcal{N}(0, \sigma \cdot \mathbf{I})$ and pass them into PLM f to receive the corresponding token label prediction $\hat{\mathbf{Y}}$ for a batch of text data $\tilde{\mathbf{X}}$. Then, we accept S samples $\{\mathbf{z}_s^{(1)}\}_{s=1}^S$ that satisfy the condition $\rho(\hat{\mathbf{Y}}, \tilde{\mathbf{Y}}) < \epsilon_1$. We use accuracy as the distance function ρ . In the next iteration, we resample embeddings from $\{\mathbf{z}_s^{(t-1)}\}_{s=1}^S$ with probability proportional to weights $w^{(t-1)}$, and perturb the sampled embeddings via a perturbation kernel to obtain a new sample, i.e., $\mathbf{z}^{(t)} \sim \mathcal{N}(\mathbf{z}^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})$. Again, we propagate these sampled embeddings through the PLM f and accept the newly proposed embeddings, $\{\mathbf{z}_s^{(t)}\}_{s=1}^S$, if $\rho(\hat{\mathbf{Y}}, \mathbf{Y}) < \epsilon_t$, where the tolerance ϵ_t is decayed by one step per iteration, i.e., $\epsilon_{t+1} = \epsilon_t - \frac{1}{N}$, where N is the total number of training data. Finally, the weights $w^{(t)}$ and the variance of the perturbation kernel are updated after each iteration (details are elaborated in Appendix B). Empirically, we find that simply using uniform weights leads to better performance (more discussion in Section 5.3). These steps are repeated for T iterations until the tolerance ϵ_T is sufficiently small. The schematic is in Figure 1 and the overall algorithm is summarized as Algorithm 1 in Appendix A.

The final collection of prompt samples $\{\mathbf{z}_s^{(T)}\}_{s=1}^S$ form an approximation to the posterior $p(\mathbf{z}|\mathcal{D})$ and we use Equation 3 to derive the approximate predictive distribution.

5 Experiment Results

In this section, we demonstrate the solid empirical performance of our proposed methods. We begin with introducing the uncertainty quantification applications and describe the experiment settings. Then, we present our main results in terms of prediction performance and UQ quality. Finally, we provide an ablation study and relevant perspectives of comparison. Detailed results and implementation steps are provided in Appendix D.

Table 1: **Prediction Performance** (Test acc \uparrow), *indicates results taken from BBT (Sun et al., 2022b)

Settings	Methods	SST-2	Yelp P.	AG’s News	DBPedia	MRPC	SNLI	RTE	Avg
Gradient-based	Prompt Tuning*	68.23 \pm 3.78	61.02 \pm 6.65	84.81 \pm 0.66	87.75 \pm 1.48	51.61 \pm 8.67	36.13 \pm 1.51	54.69 \pm 3.79	63.46
	P-Tuning v2*	64.33 \pm 3.05	92.63 \pm 1.39	83.46 \pm 1.01	97.05 \pm 0.41	68.14 \pm 3.89	36.89 \pm 0.79	50.78 \pm 2.28	70.47
	Model Tuning*	85.39 \pm 2.84	91.82 \pm 0.79	86.36 \pm 1.85	97.98 \pm 0.14	77.35 \pm 5.70	54.64 \pm 5.29	58.60 \pm 6.21	78.88
Gradient-free	Manual Prompt*	79.82	89.65	76.96	41.33	67.40	31.11	51.62	62.56
	In-Context Learning*	79.79 \pm 3.06	85.38 \pm 3.92	62.21 \pm 13.46	34.83 \pm 7.59	45.81 \pm 6.67	47.11 \pm 0.63	60.36 \pm 1.56	59.36
	Feature-MLP*	64.80 \pm 1.78	79.20 \pm 2.26	70.77 \pm 0.67	87.78 \pm 0.61	68.40 \pm 0.86	42.01 \pm 0.33	53.43 \pm 1.57	66.63
	Feature-BiLSTM*	65.95 \pm 0.99	74.68 \pm 0.10	77.28 \pm 2.83	90.37 \pm 3.10	71.55 \pm 7.10	46.02 \pm 0.38	52.17 \pm 0.25	68.29
	BBT	86.93 \pm 0.25	91.61 \pm 0.29	83.22 \pm 0.42	76.94 \pm 1.22	75.95 \pm 2.30	45.38 \pm 0.02	50.54 \pm 0.36	72.94
	Ours(ELBO)	86.81 \pm 0.47	92.07 \pm 0.17	83.96 \pm 0.22	73.25 \pm 2.35	76.35 \pm 0.94	46.78 \pm 2.92	50.78 \pm 1.39	72.86
	Ours(Ensembles)	88.61 \pm 0.78	92.35 \pm 0.16	84.62 \pm 0.20	80.12 \pm 1.06	76.77 \pm 1.13	47.95 \pm 2.76	50.34 \pm 3.40	74.39
Gradient-free & Likelihood-free	Ours(SNPE)	84.37 \pm 0.29	90.38 \pm 0.07	80.50 \pm 0.10	33.11 \pm 0.48	81.02 \pm 0.06	39.60 \pm 0.49	53.07 \pm 0.82	66.01
	Ours(ABC-SMC)	86.51 \pm 0.55	90.32 \pm 0.03	81.43 \pm 0.41	57.41 \pm 0.90	80.78 \pm 0.07	40.81 \pm 0.24	53.37 \pm 0.30	70.09

Table 2: **Calibration Performance** (ECE score \downarrow)

Settings	Methods	SST-2	Yelp P.	AG’s News	DBPedia	MRPC	SNLI	RTE	Avg
Gradient-free	BBT	0.056 \pm 0.014	0.032 \pm 0.000	0.049 \pm 0.007	0.056 \pm 0.032	0.115 \pm 0.018	0.040 \pm 0.008	0.170 \pm 0.069	0.074
	Ours(ELBO)	0.056 \pm 0.007	0.025 \pm 0.004	0.065 \pm 0.001	0.045 \pm 0.028	0.058 \pm 0.004	0.035 \pm 0.007	0.113 \pm 0.030	0.057
	Ours(Ensembles)	0.058 \pm 0.001	0.017 \pm 0.001	0.064 \pm 0.009	0.085 \pm 0.005	0.073 \pm 0.007	0.039 \pm 0.004	0.134 \pm 0.033	0.067
Gradient-free & Likelihood-free	Ours(SNPE)	0.104 \pm 0.005	0.082 \pm 0.000	0.100 \pm 0.010	0.549 \pm 0.004	0.314 \pm 0.001	0.185 \pm 0.011	0.466 \pm 0.002	0.257
	Ours(ABC-SMC)	0.106 \pm 0.009	0.084 \pm 0.001	0.108 \pm 0.001	0.278 \pm 0.026	0.309 \pm 0.009	0.178 \pm 0.002	0.458 \pm 0.004	0.217

Table 3: **Selective Classification** (AURRRC score \downarrow)

Settings	Methods	SST-2	Yelp P.	AG’s News	DBPedia	MRPC	SNLI	RTE	Avg
Gradient-free	Lower-bound	0.030	0.009	0.035	0.070	0.251	0.427	0.255	0.154
	BBT(Entropy)	0.063 \pm 0.009	0.029 \pm 0.001	0.082 \pm 0.004	0.095 \pm 0.009	0.349 \pm 0.002	0.519 \pm 0.032	0.523 \pm 0.004	0.237
	BBT(MaxP)	0.063 \pm 0.009	0.029 \pm 0.001	0.077 \pm 0.004	0.091 \pm 0.009	0.349 \pm 0.002	0.513 \pm 0.031	0.523 \pm 0.004	0.235
	ELBO(Entropy)	0.053 \pm 0.004	0.026 \pm 0.001	0.079 \pm 0.001	0.123 \pm 0.006	0.336 \pm 0.009	0.481 \pm 0.065	0.508 \pm 0.012	0.229
	ELBO(MaxP)	0.053 \pm 0.004	0.026 \pm 0.001	0.074 \pm 0.002	0.117 \pm 0.005	0.336 \pm 0.009	0.478 \pm 0.062	0.508 \pm 0.012	0.227
	Ensembles(Entropy)	0.046 \pm 0.006	0.023 \pm 0.001	0.074 \pm 0.002	0.084 \pm 0.004	0.324 \pm 0.011	0.472 \pm 0.048	0.513 \pm 0.048	0.219
	Ensembles(MaxP)	0.046 \pm 0.006	0.023 \pm 0.001	0.068 \pm 0.002	0.076 \pm 0.004	0.324 \pm 0.011	0.469 \pm 0.047	0.513 \pm 0.048	0.217
	Gradient-free & Likelihood-free	SNPE(Entropy)	0.065 \pm 0.003	0.073 \pm 0.001	0.116 \pm 0.005	0.551 \pm 0.001	0.319 \pm 0.003	0.580 \pm 0.009	0.466 \pm 0.003
SNPE(MaxP)		0.065 \pm 0.003	0.073 \pm 0.001	0.116 \pm 0.005	0.552 \pm 0.002	0.319 \pm 0.003	0.591 \pm 0.009	0.466 \pm 0.003	0.312
ABC-SMC(Entropy)		0.061 \pm 0.006	0.075 \pm 0.002	0.110 \pm 0.004	0.285 \pm 0.015	0.325 \pm 0.006	0.571 \pm 0.000	0.468 \pm 0.014	0.271
ABC-SMC(MaxP)		0.061 \pm 0.006	0.075 \pm 0.002	0.110 \pm 0.004	0.288 \pm 0.014	0.325 \pm 0.006	0.579 \pm 0.000	0.468 \pm 0.014	0.272

5.1 Settings

Uncertainty Quantification Applications. We assess the performance of the uncertainty quantification from three perspectives: (1) **Calibration** – the typical UQ quality metric that measures how well the model confidence aligned with the correctness of its prediction; (2) **Selective Classification** – aims to avoid the risk of wrong predictions by abstaining the prediction for samples with high uncertainty; and (3) **OOD Detection** – aims to identify the out-of-distribution data that is unobserved during the training stage. The OOD data can exhibit different forms of distribution shift, including the *covariate shift* where the OOD data distribution is different from the training samples; and the *semantic shift* where the OOD data contain unobserved class. In our experiment, we focus on two types of OOD tasks: the **Far OOD** detection task where both *covariate shift* and *semantic shift* happen simultaneously; the **Near OOD** detection task where the OOD data only contain *covariate shift*, but have the same class label words.

Benchmark. For a comprehensive comparison with BBT (Sun et al., 2022b), we mainly employ the same text classification benchmark datasets as BBT, including sentiment analysis datasets SST-

2 (Socher et al., 2013) and Yelp polarity (Zhang et al., 2015); topic classification datasets AG’s News (Zhang et al., 2015) and DBPedia (Zhang et al., 2015); paraphrase dataset MRPC (Dolan and Brockett, 2005); natural language inference (NLI) datasets SNLI (Bowman et al., 2015) and RTE (Wang et al., 2018).

Both calibration and selective classification tasks are conducted using the original test samples for each benchmark dataset. For the far OOD detection task, we create the ID/OOD dataset pairs by combining two datasets belonging to two different tasks, e.g., SST-2/RTE. For the near OOD detection task, we use IMDB (Maas et al., 2011) for the sentiment analysis task and MNLI (Williams et al., 2017) for the NLI task.

Baselines. For prediction performance, besides the SOTA *Gradient-free* prompt tuning approach BBT (Sun et al., 2022b), we also compare with other *Gradient-free* methods: (1) The naive **Manual Prompt** that uses the hand-crafted prompt templates; (2) **In-context Learning** (Brown et al., 2020); (3) Feature-based approaches (Peters et al., 2019) that trains auxiliary models on top of the PLM extracted features, including **Feature-MLP** training a MLP classifier and **Feature-BiLSTM**

Table 4: Far OOD Detection (AURRRC score ↓)

Settings	Methods	ID:SST-2 OOD:RTE	ID:Yelp P. OOD:RTE	ID:MRPC OOD:RTE	ID:DBPedia OOD:AG's News	ID:SNLI OOD:MRPC	ID:RTE OOD:MRPC	Avg
	Lower-bound	0.072	0.001	0.162	0.010	0.004	0.357	0.101
Gradient-free	BBT(entropy)	0.124±0.015	0.002±0.000	0.404±0.006	0.058±0.018	0.100±0.002	0.639±0.024	0.221
	BBT(MaxP)	0.124±0.015	0.002±0.000	0.404±0.006	0.059±0.014	0.098±0.002	0.639±0.024	0.221
	Ours(ELBO)(Entropy)	0.112±0.010	0.001±0.000	0.320±0.014	0.051±0.001	0.109±0.002	0.635±0.003	0.205
	Ours(ELBO)(MaxP)	0.112±0.010	0.001±0.000	0.320±0.014	0.056±0.001	0.107±0.002	0.635±0.003	0.205
	Ours(Ensembles)(Entropy)	0.097±0.008	0.001±0.000	0.350±0.038	0.057±0.003	0.110±0.001	0.606±0.047	0.204
	Ours(Ensembles)(MaxP)	0.097±0.008	0.001±0.000	0.350±0.038	0.058±0.002	0.108±0.001	0.606±0.047	0.203
Gradient-free & Likelihood-free	Ours(SNPE)(Entropy)	0.140±0.001	0.005±0.000	0.402±0.005	0.082±0.003	0.093±0.001	0.592±0.008	0.219
	Ours(SNPE)(MaxP)	0.140±0.001	0.005±0.000	0.402±0.005	0.081±0.003	0.091±0.002	0.592±0.008	0.219
	Ours(ABC-SMC)(Entropy)	0.126±0.009	0.005±0.001	0.396±0.001	0.097±0.021	0.092±0.000	0.596±0.009	0.219
	Ours(ABC-SMC)(MaxP)	0.126±0.009	0.005±0.001	0.396±0.001	0.095±0.021	0.092±0.001	0.596±0.009	0.218

Table 5: Near OOD Detection (AURRRC score ↓)

Settings	Methods	ID:SST-2 OOD:IMDB	ID:Yelp P. OOD:IMDB	ID:SNLI OOD:MNLI	ID:RTE OOD:MNLI	Avg
	Lower-bound	0.960	0.147	0.259	0.950	0.579
Gradient-free	BBT(entropy)	0.978±0.003	0.315±0.003	0.720±0.011	0.963±0.008	0.744
	BBT(confidence)	0.978±0.003	0.315±0.003	0.705±0.011	0.963±0.008	0.740
	Ours(ELBO)(Entropy)	0.976±0.002	0.308±0.006	0.692±0.039	0.968±0.005	0.736
	Ours(ELBO)(MaxP)	0.976±0.002	0.308±0.006	0.678±0.044	0.968±0.005	0.733
	Ours(Ensembles)(Entropy)	0.976±0.001	0.297±0.003	0.707±0.028	0.962±0.002	0.736
	Ours(Ensembles)(MaxP)	0.976±0.001	0.297±0.003	0.692±0.032	0.962±0.002	0.732
Gradient-free & Likelihood-free	Ours(SNPE)(Entropy)	0.984±0.000	0.365±0.001	0.715±0.002	0.951±0.000	0.754
	Ours(SNPE)(MaxP)	0.984±0.000	0.365±0.001	0.695±0.004	0.951±0.000	0.749
	Ours(ABC-SMC)(Entropy)	0.983±0.001	0.365±0.001	0.710±0.002	0.952±0.000	0.753
	Ours(ABC-SMC)(MaxP)	0.983±0.001	0.365±0.001	0.694±0.000	0.952±0.000	0.749

training a LSTM model followed by a classifier. We include additional results of *Gradient-based* approaches: (1) **Model Tuning** that fine-tunes the entire PLM; (2) **Prompt Tuning** (Lester et al., 2021) that only trains the continuous prompt without modifying PLM; (3) **P-Tuning v2** (Liu et al., 2021b) that trains the several continuous prompts injected at different layers of PLM. For uncertainty quantification tasks, few existing prompt tuning works aim to tackle this problem, so we mainly compare with BBT to justify how we can address its limitation under the *gradient-free* setting.

Implementation Details. We follow the same experiment setting as BBT. We focus on text classification as a few-shot learning problem, motivated by the fact that labeled training data can be limited in practice. Specifically, we construct few-shot training and validation data by drawing 16 random samples for each class from the original training dataset. The prediction performance is evaluated on the original development or test set, depending on the datasets. We use the same PLM model RoBERTa_{LARGE} as the backbone model and keep the hyper-parameter same as BBT. Specifically, we set the prompt length as 50, i.e., $D = 50 \times 1024$, and the subspace dimensionality as $d = 500$. The only modification is that we adapt the normal distribution (Sun et al., 2022a) to generate the random projection matrix \mathbf{A} , instead of the uniform distribution used in BBT. For a fair comparison, we reproduce the results of BBT using the random pro-

jection generated from normal distribution. More implementation details are included in Appendix C.

Performance Metrics. For prediction performance, we evaluate the prediction accuracy on the testing dataset. For calibration performance, we adopt the expected calibration error (ECE) (Guo et al., 2017) score as the metric. For both selective classification and OOD detection tasks, we compute the area under the risk vs. rejection rate curve (AURRRC) (Franc and Prusa, 2019). The risk is defined as the portion of wrong-predicted samples among the data chosen for prediction in selective classification and the portion of OOD samples among the data identified as in-distribution in the OOD detection task. The rejection rate is defined as the portion of data that abstained from the prediction based on specific uncertainty measurement. Note that an oracle with perfect knowledge of uncertainty measurement can achieve a minimum AURRRC score. This is obtained by assigning an uncertainty score based on the oracle knowledge, i.e., whether a test sample is wrong-predicted (OOD samples) or not. We denote such minimum AURRRC score as the *lower-bound*.

Given the predictive label distribution, we utilize two uncertainty measurements, including *Entropy* of the label distribution, i.e., $\mathcal{H}\left(p(\tilde{y}|\tilde{\mathbf{x}};\tilde{\mathcal{D}})\right)$, and *MaxP*, which is defined as $\max_c p(\tilde{y} = c|\tilde{\mathbf{x}};\tilde{\mathcal{D}})$.

5.2 Results

We conduct extensive evaluations of our proposed methods under both the *Gradient-free* setting and the *Gradient-free* and *Likelihood-free* setting. The results of prediction performance are shown in Table 1. For the uncertainty quantification performance, the calibration results are shown in Table 2, the selective classification results are shown in Table 3, the Far OOD detection results are shown in Table 4 and the Near OOD detection results are shown in Table 5.

Gradient-free and Likelihood-free Setting. No existing work is trying to tackle the *Gradient-free* and *likelihood-free* prompt tuning problem. However, we still compare our proposed method with other baseline methods on different problem settings to understand how well we can achieve and the price we need to pay for such a more strict constraint. In addition, we also include the results of neural net-based approach SNPE (Hermans et al., 2020; Durkan et al., 2020) for solving the SBI problem.

As shown in Table 1, our proposed method ABC-SMC can achieve competitive prediction performance as SOTA approach BBT and even outperform the other *Gradient-free* baselines without the requirement of the model likelihood. We also observe that ABC-SMC performs better than SNPE. The possible explanation is that the density estimation model adopted by SNPE usually requires a large number of simulated samples to achieve good performance, which is hindered by the slow inference speed of large PLM.

For the uncertainty quantification tasks, ABC-SMC underperforms on calibration and selective classification tasks but can still achieve comparable performance on the two OOD detection tasks. The performance gap can possibly be mitigated if we collect more samples (by increasing K) for a more accurate estimation of the empirical label distribution, but the computational cost is the price we need to pay for the *likelihood-free* constraint.

Gradient-free Setting. By relaxing the *likelihood-free* constraint, it is observed that our proposed methods, both Gradient-free Variational Inference (denoted as ELBO) and Ensembles algorithms, achieve comparable or even better prediction performance than BBT and other *gradient-free* baselines, while outperforming BBT in terms of uncertainty quantification across all

the tasks. Such empirical observation justifies the effectiveness of leveraging Bayesian and Ensemble techniques to enable more reliable *gradient-free* prompt tuning without sacrificing the prediction performance.

5.3 Discussions

In this section, we further investigate our proposed methods by exploring the use of alternate models and the effect of using uniform weights in the SMC-ABC algorithm.

Performance on other backbone models To demonstrate that our proposed methods generalize well on other PLM backbone models, we evaluate them on BERT_{LARGE} under the both *Gradient-free* setting and *Gradient-free* and *likelihood-free* setting. The results are presented in Appendix D.1. Note that our proposed methods consistently outperform BBT in terms of both prediction and uncertainty quantification performance under the *Gradient-free* setting while achieving competitive performance with a small gap under the *Gradient-free* and *likelihood-free* setting.

Ablation study of weights in ABC-SMC In practice, we observe that the ABC-SMC algorithm suffers from weight degeneracy, with weights for certain particles approaching one and effectively causing the posterior to be approximated by a single particle. Although, this issue can be mitigated by designing better proposals, we found that the heuristic of using uniform weights instead of updating the weights at each iteration of the algorithm to be far more effective. To demonstrate the efficacy, we conduct an ablation study about the sampling weights, and the results are shown in Appendix D.2. We find that with uniform weights ABC-SMC provides both improves prediction and uncertainty quantification for our application.

6 Concluding Remarks

In this work, we explore *gradient-free* prompt tuning along two under-explored angles: quantifying uncertainty in soft prompts; and tackling a more strict *likelihood-free* setting from the SBI perspective. Our developed methods demonstrate encouraging empirical performance across multiple tasks.

Investigating more modern neural SBI methods and designing more robust methods for learning prompt posteriors are exciting directions for future research. Other perspectives on *gradient-free*

prompt tuning, such as learning natural language-like interpretable prompts, are also worthy of exploration.

7 Limitations

We explored methods for learning a distribution over prompts for tuning PLMs with only API access. We rely on approximate inference algorithms to infer these distributions. Since the true posterior is intractable, effectively evaluating the quality of the inferred approximate posteriors is challenging. Here, we use downstream metrics to compare different algorithms. However, such metrics conflate the quality of posterior approximation with predictive performance. Assessing the quality of approximate posteriors remains an open problem. Another limitation of our ABC-based approach is that it is more expensive than approaches that can exploit gradient information. Improving the computational efficiency of such approaches comprises interesting future work.

8 Acknowledgements

This work was supported, in part, by the MIT-IBM Watson AI Lab under Agreement No. W1771646, and NSF under Grant No. CCF-1816209.

References

- Udit Arora, William Huang, and He He. 2021. Types of out-of-distribution texts and how to detect them. *arXiv preprint arXiv:2109.06827*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Mark A Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P Robert. 2009. Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990.
- Mark A Beaumont, Wenyang Zhang, and David J Balding. 2002. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035.
- Fernando V Bonassi and Mike West. 2015. Sequential monte carlo with adaptive weights for approximate bayesian computation. *Bayesian Analysis*, 10(1):171–187.
- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Kyle Cranmer, Johann Brehmer, and Gilles Louppe. 2020. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062.
- Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.
- Conor Durkan, Iain Murray, and George Papamakarios. 2020. On contrastive learning for likelihood-free inference. In *International Conference on Machine Learning*, pages 2771–2781. PMLR.
- Vojtech Franc and Daniel Prusa. 2019. On discriminative learning of prediction uncertainty. In *International Conference on Machine Learning*, pages 1963–1971. PMLR.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.
- David Greenberg, Marcel Nonnenmacher, and Jakob Macke. 2019. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pages 2404–2414. PMLR.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.
- Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18.
- Nikolaus Hansen and Andreas Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195.
- Joeri Hermans, Volodimir Begy, and Gilles Louppe. 2020. Likelihood-free mcmc with amortized approximate ratio estimators. In *International Conference on Machine Learning*, pages 4239–4248. PMLR.

- Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.
- Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.
- Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.
- Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.
- Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. 2017. Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, 30.
- Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.
- Jean-Michel Marin, Pierre Pudlo, Christian P Robert, and Robin J Ryder. 2012. Approximate bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.
- Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. 2003. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328.
- Trevelyan McKinley, Alex R Cook, and Robert Deardon. 2009. Inference in epidemic models without likelihoods. *The International Journal of Biostatistics*, 5(1).
- George Papamakarios, David Sterratt, and Iain Murray. 2019. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 837–848. PMLR.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.
- Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*.
- Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. 1999. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798.
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Tianxiang Sun, Zhengfu He, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022a. Bbtv2: Pure black-box optimization can be comparable to gradient descent for few-shot learning. *arXiv preprint arXiv:2205.11200*.

- Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022b. Black-box tuning for language-model-as-a-service. *arXiv preprint arXiv:2201.03514*.
- Brandon M Turner and Per B Sederberg. 2014. A generalized, likelihood-free method for posterior estimation. *Psychonomic bulletin & review*, 21(2):227–250.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.
- Simon N Wood. 2010. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104.
- Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7322–7329.
- Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.
- Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

A Omitted Algorithms

The overall algorithm of Gradient-free Variational Inference and ABC-SMC are shown in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Gradient-free Variational Inference

input Training data set $\{(\tilde{\mathbf{x}}_i, \tilde{y}_i)\}_{i=1}^N$; CMA-ES optimizer ES; Prior distribution $p(\mathbf{z})$; Number of candidate solutions m ; Total iteration T .

Initialize the initial collection of variational parameter, i.e., $\{\boldsymbol{\lambda}_j^{(0)}\}_{j=1}^m = \{(\boldsymbol{\mu}_j^{(0)}, \boldsymbol{\alpha}_j^{(0)})\}_{j=1}^m$.

for $t = 1, 2, \dots, T$ **do**

for $j = 1, 2, \dots, m$ **do**

 Generate S prompt embedding samples from the variational distribution, i.e.,

$$\{\mathbf{z}_s^{(t-1)}\}_{s=1}^S \sim \mathcal{N}(\boldsymbol{\mu}_j^{(t-1)}, \text{diag}(\boldsymbol{\alpha}_j^{(t-1)}))$$

 Evaluate the ELBO loss of j -th variational distribution i.e.,

$$\begin{aligned} \mathcal{L}_j^{(t-1)} &= \sum_{i=1}^N \sum_{s=1}^S \log \text{Cat}(\tilde{y}_i | \sigma(h_\theta(\tilde{\mathbf{x}}_i; \mathbf{A}\mathbf{z}_s^{(t-1)} + \mathbf{P}_0))) \\ &\quad - \text{KL}(q(\mathbf{z}; \boldsymbol{\lambda}_j^{(t-1)}) \| p(\mathbf{z})) \end{aligned}$$

end

 Request a new collection of variational parameter solutions, i.e.,

$$\{\boldsymbol{\lambda}_j^{(t)}\}_{j=1}^m \leftarrow \text{ES}(\{\boldsymbol{\lambda}_j^{(t-1)}\}_{j=1}^m; \{\mathcal{L}_j^{(t-1)}\}_{j=1}^m)$$

end

output Optimized collection of prompt embedding samples $\{\mathbf{z}_s^{(T)}\}_{s=1}^S$ corresponding to max ELBO loss.

B Implementation details of ABC-SMC

Updating of $w^{(t)}$ In the ABC-SMC algorithm, the sampling weights are initialized as uniform distribution at the first iteration $t = 1$ as all the samples are sampled from the prior distribution $p(\mathbf{z})$. In the later iterations, the new samples are drawing from a mixture proposal distribution consisted the previous samples and the perturbation kernel, i.e., $\sum_{s=1}^S w_s^{(t-1)} \cdot \mathcal{N}(\mathbf{z}_s^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})$. The weights are updated in an importance sampling manner as the ratio between the prior probability and the proposal probability, i.e.,

$$w_s^{(t)} = \frac{p(\mathbf{z}_s)}{\sum_{s=1}^S w_s^{(t-1)} \cdot \mathcal{N}(\mathbf{z}_s^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})}$$

Updating of $\boldsymbol{\Sigma}^{(t)}$ The covariance $\boldsymbol{\Sigma}^{(t)}$ in the perturbation kernel is a diagonal covariance matrix

Algorithm 2 ABC-SMC

input PLM f ; The fixed random projection matrix \mathbf{A} and initial prompt \mathbf{P}_0 ; Training data set $(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}) = \{(\tilde{\mathbf{x}}_i, \tilde{y}_i)\}_{i=1}^N$; Prior distribution $p(\mathbf{z})$; Initial tolerance ϵ_1 ; Distance measure function $\rho(\cdot)$; Number of samples S ; Total iteration T .

for $t = 1, 2, \dots, T$ **do**

if $t == 1$ **then**

for $s = 1, 2, \dots, S$ **do**

do

 Generate prompt embedding samples from the prior distribution, i.e., $\mathbf{z}_s^{(1)} \sim p(\mathbf{z})$;
 Obtain the corresponding prediction result $\hat{\mathbf{Y}} = f(\mathbf{A}\mathbf{z}_s^{(1)} + \mathbf{P}_0; \tilde{\mathbf{X}})$.

while $\rho(\hat{\mathbf{Y}}, \tilde{\mathbf{Y}}) > \epsilon_1$;

 Initialize the sampling probability weights $w_s^{(1)} = \frac{1}{S}$.

end

 Decay the tolerance, i.e., $\epsilon_{t+1} = \epsilon_t - \frac{1}{N}$; Initialize the perturbation kernel variance $\boldsymbol{\Sigma}^{(1)}$.

else

for $s = 1, 2, \dots, S$ **do**

do

 Draw a random sample $\mathbf{z}_s^{(t-1)}$ from $\{\mathbf{z}_s^{(t-1)}\}_{s=1}^S$ with probability $w_s^{(t-1)}$;
 Generate a new sample $\mathbf{z}_s^{(t)} \sim \mathcal{N}(\mathbf{z}_s^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})$;
 Obtain the corresponding prediction result $\hat{\mathbf{Y}} = f(\mathbf{A}\mathbf{z}_s^{(t)} + \mathbf{P}_0; \tilde{\mathbf{X}})$.

while $\rho(\hat{\mathbf{Y}}, \tilde{\mathbf{Y}}) > \epsilon_t$;

 Update the sampling probability weights $w^{(t)}$ (see Appendix B).

end

 Decay the tolerance, i.e., $\epsilon_{t+1} = \epsilon_t - \frac{1}{N}$; Update the perturbation kernel variance $\boldsymbol{\Sigma}^{(t)}$ (see Appendix B).

end

end

output Optimized collection of prompt embedding samples $\{\mathbf{z}_s^{(T)}\}_{s=1}^S$; Final sampling weights $w^{(T)}$.

$\text{diag}(\boldsymbol{\alpha}^{(t)})$, where the diagonal elements $\boldsymbol{\alpha}^{(t)}$ are updated using the weighted empirical variance of previous collection of samples, i.e.

$$\boldsymbol{\alpha}^{(t)} = \sum_{s=1}^S w_s^{(t-1)} \cdot (\mathbf{z}_s^{(t-1)} - \bar{\mathbf{z}}^{(t-1)})^2$$

Where $\bar{\mathbf{z}}^{(t-1)} = \sum_{s=1}^S w_s^{(t-1)} \cdot \mathbf{z}_s^{(t-1)}$ is the mean.

C Implementation Details

All of our experiment results are reported with means and standard deviations over three trials, each with a different random seed. The experiments are implemented in PyTorch, and each run of our proposed methods requires less than 24h of training computation time (on a single NVIDIA Tesla V100 GPU). Our proposed algorithms generate a collection of S prompt samples to estimate the predictive label distribution. We set $S = 10, 100, 100$ for Ensembles, Gradient-free Variational Inference, and ABC-SMC, respectively. The total budget for the derivative-free optimizer CMA-ES is set to be 300 with a population size of 20. We use the same prior distribution $p(z)$ for all algorithms, which is assumed to be a normal distribution with zero mean and diagonal covariance matrix, i.e., $\mathcal{N}(0, \sigma \cdot \mathbf{I})$. σ controls how concentrated the prior distribution is, and we use $\sigma = 50$ in our experiments. In ABC-SMC, the distance measure function ρ is defined as the prediction error rate, i.e., the portion of wrongly predicted data among the whole data batch. The initial tolerance ϵ_1 in ABC-SMC is initialized as the prediction error rate of an arbitrary prompt sample drawing from the prior distribution. The tolerance is decayed by one step per iteration, i.e., $\epsilon_{t+1} = \epsilon_t - \frac{1}{N}$, where N is the total number of training data.

D Additional Experiment Results

D.1 Performance on other backbone PLM

We evaluate the performance of our proposed methods on SST-2 and SNLI tasks using BERT_{LARGE} as the backbone model. We keep the hyperparameter settings the same as the original experiments. The results are shown in Table 6, 7, 8, and 9.

Table 6: **Test Performance (test acc \uparrow)**

Settings	Methods	SST-2	SNLI
Gradient-free	BBT	74.77 \pm 3.21	41.07 \pm 2.97
	Ours(ELBO)	75.38 \pm 1.74	41.20 \pm 0.39
	Ours(Ensembles)	80.05 \pm 1.79	42.64 \pm 1.96
Gradient-free & Likelihood-free	Ours(ABC-SMC)	66.40 \pm 0.46	39.00 \pm 0.22

D.2 Ablation of ABC-SMC sampling weights

We compare both the prediction and uncertainty quantification performance of our proposed ABC-SMC approaches using the updated sampling weights and the fixed uniform weights. We denote

Table 7: **Calibration Performance (ECE score \downarrow)**

Settings	Methods	SST-2	SNLI
Gradient-free	BBT	0.081 \pm 0.051	0.086 \pm 0.039
	Ours(ELBO)	0.046 \pm 0.006	0.073 \pm 0.009
	Ours(Ensembles)	0.045 \pm 0.007	0.068 \pm 0.024
Gradient-free & Likelihood-free	Ours(ABC-SMC)	0.328 \pm 0.003	0.584 \pm 0.002

Table 8: **Selective Classification (AURRRC score \downarrow)**

Settings	Methods	SST-2	SNLI
Gradient-free	BBT(Entropy)	0.146 \pm 0.028	0.564 \pm 0.036
	BBT(MaxP)	0.146 \pm 0.028	0.568 \pm 0.043
	Ours(ELBO)(Entropy)	0.132 \pm 0.009	0.542 \pm 0.006
	Ours(ELBO)(MaxP)	0.132 \pm 0.009	0.540 \pm 0.009
	Ours(Ensembles)(Entropy)	0.104 \pm 0.014	0.525 \pm 0.023
	Ours(Ensembles)(MaxP)	0.104 \pm 0.014	0.523 \pm 0.024
Gradient-free & Likelihood-free	Ours(ABC-SMC)(Entropy)	0.327 \pm 0.002	0.607 \pm 0.001
	Ours(ABC-SMC)(MaxP)	0.327 \pm 0.002	0.607 \pm 0.001

Table 9: **Far OOD Detection (AURRRC score \downarrow)**

Settings	Methods	ID:SST-2 OOD:RTE	ID:SNLI OOD:MRPC
Gradient-free	BBT(entropy)	0.402 \pm 0.015	0.076 \pm 0.016
	BBT(MaxP)	0.402 \pm 0.015	0.072 \pm 0.015
	Ours(ELBO)(Entropy)	0.365 \pm 0.037	0.089 \pm 0.007
	Ours(ELBO)(MaxP)	0.365 \pm 0.037	0.084 \pm 0.006
	Ours(Ensembles)(Entropy)	0.338 \pm 0.027	0.074 \pm 0.020
	Ours(Ensembles)(MaxP)	0.338 \pm 0.027	0.071 \pm 0.018
Gradient-free & Likelihood-free	Ours(ABC-SMC)(Entropy)	0.252 \pm 0.000	0.044 \pm 0.000
	Ours(ABC-SMC)(MaxP)	0.252 \pm 0.000	0.044 \pm 0.000

the method using updated weights as ‘‘ABC-SMC w. Weights’’. The results are shown in Table 10, 11, 12, 13, and 14.

Table 10: Prediction Performance (Test acc \uparrow)

Methods	SST-2	Yelp P.	AG's News	DBPedia	MRPC	SNLI	RTE	Avg
ABC-SMC	86.51 \pm 0.55	90.32 \pm 0.03	81.43 \pm 0.41	57.41 \pm 0.90	80.78 \pm 0.07	40.81 \pm 0.24	53.37 \pm 0.30	70.09
ABC-SMC w. Weights	84.37 \pm 0.81	90.42 \pm 0.22	79.44 \pm 0.46	50.36 \pm 0.89	80.83 \pm 0.08	42.06 \pm 1.15	53.07 \pm 0.01	68.65

Table 11: Calibration Performance (ECE score \downarrow)

Methods	SST-2	Yelp P.	AG's News	DBPedia	MRPC	SNLI	RTE	Avg
ABC-SMC	0.106 \pm 0.009	0.084 \pm 0.001	0.108 \pm 0.001	0.278 \pm 0.026	0.309 \pm 0.009	0.178 \pm 0.002	0.458 \pm 0.004	0.217
ABC-SMC w. Weights	0.156 \pm 0.008	0.091 \pm 0.005	0.160 \pm 0.023	0.506 \pm 0.009	0.316 \pm 0.002	0.182 \pm 0.005	0.463 \pm 0.003	0.268

Table 12: Selective Classification (AURRRC score \downarrow)

Methods	SST-2	Yelp P.	AG's News	DBPedia	MRPC	SNLI	RTE	Avg
ABC-SMC (Entropy)	0.061 \pm 0.006	0.075 \pm 0.002	0.110 \pm 0.004	0.285 \pm 0.015	0.325 \pm 0.006	0.571 \pm 0.000	0.468 \pm 0.014	0.271
ABC-SMC (MaxP)	0.061 \pm 0.006	0.075 \pm 0.002	0.110 \pm 0.004	0.288 \pm 0.014	0.325 \pm 0.006	0.579 \pm 0.000	0.468 \pm 0.014	0.272
ABC-SMC w. Weights (Entropy)	0.090 \pm 0.008	0.069 \pm 0.002	0.125 \pm 0.002	0.442 \pm 0.011	0.315 \pm 0.001	0.570 \pm 0.019	0.460 \pm 0.006	0.296
ABC-SMC w. Weights (MaxP)	0.090 \pm 0.008	0.073 \pm 0.003	0.133 \pm 0.008	0.479 \pm 0.020	0.315 \pm 0.001	0.570 \pm 0.017	0.460 \pm 0.006	0.303

Table 13: Far OOD Detection (AURRRC score \downarrow)

Methods	ID:SST-2 OOD:RTE	ID:Yelp P. OOD:RTE	ID:MRPC OOD:RTE	ID:DBPedia OOD:AG's News	ID:SNLI OOD:MRPC	ID:RTE OOD:MRPC	Avg
ABC-SMC(Entropy)	0.126 \pm 0.009	0.005 \pm 0.001	0.396 \pm 0.001	0.097 \pm 0.021	0.092 \pm 0.000	0.596 \pm 0.009	0.219
ABC-SMC(MaxP)	0.126 \pm 0.009	0.005 \pm 0.001	0.396 \pm 0.001	0.095 \pm 0.021	0.092 \pm 0.001	0.596 \pm 0.009	0.218
ABC-SMC w. Weights(Entropy)	0.186 \pm 0.020	0.004 \pm 0.001	0.406 \pm 0.013	0.079 \pm 0.013	0.067 \pm 0.004	0.596 \pm 0.006	0.223
ABC-SMC w. Weights(MaxP)	0.186 \pm 0.020	0.004 \pm 0.001	0.402 \pm 0.006	0.091 \pm 0.002	0.057 \pm 0.004	0.596 \pm 0.006	0.223

Table 14: Near OOD Detection (AURRRC score \downarrow)

Methods	ID:SST-2 OOD:IMDB	ID:Yelp P. OOD:IMDB	ID:SNLI OOD:MNLI	ID:RTE OOD:MNLI	Avg
ABC-SMC(Entropy)	0.983 \pm 0.001	0.365 \pm 0.001	0.710 \pm 0.002	0.952 \pm 0.000	0.753
ABC-SMC(MaxP)	0.983 \pm 0.001	0.365 \pm 0.001	0.694 \pm 0.000	0.952 \pm 0.000	0.749
ABC-SMC w. Weights(Entropy)	0.967 \pm 0.002	0.365 \pm 0.001	0.572 \pm 0.049	0.952 \pm 0.001	0.714
ABC-SMC W. Weights(MaxP)	0.967 \pm 0.002	0.365 \pm 0.001	0.534 \pm 0.032	0.952 \pm 0.001	0.705

Combining Psychological Theory with Language Models for Suicide Risk Detection

Daniel Izmaylov¹
izmaylov@post.bgu.ac.il

Amir Bialer¹
amirbial@post.bgu.ac.il

Avi Segal¹
avisegal@gmail.com

Meytal Grimland²
meytal.grimland@gmail.com

Yossi Levi-Belz²
yossil@ruppin.ac.il

Kobi Gal^{1,3}
kobig@bgu.ac.il

¹Ben-Gurion University of the Negev ²Ruppin Academic Center ³University of Edinburgh

Abstract

Recent years saw a dramatic increase in the popularity of online counseling services providing emergency mental health support. This paper provides a new language model for automatic detection of suicide risk in online chat sessions between help-seekers and counselors. The model adapts a hierarchical BERT language model for this task. It extends the state of the art in capturing aspects of the conversation structure in the counseling session and in integrating psychological theory into the model. We test the performance of our approach in a leading national online counseling service that operates in the Hebrew language. Our model outperformed other non-hierarchical approaches from the literature, achieving a 0.76 F2 score and 0.92 ROC-AUC. Moreover, we demonstrate our model's superiority over strong baselines even early on in the conversation, which is key for real-time detection in the field. This is a first step towards incorporating suicide predictive models in online support services and advancing NLP tools for resource-bounded languages.

1 Introduction

Suicide accounts for more than 700,000 lives lost across the world every year. It is the second leading cause of death for adolescents and adults from 15 to 29 years of age in many countries. A key effort in suicide prevention is to identify individuals at risk of suicide as early as possible (World-Health-Organization, 2021).

In the past decade, online counseling services for mental health support have become commonplace in many countries, providing chat support and guidance to at-risk individuals (see fictitious example in Figure 1). Online counseling services aim to provide mental support and address a variety of mental health crises through specialist counselors. These counselors are trained to detect suicide risk during conversations and intervene quickly as needed. These services have ex-

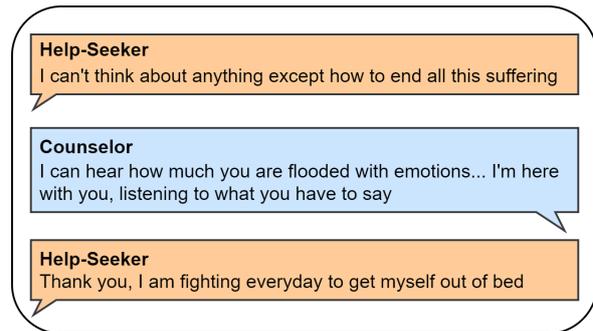


Figure 1: A fictitious example of a conversation.

perienced tremendous growth in traffic since the commencement of the COVID pandemic (Zalsman et al., 2021). Any kind of technological support to help counselors in this critical task can potentially save lives.

This paper provides a computational model for detection of suicide risk from anonymous text-based discussions between help-seekers and counselors. Our data is taken from an online counseling service in a low-resource language (Hebrew). There are several challenges towards solving suicide risk detection in our setting: State of the art pre-trained language models for suicide prevention usually focus on posts from social media which are very different in structure from online conversations between counselors and help-seekers. Existing works that do consider conversations in this domain ignore the conversation structure or are limited in the size of the conversation they consider. Also, the set of NLP resources available for low-resource languages is extremely limited when compared to English.

To address this gap, we present a hierarchical language model called SR-BERT that includes a base layer for encoding the conversation text and an additional layer for capturing aspects of conversation structure. The hierarchical structure of SR-BERT encodes each of the messages in the conversation separately and is not limited by the size

of the conversation. We hypothesized that incorporating knowledge from suicide risk theory as part of the pre-training step can improve downstream performance of the detection model. To this end, we develop a new domain knowledge-based pre-training step that embeds a Suicide Risks Factor lexicon (SRF) into SR-BERT. The SRF lexicon was created by a team of psychologists who are experts on suicide risk theory and prevention.

In empirical studies, SR-BERT significantly outperforms alternative classifiers for suicide risk (SR) detection, including the state-of-the-art (Bialer et al., 2022). We show that adding the domain-expert information to SR-BERT plays a critical part in its performance. In particular, it obtained consistently better performance than Bialer et al. (2022) when processing different portions of the conversation. These findings suggest that SR-BERT can perform well in the field when analyzing conversations in real-time.

We extend the state-of-the-art hierarchical language models to combine conversation structure and expert-based knowledge in the pretraining step. We show this approach leads to significant increases in performance for detecting suicide risk from chat conversations.

2 Related Work

This paper relates to past studies in suicide risk detection in online settings, representing domain knowledge and conversation structure in deep language models and NLP tools for low-resource languages. We expand on each of these topics in turn. For a review on using machine learning in suicide prevention, we refer the reader to Ji et al. (2021).

The majority of work using machine learning to predict suicide risk analyzes posts from social media (Coppersmith et al., 2018; Zirikly et al., 2019; Shing et al., 2018; Sawhney et al., 2018; Tadesse et al., 2019). Recent works in this social media suicide prediction space includes Cao et al. (2019) who used an LSTM with an additional attention layer to predict SR from social media posts, and Wang et al. (2021) who combined a generic BERT model with predefined rules for scoring suicide risk in social media. Additionally, Ophir et al. (2020) showed that psychological questionnaires can improve the performance of neural networks to identify at-risk individuals from Facebook posts. We significantly differ from the social media setting in our focus on conversations from online coun-

seling services, where messages are significantly longer than social media posts, and messages are part of a conversational structure and exhibit psychological dynamics. We show that capturing these aspects in the conversation model is necessary for recognizing SR in our setting.

There are few works on suicide detection in online counseling conversations, but none of these reasons about the conversation structure in the session. Most relevant to our approach is the model by Bialer et al. (2022) who combined a pre-trained language model based on BERT (Devlin et al., 2018) with a lexicon of suicide terms that were manually extracted from conversations. This model was able to represent only part of the conversation (512 tokens) and ignored the input from the counselor. Our SR-BERT model used a lexicon extracted from psychological theory, which was embedded in the pre-training process. The model significantly outperforms that of Bialer et al. (2022) on the same dataset, both for entire conversations as well as when considering early detection on parts of the conversation.

We mention two approaches for detecting SR in counseling services that did not consider early detection. Xu et al. (2021) combined a word2vec representation of suicide concepts with a bi-directional LSTM network for SR prediction in Korean online counseling service. Each side of the conversation was represented by an independent BI-LSTM. This approach used a knowledge graph to represent a psychological lexicon which may be more time-consuming for human experts to construct. Our model is shown to outperform a baseline using a similar representation (doc2vec) on our dataset. Bantilan et al. (2021) used TF-IDF embedding with XGBoost to predict SR in transcribed phone calls from an English counseling service. This model did not use a lexicon.

There is ample evidence on the benefits of incorporating domain knowledge in language models for downstream tasks (Childs and Washburn, 2019; Cao et al., 2019; Lee et al., 2020; Colon-Hernandez et al., 2021; Gaur et al., 2019). Notable examples include Gaur et al. (2019) and Wang et al. (2021) who showed that using lexicon-based features can improve machine learning prediction of suicide risk in Chinese blogs. They use lexicons to map terms from online discussions to clinically relevant sets of categories. We extend these approaches by presenting a new method for incorporating domain

knowledge in the pre-training phase of deep learning models.

In general, NLP models and solutions for low-resource languages are extremely limited. In Hebrew, two pre-trained language models were published, HeBERT (Chriqui and Yahav, 2021) and AlephBERT (Seker et al., 2022). We used AlephBERT which is freely available and was trained on a larger dataset than HeBERT and was able to outperform HeBERT on a variety of natural language tasks. We are first to use hierarchical transformer architecture to model conversation structures in a low-resource language.

3 The Sahar domain

Sahar (Hebrew acronym for Online Mental Health Support ¹) was established in 2000 and is the leading internet-based emotional support and suicide prevention organization in Israel. It provides anonymous, confidential, and free crisis support via a chat hotline (in Hebrew and in Arabic). The organization handles more than 40,000 chat sessions per year, and these numbers have increased significantly during the COVID-19 pandemic (Zalsman et al., 2021).

Sahar counselors are volunteers who receive year-round guidance and supervision from a team of mental health professionals. Shifts take place in the evening hours and are accompanied by trained therapists who monitor the conversations and provide professional support to counselors as needed. During the shifts, counselors work in a high-stress environment and usually handle multiple chat sessions in parallel at any given time. Counselors provide a written summary of each of their conversations, as well as indicate whether the conversation exhibits suicide risk.

The Sahar corpus contains more than 40,000 chat sessions (conversations) that took place over the span of five years (2017-2022). Each conversation includes the messages generated by the help-seekers and the counselors, ordered by time signatures. Table 1 presents general statistics about the dataset. We note that 39.5% of the sessions are labeled with either positive or negative SR label and 17% of these sessions are SR positive.

To validate the SR labels, a sample of 600 conversations (300 positive SR, 300 negative SR) was labeled separately by clinical psychologists with expertise in suicide theory. The Krippendorff’s α

Table 1: General statistics for Sahar corpus

Total num. of sessions	44,506
Num. of labeled sessions	17,564
SR positive label ratio	17%
Mean(Median) num. of messages	57(46)
Mean(Median) num. of turn exchanges	27(25)
Mean(Median) num. of tokens	617(566)

for inter-annotator agreement between the psychologists and the SR label in the conversation is 0.766, which is en par with other works. We note that the inconsistencies found in the samples were debated by the psychologists and resolved in the data set.

4 The SRF Psychological Lexicon

As part of our research, a team of psychology experts from a national center for suicide prevention in Israel has constructed a Suicide-Risk Factors Lexicon (SRF) in Hebrew that is based on psychological theory.

The SRF lexicon contains terms relating to personal and situational variables associated with an increase in suicidal thinking, based on valid self-report questionnaires in the psychological and psychiatric literature (Klonsky and May, 2015; Turecki and Brent, 2016; Nock et al., 2008).

Each of the 3,094 sentences in the lexicon belonged to one of 25 categories. Specifically, terms relating to depression are taken from the Patient Health Questionnaire Depression Module (PHQ-9) (Kroenke et al., 2001). Terms relating to a sense of burdensomeness are taken from the Interpersonal Needs Questionnaire (INQ) (Van Orden et al., 2012). Terms relating to a sense of hopelessness are taken from the Beck hopelessness scale (Beck et al., 1996). Terms relating to suicide behavior were taken from the Columbia questionnaire (Posner et al., 2008) which is a standard tool to measure suicide risk.

Examples of sentences for the category “perceived burdensomeness” (translated) included sentences such as “better without me”, “I am a burden”, “I spoil everything for my spouse”; and the lexicon category “explicit suicide mentions” contains phrases such as: “to die”, “to commit suicide”, “kill myself” etc.

5 The SR-BERT Language Model

Our main contribution is SR-BERT, a two-layer hierarchical language model that extends the generic DialogBERT (Gu et al., 2021) to reason about con-

¹<https://sahar.org.il>

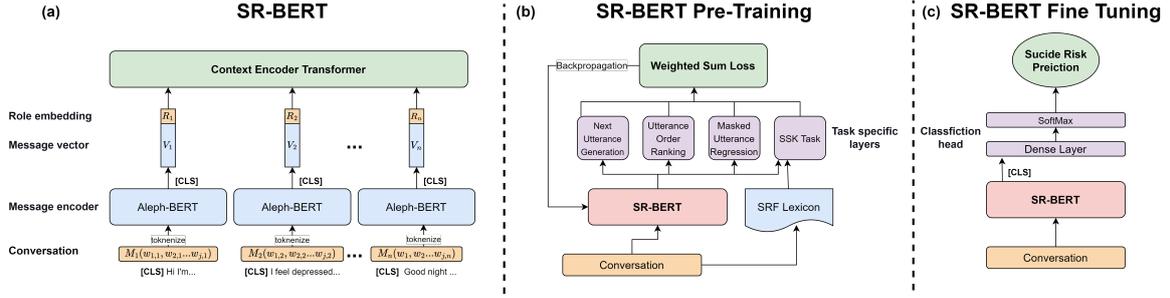


Figure 2: Model architecture. (a) SR-BERT base architecture, encoding conversation and speaker roles. (b) Pre-training procedure on 4 self-supervised tasks including psychological knowledge learning using the SRF lexicon. (c) Fine-tuning procedure learning to predict Suicide Risk (SR)

versation structure in suicide risk prediction settings and harness psychological domain knowledge. The SR-BERT architecture is shown in Figure 2(a).

The architecture is composed of two part: A transformer based layer performing message encoding, and on top of it an additional transformer layer, which captures conversation structure, named Context Encoder Transformer.

The base layer uses the AlephBERT (Seker et al., 2022) pre-trained language model to encode each message in the dialogue to a vector. The received message encoding is then combined with speaker role representation (help-seeker vs. counselor) to capture important conversation aspects such as turn-taking. The Context Encoder Transformer is a transformer based encoder applied at the message level (instead of the single token level) which transforms the series of message vectors into a context-sensitive repression of the conversation. The Context Encoder Transformer included 12 attention layers, and 12 hidden layers, each with a vector size of 780. The hidden layer size is 780 rather than 768 in AlephBert to account for the additional speaker role encoding.

The hierarchical structure of the architecture enables the model to capture multiple messages including turn exchanges and speaker roles. Furthermore, it enable the encoding of each message independently, thus avoiding the need to truncate conversations (due to AlephBert’s 512 token limit) as in past work.

5.1 Pre-training with Self Supervised Knowledge

In this section we describe the use of several pre-training tasks for adapting SR-BERT to conversation structure of online counseling, including a new pre-training task for incorporating the SRF lexicon.

This procedure uses the entire Sahar dataset, and is shown in Figure 2 (b).

The first step in this process is to represent conversations as a 25 dimension vector representing the different categories in the lexicon. For a given conversation, the value at index k is the number of sentences in the conversation with at least one occurrence in the k th lexicon category.

We also considered a reduced 5-dimension representation of conversations on the SRF lexicon space. To this end we selected the top categories using XGBoost feature selection (Chen and Guestrin, 2016) on the SR prediction task of entire conversations. We identified the top 5 categories as “self perceived burdensomeness”, “previous suicide attempt”, “loss of hope” “self injury” and “suicidal thinking”. The 5-dimension representation outperformed the 25-dimension representation on the validation set, leading us to use this representation in the subsequent pre-training phase.

The second step, called the Self Supervised Knowledge task, applies a new pre-training task for predicting Sahar conversations in the SRF representation space. For a given prefix of a conversation, we mask a message in this subset with a fixed probability of 80%. We then use SR-BERT to predict the conversation subset’s representation in the SRF space using a fully connected layer. The loss is obtained by calculating the mean squared error (MSE) between the original subset representation and the predicted (masked) representation in the SRF space. This process is repeated for increasing size of conversation prefixes, to simulate conversations of varying sizes.

In addition to the SSK task, we implemented the three pre-training tasks defined by DialogBERT (Gu et al., 2021) for capturing several aspects of the conversation structure: message-level semantics,

conversation structure, and underlying dialogue sequential order. We describe them briefly here and refer the reader to the full paper for more details.

- **Next Utterance Generation** The goal of this task is to generate the next message in the conversation when the previous messages are given. The task tries to minimize the cross-entropy loss between the predicted words and the original words of the next message.
- **Masked Utterance Regression** The goal of this task is to predict a randomly masked message in a conversation from its context. The loss is obtained by calculating the MSE between the original and the predicted message vectors.
- **Distributed Order Ranking Network** This task predicts the order index of each message from a shuffled order of a conversation. The task tries to minimize the KL divergence between the predicted order and the true order.

The calculated loss for the model propagation over the four self supervised tasks is the weighted sum of each loss function in the pre-training stage. The AdamW optimizer is employed with a linear planned warm-up technique and an initial learning rate of $5e-5$. Additionally, we use an adaptive learning-rate scheduler with 0.01 weight decay, 15,000 warm-up steps, and a batch size of 32. The model is trained for 20 epochs. All experiments are conducted on a GeForce RTX 3090 GPU using the PyTorch package.

5.2 Fine-tuning

In the fine-tuning step [Figure 2\(c\)](#), SR-BERT is adapted for the suicide risk prediction task using a standard approach ([Sun et al., 2020](#)). To this end we add a binary classification head to SR-BERT. The classification head consists of a dense layer with an output size of 2 and a softmax activation function. By maximizing the log-likelihood of the actual label, we fine-tune the Context Encoder Transformer and the classification head. We employ the AdamW optimizer with a linear planned warm-up technique and an initial learning rate of $2e-5$. Additionally, we use an adaptive learning-rate scheduler with 0.01 weight decay, and a batch size of 16. The model is trained for 10 epochs.

6 Empirical Methodology

We randomly split the labeled Sahar dataset to a train (70%) validation (15%) and test (15%) sets. These data sets were used throughout the experiments described in the following section. The validation set was used for training model hyper parameters.

We follow prior work in evaluating model performance using ROC-AUC which is widely employed in suicide detection research ([Bernert et al., 2020](#)). Additionally, we report on the F2-score ([Sokolova et al., 2006](#)) for predicting the positive SR label. This measure concentrates on reducing false negatives (rather than false positives) and is thus well suited for SR detection where missing a positive class has life threatening implications.

We compare SR-BERT with SSK to the following baseline models:

6.1 SR-BERT w.o.SSK

This model omits the SSK pre-training task from SR-BERT w. SSK. Apart from the SSK pre-training task this model is identical to SR-BERT w. SSK. including the hierarchical structure and pre-training on the other 3 tasks.

6.2 Explicit based lexicon + XGBoost

We used an XGBoost classifier that was based on an encoding of conversations over the explicit suicide related terms proposed by [Bialer et al. \(2022\)](#). This list includes 67 terms such as “commit suicide”, “cut wrists”, “wish to die” etc. We note that explicit terms carry very weak signal for SR detection.

6.3 Ensemble SI-BERT ([Bialer et al., 2022](#))

This is a non-hierarchical Hebrew language model ensembled with a classifier based on the Explicit lexicon, that represents the state of the art for SR detection. It was trained on the same dataset from the Sahar organization. To bypass BERT’s constraint of 512 tokens, Ensemble SI-BERT only utilized the help seeker text and truncated text greater than 512 tokens. We re-implemented this model with the code and parameters provided by the authors and run it on the dataset provided for this research. This is the reported state of the art for this domain in the Hebrew language.

6.4 SRF based lexicon + XGBoost

An XGBoost ([Chen and Guestrin, 2016](#)) classifier based on the 5-dimension SRF conversations rep-

Table 2: SR prediction results of compared models. Bold highlights highest value.

Model	Recall [%]	Precision [%]	ROC-AUC [%]	F2 [%]	F1 [%]
Doc2Vec+XGBoost	31.3	69.2	64.7	35.1	43.1
Explicit lexicon+XGBoost	49.2	67.1	76.9	52.3	57.7
SRF lexicon + XGBoost	55.1	67.2	76.5	57.1	60.0
Ensemble SI-BERT	60.4	70.9	91.3	62.3	65.3
SR-BERT w.o. SSK	72.9	68.4	92.1	71.9	70.6
SR-BERT w. SSK	78.3	68.9	92.1	76.2	73.3

resentation over the SRF lexicon. We note that XGBoost outperformed Random Forest and Logistic Regression as the classifier for this baseline (and for the next two baselines)

Consider for example one of the sessions which includes the statement “I am having strong stomach aches since yesterday, I want to die.”. This session includes a term from the Explicit lexicon while it is not an SR positive session.

6.5 Doc2Vec + XGBoost

An XGBoost classifier based on an encoding of each conversation to a 300-dimensional space using the Doc2Vec representation (Le and Mikolov, 2014).

7 Results

We first present the performance of the SR-BERT model in predicting SR on labeled conversations compared to the proposed baselines. Results are then reported for early SR detection, when increasing percentages of conversation information are available.

7.1 SR Detection from Complete Conversation

Table 2 compares the performance of the SR-BERT model to the baselines when predicting suicide risk from complete conversations. As seen in the table, both SR-BERT-based models (with and without SSK pre-training) outperformed the Ensemble SI-BERT model in terms of recall, F1, F2, and ROC-AUC metrics. Most notable improvement was in the recall metric where SR-BERT w.o. SSK achieved a 12.5% improvement over the Ensemble SI-BERT model, which led to a 9.6% improvement in the F2 metric. Moreover, the additional SSK pre-training improved on the SR-BERT w.o. SSK results for all metrics except the ROC-AUC score, where it hasn’t change. Ensemble SI-BERT achieved the highest precision, which was slightly

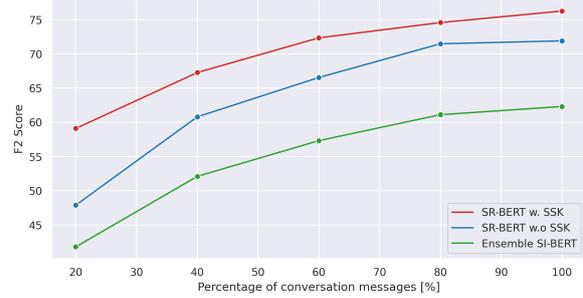


Figure 3: Classification results for early detection of top-performing SR detection approaches

better than SR-BERT w. SSK. It exhibited a substantially lower recall score, which correlates to lower F1 and F2 values.

The SRF lexicon + XGBoost based classifier was better than the Explicit lexicon + XGBoost classifier in all measures apart from ROC-AUC. We also note that the BERT based models outperformed the none BERT models on all tested metrics.

We used the McNemar paired test for labeling disagreements (Gillick and Cox, 1989) to compare between the predictions of the different models. Statistical significance with $p < 0.05$ was demonstrated for SR-BERT w. SSK vs. SR-BERT w.o. SSK and for SR-BERT w. SSK vs. Ensemble SI-BERT.

Overall SR-BERT w. SSK achieved a substantial improvement in recall and F2 compared the Ensemble SI-BERT of 17.9% and 13.9% respectively, with only a slight decrease in precision performance. This is critical in the suicide risk detection realm where recall is key to identifying help-seekers at risk and enabling targeted support.

7.2 Early SR Detection

Evaluating the ability of SR-BERT to predict SR risk from partial sessions provides an indication of its performance in real time, when only part of the session is available. To this end, Figure 3 compares the performance of the different models after

receiving the first {20, 40, 60, 80, 100} percent of messages in the session. As seen in the figure, the performance of all models improved as the sessions progressed. However, SR-BERT w. SSK model consistently outperformed the other models, followed by SR-BERT w.o. SSK. The difference in performance between SR-BERT with SSK and SR-BERT w.o. SSK was the largest at the beginning of the session and reduced as the sessions advanced. This may indicate the contribution of SR-BERT w. SSK to identify risk variables from the lexicon in early stages of the dialogue when information is lacking. In contrast, the difference in performance between SR-BERT w.o. SSK and the Ensemble SI-BERT model increases as sessions advance. This could be due to the inability of Ensemble SI-BERT to process the lengthy dialogue without having to truncate it, which may result in the loss of important information as sessions develop.

8 Conclusion and Future Work

This work has provided a new automatic approach for suicide risk detection in online conversations between help-seekers and counselors. Early detection of at-risk individuals is a key goal of suicide prevention. Our approach extends the state-of-the-art in deep language modeling by 1) incorporating domain knowledge relevant to suicide risk detection as part of the pre-training step; 2) reasoning about the structure of the conversation between help-seekers and counselors; 3) adapting to a low-resource language (Hebrew). The presented approach was able to significantly outperform the state-of-the-art approaches when detecting SR from complete conversations, as well as early detection when only part of the conversation is available. These results suggest the model may be able to support the work of counselors in real chat sessions, alerting them in real-time to at-risk individuals and enabling quick and focused responses. For future work, we intend to improve our approach by capturing more aspects of conversations, such as prosody (Wilson and Wharton, 2006; Kliper et al., 2010) as well as model the mental state dynamics of the help-seeker. We are also extending the model with explanations to be able to provide justifications for predictions made and point to key exchanges and phrases that triggered specific predictions.

9 Limitations

We note several limitations of this study.

First, our model was evaluated only in the Hebrew language. We have not directly compared SR-BERT to approaches for detecting suicidal risk in non-Hebrew domains, and note that the effectiveness of the model may vary across different languages and cultural contexts. It is difficult to make this comparison given the lack of public data sets from online counseling services.

Second, the proposed approach relies on the existence of psychological knowledge for pre-training the SR-BERT model which requires human effort. On the one hand, psychological lexicons already exist in English (Lee et al., 2020) and possibly in other languages. On the other hand, lexicons inherently suffer from limited coverage, lack of context and are expensive to maintain. Sharing domain knowledge across research tasks may go a long way to overcome these issues. We intend to make the lexicon developed for this research publicly available.

Third, the annotation of the help seekers' mental state was performed by the counselors, rather than the help seekers themselves. While the counselors underwent a thorough training process lasting several months and were monitored by certified clinical psychologists, there is still the possibility that they may have misclassified the mental state of the help seekers. This issue is prevalent in many studies that rely on observer-reported data.

Finally, the current model does not provide any explanations for its predictions, which are of high importance in order to support counselors in the field. This is essential in order to ensure that the model is not merely a means of classification but instead is able to provide valuable insights and assistance to counselors. This is a key focus of our future development plans.

10 Ethics Statement

The present study has been conducted in accordance with the highest ethical standards and has been approved by the relevant institutional review board of the participating institutions. All data utilized in this study, including the Sahar corpus of conversations between help-seekers and counselors, and the SRF psychological lexicon, have been obtained in compliance with the IRB. Specifically, the Sahar dataset has been anonymized and encrypted to protect the privacy of the participants, and all

help-seekers who have provided data for this study have given informed consent for the anonymous use of their sessions for research purposes. The counselors signed consent papers to allow the usage of their text data for the study.

It is important to note that despite the model's ability to successfully predict SR during the conversation and its demonstrated gender fairness, it is not intended to replace human volunteer counselors. We believe that human involvement is essential in providing support to help-seekers, and the role of an automated model is to serve as an aid to counselors, enhancing their ability to assess SR rather than replacing them. Our take is that in the future, when such models could be deployed in the field (after all necessary approvals and adaptations), they may only act as a "friendly parrot" on the counselors' shoulders, providing additional insights and supporting their decision-making process in the high load situations these counselors are facing on a daily basis.

References

- Niels Bantilan, Matteo Malgaroli, Bonnie Ray, and Thomas D. Hull. 2021. [Just in time crisis response: Suicide alert system for telemedicine psychotherapy settings](#). 31(3):302–312.
- Aaron T Beck, Robert A Steer, and Gregory Brown. 1996. Beck depression inventory–ii. *Psychological assessment*.
- Rebecca A Bernert, Amanda M Hilberg, Ruth Melia, Jane Paik Kim, Nigam H Shah, and Freddy Abnoui. 2020. Artificial intelligence and suicide prevention: a systematic review of machine learning investigations. *International journal of environmental research and public health*, 17(16):5929.
- Amir Bialer, Daniel Izmaylov, Avi Segal, Oren Tsur, Yossi Levi-Belz, and Kobi Gal. 2022. Detecting Suicide Risk in Online Counseling Services: A Study in a Low-Resource Language. The 29th International Conference on Computational Linguistics (COLING-22) <http://shorturl.at/crXY2>.
- Lei Cao, Huijun Zhang, Ling Feng, Zihan Wei, Xin Wang, Ningyun Li, and Xiaohao He. 2019. [Latent Suicide Risk Detection on Microblog via Suicide-Oriented Word Embeddings and Layered Attention](#).
- Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.
- Christopher M. Childs and Newell R. Washburn. 2019. [Embedding domain knowledge for machine learning of complex material systems](#). 9(3):806–820.
- Avihay Chriqui and Inbal Yahav. 2021. [HeBERT & HebEMO: A Hebrew BERT Model and a Tool for Polarity Analysis and Emotion Recognition](#).
- Pedro Colon-Hernandez, Catherine Havasi, Jason Alonso, Matthew Huggins, and Cynthia Breazeal. 2021. [Combining pre-trained language models and structured knowledge](#).
- Glen Coppersmith, Ryan Leary, Patrick Crutchley, and Alex Fine. 2018. Natural language processing of social media as screening for suicide risk. *Biomedical informatics insights*, 10:1178222618792860.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Manas Gaur, Amanuel Alambo, Joy Prakash Sain, Ugur Kursuncu, Krishnaprasad Thirunarayan, Ramakanth Kavuluru, Amit Sheth, Randy Welton, and Jyotishman Pathak. 2019. [Knowledge-aware Assessment of Severity of Suicide Risk for Early Intervention](#). In *The World Wide Web Conference on - WWW '19*, pages 514–525. ACM Press.
- Laurence Gillick and Stephen J Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. pages 532–535.
- Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. 2021. Dialogbert: Discourse-aware response generation via learning to recover and rank utterances. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12911–12919.
- Shaoxiong Ji, Shirui Pan, Xue Li, Erik Cambria, Guodong Long, and Zi Huang. 2021. [Suicidal Ideation Detection: A Review of Machine Learning Methods and Applications](#). 8(1):214–226.
- Roi Kliper, Yonatan Vaizman, Daphna Weinsahl, and Shirley Portuguese. 2010. Evidence for depression and schizophrenia in speech prosody. In *Proceedings of the 3rd ICSA Tutorial and Research Workshop on Experimental Linguistics 2010*, pages 85–88.
- E David Klonsky and Alexis M May. 2015. The three-step theory (3st): A new theory of suicide rooted in the "ideation-to-action" framework. *International Journal of Cognitive Therapy*, 8(2):114–129.
- Kurt Kroenke, Robert L Spitzer, and Janet BW Williams. 2001. The phq-9: validity of a brief depression severity measure. *Journal of general internal medicine*, 16(9):606–613.
- Quoc Le and Tomas Mikolov. 2014. [Distributed representations of sentences and documents](#). 32(2):1188–1196.

- Daeun Lee, Soyoung Park, Jiwon Kang, Daejin Choi, and Jinyoung Han. 2020. Cross-lingual suicidal-oriented word embedding toward suicide prevention. pages 2208–2217.
- Matthew K Nock, Guilherme Borges, Evelyn J Bromet, Christine B Cha, Ronald C Kessler, and Sing Lee. 2008. Suicide and suicidal behavior. *Epidemiologic reviews*, 30(1):133–154.
- Yaakov Ophir, Refael Tikochinski, Christa S. C. Asterhan, Itay Sisso, and Roi Reichart. 2020. [Deep neural networks detect suicide risk from textual facebook posts](#). 10(1):16685.
- K Posner, D Brent, C Lucas, M Gould, B Stanley, G Brown, P Fisher, J Zelazny, A Burke, MJNY Oquendo, et al. 2008. Columbia-suicide severity rating scale (c-ssrs). *New York, NY: Columbia University Medical Center*, 10.
- Ramit Sawhney, Prachi Manchanda, Puneet Mathur, Rajiv Shah, and Raj Singh. 2018. Exploring and learning suicidal ideation connotations on social media with deep learning. In *Proceedings of the 9th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 167–175.
- Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Greenfeld, and Reut Tsarfaty. 2022. Alephbert: Language model pre-training and evaluation from sub-word to sentence level. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–56.
- Han-Chin Shing, Suraj Nair, Ayah Zirikly, Meir Friedenberg, Hal Daumé III, and Philip Resnik. 2018. Expert, crowdsourced, and machine assessment of suicide risk via online postings. In *Proceedings of the fifth workshop on computational linguistics and clinical psychology: from keyboard to clinic*, pages 25–36.
- Marina Sokolova, Nathalie Japkowicz, Stan Szpakowicz, and Stan Szpakowicz. 2006. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer.
- Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. [How to Fine-Tune BERT for Text Classification?](#)
- Michael Mesfin Tadesse, Hongfei Lin, Bo Xu, and Liang Yang. 2019. Detection of suicide ideation in social media forums using deep learning. *Algorithms*, 13(1):7.
- Gustavo Turecki and David A Brent. 2016. Suicide and suicidal behaviour. *The Lancet*, 387(10024):1227–1239.
- Kimberly A Van Orden, Kelly C Cukrowicz, Tracy K Witte, and Thomas E Joiner Jr. 2012. Thwarted belongingness and perceived burdensomeness: construct validity and psychometric properties of the interpersonal needs questionnaire. *Psychological assessment*, 24(1):197.
- Rui Wang, Bing Xiang Yang, Yujun Ma, Peilin Wang, Qiao Yu, Xiaofen Zong, Zhen Huang, Simeng Ma, Long Hu, Kai Hwang, and Zhongchun Liu. 2021. [Medical-Level Suicide Risk Analysis: A Novel Standard and Evaluation Model](#). 8(23):16825–16834.
- Deirdre Wilson and Tim Wharton. 2006. Relevance and prosody. *Journal of pragmatics*, 38(10):1559–1579.
- World-Health-Organization. 2021. Live life: an implementation guide for suicide prevention in countries.
- Zhongzhi Xu, Yucan Xu, Florence Cheung, Mabel Cheng, Daniel Lung, Yik Wa Law, Byron Chiang, Qingpeng Zhang, and Paul S.F. Yip. 2021. [Detecting suicide risk using knowledge-aware natural language processing and counseling service data](#). 283:114176.
- Gil Zalsman, Yael Levy, Eliane Sommerfeld, Avi Segal, Dana Assa, Loona Ben-Dayana, Avi Valevski, and J John Mann. 2021. Suicide-related calls to a national crisis chat hotline service during the covid-19 pandemic and lockdown. *Journal of psychiatric research*.
- Ayah Zirikly, Philip Resnik, Ozlem Uzuner, and Kristy Hollingshead. 2019. Clpsych 2019 shared task: Predicting the degree of suicide risk in reddit posts. In *Proceedings of the sixth workshop on computational linguistics and clinical psychology*, pages 24–33.

Cross-Lingual Question Answering over Knowledge Base as Reading Comprehension

Chen Zhang¹, Yuxuan Lai³, Yansong Feng^{1,2*},
Xingyu Shen¹, Haowei Du¹, Dongyan Zhao^{1,4,5}

¹ Wangxuan Institute of Computer Technology, Peking University, China

² The MOE Key Laboratory of Computational Linguistics, Peking University, China

³ Department of Computer Science, The Open University of China

⁴ National Key Laboratory of General Artificial Intelligence

⁵ Beijing Institute for General Artificial Intelligence

{zhangch, fengyansong, shenxy, zhaody}@pku.edu.cn

duhaowei@stu.pku.edu.cn laiyx@ochn.edu.cn

Abstract

Although many large-scale knowledge bases (KBs) claim to contain multilingual information, their support for many non-English languages is often incomplete. This incompleteness gives birth to the task of cross-lingual question answering over knowledge base (xKBQA), which aims to answer questions in languages different from that of the provided KB. One of the major challenges facing xKBQA is the high cost of data annotation, leading to limited resources available for further exploration. Another challenge is mapping KB schemas and natural language expressions in the questions under cross-lingual settings. In this paper, we propose a novel approach for xKBQA in a reading comprehension paradigm. We convert KB subgraphs into passages to narrow the gap between KB schemas and questions, which enables our model to benefit from recent advances in multilingual pre-trained language models (MPLMs) and cross-lingual machine reading comprehension (xMRC). Specifically, we use MPLMs, with considerable knowledge of cross-lingual mappings, for cross-lingual reading comprehension. Existing high-quality xMRC datasets can be further utilized to finetune our model, greatly alleviating the data scarcity issue in xKBQA. Extensive experiments on two xKBQA datasets in 12 languages show that our approach outperforms various baselines and achieves strong few-shot and zero-shot performance. Our dataset and code are released for further research¹.

1 Introduction

Large-scale knowledge bases (KBs) such as Freebase (Bollacker et al., 2008) and DBpedia (Auer et al., 2007) store huge amounts of structured

Question:

谁在蜘蛛侠2中扮演玛丽简?

スパイダーマン2でメリージェーンを演じるのは誰ですか?

Qui joue Mary Jane dans Spiderman 2?

(Who plays Mary Jane in Spiderman 2?)

Knowledge Base (Subgraph):

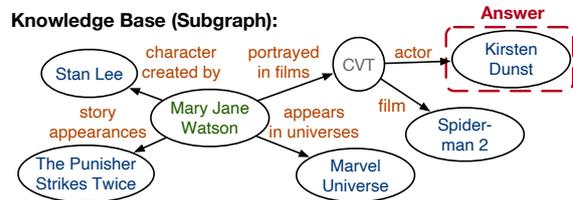


Figure 1: An example of answering questions in non-English languages over an English knowledge base.

knowledge. These KBs support a variety of natural language processing tasks, including question answering over knowledge base (KBQA), where models exploit the knowledge related to the questions and precisely identify the answers by reasoning through various KB relations. Although most large-scale KBs claim to contain multilingual information, they could not completely support non-English languages as expected. For example, Freebase has no translation for the KB relations/attributes in any non-English languages. More than half of the entities in Freebase have no Chinese translations, despite the fact that Chinese is the most spoken non-English language in the world. Therefore, these KBs could not directly support question answering in non-English languages, bringing up the problem of answering non-English questions over the KBs constructed in English.

In this work, we focus on cross-lingual KBQA (xKBQA), which aims to answer questions over a KB in another language. Figure 1 shows a KB subgraph and several factoid questions in non-English languages, which can be answered by a node in the KB subgraph. Despite considerable progress in monolingual KBQA, xKBQA receives little attention. A significant challenge in xKBQA is the

* Corresponding author.

¹<https://github.com/luciusssss/xkbqa-as-mrc>

lack of large-scale xKBQA datasets. Such datasets are quite expensive to annotate since the annotators are expected to be multilingual and have background knowledge about KBs. As a result, even the largest xKBQA dataset so far contains only a few hundred questions (Ngomo, 2018). Another challenge is that, compared to other cross-lingual tasks, the expression difference between structured KB schemas and natural language questions further hinders the learning of cross-lingual mapping.

To address these challenges, we propose to convert the KB subgraphs into natural language texts and leverage the progress in cross-lingual machine reading comprehension (xMRC) to solve the xKBQA task. Recently, there has been a series of large-scale xMRC datasets, such as MLQA (Lewis et al., 2020), MKQA (Longpre et al., 2021) and XQuAD (Artetxe et al., 2020). Multilingual pre-trained language models (MPLMs), such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), achieve competitive performance on these xMRC benchmarks. As for xKBQA, by converting KB subgraphs into natural language texts, we narrow the gap between KB schemas and natural language expressions. We then utilize the PLM-based xMRC models finetuned on xMRC datasets to learn the cross-lingual mapping efficiently, even with limited xKBQA annotations.

Specifically, we first identify the topic entity from the given question, link it to the KB, and extract its n -order neighbors to construct a KB subgraph, following traditional monolingual KBQA methods (Saxena et al., 2020; He et al., 2021). We then convert the subgraph into a question-specific passage with KB-to-text generation models, incorporating the KB triples with contextual expressions. Given the converted cross-lingual question-passage pairs, we adopt MPLMs to rank answer candidates in the passages. As a general framework, our approach can be easily applied to different languages or KBs without specialized modifications.

We empirically investigate the effectiveness of our method on two xKBQA datasets, QALD-M (Ngomo, 2018) and WebQSP-zh. QALD-M is a collection of a few hundred questions in 11 non-English languages, from a series of xKBQA evaluation campaigns. Considering its small size, we also construct a new dataset WebQSP-zh with 4,737 Chinese questions translated from WebQSP (Yih et al., 2016) by native speakers. WebQSP-zh is much larger in size and involves more natural

expressions as the annotators take into account commonsense knowledge and realistic vocabulary choices during manual translation.

Experimental results demonstrate that our method outperforms a variety of English-as-pivot baselines based on monolingual KBQA models, reaching 74.37% hits@1 on WebQSP-zh. Moreover, our method achieves strong few-shot and zero-shot performance. Using only 10% of the training data, our method performs comparably to several competitive English-as-pivot baselines trained with full training data. For the zero-shot evaluation on QALD-M, our method achieves 51.20% hits@1 on average across 11 languages.

Our main contributions are summarized as:

- We formulate xKBQA as answering questions by reading passages converted from KB subgraphs, bridging the gap between KB schemas and natural language expressions. Existing high-quality xMRC resources are further utilized to alleviate the data scarcity issue.
- We collect a large xKBQA dataset with native expressions in Chinese, i.e., WebQSP-zh. It, along with its original version, i.e., WebQSP, can be used for analyzing the gap between monolingual and cross-lingual KBQA.
- We conduct extensive experiments on two datasets with 12 languages. Our method outperforms various baselines and achieves strong few-shot and zero-shot performance.

2 Related Works

KBQA Recent efforts in KBQA generally fall into two main paradigms, either the information extraction style (Miller et al., 2016; Sun et al., 2018; Xu et al., 2019; Saxena et al., 2020; He et al., 2021; Shi et al., 2021) or the semantic parsing style (Yih et al., 2015; Lan and Jiang, 2020; Ye et al., 2022; Gu and Su, 2022). The former retrieves a set of candidate answers from KB, which are then compared with the questions in a condensed feature space. The latter manages to distill the symbolic representations or structured queries from the questions.

xKBQA Both styles of KBQA methods can be applied to xKBQA. Previous xKBQA efforts generally fall in the semantic parsing style. They rely on online translation tools (Hakimov et al., 2017) or embedding-based word-to-word translation (Zhou et al., 2021) to obtain synthetic training data. In

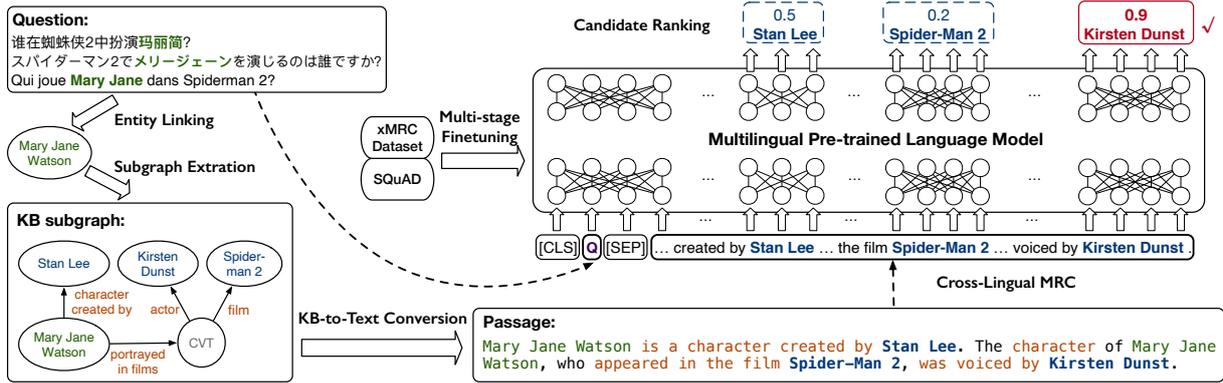


Figure 2: An illustration of our method. Given a large-scale KB in a rich-resource language such as English, to answer a question in a language with relatively fewer resources, we first extract a subgraph from KB according to entity linking results. We then convert the subgraph into a question-specific passage in natural language with KB-to-text generation models, complementing the KB triples with contextual expressions. The question and converted passage are fed into a PLM-based xMRC model, which ranks all candidate answer spans to obtain the final answer.

contrast, the information extraction based xKBQA approach is less explored. An advantage of this style of xKBQA methods is that it requires no annotation of structured queries, which is expensive to obtain for non-English languages. In this paper, we attempt to explore xKBQA approaches of the information extraction style with less reliance on machine translation tools and investigate their performance in the few-shot and zero-shot settings.

xMRC xMRC is a cross-lingual QA task receiving extensive attention recently, with considerable progress in datasets and models. There has been a stream of high-quality datasets in a wide range of languages, including MLQA (Lewis et al., 2020), MKQA (Longpre et al., 2021), XQuAD (Artetxe et al., 2020) and TyDi QA (Clark et al., 2020). Several works for xMRC adopt machine translation tools (Asai et al., 2018; Cui et al., 2019; Lee et al., 2019) or question generation systems (Riabi et al., 2021) to obtain more cross-lingual training data, while other works attempt to learn better cross-lingual mapping with MPLMs (Yuan et al., 2020; Wu et al., 2022).

KB-to-text in QA To benefit xKBQA with the progress in xMRC, we propose to convert the xKBQA task into reading comprehension. Previous works in other QA tasks attempt to convert KB triples into texts by simple concatenating heuristics (Oguz et al., 2020) or by manually-designed rules (Bian et al., 2021). Ma et al. (2022) resort to PLM-based generation models and argue that data-to-text can serve as a universal interface for open domain

QA. To the best of our knowledge, our work is the first to introduce data-to-text methods into KBQA and cross-lingual QA. Compared with Ma et al. (2022), we further address the real-world problems of complex KB structures, cross-lingual semantic gap, and data scarcity when applying data-to-text to xKBQA.

3 Methodology

We propose a novel approach to tackle xKBQA as reading comprehension. As illustrated in Figure 2, we first convert KB triples into sentences using generation models and obtain question-specific passages for reading comprehension. We then adopt MPLMs finetuned on xMRC datasets to answer cross-lingual questions according to the converted passages.

3.1 Task Formulation

In xKBQA, given a knowledge base G in language A and a question q in another language B , the model is expected to answer q by entities or literal values in G . In practice, A is often a rich-resource language such as English, and B is a language with relatively fewer resources. A knowledge base G consists of a set of knowledge triples. In a triple (h, r, t) , $h \in E$ is a head entity, $t \in E \cup L$ is a tail entity or a literal value, and $r \in R$ is the relation/predicate between h and t , where E denotes the set of all entities, L denotes the set of all literal values, and R denotes the set of all relations.

3.2 KB-to-Text Conversion

In a typical monolingual KBQA framework, one first identifies the topic entity in the question and links it to the given KB. This can be achieved by surface-level matching (Sun et al., 2018) or supervised entity linkers (Yang and Chang, 2015). In the cross-lingual setting, one can directly adopt multilingual entity linkers such as mGENRE (De Cao et al., 2022) or translate questions and KB entities into the same language for monolingual linking.

After entity linking, a KB subgraph is constructed by the neighbors within several hops around the topic entities. Based on the given question, all candidates in the subgraph are ranked to arrive at the final answers. To successfully identify from the subgraph the KB predicates leading to the answer, the KBQA models are expected to learn a mapping between KB predicates and natural language expressions in the questions. In addition to the language gap as in most cross-lingual tasks, the models have to deal with the difference in expression styles used in the KB schemas and questions.

To narrow down the gap of mapping, we propose to convert KB subgraphs to natural language passages, formulating xKBQA as an xMRC task, so that we can benefit from recent advances in xMRC. Converting KB subgraphs into natural sentences brings plausible context for candidate KB answers, facilitating the matching between questions and answers. Furthermore, with the natural language expressions of the KB subgraphs, current xMRC models can be directly adopted to solve the questions. We believe that xMRC models could benefit the xKBQA task for their strong capabilities of mapping between cross-lingual expressions. Even without annotated xKBQA data, they are able to answer a portion of xKBQA questions, utilizing their prior knowledge of the cross-lingual mapping learned from pre-training and fine-tuning on xMRC datasets.

To convert KB subgraphs into readable passages, we utilize PLM-based KB-to-text models, such as JointGT (Chen et al., 2021). A KB-to-text model converts a structured KB subgraph to natural language texts, complementing the given entities and relations with potential contextual expressions. Compared with simply concatenating the head entity, relation and tail entity of a triple, a KB-to-text model can generate more natural and coherent sentences. It also alleviates the onerous manual design of conversion rules. Moreover, the KB-to-text

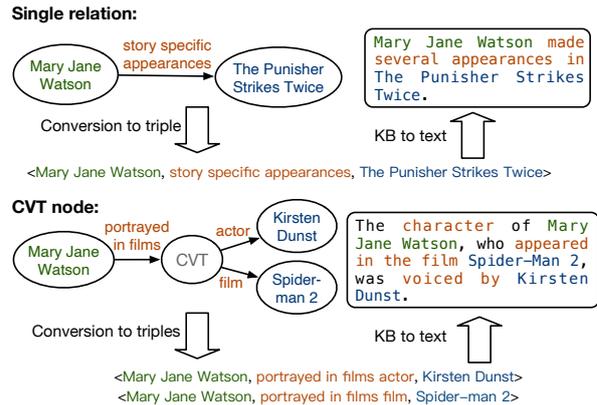


Figure 3: Examples of KB-to-text conversion for a single relation (upper) and a complex event-like fact, such as CVT nodes in Freebase (lower).

model can handle not only single-relation triples but also more complex KB structures, such as CVT nodes, which is a complex node type in Freebase referring to an event with multiple fields. Figure 3 shows examples of KB-to-text conversion for a single-relation triple and a CVT node.

After conversion, we identify the candidate answer spans from the pieces of text with fuzzy string matching tools. To form a passage, we concatenate the pieces of text, sorted by their semantic similarities to the questions.² We observe that the subgraphs around a topic entity can be very large, especially for the *hub* entities like *the USA*. Consequently, the converted passages can be very long, even up to 20k words in length. Current xMRC models struggle with such long passages. To shorten the converted passages, we fix the maximum length of the passage and discard the remaining redundant sentences.

3.3 Cross-Lingual Reading Comprehension

MPLMs are widely adopted in xMRC for their strong capabilities of understanding cross-lingual texts. They can encode different languages in a unified semantic space, relieving the reliance on translation tools. We thus use MPLMs to solve the xMRC instances converted from xKBQA.

Specifically, we concatenate the question and the converted passage as the input to the MPLMs and predict the boundary of the answer span. In the KB-to-text step, we have identified the corresponding

²Previous work shows that PLM-based MRC models are not sensitive to the order of sentences in the passage (Sugawara et al., 2020). We do not observe significant performance change after we shuffle the sentence order in the passage, which conforms to the finding by Sugawara et al. (2020).

WebQSP-zh: 安娜肯德里克出演过什么? / *What did Anna Kendrick star in?*

WebQSP-MT: 安娜肯德里克在干什么? / *What is Anna Kendrick doing?*

WebQSP: What has Anna Kendrick been in?

Freebase Predicate: film.actor.film film.performance.film

WebQSP-zh: 1945年前苏联的领导人是谁? / *Who was the leader of the former Soviet Union in 1945?*

WebQSP-MT: 1945年苏联的领导人是谁? / *Who was the leader of the Soviet Union in 1945?*

WebQSP: Who was the leader of the Soviet Union in 1945?

Freebase Predicate:

government.governmental_jurisdiction.governing_officials
government.government_position_held.office_holder

Table 1: Examples from WebQSP-zh and their corresponding questions in WebQSP. WebQSP-MT is the Chinese translation of WebQSP by Baidu Translate, a machine translation tool. The italic English texts are the literal meaning of the Chinese questions.

span in the passage for each candidate KB entity or literal value. Thus, during inference, we only need to rank the candidate answer spans. The corresponding KB entity or value for the top-ranked candidate span is selected as the final answer.

To address the data scarcity in xKBQA, we further propose to finetune the models on MRC data in multiple stages before on xKBQA data. Compared to KBQA, it is easier to acquire annotated MRC data for its straightforward annotation process without the requirement of background knowledge in KBs. Apart from large-scale English MRC datasets such as SQuAD (Rajpurkar et al., 2016), there are a series of high-quality xMRC datasets, including MLQA, MKQA and XQuAD, covering a wide range of non-English languages such as Russian, Hindi, and Dutch. In the first stage, we use large-scale English MRC datasets, e.g., SQuAD, to help MPLMs learn the language-agnostic ability to find answers from the passages. In the second stage, we finetune the models on high-quality xMRC datasets in the target language, strengthening the reading comprehension ability for the target language. In this way, the two-stage finetuning before training on xKBQA data benefits models with the rich resources in MRC and mitigate the data scarcity problem in xKBQA.

4 Experimental Setup

4.1 Datasets

We evaluate our method on two datasets, **QALD-M**, a small evaluation dataset in 11 languages, and

WebQSP-zh, a new dataset with a larger size and more realistic expressions.

QALD-M QALD-M is a series of evaluation campaigns on question answering over linked data. We use the version provided by Zhou et al. (2021) and filter the out-of-scope ones. It consists of testing questions for 11 non-English languages (fa, de, ro, it, ru, fr, nl, es, hi, pt, pt_BR) over DBPedia. The numbers of used questions for each language range from 66 to 363. We use QALD-M mainly for zero-shot evaluation. See Appendix A.1 for more details.

WebQSP-zh Considering that the size of QALD-M is small and its multilingual questions are mostly literal translations without language-dependent paraphrasing, we collect a new xKBQA dataset WebQSP-zh, with 3,098 questions for training and 1,639 questions for test.

To collect WebQSP-zh, we employ two Chinese native speakers proficient in English to manually translate all the questions in WebQSP (Yih et al., 2016), a widely-used English KBQA dataset, together with another annotator responsible for checking translation quality. To provide a more realistic benchmark for cross-lingual evaluation, the annotators are instructed to pay much attention to commonsense knowledge and natural vocabulary choices during translation. For example, in the upper example of Table 1, the phrase *be in* in the WebQSP question has multiple translations in Chinese. Based on the commonsense knowledge that *Anna Kendrick* is an actress, it is translated as *出演/star in* instead of its literal meaning *在做/be doing*. In the lower example of Table 1, the annotator chooses the Chinese word *前苏联/former Soviet Union* for translation instead of *苏联/Soviet Union* because the former is more often used by native Chinese speakers. See Appendix A.2 for more statistics, annotation details, and examples.

4.2 Baselines

Supervised A widely-adopted baseline method in cross-lingual QA tasks is translating data in non-English languages into English with machine translation tools and utilizing mono-lingual models (Asai et al., 2018; Cui et al., 2019), which we call **English-as-pivot**. For supervised experiments on WebQSP-zh, we select several competitive monolingual KBQA models for English-as-pivot evaluation. For information extraction style, we select **EmbedKGQA** (Saxena et al., 2020),

GraftNet (Sun et al., 2018), **NSM** (with its teacher-student variant, He et al., 2021), all of which require no annotation of structured KB queries, as our method does. For semantic parsing style, we select **QGG** (Lan and Jiang, 2020).³

We also provide a **Closed-book QA** baseline (Roberts et al., 2020) with generation-based MPLMs, e.g., mT5 (Xue et al., 2021). We feed the question directly into the model and expect it to output the answer based on its knowledge learned in pre-training. This method requires no external knowledge, such as KBs, and can coarsely evaluate how much parametric knowledge an MPLM may have.

Zero-shot Since the above supervised baselines are unable to answer any questions without training data, we further implement two baselines inspired from Zhou et al. (2021) for zero-shot evaluation. One is **Multilingual Semantic Matching**, which measures the similarity between questions and inferential chains with an MPLM finetuned on LC-QuAD (Trivedi et al., 2017), an English KBQA dataset. The other, based on the previous baseline, uses **Bilingual Lexicon Induction** (BLI, Lample et al., 2018) to obtain word-to-word translation in the target languages as data augmentation.

4.3 Metrics

Following previous works (Saxena et al., 2020; He et al., 2021), we use hits@1 as the evaluation metric. It is the ratio of questions whose top 1 predicted answer is in the set of golden answers.

4.4 Implementation Details

Following previous works (Sun et al., 2018; Saxena et al., 2020; He et al., 2021), we use the golden topic entities for a fair comparison with the baselines. We also discuss the effects of entity linking in Section 5.5. For KB-to-text generation, we use JointGT (Chen et al., 2021) finetuned on WebNLG (Gardent et al., 2017), a KB-to-text dataset. We use TheFuzz⁴ to identify candidate answer spans. We fix the maximum passage length to 750 words and discard the sentences with lower

³We did not include the recent semantic-parsing-style models based on Seq2Seq generation, including RnG-KBQA (Ye et al., 2022) and ArcaneQA (Gu and Su, 2022), both of which outperform QGG by 1.6% F1 on WebQSP. However, setting up an environment for them requires up to 300G memory, far exceeding our computational budgets. So we think that QGG is a suitable baseline that strikes a good balance between performance and computational resources.

⁴<https://github.com/seatgeek/thefuzz>

Model	WebQSP	WebQSP-zh
<i>English-as-pivot</i>		
EmbedKGQA (2020)	66.18	63.15 (-3.03)
GraftNet (2018)	67.79	65.61 (-2.18)
NSM (2021)	68.70	67.30 (-1.40)
NSM-student (2021)	74.30	72.54 (-1.76)
QGG (2020)	73.70	72.36 (-1.34)
<i>Closed-book QA</i>		
mT5-base		7.02
mT5-large		12.87
<i>xKBQA-as-MRC (Ours)</i>		
mBERT-base		70.53
XLM-R-base		69.92
XLM-R-large		74.37

Table 2: Hits@1 (%) of baselines and our method on the test set of WebQSP-zh using the full training data. The “WebQSP” column shows the model performance on the test set of WebQSP after training on the original English WebQSP data. The numbers in the brackets denote the performance drop of English-as-pivot models compared to their corresponding English KBQA models on WebQSP. All models except GraftNet use golden topic entities.

semantic similarity to the questions, measured by the multilingual model of SentenceTransformers (Reimers and Gurevych, 2020). For xMRC, we experiment with mBERT and XLM-R. Before finetuning on the xMRC instances converted from xKBQA datasets, we first finetune models on SQuAD 1.1, and then on three xMRC datasets, MLQA, MKQA and XQuAD. We do not search hyperparameters for the xMRC models and adopt the default configuration used by SQuAD. For English-as-pivot baselines, we use Baidu Translate API⁵ to obtain English translations. See Appendix B for more details.

5 Results and Analyses

5.1 Supervised Setting

As shown in Table 2, we first compare our method with English-as-pivot baselines using full training data of WebQSP-zh. These baselines can benefit from the development of monolingual KBQA models and achieve over 63% hits@1 on WebQSP-zh. Suppose we have perfect translation results, the English-as-pivot baselines on the WebQSP-zh should reach the performance of monolingual models on the original English WebQSP. However, the English-as-pivot baselines on WebQSP-zh drop 1.4-3.0% hits@1 compared to their monolingual per-

⁵<https://fanyi.baidu.com/>

Model	fa	de	ro	it	ru	fr	nl	es	hi	pt	pt_BR	Avg.
<i>Multilingual Semantic Matching</i>												
LC-QuAD	43.41	44.90	48.55	47.93	36.84	47.38	43.93	46.53	41.60	37.43	48.48	44.27
+ Sing. BLI	46.41	50.41	50.87	51.24	40.35	48.76	48.55	49.42	34.73	40.35	54.54	46.88
+ All BLI	46.41	49.31	50.58	49.04	41.52	49.59	47.40	48.55	41.98	40.94	51.51	46.98
<i>xKBQA-as-MRC (Ours)</i>												
SQuAD	39.22	48.21	44.48	45.45	33.33	45.17	48.27	47.11	43.89	35.67	51.51	43.85
+ Sing. xMRC	39.22	52.07	52.91	56.20	45.61	51.24	52.02	54.62	50.76	42.69	59.09	50.59
+ All xMRC	48.50	55.10	52.03	54.27	44.44	53.44	52.89	53.47	46.95	41.52	60.61	51.20

Table 3: Hits@1 (%) of the baseline and our method with XLM-R-large on QALD-M under the zero-shot setting. “LC-QuAD” and “SQuAD” means using LC-QuAD and SQuAD for finetuning, respectively. “BLI” and “xMRC” means using BLI translation and xMRC datasets for finetuning, respectively. “Sing.” means using the data in the target language only while “All” means combining the data in all the languages. We do not find available xMRC datasets for Persian (fa), so the performance of “+ Sing. xMRC” on Persian is the same as that of “SQuAD”.

formance on the original WebQSP. This is because the English-as-pivot baselines are highly dependent on machine translation tools, whose outputs may contain unnatural expressions or even errors.

As for the closed-book QA baselines, mT5-large correctly outputs the answers in English for even 12.9% of the WebQSP-zh questions, without resorting to any external knowledge. This proves that MPLMs have learned a large amount of factual knowledge and strong cross-lingual capabilities, which can be properly utilized for xKBQA, as our method does.

All our models reach over 69% hits@1 on WebQSP-zh. Our two base-size models outperform EmbedKGQA by approximately 6% hits@1, an English-as-pivot baseline that utilizes RoBERTa-base and the KB embedding ComplEx (Trouillon et al., 2016). Our model with XLM-R-large outperforms all baselines, achieving 74.37% hits@1 thanks to the strong cross-lingual capability from MPLMs and rich resources in xMRC. Moreover, these results demonstrate another merit of our approach that it can directly answer non-English questions over KBs in English, reducing the reliance on machine translation systems. Although NSM-student, which does not use PLMs itself, performs better than our two base-size models, the parameters and computational complexity introduced by the translation system are much heavier than the MPLM used in our method. Furthermore, our approach demonstrates its advantage with fewer or even no training data, as we will discuss next.

5.2 Few-Shot and Zero-Shot Settings

Consider the high cost of annotating high-quality xKBQA data, we investigate the capabilities of our method under few-shot and zero-shot settings.

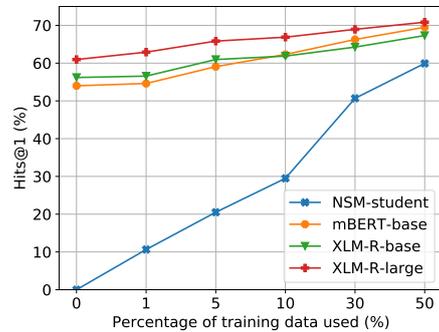


Figure 4: Few-shot and zero-shot performance of our method and NSM-student on the test set of WebQSP-zh.

Figure 4 shows the performance of our method and NSM-student on WebQSP-zh under few-shot and zero-shot settings. For NSM-student, its performance drops drastically with the decrease in training data. and it is totally incapable of zero-shot xKBQA. By contrast, when trained with half of the training data, our method still performs well, with less than 3% decrease in hits@1 compared with those trained with full data. With only 10% of the training data, i.e., 310 instances, our models reach over 62% hits@1, comparable with EmbedKGQA trained with full training data. Even under the zero-shot setting, our method can achieve 53-61% hits@1. The high performance of our method with limited training data is attributed to the KB-to-text conversion, which in turn makes it possible to benefit from the rich resources in xMRC. The MPLMs for xMRC have learned to encode different languages in the same semantic space during pre-training. After finetuning on xMRC datasets, the models can learn the ability to seek information from passages in a different language. By combining the prior knowledge of cross-lingual

mapping and reading comprehension abilities, our models can successfully answer a large portion of the xMRC-like questions converted from xKBQA.

To demonstrate that our method can generalize to different languages without specialized modifications, we test our approach on QALD-M in 11 typologically-diverse languages under the zero-shot setting. We evaluate the model on QALD-M after finetuning (1) on SQuAD only, (2) on SQuAD and xMRC datasets of a single language, and (3) on SQuAD and xMRC datasets of all the languages. As shown in Table 3, after finetuning XLM-R-large with SQuAD, our models achieve 43.9% hits@1 on average across 11 non-English languages, demonstrating our method’s strong generalization ability from English MRC datasets. After further finetuning on xMRC datasets for each language, we observe a 6.7% hits@1 boost in the average performance, showing the benefit of xMRC datasets in the absence of xKBQA data. If we combine the xMRC of all the languages for finetuning, the average hits@1 further increases slightly by 0.6%, probably due to the potential complementary effects between data in different languages. Compared with the semantic matching baseline finetuned with LC-QuAD and BLI-based translations, our best model outperforms it by 4.2% hits@1 on average. This is because the KB-to-text process of our method provides richer context than single inferential chains and the xMRC data are of higher quality than the BLI-based word-to-word translation.

5.3 Ablation Study

To evaluate the effectiveness of the designs in our approach, we conduct experiments in several ablated settings on WebQSP-zh with full xKBQA training data. We additionally conduct an ablation study with only 10% of the training to investigate what is behind the promising few-shot performance. The results are shown in Table 4.

With full training data, after we replace the PLM-based KB-to-text model with the simple heuristic of concatenating the head, predicate, and tail (w/o KB to text), the performance drops by 2.13% hits@1. Although the xMRC models can to some extent learn the mapping between questions and sentences converted by heuristics, the coherence and readability of KB-to-text generation results contribute to the final performance. Skipping the finetuning on either SQuAD (w/o SQuAD) or xMRC datasets (w/o xMRC data) leads to a performance drop,

Model	100%	10%
XLM-R-large (Ours)	74.37	67.60
- w/o KB to text	72.24 (-2.13)	65.58 (-2.02)
- w/o xMRC data	71.81 (-2.56)	65.53 (-2.07)
- w/o SQuAD	71.02 (-3.35)	65.10 (-2.50)
- w/o xMRC data, SQuAD	66.69 (-7.68)	54.79 (-12.81)

Table 4: Ablation study of our method with XLM-R-large on WebQSP-zh, using 100% or 10% of the training data (Hits@1 in percent).

showing the importance of high-quality data augmentation in absence of large-scale xKBQA data.

In the setting with 10% of the training data, both KB-to-text generation and finetuning on the MRC data contribute to the high few-shot performance, similar to the full training data setting. We observe a drastic drop of 12.81% hits@1 if the model is not finetuned on any MRC data (w/o xMRC data, SQuAD). This indicates that MRC data, no matter monolingual or cross-lingual, can greatly relieve the problem of data scarcity in xKBQA.

5.4 Error Analysis

We sample 50 error cases in WebQSP-zh and analyze their sources of error, as shown in Table 5.

34% of the errors result from the annotation of the original WebQSP dataset, where the annotated answer sets may be incomplete or incorrect. Another common source of error is the MRC model, which incorrectly answers 34% of the sampled questions. Among them, many are complex questions involving constraints or multiple relations. In the future, multi-hop MRC models can be adopted for addressing them. Besides, there are also several error cases resulting from KB-to-text generation and sentence filtering. We believe that our model will achieve better performance if each module in our framework is carefully optimized for the datasets.

5.5 Effect of Entity Linking

Entity linking (EL) is a crucial issue in KBQA, which requires linking the entity mentions in the questions to the entities in a KB. It becomes even more difficult in the cross-lingual setting. In the experiments above, we use golden entity linking results following previous works. To further investigate the effect of entity linking in xKBQA, we conduct pilot experiments with two EL methods. One is surface-level matching after translating the questions, and the other is mGENRE (De Cao

Source	Example	Explanation	%
Answer Annotation	Question: 沃尔玛经营什么产业? / <i>What industry does Walmart operate in?</i> Passage: ... The industry of Walmart is <u>Retail-Store</u> , <u>Variety Stores</u> and <u>Department Stores</u> Answer: Variety Stores Prediction: Retail-Store	The annotated answers in the original WebQSP dataset are incomplete or incorrect. In the left case, the annotated answer set fails to include two correct answers, <i>Retail-Store</i> and <i>Department Stores</i> .	34
KB-to-text Generation	Question: 凯南·鲁兹在灯红酒绿杀人夜中扮演谁? / <i>Who does Kellan Lutz play in Prom Night?</i> Passage: ... Kellan Lutz, a character in the film “Prom Night”, played with <u>Rick Leland</u> Kellan Lutz, a character in <i>Twilight</i> , played the role of <u>Emmett Cullen</u> Answer: Rick Leland Prediction: Emmett Cullen	The KB-to-text model converts a KB schema to a wrong natural language expression or omits the entities in the given triple. In the left case, the model incorrectly converts the KB schema <i>character</i> to the expression <i>play with</i> .	12
Sentence Filtering	Question: 爱德华多·包洛奇在他的工作中使用了什么材料? / <i>What Materials did Eduardo Paolozzi use in his work?</i> Passage: ... The art forms of Eduardo Paolozzi are <u>Sculpture</u> Answer: Bronze Prediction: Sculpture	The answers are missing in the passages because the model for sentence similarity calculation incorrectly filters out the sentences containing answers. In the left case, the sentence containing the answer <i>Bronze</i> is mistakenly filtered out.	20
Reading Comprehension	Question: 谁是杰拉尔德福特的副总裁? / <i>Who was the vice president of Gerald Ford?</i> Passage: ... David Gergen was appointed as the White House Communications Director by President Gerald Ford The vice president of Gerald Ford was <u>Nelson Rockefeller</u> Answer: Nelson Rockefeller Prediction: Staff Dick Cheney	The xMRC model fails to select the correct answer span. In the left case, the xMRC model incorrectly maps the word 副总裁/ <i>vice president</i> to the expression <i>White House Communications Director</i> in the passage.	34

Table 5: Examples, explanations and percentages of different sources of error in the 50 sampled WebQSP-zh question that XLM-R-large fails to answer. The underlined spans in passages are answer candidates.

et al., 2022), a cross-lingual EL tool that does not rely on machine translation tools. On the test set of WebQSP-zh, two EL methods achieve 89.1% and 76.8% recall@5, respectively. With the results from two EL methods, our xMRC model with XLM-R-large achieves 65.9% and 56.5% hits@1, respectively. The large gap compared to the results with golden topic entities indicates that more future research on cross-lingual EL is desired.

6 Conclusion

In this paper, we propose to formulate xKBQA as answering questions by reading passages, benefiting from the recent advance in xMRC. By converting KB subgraphs into passages, we narrow the gap between KB schemas and natural questions under cross-lingual settings. The cross-lingual knowledge in MPLMs and the rich resources in xMRC alleviate the problem of data scarcity in xKBQA. To facilitate the evaluation of xKBQA, we collect WebQSP-zh, a new large-scale xKBQA dataset with more natural expressions. Extensive experiments on two datasets with 12 languages show

the strong performance of our method under both supervised and zero-shot settings.

We hope that our work will inspire more efforts into xKBQA. Several promising research directions under our framework include generating better passages for KB subgraphs, supporting more types of KBQA questions, and exploring better EL strategies for xKBQA.

Limitations

We discuss the limitations of our work from the following four aspects:

First, our work mainly focuses on single-relation questions and CVT questions in KBQA. We construct a new dataset WebQSP-zh based on WebQSP, which lacks complex questions with multiple constraints or relations. Since we use a vanilla BERT-based MRC model in our framework, it has a limited capacity for solving complex KBQA questions. As future work, multi-hop MRC models can be adopted to address complex questions in cross-lingual KBQA.

Second, our method is mainly designed for

entity-centric QA. It can handle well the answer types of KB entities or attribute values in KBQA. Yet its capability on other types of answers is currently unknown. We will consider extending our method with more diverse answer types in the future.

Third, the size of retrieved KB subgraphs is constrained by the maximum input length of PLMs. This could, to some extent, lower the answer coverage of the converted passages and hurt the overall performance. In the future, Longformer-based encoders or text summarization techniques could be explored to address this limitation.

Fourth, although using existing xMRC datasets can alleviate the data scarcity problem in xKBQA, it cannot fundamentally solve the problem of insufficient and expensive cross-lingual datasets. With more powerful cross-lingual PLMs, we may reduce the reliance on xMRC data. We will explore more strategies for tackling the data scarcity problem in future work.

Acknowledgments

This work is supported by NSFC (62161160339, 62206070). We would like to thank the anonymous reviewers for their valuable suggestions. Also, we would like to thank Xiao Liu and Quzhe Huang for their great help in this work. For any correspondence, please contact Yansong Feng.

References

- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. [On the cross-lingual transferability of monolingual representations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.
- Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. [Multilingual extractive reading comprehension by runtime machine translation](#). *ArXiv preprint*, abs/1809.03275.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.
- Ning Bian, Xianpei Han, Bo Chen, and Le Sun. 2021. Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12574–12582.
- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.
- Yubo Chen, Yunqi Zhang, Changran Hu, and Yongfeng Huang. 2021. [Jointly extracting explicit and implicit relational triples with reasoning pattern enhanced binary pointer network](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5694–5703, Online. Association for Computational Linguistics.
- Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. [TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages](#). *Transactions of the Association for Computational Linguistics*, 8:454–470.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2019. [Cross-lingual machine reading comprehension](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1586–1595, Hong Kong, China. Association for Computational Linguistics.
- Nicola De Cao, Ledell Wu, Kashyap Papat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. [Multilingual autoregressive entity linking](#). *Transactions of the Association for Computational Linguistics*, 10:274–290.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. [Creating training corpora for NLG micro-planners](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

- Yu Gu and Yu Su. 2022. [ArcaneQA: Dynamic program induction and contextualized encoding for knowledge base question answering](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1718–1731, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Sherzod Hakimov, Soufian Jebbara, and Philipp Cimiano. 2017. Amuse: multilingual semantic parsing for question answering over linked data. In *International Semantic Web Conference*, pages 329–346. Springer.
- Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 553–561.
- Guillaume Lample, Alexis Conneau, Marc’ Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.
- Yunshi Lan and Jing Jiang. 2020. [Query graph generation for answering multi-hop complex questions from knowledge bases](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974, Online. Association for Computational Linguistics.
- Kyungjae Lee, Sunghyun Park, Hojae Han, Jinyoung Yeo, Seung-won Hwang, and Juho Lee. 2019. [Learning with limited data for multilingual reading comprehension](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2840–2850, Hong Kong, China. Association for Computational Linguistics.
- Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. [MLQA: Evaluating cross-lingual extractive question answering](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.
- Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. [MKQA: A linguistically diverse benchmark for multilingual open domain question answering](#). *Transactions of the Association for Computational Linguistics*, 9:1389–1406.
- Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2022. [Open domain question answering with a unified knowledge interface](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1605–1620, Dublin, Ireland. Association for Computational Linguistics.
- Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. [Key-value memory networks for directly reading documents](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.
- Ngonga Ngomo. 2018. 9th challenge on question answering over linked data (qald-9). *language*.
- Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. [Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering](#). *ArXiv preprint*, abs/2012.14610.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.
- Arij Riabi, Thomas Scialom, Rachel Keraron, Benoît Sagot, Djamé Seddah, and Jacopo Staiano. 2021. [Synthetic data augmentation for zero-shot cross-lingual question answering](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7016–7030, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. [How much knowledge can you pack into the parameters of a language model?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.
- Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. [Improving multi-hop question answering over knowledge graphs using knowledge base embeddings](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics.
- Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. [TransferNet: An effective and transparent framework for multi-hop question answering over relation graph](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4149–4158, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

- Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8918–8927.
- Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. Association for Computational Linguistics.
- Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218. Springer.
- Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.
- Linjuan Wu, Shaojuan Wu, Xiaowang Zhang, Deyi Xiong, Shizhan Chen, Zhiqiang Zhuang, and Zhiyong Feng. 2022. Learning disentangled semantic representations for zero-shot cross-lingual transfer in multilingual machine reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 991–1000, Dublin, Ireland. Association for Computational Linguistics.
- Kun Xu, Yuxuan Lai, Yansong Feng, and Zhiguo Wang. 2019. Enhancing key-value memory neural networks for knowledge based question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2937–2947, Minneapolis, Minnesota. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Yi Yang and Ming-Wei Chang. 2015. S-MART: Novel tree-based structured learning algorithms applied to tweet entity linking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 504–513, Beijing, China. Association for Computational Linguistics.
- Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6032–6043, Dublin, Ireland. Association for Computational Linguistics.
- Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China. Association for Computational Linguistics.
- Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.
- Fei Yuan, Linjun Shou, Xuanyu Bai, Ming Gong, Yaobo Liang, Nan Duan, Yan Fu, and Daxin Jiang. 2020. Enhancing answer boundary detection for multilingual machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 925–934, Online. Association for Computational Linguistics.
- Yucheng Zhou, Xiubo Geng, Tao Shen, Wenqiang Zhang, and Daxin Jiang. 2021. Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5822–5834, Online. Association for Computational Linguistics.

A Dataset Details

A.1 QALD-M

Statistic The QALD-M dataset used in our paper is based on the version released by Zhou et al. (2021), composed of questions from QALD-M 4 to QALD-M 9 in 11 non-English languages. We filter the yes/no questions, counting questions, and the questions whose answers cannot be found in the knowledge base. The sizes of testing questions for each language are shown in Table 7, ranging from 66 to 363.

Knowledge Base For QALD-M, we use the 2016-10 version of DBPedia⁶. We discard the KB triples that are unlikely to contain answers such as page IDs and revision history, and only include information about article categories and object properties. For each question, we include in the subgraph the triples where the topic entity is the head entity or the tail entity, namely its one-hop neighbors.

A.2 WebQSP-zh

Statistics The WebQSP-zh dataset proposed in our paper consists of 4,737 question-answer pairs, of which 3,098 instances are for training and the remaining 1,639 instances are for testing. The average length of questions is 12.7 characters. The average number of answers per question is 9.8.

Knowledge Base For WebQSP-zh, we use a pre-processed version of Freebase⁷. Following previous works (Sun et al., 2018; Saxena et al., 2020), we further prune it to contain only those relations that are mentioned in the dataset. For each question, we obtain the neighborhood graph within two hops of topic entities.

Annotation Details We recruited the annotators from a Chinese campus BBS, who are proficient in both Chinese and English. They are instructed to translate the questions in WebQSP into Chinese and to pay attention to commonsense knowledge and natural vocabulary choice. They are paid 3 CNY for each question annotated, which is adequate given the participants’ demographic. The annotators are informed of how the data would be used.

⁶<http://downloads.dbpedia.org/wiki-archive/downloads-2016-10.html>

⁷<https://github.com/hugochan/BAMnet>

More Examples We provide more examples from WebQSP-zh in Table 6 to show that WebQSP-zh is a more realistic benchmark for cross-lingual evaluation, incorporated with commonsense knowledge and realistic vocabulary choices.

In the first example of Table 6, based on the knowledge that Aldi is a company, the word *originate* is translated as 创建/*found* instead of its literal translation 起源/*originate*. In the second example of Table 6, the annotator uses 范德堡大学/*Vanderbilt University* instead of 范德堡/*Vanderbilt* because native Chinese speakers often call Western universities by their full names and rarely drop the word 大学/*university*.

WebQSP-zh: 阿尔迪是什么时候创建的? / *When was Aldi founded?*

WebQSP-MT: 阿尔迪是什么时候起源的? / *When did Aldi originate?*

WebQSP: When did Aldi originate?

Freebase Predicate:

business.employer.employees
business.employment_tenure.from

WebQSP-zh: 范德堡大学的吉祥物是什么? / *What is Vanderbilt University’s mascot?*

WebQSP-MT: 范德堡的吉祥物是什么? / *What is Vanderbilt’s mascot?*

WebQSP: What is Vanderbilt’s Mascot?

Freebase Predicate:

education.educational_institution.mascot

Table 6: Examples from WebQSP-zh and their corresponding questions in WebQSP. WebQSP-MT is the Chinese translation of WebQSP by the machine translation tool Baidu Translate. The italic English texts are the literal meaning of the Chinese questions.

A.3 xMRC datasets

We use three xMRC datasets for data augmentation. Their preprocessing details and statistics are as follows.

In terms of MLQA and XQuAD, we directly use the officially released data with English passages paired with non-English questions. In terms of MKQA, the passages for reading comprehension are full-length English Wikipedia articles. Since the Wikipedia articles are too long for PLM-based xMRC models to handle, we use the annotated non-tabular long answers as passages, which are generally a few hundred words long.

For each language, we combine the data from different xMRC for finetuning. Specifically, we use MLQA for zh, de, es, hi; MKQA for zh, de, es, fr, it, nl, pt, pt_BR, ru; XQuAD for de, es, hi, ro,

Language	fa	de	ro	it	ru	fr	nl	es	hi_IN	pt	pt_BR
Size	334	363	344	363	171	363	346	346	262	171	66

Table 7: The sizes of QALD-M testing questions in 11 languages used in our paper.

Language	zh	de	ro	it	ru	fr	nl	es	hi_IN	pt	pt_BR
Size	5,641	8,904	1,190	2,685	3,875	2,685	2,685	9,628	6,615	2,685	2,685

Table 8: The number of questions in the combined xMRC datasets used in our paper.

ru. The statistics of the combined xMRC data are shown in Table 8.

B Implementation Details

B.1 KB-to-Text

We use JointGT (Chen et al., 2021) based on BART-base for KB-to-text generation. It is finetuned on WebNLG with the same hyperparameters in the original paper. In sentence filtering, we use the paraphrase-multilingual-mpnet-base-v2 model in SentenceTransformers for cross-lingual semantic similarity calculation.

B.2 xMRC

Our implementation of xMRC models is based on the Transformers⁸. For the finetuning on SQuAD, we set the batch size to 12, the learning rate to 3e-5, the number of training epochs to 2, the maximum input length to 384, and the document stride to 128. For the finetuning on xMRC datasets and the data converted from xKBQA, we use the same hyperparameters as the finetuning on SQuAD. The results are from single runs. We use an NVIDIA A40 GPU for experiments. An epoch on the data converted from xKBQA takes about 9 minutes.

C Licenses of Scientific Artifacts

The licenses for each dataset used are as follows: CC BY-SA 4.0 for SQuAD, Apache-2.0 License for MKQA, CC BY-SA 4.0 for XQuAD, CC-BY-SA 3.0 for MLQA, CC-BY 4.0 for WebQSP, GPL-3.0 License for LC-QuAD, and MIT License for QALD. The licenses for each model used are as follows: Apache-2.0 License for EmbedKGQA, BSD-2-Clause License for GraftNet, and Apache-2.0 License for Transformers. No license is provided by other models.

⁸<https://github.com/huggingface/transformers>

Delving Deeper into Cross-lingual Visual Question Answering

Chen Cecilia Liu¹, Jonas Pfeiffer¹, Anna Korhonen², Ivan Vulić², Iryna Gurevych¹

¹Ubiquitous Knowledge Processing Lab,

Department of Computer Science and Hessian Center for AI (hessian.AI),

Technical University of Darmstadt

²Language Technology Lab, University of Cambridge

www.ukp.tu-darmstadt.de

Abstract

Visual question answering (VQA) is one of the crucial vision-and-language tasks. Yet, existing VQA research has mostly focused on the English language, due to a lack of suitable evaluation resources. Previous work on cross-lingual VQA has reported poor zero-shot transfer performance of current multilingual multimodal Transformers with large gaps to monolingual performance, without any deeper analysis. In this work, we delve deeper into the different aspects of cross-lingual VQA, aiming to understand the impact of 1) modeling methods and choices, including architecture, inductive bias, fine-tuning; 2) learning biases: including question types and modality biases in cross-lingual setups. The key results of our analysis are: **1)** We show that simple modifications to the standard training setup can substantially reduce the transfer gap to monolingual English performance, yielding +10 accuracy points over existing methods. **2)** We analyze cross-lingual VQA across different question types of varying complexity for different multilingual multimodal Transformers, and identify question types that are the most difficult to improve on. **3)** We provide an analysis of modality biases present in training data and models, revealing why zero-shot performance gaps remain for certain question types and languages.

1 Introduction

The lack of multilingual resources has hindered the development and evaluation of Visual Question Answering (VQA) methods beyond the English language until recently. A rise in interest in creating multilingual Vision-and-Language (V&L) resources has inspired more research in this area (Srinivasan et al., 2021; Su et al., 2021; Liu et al., 2021a; Pfeiffer et al., 2022; Wang et al., 2022; Bugliarello et al., 2022, *inter alia*). Large Transformer-based models pretrained on images and text in *multiple* different languages have been proven as a viable vehicle for the development of

multilingual V&L task architectures through transfer learning, but such models are still few and far between (M3P, UC2; Ni et al., 2021; Zhou et al., 2021). Large decreases in task performance between monolingual and (zero-shot) cross-lingual transfer setups have been measured and reported, among other multilingual V&L tasks, in VQA (Pfeiffer et al., 2022). Yet, the reasons for such low results in this pivotal V&L task have not been investigated in depth.

In this work, we aim to shed new light on the cross-lingual performance gap of cross-lingual VQA models from multiple angles. To the best of our knowledge, we are the first to provide a comprehensive analysis of multilingual VQA, with a focus on cross-lingual transfer.

We first assess and discuss the impact of modeling methods and choices on the final cross-lingual VQA performance, aiming to mitigate the present performance gap. This includes experimenting with diverse prediction head architectures, incorporating inductive bias by extending input signals, as well as more sophisticated fine-tuning strategies. We analyze cross-lingual VQA across different question types of varying complexity for different multilingual multimodal Transformers, and in zero-shot and few-shot scenarios.

Next, we focus on the learning biases, where we investigate whether current multilingual multimodal models suffer from the so-called unimodal bias: that is, we probe if the models truly reason over both images and questions to solve the VQA task, or if they take unimodal ‘shortcuts’ instead, exploiting spurious correlations and artifacts of data creation. Our analysis allows us to identify the most difficult question types and reveals a shortcoming of the current evaluation scheme.

We find that standard approaches from text-only cross-lingual transfer scenarios (Pires et al., 2019; Hu et al., 2020) do not leverage the full multilingual capabilities of the pretrained models; we mea-

sure the considerably worse performance of ‘standard’ fine-tuning compared to a simple modified fine-tuning regime. Interestingly, we report a discrepancy between monolingual and cross-lingual performance in the modified fine-tuning regime: while they do not have any substantial impact on the model performance in the *source* language (English), they considerably improve *cross-lingual* VQA capabilities, achieving gains of more than 10 absolute accuracy points over the baselines.

Code is available at github.com/UKPLab/eacl2023-xlingvqa.

2 Preliminaries

The VQA task is typically framed as a classification problem with a large number of classes. For instance, in the VQA task on the standard English GQA dataset (Hudson and Manning, 2019), given a pair of an image and a question, a model needs to predict a correct answer from 1,853 possible classes. GQA consists of diverse structural and semantic patterns, in which the questions are visually grounded in the image. In multilingual and cross-lingual VQA, the goal is to make similar predictions, but the questions can be posed in different *target* languages (Pfeiffer et al., 2022): e.g., the VQA task on the multilingual xGQA dataset (Pfeiffer et al., 2022) relies on the same set of 1,853 classes as English GQA.

We base all our analyses and experiments on the xGQA dataset, which is, due to its size and language coverage, arguably the most comprehensive evaluation resource for cross-lingual VQA to date. It has also been included in the multimodal multilingual evaluation benchmark IGLUE (Bugliarello et al., 2022). xGQA is the multilingual extension of the English GQA dataset (Hudson and Manning, 2019) to 7 typologically diverse languages.

In this work, we use and empirically compare two state-of-the-art pretrained multimodal multilingual Transformer architectures: **M3P** (Ni et al., 2021) and **UC2** (Zhou et al., 2021).¹ The standard cross-lingual *zero-shot* transfer setup for VQA involves fine-tuning all the weights of the pretrained model on the downstream task data in the source language only. In the *few-shot* setup, after the source-language fine-tuning, the model is additionally optimized on a handful of task-annotated examples in the target language (Pfeiffer et al., 2022).

¹For technical details of the two models, we refer the reader to their respective papers.

3 Modeling Methods

Motivation. Recent work on VQA in cross-lingual settings (Pfeiffer et al., 2022; Bugliarello et al., 2022) benchmarked standard multimodal architectures in zero-shot and few-shot transfer scenarios on the xGQA dataset, without aiming to provide a deeper understanding of the particulars of the cross-lingual VQA task. At the same time, they report large gaps of cross-lingual transfer performance when compared to monolingual English performance, suggesting that there is ample room for improvement. In this work, we aim to leverage novel insights into different aspects of the cross-lingual VQA task (e.g., analyses over different question types or classification architectures) to guide improved cross-lingual VQA methods. In particular, we assess the impact of three orthogonal directions: **1)** classification architectures (§3.1); **2)** (richer) input signals (§3.2); **3)** fine-tuning strategies (§3.3).

3.1 Classification Architecture Variants

The original work on xGQA (Pfeiffer et al., 2022) evaluated only a simple ‘shallow’ linear classification head, termed **Linear** here: the output [CLS] token of the pretrained Transformer-based model (which has cross-attended over all text and image features) is simply passed into a linear classification head. However, we hypothesize that this choice might have a substantial impact on transfer performance. Therefore, in the so-called **Deep** variant, instead of a linear classification head, we add a 2-layer transformation network (f_{trans}) with the GELU activation function (Hendrycks and Gimpel, 2016), dropout and a layer-normalization layer, before feeding the representations into a linear layer for classification. The first layer of f_{trans} uses an orthogonal initializer (Saxe et al., 2014). Unless noted otherwise, all of our following experiments are based on this ‘deeper’ architecture; we illustrate the architecture in Figure 3 in Appendix C.

3.2 Incorporating Inductive Bias into the Input Signal

A large number of output classes (see §2) potentially amplifies the difficulty of zero-shot and few-shot cross-lingual transfer due to the need of aligning contextual representations in multiple languages for multi-class classifications. Standard VQA datasets such as GQA and xGQA contain questions of five different structural types (*Verify*,

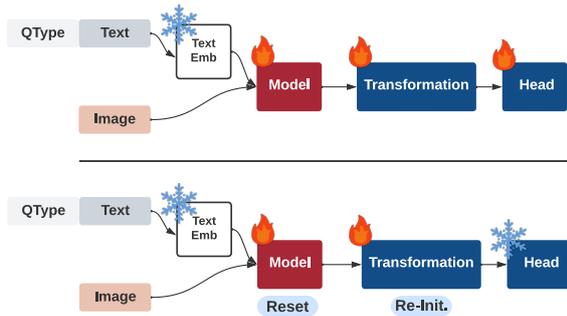


Figure 1: Self-Bootstrapping (§3.3). *Top*: Fine-tuning with frozen text embeddings (Stage 1). *Bottom*: Fine-tuning with text embeddings and classification head frozen. Other parameters are reset to their pretrained values or randomly initialized (Stage 2).

Logical, Query, Choose, Compare).² Pfeiffer et al. (2022) have demonstrated a considerable performance variation over different question types, e.g., there is a large cross-lingual performance drop especially for *Choose*-type questions.

To help alleviate this issue, we propose to feed the model with designated question-type tokens (Prager, 2006; Murdock et al., 2012) which appear in GQA and xGQA. The idea is to influence the label distribution for the VQA classification task by conditioning on a question-type token.

More concretely, we prepend a question-type token *QType* in English to the text input. We use structural question types as the question-type tokens; the text input then takes the following format: ‘[QType] : [Question]’.

As the xGQA data contains questions with binary answers (i.e. *Yes/No* questions). We anticipate that for a large fixed number of output classes, these questions should benefit the most from using the question-type tokens. The models which rely on this question-type conditioning are denoted with the superscript Q , e.g., M3P Q , see also later §3.4.

Recent work (Schick and Schütze, 2021; Li and Liang, 2021; Liu et al., 2021b; Shin et al., 2020) suggests that there exist more sophisticated prefixes/prompts and prompt-tuning methods. As our focus is not on conducting a large-scale analysis over different prompt-based conditioning, we leave this topic for future work.

²See Appendix B for example questions for each of the five question types.

3.3 Fine-Tuning Strategy

Misalignment of multilingual text embeddings (Søgaard et al., 2018; Dubossarsky et al., 2020) has been indicated by Pfeiffer et al. (2022) as one of the principal causes for reduced zero-shot performance in the cross-lingual VQA tasks. Therefore, we propose two fine-tuning strategies, tailored exactly towards mitigating such undesired shifts in the multilingual embedding space.

Freezing Text Embeddings. In the first variant, we freeze text embeddings during fine-tuning and only optimize the Transformer weights and the classification head. This should prevent misalignment of the text embedding space during fine-tuning, as only the alignments between image and text embeddings change, but not text-to-text alignments. This strategy, labeled **+FT**, is referred to as contrastive tuning by Zhai et al. (2022).

Self-Bootstrapping. Zero-shot cross-lingual transfer via standard fine-tuning is known to be sensitive to parameter initialization (Bugliarello et al., 2022). Previous work has shown that fine-tuning a classification head first, then fine-tuning the model can effectively improve the generalization of the model (Kumar et al., 2022; Yang et al., 2022). Motivated by these insights from prior research, we first train the network to learn the classification head, then reset and fine-tune the remaining model parameters. This leads to a two-stage fine-tuning process, termed *self-bootstrapping* (labeled **+SB**), illustrated in Figure 1 and outlined here:

Stage One: We fine-tune all parameters (with text embeddings frozen) on the task data.

Stage Two: We **1)** freeze the classification head (excluding the bias parameters) and text embeddings, **2)** reset the remaining parameters in the multimodal multilingual model to pretrained weights, and **3)** re-initialize the f_{trans} network (see §3.1). We then fine-tune the transformer weights on the task data.³

In order to make fair comparisons between +FT fine-tuning and self-bootstrapping, we define two extra +FT variants that match the fine-tuning budget of self-bootstrapping. In +FT_{short} we fine-tune until the budget of self-bootstrapping’s Stage 1 is matched. In +FT_{long}, we fine-tune until the total

³In our preliminary experiments, we found that self-bootstrapping-based fine-tuning still achieves better performance even if we perform Stage 1 with tunable text embeddings (i.e., standard fine-tuning). Freezing text embeddings in Stage 1 is an empirical decision, freezing them in Stage 2 is essential for self-bootstrapping to work.

training budget of self-bootstrapping is matched.

3.4 Model Configurations and Notation

Different choices across the orthogonal axes of classification architecture, input, and fine-tuning strategy give rise to a wide spectrum of *model configurations*. In particular, we can independently choose **1**) between the Linear or Deep classification architecture; **2**) whether to include the information on the question type at input (Q) or not; **3**) whether to apply standard fine-tuning from prior work (Pfeifer et al., 2022), or rely on +FT or +SB fine-tuning strategies. On top of this, we can also vary **4**) the underlying model (M3P or UC2), and **5**) the transfer scenario (zero-shot versus few-shot). For clarity of presentation, unless noted otherwise, we always assume zero-shot scenarios and Deep classification architecture. Moreover, different variants are also labeled in a systematic manner using abbreviations introduced in §3.1-§3.3: e.g., *M3P+SB* means that we apply self-bootstrapping on the underlying M3P model (with Deep architecture assumed). In another example, *UC2^Q+FT_{long}* means that we apply the *long* variant of +FT fine-tuning (see §3.3) with UC2 as the underlying model, and we condition the model on the information about question types.

4 Analysis Methods

The VQA task is inherently multimodal—a model is required to reason over both images and questions in textual form to solve the task. However, as with some unimodal text-only tasks (Gururangan et al., 2018; Poliak et al., 2018) VQA models might also be prone to ‘taking shortcuts’, that is, exploiting spurious correlations and artifacts of data creation. In other words, the VQA model could circumvent the multimodal aspect and only focus on a single modality to solve the task (Agrawal et al., 2016, 2018). Therefore, to better understand the multimodal reasoning abilities of VQA models in cross-lingual transfer, we propose several diagnostic approaches and methods that ablate the input features of the models, inspired by the diagnostic methods of Frank et al. (2021) and Shrestha et al. (2020) in monolingual setups. They should provide us with deeper insights into the inner workings of cross-lingual VQA models.

4.1 Unimodal Evaluation

The first set of analyses involves a combination of standard multimodal (MM) training with unimodal

inference/evaluation. During training, we pass both visual features and text tokens into the model. However, at inference, we provide the model with features of only one modality (Visual modality: **V** or Text: **T**). This naturally gives rise to the following two experimental setups:

MM-V: When evaluating on xGQA’s test set, we pass only a single ‘?’ as textual input to the model, while the standard visual features are used.

MM-T: At inference, we zero out all visual features (e.g., object features, spatial features), only providing the model with the total number of objects detected; the unchanged questions in the textual form are provided to the model.

4.2 Unimodal Training and Evaluation

Next, we probe purely unimodal models *trained* on a single modality (**V** or **T**): during training, the model is provided only with visual features or text tokens; at inference, we again only provide the model with unimodal features from the same modality. This creates three experimental setups:

V-V: We pass only ‘?’ as a (placeholder) textual input to the model, while the standard visual features (from the full multimodal model) are used.

T-T: All visual features are zeroed out; we only provide the number of objects detected; the unchanged questions in the textual form are provided.

T^G-T^G: We randomly sample object features from a Gaussian distribution with a mean and a standard deviation that match the actual object feature distribution for that image. Spatial features and the number of objects detected are kept as in the full MM model. The standard unchanged questions in the textual form are provided to the model.

5 Experimental Setup

Pretrained Models and Data. As introduced in §2, we 1) rely on two standard state-of-the-art multimodal multilingual transformers (M3P, UC2; Ni et al., 2021; Zhou et al., 2021) as the underlying pretrained models, and 2) conduct all evaluations on the standard monolingual English GQA dataset, and its multilingual extension: xGQA.

The GQA dataset consists of two training sets: **full** and **balanced**. The full dataset contains 113K images and 22M questions, whereas the balanced dataset consists of 1.7M data samples. The dataset also contains a balanced test-dev set with 12,578 questions and 398 images for evaluation. In xGQA, the questions are manually translated from the

Method	En	De	Zh	Ko	Id	Bn	Pt	Ru	Avg
G1 M3P* (Linear)	51.88±0.7	27.45±5.8	16.33±8.3	13.70±5.4	25.25±11.4	10.59±3.4	21.10±3.4	20.95±3.3	19.34
M3P*	51.66±0.6	35.33±5.4	27.80±10.9	25.55±11.4	30.54±9.8	17.94±8.6	30.61±7.2	29.74±6.6	28.22
G2 M3P ^Q	50.90±0.5	37.95±1.5	35.06±2.6	32.31±3.4	36.56±2.0	27.69±1.8	36.64±2.4	37.30±4.6	34.79
G3 M3P + SB	47.26±1.0	35.71±6.1	29.70±8.2	30.33±8.3	28.16±2.7	20.70±3.9	34.65±6.5	34.63±6.9	30.56
G4 M3P ^Q + FT _{short}	49.48±0.3	38.68±2.6	34.94±2.2	34.17±2.6	37.18±2.4	30.00±2.2	37.35±1.9	37.57±2.4	35.56
M3P ^Q + FT _{long}	51.00±0.9	38.42±2.1	35.05±2.1	33.38±2.5	36.24±2.3	27.77±1.7	36.78±2.3	37.42±2.0	35.01
M3P ^Q + SB	46.70±0.7	39.52±1.3	36.15±0.9	35.67±1.1	36.73±1.6	29.75±1.4	37.59±0.8	37.93±0.9	36.19
G1 UC2* (Linear)	57.83±0.3	40.57±1.7	35.54±3.4	16.95±6.1	34.18±0.8	8.53±1.9	24.90±3.7	24.05±4.6	26.39
UC2*	58.31±0.2	41.33±1.6	34.77±2.2	23.87±1.5	34.79±1.3	11.82±1.9	29.30±4.5	29.41±3.7	29.33
G2 UC2 ^Q	58.35±0.4	45.13±0.8	42.85±0.9	31.33±1.0	35.64±0.9	24.86±0.6	37.19±0.6	38.61±0.9	36.52
G3 UC2 + SB	58.52±0.4	48.51±1.3	43.97±0.3	35.08±2.0	37.33±3.2	19.09±4.5	35.29±2.9	35.99±3.5	36.46
G4 UC2 ^Q + FT _{short}	57.83±0.5	47.17±1.6	45.59±0.9	34.19±0.7	37.04±1.1	24.94±0.5	38.32±1.2	39.96±1.4	38.17
UC2 ^Q + FT _{long}	58.15±0.6	44.27±0.5	42.49±0.4	29.75±0.3	36.81±0.4	24.48±0.2	35.39±0.4	37.32±0.4	35.79
UC2 ^Q + SB	58.57±0.2	49.51±1.1	46.52±0.9	36.48±1.3	38.92±1.3	26.23±1.5	39.76±0.6	41.72±0.3	39.87

Table 1: Zero-shot transfer results on xGQA. Avg. refers to the average accuracy across languages excluding English. Group G1: baselines. *: our runs of baselines trained on balanced GQA. Group G2: results using a question-type token. Group G3: results using self-bootstrapping (+SB). Group G4: combining different fine-tuning strategy with the use of question-type tokens. Best results in each column and per each pretrained model across Groups G1-G4 are shown in **bold**. Results are averaged across four random seeds.

GQA test-dev set into 7 different languages: Bengali, Chinese (simplified), German, Indonesian, Korean, Portuguese, and Russian. xGQA provides a zero-shot evaluation set and a different training/evaluation set for the few-shot setting. Please see the original paper for details.

Training Details and Hyperparameters. Following the recommendations from Bugliarello et al. (2022), we predominantly run training on the more lightweight *balanced* subset of GQA.⁴ We also define a total training budget of 6 epochs (less than 24 hours of training). For the self-bootstrapping procedure, this means the total training time (Stage 1 + Stage 2) is equal to 6 epochs. See Appendix A for further details.

6 Results and Discussion

In §6.1, we discuss the results of the different modeling approaches across the three dimensions (see

⁴Another established yet less efficient training procedure is to train on the full GQA dataset first, then further train on the balanced dataset (Li et al., 2020). This procedure can produce good results on the English evaluation dataset at the cost of a substantial increase in computation demands (~4 days on one NVIDIA V100 for one model). Furthermore, our initial experiments have indicated that training with the balanced set performs similarly to the previously reported baselines in the xGQA paper while using substantially less computing. We stress that we also further run experiments under the more demanding training regime (Li et al., 2020) with the best-performing model configuration from our experiments. For more details, we refer the reader to §7.

§3): classification architectures, input signals and fine-tuning strategies. A finer-grained analysis concerning different structural question types is provided in §6.1. Finally, in §6.2 we delve deeper into the VQA models’ susceptibility towards exploiting unimodal biases and artifacts of the VQA datasets, relying on model variants discussed in §4.

6.1 Impact of Modeling Methods

A summary of the results with a wide spectrum of possible model configurations (see §3.4) is provided in Table 1, with accuracy as the main metric.

First, an interesting trend emerges: different model configurations have *no* significant effect on performance in the source language (English), especially so for the better-performing pretrained model UC2. However, variations in different modeling choices from §3 do show *considerable* impact on cross-lingual transfer performance: we report gains by more than 16 and 13 absolute accuracy points for M3P and UC2, respectively.

Classification Architectures. Surprisingly, simply adding additional non-linear layers to the prediction head has a considerable impact on the cross-lingual transfer performance of the baseline models (especially for the M3P model) while performance in the source language stays nearly the same (Table 1, Group G1). Put simply, a deeper classification architecture seems to benefit cross-lingual

transfer performance, and the extent of its impact cannot be captured by monolingual English-only evaluation. Further, to isolate the source of these improvements, we conducted additional experiments by removing the layer normalization component from the deeper architecture. The results are provided in Table 8 in the Appendix. Another key observation is that the impact of depth is model-dependent with stronger configurations. While it yields large gains when we start from the baseline transfer models (G1), the gains from the classification architecture are less pronounced or even non-existent, e.g., for the best-performing UC2^Q + SB model variant (see Group G4): 39.87 (Deep) versus 40.89 (Linear). The gains from classification architecture remained for the M3P model variant: 36.19 (Deep) versus 18.24 (Linear).

Input Signal. The large number of output classes of GQA potentially results in a noisy distribution over the predicted labels when sentences in a different language are passed into the model. We find that including the question-type token (Q) improves the average cross-lingual zero-shot transfer accuracy by more than 10% relatively for both M3P^Q and UC2^Q (Table 1, Group G2). This modeling decision again has an inconsequential impact on the source language but suggests that the question-type token can partially mitigate the poor performance of cross-lingual transfer. A comparison of G3 versus G4 models in Table 1 demonstrates that including the question type at input yields gains of almost 6 accuracy points with M3P, and more than 3 points with UC2, with especially large gains for Bengali as the lowest-performing language.

Fine-tuning Strategy. Freezing the embeddings to mitigate a shift in the multilingual embedding space results in positive gains for cross-lingual scenarios (Table 1, Group G4). The self-bootstrapping strategy (+SB with and without Q) achieves further gains over both +FT embedding-freezing experimental setups. At the same time, it also yields much lower variance across languages (with Q). This validates that resetting parameters with self-bootstrapping positively impacts model performance, and supports our hypothesis that first fixing the classifier weights to good values leads to better performance and lower variance. Note that the average +SB results of UC2 are statistically significant against UC2^Q and UC2^Q + FT_{short} ($p < 0.05$).

Performance across Question Types. Finer-

grained results per individual question type are summarized in Figure 2, where we compare the baseline models with the best-performing variant, which utilizes the question-type at the input and the self-bootstrapping strategy. In sum, we observe gains across all structural question-types for such ^Q+SB model configurations, both for M3P and UC2. Performance on *Query* and *Choose* questions meets substantial gains, suggesting that improving the alignment between multilingual text embeddings has a positive effect on performance, especially for non-binary, free-form question-types. Complete results are available in Appendix B).

6.2 Learning Biases: Multi-Modal versus Unimodal VQA?

We use the analysis methods from §4 to determine whether the underlying models have learned to rely on a single modality to make predictions, either due to spurious correlations in the data or the model’s inability to effectively combine multi-modal features. The main results are provided in Table 2.

Unimodal Evaluation. The scores of MM-T/MM-V ablations reveal the sensitivity to missing features in each input modality at test time. We observe a drop in accuracy of more than 50% across all question types in the MM-T/MM-V experiments compared to their counterparts that assume ‘full-feature’ multi-modal input at inference. Moreover, *Verify*, *Logical* and *Compare* questions seem more dependent on text features. The results confirm that the trained model needs both modalities to achieve good cross-lingual performance, although not at equal proportions. In other words, high zero-shot transfer performance observed in our experiments are obtained by leveraging both modalities in synergy, and not by ‘taking unimodal shortcuts’ (§4).

Unimodal Training and Evaluation. V-V/T-T/T^G-T^G experiments reveal the worst-case exploitation of the data biases in modalities by the models. The results suggest that a majority of the final performance can be attained with text features in fine-tuned models for the *Logical*, *Verify*, and *Compare* question types. Therefore, the results indicate that these question types contain modality biases that can be exploited by unimodal VQA architectures. The exploitable data biases could also explain the observations from prior experiments. We suspect this could also explain the asymmetrical attention over modalities, observed by Frank et al. (2021) in monolingual multi-modal models.

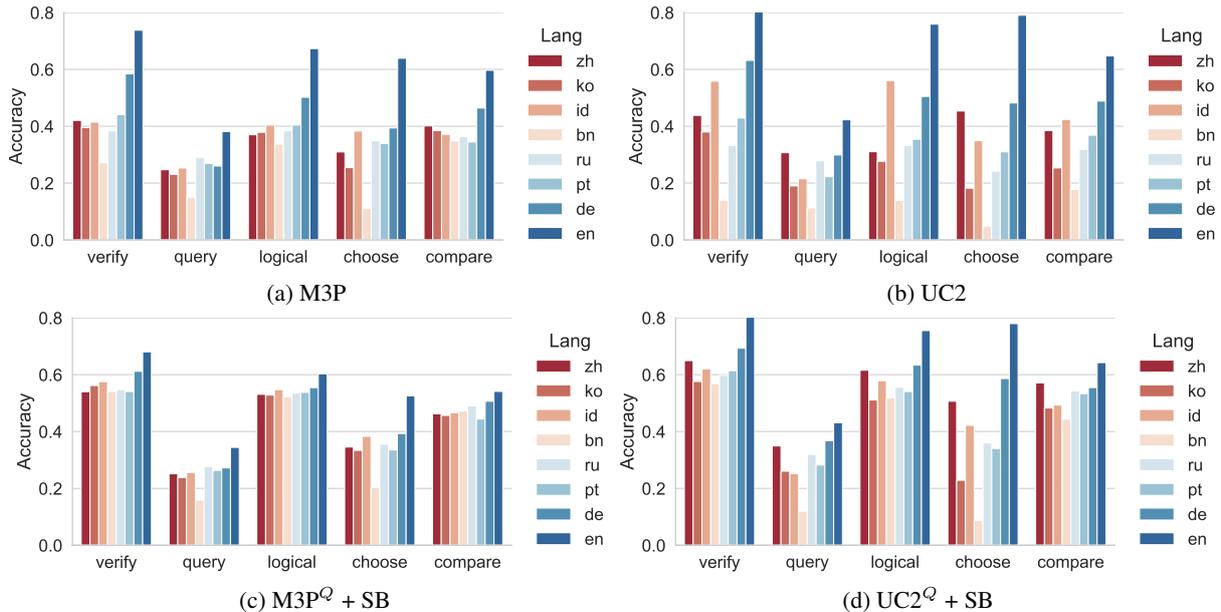


Figure 2: Zero-shot cross-lingual transfer performance across individual question types in GQA and xGQA.

Biases across Question Types.. Unimodally trained models can only attain $\sim 20\%$ (M3P) and $\sim 26\%$ (UC2) accuracy at best for the *Query* question type, with similar trends observed for *Choose*. Exposing the models to increasingly more visual features (from T-T over T^G-T^G to the full multi-model) yields significant performance gains. It thus indicates that *Query* and *Choose* questions contain fewer exploitable data biases, and additional image-text grounding could help improve predictions. Previous work in monolingual settings (Keravade et al., 2021) concludes that *Compare* and *Query* questions should be focused on for future improvements. Here, in the cross-lingual setting, we found *Query* and *Choose* questions as the most difficult questions with the largest gaps in monolingual English performance.

Table 2 also reveals that more sophisticated fine-tuning strategies such as self-bootstrapping, which prevent multilingual text embedding shifts, are an effective way to improve performance on these two (most challenging) question types.

In summary, it is crucial to conduct such finer-grained analyses across different question types in the multilingual VQA tasks, and not treat them equally with only a global accuracy metric. In particular, our results render *Query* and *Choose* question types as by far the most challenging question types for cross-lingual transfer and the types that do not suffer from exploitable data biases. Future research in multilingual VQA should put more

M3P	V-V	T-T	T^G-T^G	M3P ^Q	M3P ^Q +SB	MM-V	MM-T
Verify	45.19	53.98	54.88	58.35	55.98	0.1	18.59
Logical	43.18	51.66	53.06	53.89	53.65	0.0	19.87
Compare	27.76	46.22	39.64	45.82	47.14	0.1	17.85
Query	2.63	4.39	11.42	21.86	24.50	6.81	4.46
Choose	1.21	8.52	22.26	29.43	33.57	2.08	12.22

UC2	V-V	T-T	T^G-T^G	UC2 ^Q	UC2 ^Q +SB	MM-V	MM-T
Verify	44.60	51.87	57.00	59.94	61.70	4.21	24.91
Logical	44.26	50.78	52.57	54.87	56.49	6.27	21.12
Compare	33.45	40.55	46.91	49.15	51.73	2.85	21.08
Query	3.39	6.23	12.11	23.94	27.88	7.30	0.02
Choose	1.39	17.24	23.76	29.66	36.14	2.27	0.14

Table 2: Zero-shot transfer results of M3P^Q/UC2^Q trained and tested with visual features only (V-V), text features only (T-T), text features with partial visual features (T^G-T^G), as well as of M3P^Q+SB/UC2^Q+SB trained using all features, but exposed only to visual features (MM-V) or text features (MM-T) at inference (§4). The scores are averaged over all target languages in xGQA, excluding English.

emphasis on such questions, and approaches that prevent the exploitation of unimodal data biases. Future research should also look beyond the question types currently covered by xGQA, and introduce even more challenging types.

7 Additional Results

Training with Full English GQA. To validate the effectiveness of our approach in setups where more data in the source language is available, we additionally run experiments in another VQA setup: we train the best-performing method UC2^Q+SB for 5

Method	0	1	5	48
M3P	35.58	37.62	39.29	42.28
M3P + SB	33.73	35.89	39.27	42.46
M3P ^Q	33.81	35.40	37.80	41.87
M3P ^Q + SB	37.14	37.50	38.16	40.00
UC2	30.15	36.09	38.67	44.37
UC2 + SB	38.09	40.51	42.14	46.68
UC2 ^Q	37.28	39.24	40.88	45.11
UC2 ^Q + SB	39.83	42.35	43.68	46.62

Table 3: Averaged few-shot (0/1/5/48-shot) accuracy scores on xGQA (excl. English) for selected models.

epochs on the unbalanced English GQA dataset, followed by 2 epochs on the balanced dataset. Despite the fact that this variant leverages more source-language training data and consumes considerably more compute, we do not observe any gain on monolingual English performance, and observe only a small gain in the cross-lingual zero-shot setup: the accuracy score, averaged across all the target languages, increases from 39.87 to 40.51.⁵

Few-shot Experiments. Besides the zero-shot transfer scenario—which is the primary focus of this work—we also evaluate whether similar findings extend to few-shot scenarios, where a handful of annotated examples in the target language is assumed. Following the standard setup of Lauscher et al. (2020) we start from the weights of the best-performing model, already fine-tuned on English VQA data. We then further fine-tune it on the few examples in the target language. In particular, we conduct few-shot experiments with 1, 5, and 48 images.⁶ Following Pfeiffer et al. (2022) we fine-tune for 10 epochs, with a learning rate of 5e-5.

The results are summarized in Table 3 (see Table 10 in Appendix F for full results), and indicate two key findings. First, we corroborate findings from prior work, where it was shown that fine-tuning on an increasing number of shots/examples in the target language generally improves model performance. Second, although baseline models are able to recover more performance from zero-shot to few-shot setups, our best-performing configuration with UC2 still significantly outperforms the baseline. We attribute the on-par performance across M3P variants to M3P’s sensitivity to initialization and high variance. These results indicate that few-shot fine-tuning is an *additional*

⁵Table 9 in Appendix E provides per-language accuracy.

⁶We choose 1 and 5 shots because these are typical in few-shot training setups (Zhao et al., 2021). 48 shots are the maximum available training data for the few-shot evaluation.

cost-efficient approach, orthogonal to our modeling enhancements from §3, to further improve VQA model performance in the target language.

8 Related Work

Transformer-based models trained on multimodal data (Tan and Bansal, 2019; Li et al., 2020; Cho et al., 2021; Shen et al., 2022; Kamath et al., 2021, *inter alia*) have demonstrated impressive results on English-only VQA tasks. However, as training and evaluation data has previously only been available in high resource languages (Elliott et al., 2016, 2017; Barrault et al., 2018; Gao et al., 2015), progress in multilingual vision-and-language learning has not kept pace.

More comprehensive multilingual multimodal benchmarks have been developed only recently (Srinivasan et al., 2021; Su et al., 2021; Liu et al., 2021a; Pfeiffer et al., 2022; Wang et al., 2022; Bugliarello et al., 2022, *inter alia*) making it possible to evaluate multimodal models which have either been pretrained on multilingual data (Ni et al., 2021; Zhou et al., 2021) or extended to unseen languages (Liu et al., 2021a; Pfeiffer et al., 2022).

Our work complements this recent line of work by delving deeper into cross-lingual visual question answering, again highlighting the inherent difficulty of multilingual multimodal learning.

9 Conclusion

In this work, we provide an extensive analysis of the issues present in VQA-related multilingual vision-and-language learning, aiming to inspire new solutions that can improve cross-lingual VQA performance. To this end, we studied simple yet effective methods that increase previously low transfer performance and thus substantially reduce the gap to monolingual English performance. This has been achieved through more sophisticated classification architectures, fine-tuning strategies, and introducing inductive biases to input via question-type conditioning. We also conducted further analyses and empirical comparisons, including detection of unimodal biases in training and evaluation data, fine-grained analyses across different question types, and comparisons across different multilingual Transformer models and transfer scenarios. We hope that this work will spark more interest and inspire future research on cross-lingual VQA tasks in particular, as well as on multilingual multimodal learning in general.

10 Limitations

Our study focuses on the cross-lingual VQA task relying on the xGQA dataset only. xGQA contains seven typologically different languages and many low-resource languages are not included. We aim to extend this study to other low-resource languages in the future, and to other datasets that were made publicly available after the completion of this study (Changpinyo et al., 2022).

In one part of our study, we assume gold question-type information is available during training and testing. This assumption is made for analysis purposes, in practice, one could train a classifier for question-type classification first.

The proposed self-bootstrapping method requires the ability to divide training into stages and reset weights during training.

We have averaged our results over four runs. From our experiments, we noticed that both of the underlying multilingual multimodal Transformers produced high variance results. We plan to investigate the causes of the variance in detail as part of future research. Currently, we relied on one established few-shot learning paradigm, recently Schmidt et al. (2022) shows that combining English and target-language data might yield more robust transferring results, which we plan to investigate into future.

Acknowledgements

■ The Ubiquitous Knowledge Processing Lab acknowledges the financial support of the German Federal Ministry of Education and Research (BMBF) under the promotional reference 13N15897 (MISRIK), and the LOEWE initiative (Hesse, Germany) within the emergenCITY center. The work of Ivan Vulić and Anna Korhonen has been supported by the ERC PoC Grant Multi-ConvAI: Enabling Multilingual Conversational AI (no. 957356) and a research donation from Huawei. The work of Ivan Vulić was also supported in part by a personal Royal Society University Research Fellowship (no 221137; 2022-).

We thank Gregor Geigle for insightful feedback and suggestions on a draft of this paper.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. [Analyzing the behavior of visual question](#)

[answering models](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas. Association for Computational Linguistics.

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. [Don’t just assume; look and answer: Overcoming priors for visual question answering](#). In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4971–4980. Computer Vision Foundation / IEEE Computer Society.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. [Bottom-up and top-down attention for image captioning and visual question answering](#). In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. Computer Vision Foundation / IEEE Computer Society.

Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott, and Stella Frank. 2018. [Findings of the third shared task on multimodal machine translation](#). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 304–323, Belgium, Brussels. Association for Computational Linguistics.

Emanuele Bugliarello, Fangyu Liu, Jonas Pfeiffer, Siva Reddy, Desmond Elliott, Edoardo Maria Ponti, and Ivan Vulić. 2022. [IGLUE: A benchmark for transfer learning across modalities, tasks, and languages](#). In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2370–2392. PMLR.

Soravit Changpinyo, Linting Xue, Idan Szepes, Ashish V. Thapliyal, Julien Amelot, Xi Chen, and Radu Soricut. 2022. [Towards multi-lingual visual question answering](#). *CoRR*, abs/2209.05401.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. [Unifying vision-and-language tasks via text generation](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1931–1942. PMLR.

Haim Dubossarsky, Ivan Vulić, Roi Reichart, and Anna Korhonen. 2020. [The secret is in the spectra: Predicting cross-lingual task performance with spectral similarity measures](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2377–2390, Online. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. [Findings of the second shared task on multimodal machine translation and multilingual image description](#). In *Proceedings of the Second Conference on Machine Transla-*

- tion, pages 215–233, Copenhagen, Denmark. Association for Computational Linguistics.
- Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. **Multi30K: Multilingual English-German image descriptions**. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.
- Stella Frank, Emanuele Bugliarello, and Desmond Elliott. 2021. **Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9847–9857, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. 2015. **Are you talking to a machine? dataset and methods for multilingual image question answering**. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, page 2296–2304, Cambridge, MA, USA. MIT Press.
- Gregor Geigle, Jonas Pfeiffer, Nils Reimers, Ivan Vulić, and Iryna Gurevych. 2022. **Retrieve fast, rerank smart: Cooperative and joint approaches for improved cross-modal retrieval**. *Transactions of the Association for Computational Linguistics*, 10:503–521.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. **Annotation artifacts in natural language inference data**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Dan Hendrycks and Kevin Gimpel. 2016. **Gaussian error linear units (GELUs)**. *arXiv*, abs/1606.08415.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. **XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation**. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 4411–4421.
- Drew A. Hudson and Christopher D. Manning. 2019. **GQA: A new dataset for real-world visual reasoning and compositional question answering**. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6700–6709. Computer Vision Foundation / IEEE.
- Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. **MDETR - modulated detection for end-to-end multi-modal understanding**. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1780–1790.
- Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. 2021. **Roses are red, violets are blue... but should VQA expect them to?** In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2776–2785. Computer Vision Foundation / IEEE.
- Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. **Fine-tuning can distort pretrained features and underperform out-of-distribution**. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. **From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4483–4499, Online. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. **Prefix-Tuning: Optimizing continuous prompts for generation**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.
- Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. **Oscar: Object-semantics aligned pre-training for vision-language tasks**. In *European Conference on Computer Vision (ECCV) 2020 - 16th European Conference, Glasgow, UK, volume 12375*, pages 121–137. Springer.
- Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. 2021a. **Visually grounded reasoning across languages and cultures**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10467–10485, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. **GPT understands, too**. *arXiv*, abs/2103.10385.
- Francisco Massa and Ross Girshick. 2018. **maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch**. <https://github.com/facebookresearch/maskrcnn-benchmark>.

- J. William Murdock, Aditya Kalyanpur, Chris Welty, James Fan, David A. Ferrucci, David Gondek, Lei Zhang, and Hiroshi Kanayama. 2012. [Typing candidate answers using type coercion](#). *IBM J. Res. Dev.*, 56(3):7.
- Minheng Ni, Haoyang Huang, Lin Su, Edward Cui, Taroon Bharti, Lijuan Wang, Dongdong Zhang, and Nan Duan. 2021. [M3P: learning universal representations via multitask multilingual multimodal pre-training](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3977–3986. Computer Vision Foundation / IEEE.
- Jonas Pfeiffer, Gregor Geigle, Aishwarya Kamath, Jan-Martin O. Steitz, Stefan Roth, Ivan Vulić, and Iryna Gurevych. 2022. [xGQA: Cross-lingual visual question answering](#). In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. [How multilingual is multilingual BERT?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. [Hypothesis only baselines in natural language inference](#). In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics, *SEM@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 180–191. Association for Computational Linguistics.
- John M. Prager. 2006. [Open-domain question-answering](#). *Found. Trends Inf. Retr.*, 1(2):91–231.
- Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2014. [Exact solutions to the nonlinear dynamics of learning in deep linear neural networks](#). In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Timo Schick and Hinrich Schütze. 2021. [Exploiting cloze-questions for few-shot text classification and natural language inference](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.
- Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2022. [Don’t stop fine-tuning: On training regimes for few-shot cross-lingual transfer with multilingual language models](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10725–10742, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2022. [How much can CLIP benefit vision-and-language tasks?](#) In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual, April 25-29, 2022*. OpenReview.net.
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. [AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4222–4235, Online. Association for Computational Linguistics.
- Robik Shrestha, Kushal Kafle, and Christopher Kanan. 2020. [A negative case analysis of visual grounding methods for VQA](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8172–8181, Online. Association for Computational Linguistics.
- Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. [On the limitations of unsupervised bilingual dictionary induction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788, Melbourne, Australia. Association for Computational Linguistics.
- Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. [WIT: Wikipedia-Based Image Text Dataset for Multimodal Multilingual Machine Learning](#), page 2443–2449. Association for Computing Machinery, New York, NY, USA.
- Lin Su, Nan Duan, Edward Cui, Lei Ji, Chenfei Wu, Huaishao Luo, Yongfei Liu, Ming Zhong, Taroon Bharti, and Arun Sacheti. 2021. [GEM: A general evaluation benchmark for multimodal tasks](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2594–2603, Online. Association for Computational Linguistics.
- Hao Tan and Mohit Bansal. 2019. [LXMERT: Learning cross-modality encoder representations from transformers](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.
- Josiah Wang, Josiel Figueiredo, and Lucia Specia. 2022. [Multisubs: A large-scale multimodal and multilingual dataset](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 6776–6785. European Language Resources Association.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz,

- Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Zhuoyi Yang, Ming Ding, Yanhui Guo, Qingsong Lv, and Jie Tang. 2022. [Parameter-efficient tuning makes a good classification head](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. 2022. [LiT: Zero-shot transfer with locked-image text tuning](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18102–18112. IEEE.
- Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021. [A closer look at few-shot crosslingual transfer: The choice of shots matters](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5751–5767, Online. Association for Computational Linguistics.
- Mingyang Zhou, Luwei Zhou, Shuohang Wang, Yu Cheng, Linjie Li, Zhou Yu, and Jingjing Liu. 2021. [UC2: Universal cross-lingual cross-modal vision-and-language pre-training](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4155–4165. Computer Vision Foundation / IEEE.

A Details of Training Setup and Hyperparameters

The hyperparameters used to train M3P and UC2 models are summarized in Table 4. We conducted all experiments with either an NVIDIA V100 or A100 GPU. The numbers of training epochs across different model configurations are summarized in Table 5. Training time for the rest of our zero-shot experiments ranges from 8 to 24 hours.

We searched over the following learning rates: $2e-5$, $5e-5$, and $1e-4$.

We note that experiments which rely on Full GQA data (*-full*) have a significantly different training budget. This setup followed the previously recommended training setup of Li et al. (2020).

We use pretrained, state-of-the-art Transformer-based M3P and UC2 models (open-sourced), which build on pre-extracted image features from pretrained object detectors. M3P was pretrained via masked language modeling, cross-lingual masked language modeling and cross-modal text-image region alignments objectives. UC2 was trained similarly to M3P with an additional auxiliary task (i.e. translation).

We extracted image features for M3P using the ResNet-101 backbone using the vqa-maskrcnn-benchmark model (Massa and Girshick, 2018) (100 bounding boxes), and we extracted image features for UC2 using the bottom-up-attention (Anderson et al., 2018) (100 bounding boxes). The feature extraction procedures are different because the pretrained M3P and UC2 use different features. For experiments, we implemented everything in PyTorch, and we utilized Hugging Face Transformers (Wolf et al., 2020) and MMT-Retrieval (Geigle et al., 2022).

Name	Value
learning rate (M3P)	0.00002
learning rate (UC2)	0.0001
train batch size	192
warmup steps	0
weight decay	0.05
max grad norm	1
dropout rate	0.5
max seq length	70
max img seq length	50
f_{trans} hidden dim	768
optimizer	AdamW

Table 4: Hyperparameters.

Exp.	Balanced		Total Ep.	Time
	Stage 1	Stage 2		
M3P ^Q	6	-	6	<24hrs
M3P ^Q + FT _{short}	4	-	4	<24hrs
M3P ^Q + FT _{long}	6	-	6	<24hrs
M3P ^Q + SB	4	2	6	<24hrs
UC2 ^Q	6	-	6	<24hrs
UC2 ^Q + FT _{short}	3	-	3	<24hrs
UC2 ^Q + FT _{long}	6	-	6	<24hrs
UC2 ^Q + SB	3	3	6	<24hrs

Exp.	Full	Balanced	Total Ep.	Time
	Stage 1	Stage 2		
<i>-full</i>	5	2	7	4 days

Table 5: Training epochs and times. Full and Balanced indicate the GQA subset used for training. The self-bootstrapping experiments are initialized from the weights of *short* experiments.

Question Type	Count
Verify	2,251
Logical	1,803
Compare	5,89
Query	6,804
Choose	1,129

Table 6: GQA test-dev set: distribution of questions over question types.

B Structural Question Types in GQA and xGQA

There are 5 different structural question-types in GQA and, consequently, in xGQA. We used the exact lowercase name of each question type as the QType token in our experiments, namely: *verify*, *logical*, *compare*, *query*, and *choose*. The text input follows the format of: ‘[QType] : [Question]’ (see again §3.2). Some example questions for each question type are as follows:

Verify: Yes/No questions. E.g. *Do you see books near the device that looks gray? Is the bus blue?*

Logical: Questions that require logical inference. E.g. *Is there any motorcycle or ball in the scene? Does the dirt look brown and fine?*

Compare: Comparison questions between two or more objects. E.g. *Who seems to be younger, the boy or the woman?*

Query: Open questions. E.g. *What color are the pants? What is the animal that is standing on the grass called?*

Choose: Choose from two presented alternatives. E.g. *Is it red or blue? What size is the jacket, small*

Question Type	M3P	M3P + SB	M3P ^Q	M3P ^Q + SB
Verify	40.15	44.45	58.35	55.98
Logical	39.15	45.29	53.89	53.65
Compare	35.95	40.75	45.82	47.14
Query	24.57	21.42	21.86	24.50
Choose	30.63	29.07	29.43	33.57

Question Type	UC2	UC2 + SB	UC2 ^Q	UC2 ^Q + SB
Verify	41.55	51.27	59.94	61.70
Logical	35.40	48.32	54.87	56.49
Compare	34.48	44.71	49.15	51.73
Query	23.18	27.68	23.94	27.88
Choose	29.51	36.62	29.66	36.14

Table 7: Average accuracy on different structural question types from xGQA (excluding English). M3P and UC2 are using Deep architecture.

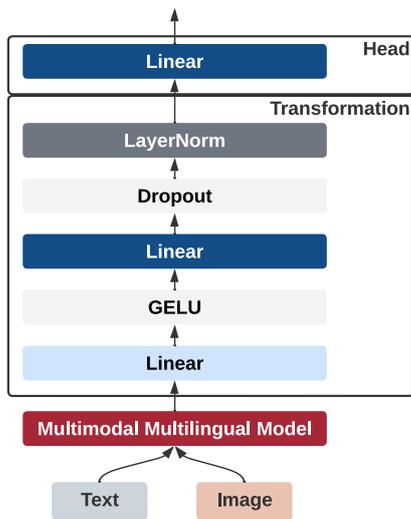


Figure 3: The deep(er) classification architecture (see §3.1). The first linear layer in the transformation uses an orthogonal initializer.

or large?

Verify and Logical question types are binary question types (Yes/No). The question type distribution in the test-dev set of GQA is given in Table 6, while we provide average accuracy scores overall target languages in xGQA (excluding English), with a representative set of models, in Table 7.

C Classification Architecture with and without Layer Normalization

The deeper variant of the classification architecture from §3.1 is illustrated in Figure 3. The *Multimodal Multilingual Model* block in Figure 3 denotes one of the two pretrained multimodal multilingual models used throughout the (main) paper: UC2 and M3P.

We further experimented with another variant of the architecture, where we removed the layer normalization (LayerNorm) layer. The results of this variant are available in Table 8.

In a nutshell, LayerNorm has more impact on M3P’s zero-shot transfer accuracy scores than on UC2. However, the variance of UC2 results increases with the removal of LayerNorm.

D Accuracy vs. Total Training Epochs

We conducted experiments with different total numbers of training epochs with M3P to understand the effect of the self-bootstrapping fine-tuning strategy. We experimented with the following three model configurations across different setups:

1. M3P^Q + FT: We train the M3P^Q model with text embeddings frozen for 4, 6 and 10 epochs.
2. M3P^{Q*} + FT: We initialize the M3P^Q model with fine-tuned weights (including transformation, classification head) from 1 (i.e., the variant above), and train for 4 epochs. We continue to fine-tune the model for 2 or 5 more epochs after resetting the learning rate and the optimizer.
3. M3P^Q + SB: We train the M3P^Q model with self-bootstrapping and the classification head weights from variant 1 above and do it for 4 epochs. We continue to fine-tune the model for 2 or 5 epochs.

We also run similar variants with UC2 as the underlying model with shorter training epochs. These variants are UC2^Q + FT / UC2^{Q*} + FT / UC2^Q + SB where superscripts and acronyms remain the same as the M3P variants. Results of these experiments are provided in Figure 4a (M3P) and Figure 4b (UC2).

We observe that the gains in cross-lingual transfer with +FT variants diminish or even start decreasing with the increase of training time. Similar results are observed when we reset the learning rate, weight decay and optimizer after training for 4 epochs. We also find that self-bootstrapping training continually improves the results, even with less additional total training epochs.

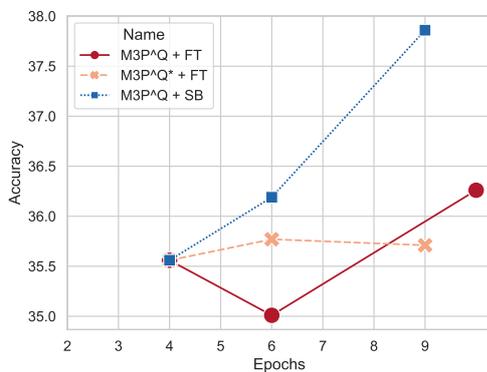
Moreover, the performance of self-bootstrapping is considerably more stable (lower variance) across random seeds, even though its classification heads are initialized from the corresponding trained weights from the M3P^Q + FT experiments.

Method	En	De	Zh	Ko	Id	Bn	Pt	Ru	Avg
M3P (Linear)	51.88 \pm 0.7	27.45 \pm 5.8	16.33 \pm 8.3	13.70 \pm 5.4	25.25 \pm 11.4	10.59 \pm 3.4	21.10 \pm 3.4	20.95 \pm 3.3	19.34
M3P w/ LN	51.66 \pm 0.6	35.33 \pm 5.4	27.80 \pm 10.9	25.55 \pm 11.4	30.54 \pm 9.8	17.94 \pm 8.6	30.61 \pm 7.2	29.74 \pm 6.6	28.22
M3P w/o LN	50.89 \pm 1.0	32.92 \pm 5.6	22.14 \pm 8.0	20.33 \pm 9.1	25.44 \pm 6.5	16.88 \pm 8.0	29.40 \pm 7.8	29.31 \pm 7.9	25.20
UC2 (Linear)	57.83 \pm 0.3	40.57 \pm 1.7	35.54 \pm 3.4	16.95 \pm 6.1	34.18 \pm 0.8	8.53 \pm 1.9	24.90 \pm 3.7	24.05 \pm 4.6	26.39
UC2 w/ LN	58.31 \pm 0.2	41.33 \pm 1.6	34.77 \pm 2.2	23.87 \pm 1.5	34.79 \pm 1.3	11.82 \pm 1.9	29.30 \pm 4.5	29.41 \pm 3.7	29.33
UC2 w/o LN	58.03 \pm 0.5	42.74 \pm 1.4	37.84 \pm 3.0	24.91 \pm 5.2	33.56 \pm 1.6	13.21 \pm 4.5	29.99 \pm 4.5	29.47 \pm 6.3	30.25

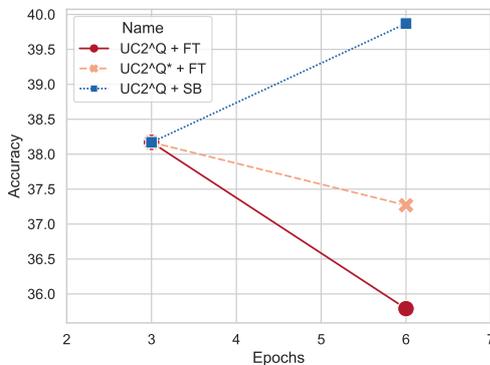
Table 8: Zero-shot cross-lingual transfer results with and without LayerNorm.

Method	En	De	Zh	Ko	Id	Bn	Pt	Ru	Avg
UC2 ^Q + SB - full	57.88 \pm 0.2	50.52 \pm 0.5	47.63 \pm 0.2	37.56 \pm 1.7	40.37 \pm 1.6	25.25 \pm 1.4	40.56 \pm 0.2	41.67 \pm 0.8	40.51

Table 9: Zero-shot results when the models are trained with Full GQA data.



(a) M3P



(b) UC2

Figure 4: Average accuracy versus total training epochs.

We also observe an increase in zero-shot transfer accuracy scores with more epochs of training in Stage 2 of self-bootstrapping. However, this results in much longer training times, which may not be realistic for academic and even some industry settings.

E Results with Full GQA Data

It is worth noting that the experiments trained with full GQA data (*-full*) have a significantly differ-

ent (and larger) training budget (see §7). We follow the previously recommended total training budget of Li et al. (2020) and combine it with our self-bootstrapping fine-tuning strategy. Table 9 shows the detailed results.

F Few-shot Experiments: Full Results

Table 10 shows the detailed results of our few-shot experiments, where the summary table is provided in the main paper: Table 3 in §7.

Lang	Method	0	1	5	48
de	M3P	39.45	40.76	41.88	44.20
	M3P + SB	39.25	39.72	40.98	43.08
	M3P ^Q	36.84	38.28	40.11	43.18
	M3P ^Q + SB	40.99	40.74	40.39	41.71
zh	M3P	35.76	37.65	40.28	42.18
	M3P + SB	32.96	35.55	38.24	41.15
	M3P ^Q	33.74	35.97	37.95	41.28
	M3P ^Q + SB	36.95	36.88	37.60	39.38
ko	M3P	34.53	36.58	36.79	39.61
	M3P + SB	36.04	36.92	37.31	39.41
	M3P ^Q	31.96	32.77	35.39	40.45
	M3P ^Q + SB	35.78	35.38	37.46	38.99
id	M3P	38.38	39.39	40.63	42.57
	M3P + SB	29.17	36.94	39.49	41.16
	M3P ^Q	34.69	35.37	38.50	42.12
	M3P ^Q + SB	37.75	36.25	38.57	39.92
bn	M3P	24.27	30.53	34.72	40.73
	M3P + SB	22.71	25.94	33.96	40.46
	M3P ^Q	27.67	29.95	33.15	40.36
	M3P ^Q + SB	30.50	31.77	34.08	39.24
pt	M3P	38.19	38.35	40.54	44.27
	M3P + SB	38.17	37.98	39.35	43.01
	M3P ^Q	36.87	37.93	39.72	43.08
	M3P ^Q + SB	38.56	39.24	39.71	40.56
ru	M3P	38.46	40.06	40.22	42.38
	M3P + SB	37.84	38.20	38.54	41.95
	M3P ^Q	34.86	37.51	39.82	42.64
	M3P ^Q + SB	38.76	39.74	39.39	40.19
de	UC2	40.39	44.23	46.03	49.51
	UC2 + SB	49.52	50.10	50.30	51.42
	UC2 ^Q	46.26	46.95	46.94	49.42
	UC2 ^Q + SB	50.23	50.70	50.53	51.39
zh	UC2	37.26	41.70	42.68	46.32
	UC2 + SB	43.54	46.30	47.17	48.80
	UC2 ^Q	43.89	44.90	45.56	47.24
	UC2 ^Q + SB	46.37	47.82	48.32	48.47
ko	UC2	25.93	32.63	36.11	41.11
	UC2 + SB	36.48	36.73	37.84	43.90
	UC2 ^Q	32.45	35.79	37.37	42.04
	UC2 ^Q + SB	37.80	39.05	40.68	43.38
id	UC2	35.76	39.35	40.12	44.24
	UC2 + SB	32.70	38.18	42.88	47.06
	UC2 ^Q	36.70	39.54	41.40	45.78
	UC2 ^Q + SB	38.34	42.16	42.33	47.01
bn	UC2	12.00	21.91	25.95	39.75
	UC2 + SB	24.66	29.76	32.31	42.08
	UC2 ^Q	25.29	27.68	32.75	39.82
	UC2 ^Q + SB	24.07	31.67	35.77	42.83
pt	UC2	29.79	33.86	40.18	45.23
	UC2 + SB	38.79	40.49	41.95	47.34
	UC2 ^Q	36.60	39.56	40.67	46.45
	UC2 ^Q + SB	40.36	42.65	43.79	47.63
ru	UC2	29.94	38.97	39.66	44.41
	UC2 + SB	40.93	42.02	42.54	46.15
	UC2 ^Q	39.76	40.26	41.46	45.04
	UC2 ^Q + SB	41.62	42.42	44.32	45.63

Table 10: Few-shot transfer average accuracy with different amounts of training data. M3P and UC2 are using the deeper classification architecture.

Bridging Argument Quality and Deliberative Quality Annotations with Adapters

Neele Falk and Gabriella Lapesa

Institute for Natural Language Processing, University of Stuttgart

{neele.falk,gabriella.lapesa}@ims.uni-stuttgart.de

Abstract

Assessing the quality of an argument is a complex, highly subjective task, influenced by heterogeneous factors (e.g., prior beliefs of the annotators, topic, domain, and application), and crucial for its impact in downstream tasks (e.g., argument retrieval or generation). Both the Argument Mining and the Social Science community have devoted plenty of attention to it, resulting in a wide variety of argument quality dimensions and a large number of annotated resources. This work aims at a better understanding of how the different aspects of argument quality relate to each other from a practical point of view. We employ adapter-fusion (Pfeiffer et al., 2021) as a multi-task learning framework which a) can improve the prediction of individual quality dimensions by injecting knowledge about related dimensions b) is efficient and modular and c) can serve as an analysis tool to investigate relations between different dimensions. We conduct experiments on 6 datasets and 20 quality dimensions. We find that the majority of the dimensions can be learned as a weighted combination of other quality aspects, and that for 8 dimensions adapter fusion improves quality prediction. Last, we show the benefits of this approach by improving the performance in an extrinsic, out-of-domain task: prediction of moderator interventions in a deliberative forum.

1 Introduction

Although people have been dealing with the art of persuasion since ancient times, there are many answers to the question of what constitutes a good argument or good argumentation, and none can be considered the best: the quality of arguments is complex, subjective and depends on the context in which the quality is assessed and on the prerequisites (attitudes and values) of the one who judges it. Despite the high complexity of this task, a considerable research effort has been done to automatically model argument quality in different contexts

(Wachsmuth et al., 2017a) due to its usefulness in downstream applications, such as automatic writing assistants (Wambsganss et al., 2020), argument extraction (Alshomary et al., 2021) and generation (Gurcke et al., 2021). Social Science offers another whole field of theories and definitions about argument quality in which the focus is usually not only on the argument itself but on the discussion between participants thus emphasizing the deliberative goal of the discourse (Gerber et al., 2018).

These two research communities, Argument Mining (AM) and Deliberative Theory (DT), have not only produced different theories of argument quality, but also a number of annotated datasets on the basis of which the models for the automatic assessment can be (or have been) trained. In both AM and DT, argument quality (AQ) and its Social Science counterpart, deliberative quality (DQ) are broken down into finer-grained dimensions. Such dimensions map, for example, whether an argument is logically constructed (micro-level), or constructive in the context of an overall discussion (macro-level). However, neither individual quality dimensions, nor an aggregated score can do justice to the complexity of this concept. Besides, a model that represents different aspects but has only been trained on one dataset will reproduce data-specific biases and may be less robust on other domains.

Multi-task learning (in this work, each quality dimension, e.g., logical cogency, clarity, persuasiveness, is a task), and the training data drawn from different datasets are a solution, as they allow to integrate the different dimensions and data sources from the two research communities. We propose to implement it using adapters (Houlsby et al., 2019), modules added between the layers of a transformer model. Differently from fine-tuning of the full model, adapters allow to use a minimal amount of parameters while still achieving good performance. Differently from standard multi-task learning, adapters do not require all the tasks to be

learnt simultaneously, but they can be learnt as specific modules that can also be combined (fused or stacked). The modular design of the adapters then allows for flexible composition of the individual quality aspects and thus can be used in various configurations; this property facilitates future research of argument quality in different domains and lends itself as tool for the investigation of the relationship between different quality dimensions, both within and across disciplines. We experiment with 6 datasets containing AQ and DQ annotations, for a total of 20 dimensions (AQ:8; DQ: 12) covering a wide range of logical, rhetorical and dialectical aspects and a variety of domains and topics. Our work proceeds in two steps.

In the first step, we employ adapter fusion to learn a target dimension as a weighted combination of single-task adapters (e.g., clarity as a combination of cogency and effectiveness). We improve the results with respect to single-task learning on 8 dimensions out of 20 in a low-resource scenario. Furthermore, fusion activation patterns provide us a tool to investigate the relationship between different quality dimensions.

In the second step, we employ quality adapters for a new task on a new dataset: predicting that a post in a deliberative forum needs moderation (Park et al., 2012). A fusion based on quality adapters is compared to baseline and full fine-tuning, outperforming both. Moreover, our analysis shows that in solving the task, the models exploit information from all major quality sub-categories. Crucially for the downstream application, casting moderation prediction as a fusion of quality adapters allows us to provide recommendation explained along specific quality dimensions (e.g., "this comment has major issues with the logical side of argumentation and it is disrespectful").

The contributions of this paper are at multiple level: a) at the level of task and methods, this is the first work which employs adapters for finer-grained AQ dimensions; b) at the conceptual/theoretical level, we make a first step in the integration of theories of AQ and DQ, bridging between the annotations produced by the two communities and proposing adapter activation as a tool to empirically compare the conceptual core shared by AQ and DQ dimensions c) at the level of application, we show that quality adapters can support the task of predicting moderation of user comments, additionally contributing a theory-based explanation layer.

2 Related Work

Argument Quality Much work on automatic modeling and annotation of argument quality (AQ) in the Argument Mining community focuses either on a specific aspect of quality (e.g. argument relevance (Wachsmuth et al., 2017b), sufficiency (Stab and Gurevych, 2017)) or a more general notion of argument quality based on human intuition (Habernal and Gurevych, 2016). Wachsmuth et al. (2017a) proposes a holistic taxonomy based on different theories of argument quality, inspired from rhetoric and linguistics, which divides AQ into three main sub-categories. The logical dimension measures whether an argument has premises and a valid conclusion (cogency) thus takes the content and structure of a single argument into account. The rhetorical dimension (effectiveness) measures the persuasiveness of the argument and takes into account *how* it is presented (style, emotional appeal). The dialectical dimension (reasonableness) plays a more important role in the context of a discourse and reflects whether an argument is valid towards a universal audience (e.g. whether the reasoning is based on values generally accepted by the society) or whether it is constructive in helping to resolve issues. Wachsmuth et al. (2017a) construct a corpus consisting of 302 arguments annotated with the three core and 15 sub-dimensions. Wachsmuth and Werner (2020) investigate which linguistic features are predictive of the fine-grained dimensions and which of the dimensions can be automatically assessed based on the textual input representations alone. The work by Fromm et al. (2022) are the first that try to combine AQ definitions from different corpora and based on different annotation schemas into one model. They investigate the generalizability of AQ when combining different sources and explore multi-task learning for assessing it in four different datasets. On top of that they investigate the relationship between AQ and other AM tasks such as evidence detection.

While most work on AQ in the Argument Mining community focuses on the logical dimension or specific aspects of persuasion, research on deliberative quality (DQ) from Social Science puts the discourse as a whole and the interaction between discourse participants into the focus. Here, argument quality (or discourse / deliberative quality) is investigated to find out which tools and solutions (e.g. moderation, platform design, structured overviews) can contribute to a more productive and

respectful public discourse. Thus, the annotated datasets from this domain complement the ones from the AM community providing many aspects of the rhetorical and dialectic dimensions.

Adapters Adapters (Houlsby et al., 2019) are a set of task-specific parameters that are introduced in every layer of a transformer (Vaswani et al., 2017) and updated for a specific task while the rest of the pre-trained language-model parameters is kept frozen. Besides being more efficient than full fine-tuning, adapters can be used as building blocks for other tasks due to their modular architecture and are therefore particularly well suited for transfer- and multi-task learning (He et al., 2021) and to inject external knowledge sources to solve downstream tasks (Lauscher et al., 2020a). Pfeiffer et al. (2021) propose to train task-specific adapters first (knowledge extraction) and combine them in a second step (knowledge composition) using self-attention to mitigate catastrophic interference, a problem which often occurs with traditional multi-task learning approaches. In their work, this approach has proven to be useful especially in low-resource settings which is often the case for complex annotations such as the AQ ones. To the best of our knowledge, our work is the first to employ adapters for AQ to conduct a systematic comparison of AQ and DQ on different data sources.

3 Datasets

For our experiments we rely on diversity, both in terms of data sets and different conceptualizations of argument quality. Therefore, we also integrate two datasets from the Social Sciences, which are not established in the argument quality community, but show a particularly large variety of dimensions. **Europolis** (Gerber et al., 2018): consists of transcriptions of a face-to-face discussion about the topic immigration, initiated by the European Union in order to enable deliberation on a European level. The spoken multi-lingual contributions have been transcribed, partially translated and annotated with five different dimensions of DQ by political scientists, each dimension between two to five labels that can be arranged on a scale from a low to a high standard of deliberative abilities. The dimensions capture the logical aspect (*justification*), rhetorical aspects (*storytelling*) and dialectic aspects (*common Good*, *interactivity* and *respect*).

THF/BK (Esau, 2022): this dataset contains comments from two online citizen dialogues on munic-

ipal issues: one on the further development of the “Tempelhofer Feld” site in Berlin and the other on the use of the former lignite area in North Rhine-Westphalia. The data was annotated by political scientists with different dimensions of DQ using a binary label for each dimension. The goal of the work was to investigate the relationship between “classic standards of deliberation”, such as rationality and constructiveness and alternative forms of deliberation, such as humor, narratives and the use of emotions. This dataset therefore offers annotations for the so far rather underexplored and more affective dimensions of argument quality, such as *positive emotions*, *narration* and *empathy*.

Kialo (Durmus et al., 2019): This dataset was created based on the online discussion platform Kialo <https://www.kialo.com> on which users engage in structured discussions about a certain statement. Users are able to rate the *impact* of an argument given its context. The dataset contains arguments about a large number of different topics together with their impact – a label which aggregates impact votes by all users. Durmus et al. (2019) and Li et al. (2020) report F-macro scores between 0.56 and 0.58 using different transformer-based models.

Grammarly Argument Quality Corpus (GAQ) (Ng et al., 2020): this dataset contains online contributions from four different domains annotated with the coarse-grained levels of the taxonomy introduced by Wachsmuth et al. (2017a) on a five-point scale. Lauscher et al. (2020b) evaluate different systems for automatic prediction of the quality scores, also experimenting with different multi-task architectures showing that multi-task learning can lead to improvements for all dimensions.

IBM-Rank-30k (Gretz et al., 2020): the largest available corpus with AQ annotations has been created based on a large quantity of binary annotations for human-generated arguments. The authors evaluate different methods of aggregating the annotations into a continuous score and conduct experiments on the automatic prediction of these scores with a Pearson correlation of around 0.48 on a test set with unseen topics. Lauscher et al. (2020b) found positive correlations between this aggregated AQ score and automatically generated scores for *cogency*, *effectiveness* and *reasonableness* on this corpus.

SwanRank (Swanson et al., 2015): as one of the first datasets with AQ annotations in the AM community this corpus contains arguments from on-

Dataset	size	genre	topics	mean length
SwanRank	5k	online discussion	gay marriage, gun control, death penalty, evolution	19
GAQ	5k	Debates, CQA, Reviews	diverse	109
IBM-Rank-30k	30k	crowd-sourced arguments	71 common controversial topics	18
Kialo	7k	argument maps	741 topics	23
Europolis	1k	face-to-face deliberation	immigration in Europe	131
THF/BK	1k	online deliberation	Redevelopment Tempelhofer Feld (THF) and lignite mining (BK)	124

Table 1: Overview of the datasets: original size, genre, topics and mean length in tokens of contributions

dimension	short description	measured	corpus
overall	general argument quality	score (1-5)	GAQ
cogency	acceptable and sufficient premises to draw a conclusion	score (1-5)	GAQ
reasonableness	contribution to resolution of issues, argument is accepted by universal audience	score (1-5)	GAQ
effectiveness	persuasion, rethorical, emotional appeal	score (1-5)	GAQ
quality	general argument quality	score (0-1)	IBM-Rank-30k
clarity	is it hard or easy to interpret the argument?	score (0-1)	Swanson
justification	rationality, providing reasons, reflection	multi-class (4)	Europolis
respect	empathy or respect towards groups (e.g. immigrants)	multi-class (3)	Europolis
storytelling	personal experience, subjective description of an event or situation	binary	Europolis
interactivity	respect towards other participants, reference to other participants arguments	multi-class (4)	Europolis
common good	taking interests of the broader community or utilitarianism based values (justice, equality) into account	multi-class (3)	Europolis
posEmotion	positive emotions are contained in the utterance	binary	THF/BK
proposal	a statement about what or how something is to be done	binary	THF/BK
narration	personal experience, subjective description of an event or situation	binary	THF/BK
reference	participant refers to another discourse participant	binary	THF/BK
argument	providing reasons and/or evidence in favor of or against a claim	binary	THF/BK
negEmotion	negative emotions are contained in the utterance	binary	THF/BK
empathy	Speaker puts himself in the perspective or emotional state of others	binary	THF/BK
Q(uestion) for justification	asks for the reasons for a statement or action	binary	THF/BK
impact	user likes / recommendations	multi-class (3)	Kialo

Table 2: Overview of the datasets with their respective argument quality dimensions

line discussion fora about four controversial topics. The corpus was annotated using crowd-sourcing on a continuous scale expressing whether an argument is easy or hard to interpret, thus reflecting the *clarity* of an argument. More recent experiments on this dataset are for example reported in Gretz et al. (2020) who experiment with fine-tuning transformer-based models after pre-training them on the IBM-Rank-30k dataset.

Table 2 shows an overview of the mentioned datasets and their corresponding quality dimensions, an example with the annotated label / score for each dimension can be found in Tables 13 to 16 in the appendix. Table 1 shows an overview of the six datasets with their respective size and number of topics. While the two datasets from Social Science offer the largest amount of different annotations they are also the smallest in size. On the other hand they consist of full discussions whereas the datasets from Argument Mining consist of single arguments without their broader context.

4 Experiment 1: Modeling AQ and DQ using adapters and adapter-fusion

In the following experiment we are interested in the relationship between different conceptualizations of AQ and DQ from a modeling perspective: does injecting knowledge about other dimensions help to improve the predictions on a target dimension? If so, which dimensions are especially helpful? To investigate this we treat each of the 20 dimensions as a task which we aim to model. We want to compare how the models perform without external information (using only a single-task adapter) with those using information about other dimensions (using multi-task learning with adapter-fusion).

4.1 Experimental setup

The input for all adapter models is the argumentative text, which consists of a sentence, a comment, or a spoken contribution, depending on the data set. We use RoBERTa (Liu et al., 2019) (`roberta-base`) as the backbone transformer model for all dimensions. Note that for each of the 20 single-task adapters we train a task-specific prediction head, depending on the underlying classifi-

cation problem (binary-, multi-class classification or regression). We pick the model with the best results on the validation set (lowest mean-squared error for the regression-, highest F1 macro for the classification tasks using class weights to counteract class imbalance).

Heuristic: how to select source tasks for adapter fusion? As the number of existing dimensions is large (20) we apply a heuristic to select different pools of source tasks for a target quality dimension. For this, we use predictions of dimension-specific adapters as proxies to uncover relationships between different quality aspects. We train an adapter for each dimension on the whole corresponding source dataset, generate predictions on all other datasets and measure pair-wise correlations across the datasets. We hypothesize that dimensions that have a clear positive or negative correlation to the target dimension will be most useful to support modeling that quality aspect, thus we add a dimension as source task if the absolute value of the correlation to the target dimension exceeds a threshold.

We sample source tasks from correlations between all 20 dimensions (*fusion corr ALL*), from correlations between dimensions from datasets with a focus on deliberation (Europol, THF/BK, Kialo: *fusion corr DQ*) and from those originating from established datasets from Argument Mining (GAQ, IBM, SwanRank: *fusion corr AQ*). We use the third quartile of the correlations of the respective dimensions as the threshold in each case (more details and correlation matrix in Appendix Section C and Figure 6). Appendix table 12 displays the output of the selection based on this heuristic. For most of the target dimensions, it indicates a fusion with between 2 and 9 source dimensions: more logical or general dimensions are more often selected (e.g. most frequent source dimension is *justification*, which gets selected for 14 target dimensions). A qualitative inspection of the suggested combinations shows that the heuristic is picking up sensible conceptual patterns. For example, for the target dimension *empathy* the candidates for fusion in the *fusion corr ALL* setup are *negEmotion*, *story* and *narration*.

For each setup we experiment whether we need to add the adapter of the target dimension as source task (*w own adapter*) as it has been done in Pfeiffer et al. (2021) or whether we can learn a target dimension as a weighted combination from

other source dimensions (*w/o own adapt.*) As the multi-tasking approach should be most helpful for low-resource scenarios, we down-sample the larger datasets (Kialo, IBM-Rank-30k, GAQ, SwanRank) to 1000 instances. We use the original train/val/test split for IBM-Rank-30k, GAQ and Kialo and create our own split for THF/BK, SwanRank and Europol. We train the fusions similar to the single-task adapters with a lower number of epochs and a smaller learning rate ($5e - 5$).¹ We train each model with 3 different seeds and report mean and standard deviation of F1 macro score and Pearson correlation in Table 11.

5 Results

Can we improve modeling AQ with adapter-fusion? Table 3 shows the results comparing single-task adapters with the fusion-based models, averaged over three seeds. We use the Almost Stochastic Order test (Del Barrio et al., 2018; Dror et al., 2019) as implemented by Ulmer et al. (2022) to identify for which dimensions multi-task learning can lead to significant improvements.²

Our results show that: a) Information about related quality dimensions can improve modeling for individual dimensions (significant improvements for 8 of 20 dimensions). These stem from 4 different datasets, so the trend holds across different datasets from both communities (AM and DT). b) For most dimensions the fusion does not lead to performance drops, which confirms the fact that adapter-fusion is more robust than traditional multi-task learning (no catastrophic forgetting / interference). The individual modules for different dimensions can thus be tried out without major disadvantages for new data sets or quality annotations. c) we gain improvements, even when the target adapter is not provided to the fusion (GAQ dimensions, *narration* and *argumentative*). Thus the target dimension can be learned as a weighted combination of source dimensions that are different. This can be especially useful when we only have little or noisy data for the target dimension available.

¹For implementation details refer to Appendix Section A.

²The test compares two score distributions by quantifying to which extend stochastic order is being violated. If the amount of violation is small enough, one model can be considered as superior (stochastically dominant) over the other.

dimension	ST	fusion corr ALL		fusion corr DQ		fusion corr AQ	
		w own adapt.	w/o own adapt.	w own adapt.	w/o own adapt.	w own adapt.	w/o own adapt.
overall	0.63 \pm 0.01	0.64 \pm 0.02	0.64 \pm 0.02			0.61 \pm 0.06	0.65\pm0.02*
cogency	0.41 \pm 0.10	0.47\pm0.02*	0.45 \pm 0.05			0.48\pm0.01*	0.49\pm0.01*
reasonableness	0.56 \pm 0.03	0.55 \pm 0.03	0.55 \pm 0.05			0.57 \pm 0.02	0.56 \pm 0.04
effectiveness	0.49 \pm 0.13	0.59\pm0.02**	0.57\pm0.02*			0.57\pm0.02*	0.58\pm0.01**
quality	0.38 \pm 0.16	0.48 \pm 0.05	0.43 \pm 0.04			0.45 \pm 0.06	0.43 \pm 0.07
clarity	0.64 \pm 0.01	0.63 \pm 0.03	0.63 \pm 0.01				
justification	0.46 \pm 0.04	0.45 \pm 0.03	0.45 \pm 0.02	0.46 \pm 0.03	0.46 \pm 0.02		
story	0.75 \pm 0.02	0.76 \pm 0.02	0.74 \pm 0.04	0.75 \pm 0.03	0.73 \pm 0.04		
interactivity	0.35 \pm 0.05			0.39\pm0.02*	0.36 \pm 0.04		
cgood	0.60 \pm 0.04			0.61 \pm 0.05	0.60 \pm 0.02		
posEmotion	0.64 \pm 0.03	0.63 \pm 0.03	0.61 \pm 0.03	0.64 \pm 0.03	0.60 \pm 0.01		
proposal	0.79 \pm 0.01	0.80 \pm 0.03	0.79 \pm 0.02	0.79 \pm 0.02	0.78 \pm 0.02		
narration	0.76 \pm 0.02	0.76 \pm 0.01	0.77 \pm 0.01	0.77 \pm 0.02	0.78\pm0.02*		
reference	0.80 \pm 0.01	0.80 \pm 0.02	0.80 \pm 0.02	0.81 \pm 0.01	0.80 \pm 0.01		
argumentative	0.77 \pm 0.01	0.77 \pm 0.02	0.78\pm0.02*	0.76 \pm 0.03	0.76 \pm 0.01		
negEmotion	0.70 \pm 0.01	0.72\pm0.04*	0.70 \pm 0.02	0.71\pm0.01*	0.70 \pm 0.02		
empathy	0.69 \pm 0.04	0.71 \pm 0.02	0.69 \pm 0.04	0.69 \pm 0.03	0.67 \pm 0.02		
Qjustification	0.89 \pm 0.01			0.89 \pm 0.01	0.87 \pm 0.01		
impact	0.47 \pm 0.02	0.49\pm0.02*	0.47 \pm 0.01				

Table 3: Comparison between task-specific adapter and fusions. Average performance (F1 macro and pearson correlation) on the test set. * denotes almost stochastic dominance ($\epsilon_{\min} < \tau$ with $\tau = 0.5$) and ** denotes truly stochastic dominance ($\epsilon_{\min} < \tau$ with $\tau = 0.0$)

6 Analysis: relationship between AQ and DQ dimensions

For each target dimension we analyze which adapters get activated during inference. We extract the attention scores for source dimensions for each target dimension based on the test set. Similar to Pfeiffer et al. (2021) we assume that high activations indicate more useful source tasks.

General AQ / AQ based on intuition First we compare two very general conceptualizations of general AQ: *quality* (from the IBM-rank dataset) which was trained on a wide variety of controversial topics and *clarity* with a slightly more tolerant conceptualization of quality (is the argument clear / understandable?) on 4 different topics. Both conceptualizations are rather under-specified and based on human intuition, we can thus gain insights into which dimensions play a particularly important role for the intuitive understanding of AQ. Figure 1 visualizes the most activated dimensions. For both dimensions different aspects, logical (*justification*, *cogency*) and rhetorical (*effectiveness*) are activated. Emotions play a role (high activation for *posEmotion*) and all dimensions from GAQ receive high activation indicating that they provide useful information in general. Interestingly, the adapter for *quality* (IBM) gets the most activation when modeling *clarity*, while the other way around is not the case. This may indicate that *clarity* represents a somewhat more specific conceptualization of argument quality, while *quality* reflects a more general.

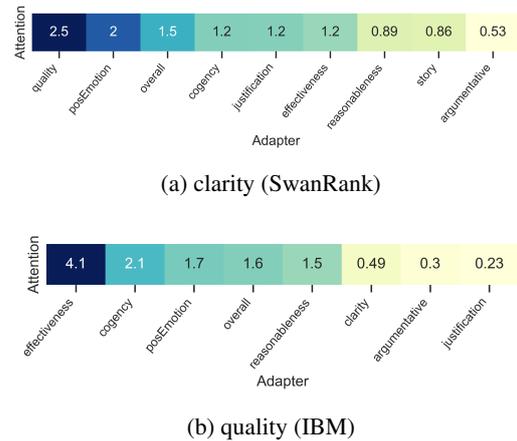
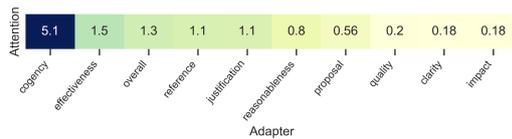


Figure 1: General conceptualizations of quality: sum of adapter activations over all layers.

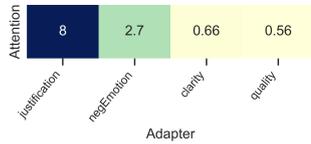
Recall also that the source corpus for *quality* is IBM rank, which covers 71 topics thus resulting in representations that are more applicable to corpora of other domains. Dialectical dimensions are less relevant, as both datasets contain single arguments without a discussion context.

Logical aspect of AQ and DQ With Figure 2 we can compare two logical conceptualizations, one from the AM community (*cogency* from GAQ) and one from the Social Science (*argumentative* from THF/BK). This allows us to explore the extent to which a similar conceptualization of logical argument quality varies between the two datasets from the different research communities.

Argumentative benefits mostly from the other logical dimension of the DQ dataset (*justification*),



(a) cogency (GAQ)



(b) argumentative (THF)

Figure 2: Logical conceptualizations of quality: sum of adapter activations over all layers.

while *cogency* benefits mostly from other dimensions from the same dataset. However *justification* also provides useful information for *cogency* hence seems to be the connecting element between the two conceptualizations. Other useful dimension from a deliberative source are *references* to other people for *cogency* and *negative emotions* for *argumentative*. Having a look at concrete correlation values reveals that the models pick up on positive and negative correlations: arguments with high cogency are less likely to focus on interaction with other people (refer to other peoples arguments) while more argumentative arguments in the THF/BK corpus express more negative sentiment.

Rhetorical aspects: narratives as alternative form of deliberation Finally, we examine a deliberative dimension that is rather rarely studied in the context of argument quality: *narration* (Figure 3). Moreover, this represents a rhetorical quality dimension, which enables us to compare how this kind of quality dimension differs from logical and general argument quality. Emotions as well as classical argumentative properties play a major role (high activation for positive, negative *Emotions* and *argumentative*), indicating that narration and argumentation are often intertwined. The high activation for *empathy* and *reference* (reference to others) illustrates perspective taking, which is characteristic for narrative. Overall, rhetorical and dialectical aspects play more of a role for this dimension.

We can summarize the following trends: either dimensions that come from similar or the same datasets or conceptually related dimensions are particularly activated. However, we also find empirical evidence that emotions play a role in modeling all

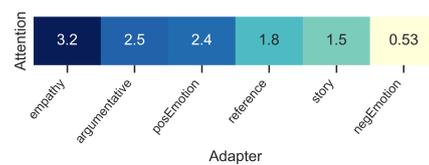


Figure 3: rhetorical conceptualization of quality – narration (THF): sum of adapter activations over all layers.

kinds of dimensions. We suspect that the relationship between emotions and AQ strongly depends on discourse/context, but further research is needed to investigate the relationships more precisely.

7 Experiment 2: predicting moderator interventions

In this experiment, we evaluate the models using a new down-stream application: we want to predict whether a comment in an online discussion should be moderated. Our hypothesis is that we can use information about different quality dimensions to solve the task. Moderation, especially in deliberative discussions such as on civic participation platforms, is a complex task that generally consists of facilitating a productive and fair discussion with respectful interaction. Since the task becomes more difficult for human moderators to perform as the number of participants and comments increases, automatic models can be useful for predicting whether a comment should be moderated. Here, the logical quality dimensions can help distinguish less argumentative from argumentative comments, the rhetorical dimensions are important for ensuring civil interaction, and the dialectical dimensions can identify valuable comments (is a solution proposed or the common good considered?).

We use the dataset from Park et al. (2012), in which the authors annotated the functions of moderation in discussions on a deliberative platform and identified ‘quality of comments’ as a common reason for intervention. The dataset was used in Falk et al. (2021), who obtained an F1 score of 0.34 using a full fine-tuning approach with roberta-base. The dataset is small and consists of 876 negative and 222 positive instances, a further motivation for a multi-task based approach. We train and test the models on the 5-fold split provided by Falk et al. (2021). As moderator interventions are the minority class we use class weights for all models. We compare the following models:

(*quality*) scores *ST*: we generate predictions for

each quality dimension and convert them into scores.³ Classifier: logistic regression.

(*quality*) *scores-MT*: similar to *quality scores* but we generated with the fusion-based adapters. Classifier: logistic regression.

model	F1 intervention	F1 macro
random baseline	0.29±0.06	0.45 ±0.04
scores-ST	0.37±0.05	0.55±0.04
scores-MT	0.38±0.04	0.54±0.04
moderation-ST	0.34±0.03	0.57±0.03
fusion-AQ	0.35±0.04	0.56±0.01
fusion-all	0.38±0.05	0.57±0.03
(Falk et al., 2021)	0.34±0.05	0.57±0.03

Table 4: F1 positive (moderator intervention) and F1 macro: average and standard deviation over 5 test sets.

moderation-ST: we train a single-task adapter on the task of moderation intervention.

fusion-AQ: we train a fusion on the task of moderation intervention using only quality adapters as input representations.

fusion-all: we train a fusion on the task of moderation intervention using all quality adapters and the adapter for moderation.

roberta-full: we report the result of Falk et al. (2021) who predict interventions on the same data split with full fine-tuning RoBERTa.

We use the same hyperparameters for the fusion and the single-task adapter as in experiment 2.

Can AQ adapters be applied to predicting moderator interventions? Table 4 shows F1 for interventions and F1 macro as average over the 5 splits. We consider F1 for interventions to be more important because it represents the minority class and only the positive instances are suggested to a human moderator for further evaluation. Figures 5 and 7 (Appendix) show the model-to-model calculated significance values for the almost stochastic order test. All models are outperforming the baseline. The single-task adapter yields similar results to full fine-tuning, the two feature-based models with the scores for the quality dimensions yield better results for interventions, indicating that the information on the quality dimensions is useful for this task. The best results are obtained with an adapter-fusion, provided that it also includes the adapter for moderation. This indicates that the information about the quality dimensions is comple-

³For dimensions based on binary classification we use the probability of the positive class, for the multi-class dimensions, we use the probability for each class (e.g. *common good* will be converted into three features: probability for class 1 ('no reference'), class 2 ('reference to own country') and class 3 ('reference to common good'))

mentary with a data and task specific representation (*moderation-ST*).

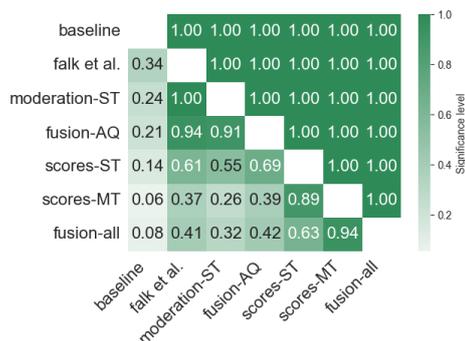


Figure 5: Almost Stochastic Order Scores (ϵ) for moderation test data for the F1 positive class, adjusted by using the Bonferroni correction. $\epsilon = 0.0$ means model in row is stochastically dominant over model in column, $\epsilon < 0.5$ denotes *almost stochastic dominance*.

Which aspects of AQ are important for predicting moderation? As discussed for the analysis of the relationship between quality dimensions in Experiment 1, an additional advantage of fusion-based models is the additional level of interpretability they provide. We investigate the relevance of each quality dimension for predicting moderation interventions using activation patterns of quality adapters. We compute the activation of each adapter of our best model (*fusion-all*) and visualize this as a heat-map (Fig. 4). The adapter for *impact* is the most activated. This is probably because this adapter is a good representation for distinguishing high vs. low quality comments, since the underlying dataset provides a high number of different topics (and thus can provide a good domain-independent representation). This is followed by dimensions that are important for a civic and appreciative interaction (*empathy* and *respect*) or for a solution-oriented discourse that considers the common good (*proposal*, *cgood*). The adapters for *argumentative* and *quality* add the more rational dimensions of two very different data sources, followed by the more affective and rhetorical dimensions (*story*, *narration*, *emotion*).

8 Conclusion

This work targeted the relationships between different aspects of argument and deliberative quality. We experimented with 6 datasets and 20 quality dimensions, employing adapters we learn modular representations of the targeted dimensions. We

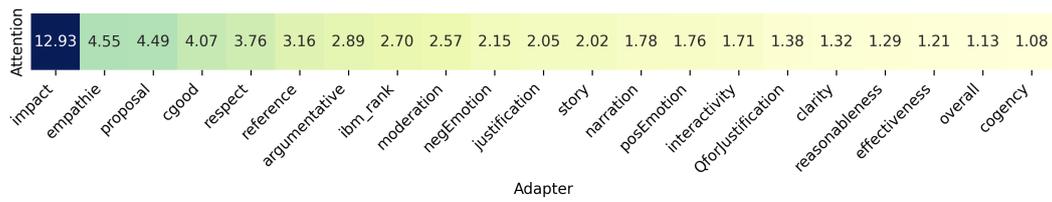


Figure 4: Predicting moderator intervention: activation for each dimension (avg. over all test instances, sum over all layers)

show that adapter fusion improves predictions in 8 dimensions out of 20. We then use the learnt adapters in the task of predicting moderator interventions - we show that information about different argument quality dimensions helps to improve the performance. Having more insights about which aspects of argument quality were activated more or less when the model that a user contribution should be moderated could help human moderators decide which aspects to focus on and information about why that model fired can increase human moderators awareness of and (ideally) confidence in automated support methods. We make models for single-task adapters and fusions and the code to train and test them available: <https://github.com/Blubberli/ArgQualityAdapters.git>.

9 Limitations

The datasets were created with very different motivations, the annotations were partly created by experts, partly via crowd-sourcing. The definitions of the different aspects of argument quality are also based on different theories or merely on human intuition. This work is only a first step to collect the existing data, to use it and to gain first insights about overlaps between relations based on empirical experiments. A deeper analysis of the underlying annotations and definitions is an urgent next step. Another limitation is that we compare the benefits of adapter-fusion to single-task adapters in a low-resource scenario. Because we are dealing with a large amount of different dimensions (20) additional experiments that compare this approach to full fine-tuning or traditional MLT-learning were not feasible in this work but can be conducted in the future, potentially on a smaller set of selected dimensions. On top of that we do not try to improve the state-of-the-art results for each quality dimension for each dataset. This is for the following reasons: the main focus of this work is to

investigate whether adapter-fusion improves the results compared to single-task adapters, not which model works best for which data set. The SOTA results for individual dimensions in our case are either not available (Social Science datasets) or based on data-specific optimizations of the hyperparameters / architectures. We focus on a variety of dimensions and datasets, especially those coming from the social sciences. In addition to the potential improvements in results through MLT with adapter-fusion, we see the advantage above all in the modular design (depending on the annotation from future datasets, dimensions can simply be added or omitted) and the insights we can gain about the contribution of individual dimensions through attention patterns. The models in this work were partially trained on small datasets. It is necessary to investigate to what extent the models are applicable to other domains. Also the influence of the topics in the discussions (topic bias) should be investigated.

Potential Negative Societal Impacts The automatic modeling of Argument Quality bears the danger that what is considered as "high quality arguments" will be closely related to what is represented as high quality in the existing datasets. This might disadvantage certain styles of argumentation but also certain opinions that are so far underrepresented in the data. It is therefore necessary to investigate how these models behave with data with such underrepresented styles and opinions and to create new datasets with AQ with greater diversity.

Acknowledgments

We would like to thank Anne Lauscher and Agnieszka Faleńska who provided valuable feedback at various points of this work. This research has been funded by Bundesministerium für Bildung und Forschung (BMBF) through the project E-DELIB (Powering up e-deliberation: towards AI-supported moderation).

References

- Milad Alshomary, Timon Gurcke, Shahbaz Syed, Philipp Heinisch, Maximilian Spliethöver, Philipp Cimiano, Martin Potthast, and Henning Wachsmuth. 2021. [Key point analysis via contrastive learning and extractive argument summarization](#). In *Proceedings of the 8th Workshop on Argument Mining*, pages 184–189, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Eustasio Del Barrio, Juan A Cuesta-Albertos, and Carlos Matrán. 2018. An optimal transportation approach for assessing almost stochastic order. In *The Mathematics of the Uncertain*, pages 33–44. Springer.
- Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. [Deep dominance - how to properly compare deep neural models](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2773–2785. Association for Computational Linguistics.
- Esin Durmus, Faisal Ladhak, and Claire Cardie. 2019. [The role of pragmatic and discourse context in determining argument impact](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5668–5678, Hong Kong, China. Association for Computational Linguistics.
- Katharina Esau. 2022. *Kommunikationsformen und Deliberationsdynamik*. Nomos Verlagsgesellschaft mbH & Co. KG.
- Neele Falk, Iman Jundi, Eva Maria Vecchi, and Gabriella Lapesa. 2021. [Predicting moderation of deliberative arguments: Is argument quality the key?](#) In *Proceedings of the 8th Workshop on Argument Mining*, pages 133–141, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Michael Fromm, Max Berrendorf, Johanna Reiml, Isabelle Mayerhofer, Siddharth Bhargava, Evgeniy Faerman, and Thomas Seidl. 2022. [Towards a holistic view on argument quality prediction](#). *CoRR*, abs/2205.09803.
- Marlène Gerber, André Bächtiger, Susumu Shikano, Simon Reber, and Samuel Rohr. 2018. [Deliberative abilities and influence in a transnational deliberative poll \(europolis\)](#). *British Journal of Political Science*, 48(4):1093–1118.
- Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Asaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2020. [A large-scale dataset for argument quality ranking: Construction and analysis](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7805–7813.
- Timon Gurcke, Milad Alshomary, and Henning Wachsmuth. 2021. [Assessing the sufficiency of arguments through conclusion generation](#). In *Proceedings of the 8th Workshop on Argument Mining*, pages 67–77, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ivan Habernal and Iryna Gurevych. 2016. [What makes a convincing argument? empirical analysis and detecting attributes of convincingsness in web argumentation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223, Austin, Texas. Association for Computational Linguistics.
- Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. [On the effectiveness of adapter-based tuning for pretrained language model adaptation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online. Association for Computational Linguistics.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for nlp](#). In *ICML*.
- Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. 2020a. [Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers](#). In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 43–49, Online. Association for Computational Linguistics.
- Anne Lauscher, Lily Ng, Courtney Napoles, and Joel Tetreault. 2020b. [Rhetoric, logic, and dialectic: Advancing theory-based argument quality assessment in natural language processing](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4563–4574, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Jialu Li, Esin Durmus, and Claire Cardie. 2020. [Exploring the role of argument structure in online debate persuasion](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8905–8912, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Lily Ng, Anne Lauscher, Joel Tetreault, and Courtney Napoles. 2020. [Creating a domain-diverse corpus for](#)

- theory-based argument quality assessment. In *Proceedings of the 7th Workshop on Argument Mining*, pages 117–126, Online. Association for Computational Linguistics.
- Joonsuk Park, Sally Klingel, Claire Cardie, Mary Newhart, Cynthia Farina, and Joan-Josep Vallbé. 2012. *Facilitative moderation for online participation in erulemaking*. In *Proceedings of the 13th Annual International Conference on Digital Government Research*, dg.o '12, page 173–182, New York, NY, USA. Association for Computing Machinery.
- Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. *AdapterFusion: Non-destructive task composition for transfer learning*. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.
- Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. *AdapterHub: A framework for adapting transformers*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.
- Christian Stab and Iryna Gurevych. 2017. *Recognizing insufficiently supported arguments in argumentative essays*. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 980–990, Valencia, Spain. Association for Computational Linguistics.
- Reid Swanson, Brian Ecker, and Marilyn Walker. 2015. *Argument mining: Extracting arguments from online dialogue*. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 217–226, Prague, Czech Republic. Association for Computational Linguistics.
- Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2022. *deep-significance-easy and meaningful statistical significance testing in the age of neural networks*. *arXiv preprint arXiv:2204.06815*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. *Attention is all you need*. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. 2017a. *Computational argumentation quality assessment in natural language*. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 176–187, Valencia, Spain. Association for Computational Linguistics.
- Henning Wachsmuth, Benno Stein, and Yamen Ajour. 2017b. *“PageRank” for argument relevance*. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1117–1127, Valencia, Spain. Association for Computational Linguistics.
- Henning Wachsmuth and Till Werner. 2020. *Intrinsic quality assessment of arguments*. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6739–6745, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Thiemo Wambsganss, Christina Niklaus, Matthias Söllner, Siegfried Handschuh, and Jan Marco Leimeister. 2020. *A corpus for argumentative writing support in German*. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 856–869, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Appendix

A Implementation details

For training the adapters and the adapter-fusion models we use the `adapter-transformers` library (Pfeiffer et al., 2020) with `roberta-base` as a backbone. We use the default hyperparameters (learning rate of 0.0001 which was found to work empirically best in most setups (Pfeiffer et al., 2020, 2021) and a reduction factor of 16). As a maximum sequence length we use 256 which is higher than all means of the 6 datasets. We train the adapters for a maximum of 40 epochs for classification and 25 epochs for regression and use the model with the best performance (lowest MSE or highest F1 macro) on the validation set. For the adapter-fusion we also rely on the best learning rate according to Pfeiffer et al. (2021), which is $5e-5$. We lower the maximum number of epochs (25 for classification and 15 for regression). We train all models on 3 GPUs (*NVIDIA RTX A6000, each GPU has 49GB, CUDA Version 11.7*) with a batch size of 16, each model is trained with 3 seeds (5, 42, 108) for the experiments reported in section 4. We use the the adapters of one seed (42) to generate the predictions and the single-task adapters trained with that seed for the AdapterFusion in the experiment in section 7. The largest model is the AdapterFusion with 21 adapters (all quality dimensions and moderation). The training run time for this is 15.349 samples per second and 44.282 samples per second during inference. In experiment 3, for the logistic regression classifiers, we find the best hyperparameters using grid search and 3-fold cross-validation on a separate data split (L2 penalty, class weights and $C=0.1$).

B Datasets

The tables in the end of this Appendix (Table 13 for THF/BK, Table 14 for Europolis, Table 15 for GAQ and Table 16 for SwanRank, IBM-Rank-30k and Kialo) illustrate examples of each dataset, each example exhibits a high score (or label) of a different dimension of AQ.

Parts of the transcriptions of the Europolis dataset were not in English and automatically translated using DeepL (<https://www.deepl.com/translator>). Similarly, the online-comments from THF/BK are originally German and have been automatically translated

using DeepL. Samples of the automatic translations were verified by native speakers.

Data splits for Experiment 1: Table 5 shows the amount of training / development and test data for each corpus.

	train	dev	test
THF/BK	788	198	247
Europolis	546	140	175
Kialo	650	150	200
GAQ	650	150	200
SwanRank	650	150	200
IBM-Rank-30k	650	150	200

Table 5: Amount of train, validation and test data for each dataset. The amount for *Kialo, GAQ, SwanRank, IBM-Rank-30k* has been down-sampled to 1000 instances.

Table 6 gives an overview the positive amount of instances for each quality dimension in the training data. Most of the dimensions (except argumentative) are the minority class.

dimension	relative amount in train
posEmotion	13 %
proposal	38 %
narration	31 %
reference	41 %
argumentative	75 %
negEmotion	21 %
empathy	11 %
Qjustification	20 %

Table 6: Relative amount of positive instances for each quality dimension in the THF/BK training set.

Table 7 and 8 show the distribution of each class label for the dimensions in Europolis and the one in Kialo.

Table 9 and 10 show the mean and standard deviation for the point-wise quality scores in the training data of GAQ, SwanRank and IBM-Rank-30k.

C Experiment 1

Heuristic The following describes more details about the heuristic used to select source tasks for the multi-task experiment in section 4. To generate predictions we first train single-task adapters on the original datasets. We use the original train/val/test

Dimension and labels	amount	Dimension and labels	amount
interactivity		respect	
negative reference	41 %	disrespectful	10 %
no reference	4 %	implicit respect	75 %
neutral reference	35 %	explicit respect	15 %
positive reference others	20 %	justification	
cGood		no justification	16 %
no reference	9 %	inferior justification	40 %
own country	76 %	qualified justification	34 %
common good	15 %	sophisticated	10 %
storytelling			
storytelling	33 %		
no storytelling	67 %		

Table 7: Distribution of class labels for each dimension in the Europolis training set.

impact labels	relative amount in train
not impactful	22 %
medium impactful	23 %
impactful	55 %

Table 8: Distribution of class labels for *impact* in the kialo training set.

split for IBM-Rank-30k (train=20974, val=3208, test=6315), GAQ (train=2746, val=1177, test=538) and Kialo (train=5170, val=1108 test=1108) and create our own split for SwanRank (train=3440, val=860, test=1075), THF/BK and Europolis (splits in Table 5).

Table 11 shows the results of each single-task adapter on the original-sized dataset. We report the mean and standard deviation across 3 seeds.

We then take the adapter for each dimension and generate predictions for all other datasets. For feasibility, we sample 3000 instances for Kialo, IBM-rank-30k and SwanRank to generate predictions on these subsets. Based on the predictions we compute the pair-wise Spearman correlations between the AQ dimensions for each dataset. For binary classes we use the probability of the positive class as a continuous score, for dimensions with 3 to 4 classes we convert the predicted class labels into scores on a linear scale, e.g. *impact* has 3 class

dimension	mean	std
cogency	3.29	0.65
effectiveness	3.13	0.76
reasonableness	3.05	0.72
overall	3.14	0.72

Table 9: Mean and standard deviation of point-wise quality for each dimension in the GAQ corpus.

dimension	mean	std
quality	0.79	0.20
clarity	0.53	0.24

Table 10: Mean and standard deviation of point-wise quality for *clarity* and *quality* in the corresponding training sets.

dimension	performance
<i>Pearson correlation</i>	
overall	0.56±.00
cogency	0.54±.01
effectiveness	0.59±.01
reasonableness	0.49±.00
quality	0.55±.00
clarity	0.73±.01
<i>F1 macro</i>	
justification	0.46±.04
interactivity	0.35±.05
respect	0.50±.04
cgood	0.60±.04
story	0.75±.02
Q(uestion) for justification	0.89±.01
reference	0.80±.01
argument	0.77±.01
narration	0.76±.02
proposal	0.79±.01
negEmotion	0.70±.01
posEmotion	0.64±.03
empathy	0.69±.04
impact	0.52±.01

Table 11: Results on the test set for each quality dimension. Performance with standard deviation, averaged over 3 seeds.

labels: low impact, medium impact, high impact which we convert into 1, 2 and 3 respectively. We compute the pair-wise correlations for each dataset (we take the gold annotations when available) and average them across datasets. Figure 6 shows the pair-wise correlations as a correlation matrix.

Next we sample source tasks for each target task based on the correlations. Taking different samples of dimensions we compute a threshold based on absolute correlation values and add a dimension as source task if the correlation to the target dimension exceeds the computed threshold. We consider the following setups:

fusion corr ALL We select the source tasks from all dimensions if the absolute correlation value is

higher than 0.24 (corresponds to the third quartile of all correlations).

fusion corr DQ We consider only source tasks from datasets with a deliberative focus (THF/BK, Europolis, Kialo). The tasks are sampled from 14 dimensions and the threshold is 0.15 (third quartile of all correlations between the 14 dimensions).

fusion corr AQ We extract the source tasks from all dimensions that stem from more general argumentative contexts (IBM-rank-30-k, GAQ, swanson). The threshold is based on the correlations between the 6 dimensions (0.54, the second quartile due to the high correlations between the GAQ dimensions).

Table 12 shows the source task dimensions for each target dimension, depending on the setup (*fusion corr ALL*, *fusion corr DQ*, *fusion corr AQ*). Each dimension is learned using between 1 and 9 other dimensions as source tasks. For *respect* there were no dimensions with a high enough correlation in any of the setups.

D Experiment 2: predicting moderator interventions.

Figure 7 shows the significance matrix between all models for the task of predicting moderator interventions for the F1 macro score.

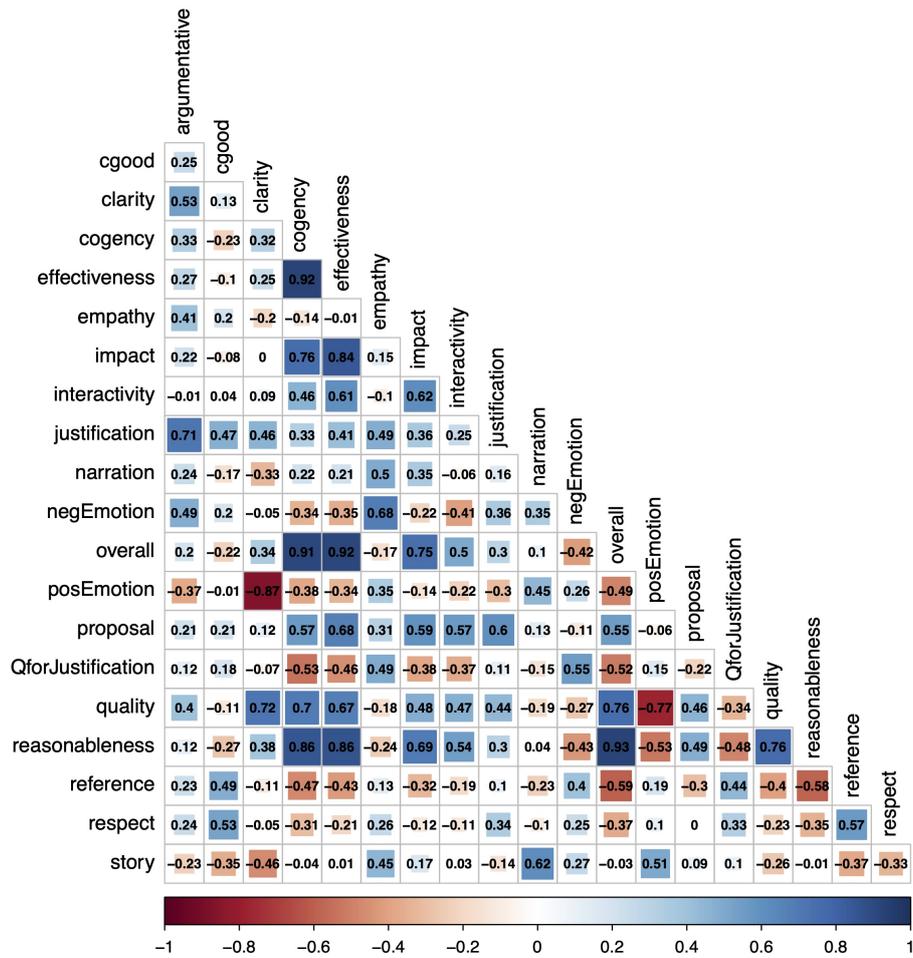
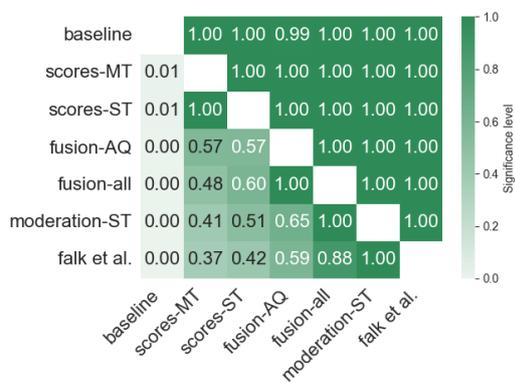


Figure 6: Pairwise Spearman correlations between all quality dimensions based on single-task adapter predictions. Average across all 6 datasets.

target task	additional source tasks using all dimensions	additional source tasks using only dimensions from deliberative context	additional source tasks using only dimensions from general argumentation
overall	impact, effectiveness, proposal, quality, reference, reasonableness, clarity, cogency, justification	-	effectiveness, reasonableness, cogency
cogency	impact, effectiveness, overall, quality, reference, reasonableness, clarity, justification	-	effectiveness, overall, quality, reasonableness
reasonableness	impact, effectiveness, overall, quality, reference, clarity, cogency, justification	-	effectiveness, overall, cogency
effectiveness	impact, proposal, overall, quality, reference, reasonableness, clarity, cogency, justification	-	overall, quality, reasonableness, cogency
quality	effectiveness, overall, posEmotion, reasonableness, clarity, argumentative, cogency, justification	-	effectiveness, cogency
clarity	effectiveness, overall, posEmotion, quality, reasonableness, story, argumentative, cogency, justification	-	-
justification	effectiveness, proposal, overall, quality, negEmotion, reasonableness, clarity, argumentative, cogency	proposal, empathy, negEmotion, cgood, argumentative	-
story	empathy, reference, clarity, narration	empathy, posEmotion, cgood, reference, narration	-
interactivity	-	negEmotion	-
cgood	-	story, justification	-
posEmotion	quality, clarity, narration	story, argumentative, narration	-
proposal	effectiveness, overall, reference, justification	reference, justification	-
narration	empathy, posEmotion, negEmotion, story	empathy, posEmotion, negEmotion, reference, story, argumentative	-
reference	effectiveness, proposal, overall, reasonableness, story, cogency	proposal, story, narration	-
argumentative	quality, negEmotion, clarity, justification	empathy, posEmotion, negEmotion, narration, justification	-
negEmotion	empathy, argumentative, narration, justification	empathy, interactivity, argumentative, narration, justification, QforJustification	-
empathy	negEmotion, story, narration	negEmotion, story, argumentative, narration, justification, QforJustification	-
Qjustification	-	empathy, negEmotion	-
impact	overall, reasonableness, cogency, effectiveness	-	-

Table 12: Multi-task experiments: target dimension with source dimensions used as input adapters for adapter-fusion.



(a) F1 macro

Figure 7: Almost Stochastic Order Scores (ϵ) for moderation test data for the F1 macro score, adjusted by using the Bonferroni correction. $\epsilon = 0.0$ means model in row is stochastically dominant over model in column, $\epsilon < 0.5$ denotes *almost stochastic dominance*.

example	dimension
<p>On the one hand, our lignite is needed to maintain an affordable and reliable energy supply (and based on physical and economic laws, will still be needed in 50 years) and on the other hand, our lignite can do more than just be burned to generate electricity.</p>	argumentative
<p>In New Zealand, residents of the Pacific island of Tuvalu have already been granted the right to asylum - on the grounds of climate change. Who is asking for their recovery? How do people who have been forced to flee their homes due to the global burning of fossil fuels and the resulting DECREASED global warming read our news and debates? "Act only according to that maxim by which you can at the same time will that it become a general law." If we include Immanuel Kant's thoughts in the guiding decision, shouldn't lignite mining really end at the A61 and RWE workers be supported in the corporation's structural transformation in a way that provides well for them and their families?</p>	empathy
<p>Classical music for all Once a week the Berlin Philharmonic Orchestra should play at THF for ALL Berliners. This way even families with little money can enjoy classical music. The prices at the Philharmonie or concerts by other great musicians are so immensely high that only higher earners can afford it. This is an outrage because they are subsidized by us and we can't even afford to go.</p>	proposal
<p>I think #person is more than right and I share his opinion... Lignite has and should continue to have a place here in the region. Good luck</p>	reference (to other discourse participants)
<p>I have been to Holzweiler many times. The experiences from Immerath and Borschemich show that the club life in an intact village does not suffer due to the resettlement. On the contrary, it strengthens the feeling of togetherness and allows the clubs to flourish.</p>	narration
<p>But I also think what #person wrote is great. One notices from it that not immediately a rejection against it prevails but rather a certain concern. In particular here around animals. You also notice that there is still a great ignorance. I find great that you have expressed yourself. I think the discussions here should be there to reduce possible worries and prejudices. Thank you</p>	positive emotion
<p>I have been following what has happened to lignite for many years and I think it is terrible. I've lived in the Rhineland for years and it's easy to live with the changes caused by lignite. More and more good jobs are disappearing in Germany. My last employer is already cutting well-paid jobs due to the low oil price.</p>	negative emotion

Table 13: Examples for each Quality Dimension in THF/BK

example	dimension
<p>I have friends from Latin America and many other places and they work and they pay in a pot. So, I'm a mother and well, unfortunately, if I have to go to another country, well, I try to integrate in the country I'm going to. I'm not going to go there, to impose my goals [?], my way of seeing, no. I'm going there to work and not to steal. And there is something else. But again, I'm holding back.</p>	storytelling
<p>I don't know if you can regulate it well, how many people immigrate or emigrate or whatever. I think it's important to create a basis for all people to be able to live in their country. Because I think that is actually the main cause. That many industrialized countries are bleeding small countries or poor countries dry and taking away their livelihood. And that's why people emigrate, because they no longer have anything to eat, because they can no longer find work in their country, and because life in the industrialized countries is simply made out to be nicer or better. In order to be able to ultimately prevent an immigration policy, illegals, I believe that you first have to change the basis in the other countries, that is, the countries of origin. Create a basis. Life base.</p>	sophisticated justification
<p>Well, I am of the opinion that simply in the population the term EU is seen completely wrong; one always wants only something and one wants to give nothing. I am simply of the opinion that it should be a community and a community simply has to support the weaker ones and the stronger ones simply have to give. I think this is the basic problem of the EU and I think it's very nice that today and in the next few days this could contribute to the fact that this spirit, which was really brought into being by Robert Schuman and by all those who have worked so hard for the EU, could be recognized and a community could really take place; at least in the microcosm now.</p>	reference to common good
<p>Here, when we are talking about immigration, it should be first identified why a certain person left his country. Just like my friend before said to be a refugee is also a man, which probably feels bad in his own country. For example, when his country's situation does not give him a life in dignity. So I think that every country should identify immigrants and help them in certain ways, for example with social benefits. I know that some countries for example Poland are not rich countries, so they need EU help in such a matter. Especially the countries where immigration is quite high.</p>	explicit respect
<p>Yes I completely agree with what this gentleman just said because I think we have created ghettos, we have - at the moment - people who live very very badly, immigrants who live very very badly, who are already unemployed, who have enormous problems of integration and I think we should already make an effort to integrate these people who are well in our countries and then we see what we can do to bring in others, we must already take care of the people who are on our territory and who are living very badly and who are unemployed, who are poorly cared for, who have problems with their children, school problems, problems with papers and I think that we must already arrive once we have properly resolved these problems and that we will have sooner than bring people in and make them unhappy - I think that it is perhaps worse than doing something more moderate.</p>	positive reference to other discourse participant
positive reference	

Table 14: Examples for each Quality Dimension in Europolis

example	dimension
<p>I'm a fairly tolerant human being and am in no way an advocate of the death penalty. I also understand that a short sentence and rehabilitation is also an effective form of justice in terms of re-offending in many countries. However I think there's gotta be a line drawn somewhere in which a person entirely loses their liberty and autonomy if the crime they committed was as heinous as the one committed by Mr Breivik. Also, even though I am aware that there is a good chance he will still remain behind bars for the rest of his life, the possibility that he won't baffles and worries me. Please CMV.</p>	high cogency (5)
<p>Well, this topic has raised lots of questions lately particularly in France. This is where I stand: - Wearing a burqa should be a matter of choice, just as women choose to wear anything else, regardless of any religious manifestations. - Wearing the burqa shouldn't be banned, and shouldn't be forced to women either; it should be a personal choice. - When talking about choices, it's the society that gives these choices according to what the majority thinks, hence the more civilized and democratic a society is the more choices people have. - It's basically a matter of respect, if a woman chooses to wear it then we should respect that, we can't force her not to wear it, as we can't force her to wear it: free will :) , on the other hand that woman should respect and obey all the security issues that comes along with wearing it.</p>	high effectiveness (5)
<p>The point of daylight savings is to make our numeric time cycle fit with the Sun's time cycle. In other words, standardize the time of day in which the sun is shining. This way, people and businesses can keep their operating hours steady without working in the dark, and less electricity is used. Most arguments I've heard against it pertain to the inconvenience of changing clocks and accounting for gained/lost hour, but with most clocks being digital and synced up to DST nowadays, that's becoming less and less of a problem. And besides, one day of inconvenience in exchange for a whole season of "correct" daylight seems like a pretty good deal to me.</p>	high reasonableness (5)
<p>I believe property is a social construct that is only justified through appeals to utility. In other words, any particular set of property laws are only justified insofar as they make people better off, in terms of their capabilities. Most Libertarians I've debated with either believe property rights are somehow fundamental(natural or God-given) or develop out of other moral principles, like the NAP. The first option appeals to non-existent entities. The second is circular, as what NAPER's define as aggression is violation of property rights, and violations of property rights is defined in terms of the NAP.</p>	high overall (5)

Table 15: Examples for each Quality Dimension in GAQ

example	dimension	Dataset
A basic principle of punishment is that it should be proportional to the crime, and therefore capital punishment is the only legitimate response to a crime such as first degree murder.	high (1.0) quality	IBM-Rank-30-k
When voters are able to make an impact and change their votes more often they will feel more engaged with the political process, and get more involved in politics.	high impact	Kialo
First a prediction is made from an hypothesis of some observation that must be true if the hypothesis is correct.	high (1.0) clarity	SwanRank

Table 16: Examples for each Quality Dimension in SwanRank, Kialo and IBM-rank-30-k

Interventional Probing in High Dimensions: An NLI Case Study

Julia Rozanova¹, Marco Valentino², Lucas Cordeiro¹, André Freitas^{1,2}

University of Manchester, United Kingdom¹

{firstname.lastname}@manchester.ac.uk¹

Idiap Research Institute, Switzerland²

{firstname.lastname}@idiap.ch²

Abstract

Probing strategies have been shown to detect the presence of various linguistic features in large language models; in particular, semantic features intermediate to the “natural logic” fragment of the Natural Language Inference task (NLI). In the case of natural logic, the relation between the intermediate features and the entailment label is explicitly known: as such, this provides a ripe setting for *interventional* studies on the NLI models’ representations, allowing for stronger causal conjectures and a deeper critical analysis of interventional probing methods. In this work, we carry out new and existing representation-level interventions to investigate the effect of these semantic features on NLI classification: we perform *amnesic* probing (which removes features as directed by learned linear probes) and introduce the *mnesic* probing variation (which forgets all dimensions *except* the probe-selected ones). Furthermore, we delve into the limitations of these methods and outline some pitfalls have been obscuring the effectivity of interventional probing studies.

1 Introduction

The *probing* paradigm has emerged as a useful interpretability methodology which has been shown to have reasonable information-theoretic underpinnings (Pimentel et al., 2020; Voita and Titov, 2020; Zhu and Rudzicz, 2020), indicating whether a given feature is captured in the intermediate vector representations of neural models. It has been noted many times that this does not generally imply that the models are *using* these learnt features, and they may represent vestigial information from earlier training steps (Ravichander et al., 2021; Elazar et al., 2020).

Only through interventional analyses can we start to make claims about which modelled features are used for a given downstream task: this is the aim of works such as Elazar et al. (2020);

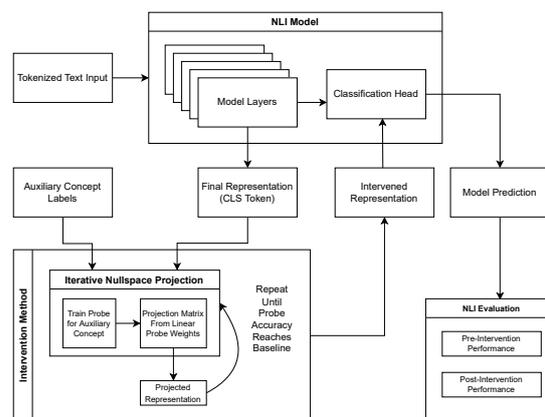


Figure 1: Workflow for interventional probing for NLP classification models: a basis for both the *amnesic* and *mnesic* intervention strategies.

Giulianelli et al. (2018) and Geiger et al. (2021). We refer to the case where the interventions are guided by trained probes as *interventional probing*.

It has been suggested in Elazar et al. (2020) (as the guidance for their *amnesic probing* methodology) that if features are strongly detected by probes, one may use debiasing methods such as *iterative nullspace projection (INLP)* (Ravfogel et al., 2020) to intervene on the corresponding vector representations and effectively “remove” the features before re-insertion into the given classifier. Investigating the effect of these intervention operations on the classifier performance could allow for stronger causal claims about the role of the probe-detected features.

In this work, we delve deeper into the amnesic probing methodology with an NLI case study and identify two key limitations. Firstly, there is an issue of dimensionality: when the number of dimensions is high and the number of auxiliary fea-

ture classes is low, it seems that amnesic probing is not sufficiently informative. In particular, we cannot rely on the same control baselines to reach the kind of conclusions discussed in (Elazar et al., 2020), as nulling out small numbers of random directions consistently has no impact on the downstream performance. Secondly, in the linguistic settings explored in Elazar et al. (2020), we do not have expectations for exactly *how* or even *if* the explored features should be affecting the downstream task. This makes it difficult to explore the effectivity of the methodology itself.

To this end, we propose the use of a controlled subset of NLI called *natural logic* (MacCartney and Manning, 2007). In this setting, the intermediate linguistic features of *context monotonicity* and *lexical relations* are already known to be highly extractable from certain NLI models’ hidden layers (Rožanova et al., 2021b), allowing us a certain amount of understanding and control of these features’ representations in the latent space. Using the deterministic and well-understood nature of the problem space where we have concrete *expectations* about the theoretical interaction between the intermediate features and the downstream label, we may critically analyse the effectivity of interventional probing.

Through the application of probe-based interventions in this setting, we show that blindly applying the amnesic probing argument structure leads to unexpected and contradictory conclusions: the two features which the final label is *known* to depend on are shown to have no influence on the final classification (both jointly and independently). This further calls into question the suitability of these methods for situations where a small number of feature label classes and high dimensionality of representations is concerned. Even more perplexingly, when we treat the NLI gold label itself as an intermediate feature which can be nulled out with INLP, we yet again observe *almost no change to the NLI performance*. As such, the feature removal strategy appears ineffective here: we attribute this to the disproportionate size of probe-selected feature subspaces to the very high-dimensional representations.

In response, we introduce and study a variation which we call *mnesic* probing, which we show to be more informative in the high-dimensional, low-class-count setting: the core idea is to *keep only* the directions identified by the iteratively trained

probes. This allows us to analyse much lower dimension subspaces, while making better use of the outputs of the INLP strategy used in amnesic probing.

We find that *mnesic probing* leads to more informative observations which are a) in line with expected behaviour for natural logic, and b) yield results which seem to better discriminate between model behaviours.

In summary, the contributions of the paper are as follows:

1. We propose the setting of *natural logic* to be ripe territory for exploration of interventional probing strategies.
2. We note two limitations of the amnesic probing methodology, demonstrating both dimensionality limitations for the control baselines 4.4 and contradictory behaviour in the NLI setting 4.2 (namely that that the expected effects of semantic features on the downstream NLI task are notably absent).
3. Building upon previous interventional methodologies, we introduce an additional *mnesic* intervention operation which uses the outputs of the INLP process in the opposite way.
4. We contrast the mnesic probing strategy with the amnesic probing results, and demonstrate it presents more informative results which are aligned with the constructed expectations in our high dimensional, low label class count setting.

2 Interventional Probing

We may summarise the general setup of interventional probing as follows: suppose we start with a classification model that may be decomposed as $f \circ g : \mathcal{X} \rightarrow \mathbb{R}^n$, where g is an encoder module which yields a representation which serves as an input to the classifier head f , and n is the number of output classes of the final classifier. We aim to intervene on the output of g and observe the change in the performance of f (usually in comparison with some kind of random control baseline intervention).

Linear probes (also known as *diagnostic classifiers*) are able to identify subspaces in which a given intermediate feature set is found to be represented. These may be used as a guide for vector-

level interventions on the representation space; we are specifically concerned with interventions which are vector *projections*. Otherwise, The exact nature of this intervention is interchangeable. We consider two projection strategies in particular: the *amnesic* intervention introduced in Elazar et al. (2020) (described further in section 2.2) and our *mnesic* variation which uses the same INLP technique (section 2.3).

2.1 What Should it Tell Us?

The interventional probing steps are performed on exactly the representation that would have been an input to the classifier head f . We may re-insert the intervened representations and re-calculate the classifier accuracy (note that the iterative projections in sections 2.2 and 2.3 maintain the original dimensionality of the vector set but reduce the *rank*).

We are looking to see if the downstream performance of the classifier f drops. If it does, the interventions have removed information that was necessary for successful classification. However, as any projection would remove some information, these results must be viewed in the context of a control intervention: if the INLP process ends up removing n directions, a sample of n randomly chosen directions is selected from the original representation, Elazar et al. (2020) argue that if the amnesic downstream performance drops significantly more than the random removal control performance, we may conclude that the features were necessary for the final downstream classification. On the other hand, if the performance does not drop at all, the features were not useful for the classifier in the first place. In the ensuing sections and results, we demonstrate that this is not necessarily a valid conclusion.

2.2 The Amnesic Intervention

We follow the procedure in (Elazar et al., 2020) (in turn based on *iterative nullspace projection* (Ravfogel et al., 2020)): given a set X of encoded representations for the textual input (with dimensions $\text{num_examples} \times \text{embedding_dimension}$), we iteratively train linear SVM classifiers according to a set of auxiliary feature labels. For each INLP step i , This yields a linear transformation $W_i X + B$, where the vectors of W_i define directions onto which the probe projects the representations for auxiliary label classification (i.e., these are the chosen directions most aligned with auxiliary class separation). For each step i , an orthogonal basis denoted R_i is found for this rowspace. The

projection to the intersection of the nullspaces is given by a matrix

$$PX = (I - (R_0 + \dots + R_n))X.$$

The matrix product PX is a matrix in the original dimensions of X , but with reduced rank by the number of iteration steps (as each projection "flattens out" the representation in these directions).

Projection to the intersection of nullspaces is thus the removal of any information pertaining to the auxiliary feature labels (or at least, the information which allows high performance for a linear probe). The training terminates these auxiliary task classifiers start consistently performing at the majority class baseline, indicating that there is no further linearly information to be extracted from the remaining representation. As such, the resulting representation is treated as an altered representation where this feature is *removed* or forgotten.

2.3 A Variation: The Mnesic Intervention

Elazar et al. (2020) perform a series of experiments on various linguistic features which had previously been shown to be well-captured in language model representations and use the amnesic probing methodology to distinguish between features that are *used* by the model and those that are not by comparing post-intervention downstream task performance to a baseline of randomly removed directions.

Rather than projecting the embedded representations to the intersection of nullspaces of the trained probes (removing the target property), we project them to the *union of the rowspaces* with the transformation:

$$\begin{aligned} (I - P)X &= (I - (I - (R_0 + \dots + R_n)))X \\ &= (R_0 + \dots + R_n)X \end{aligned}$$

This has the opposite effect: we use projection to null out *everything except* the directions identified by the probes as indicative of the target feature. As such, we "remember" only that feature rather than forgetting it.

3 Experimental Setup

In this study, we use interventional methods ¹ to study the internal behaviour of NLI models.

¹We reuse much of the code included with (Elazar et al., 2020), but we include our data and reproducible experimental code at https://github.com/juliarozanova/mnesic_probing.

We compare amnesic and mnestic variations of the INLP strategy, evaluating intermediate feature probing performance and downstream NLI performance after every step of the intervention process.

For each auxiliary feature label and model, we perform the *interventional probing* as outlined in figure 1.

3.1 Dataset

Our setting for this study is a fragment of NLI called *Natural Logic* (MacCartney and Manning, 2007). In particular, we focus on single-step natural logic inferences in which entailment examples are generated by replacing a noun phrase in a sentence with a hyponym, hypernym or unrelated noun phrase. The context of the substituted term is either *upward* or *downward* monotone, as determined by the composition of negation markers, generalized quantifiers or determiners present in the context. The entailment label of the example is a consequence of this feature and the lexical relation between the substituted terms.

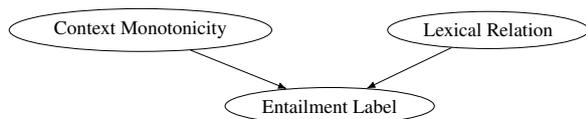


Figure 2

We use the NLI_{XY} dataset from (Rožanova et al., 2021b,a). By construction, the NLI_{XY} dataset consists of NLI examples which rely on exactly these two abstract features: context monotonicity and the lexical relation of the substituted terms.

We perform two flavours of probe-based interventions (described fully in section 2) with four feature label sets (described next).

Auxiliary Feature Labels We begin with the two relevant intermediate features (respectively, context monotonicity and lexical relation) which are already known to correlate with stronger performance on the downstream NLI_{XY} task (Rožanova et al., 2021b). We will refer to this as *single-feature* interventional probing, as the probing and intervention steps are only applied to one feature set at a time. Next, we combine the two features in a cross product, creating a new feature label set with all possible combinations of these intermediate features (in the dataset, they are completely independent variables by construction (Rožanova

et al., 2021a)). We refer to this as the *composite feature label*.

Lastly, we also consider the *entailment label* itself (the downstream task label) as an input to the interventional probing process. The latter is particularly useful as a diagnostic sanity check, and aids the critical nature of our findings.

3.2 NLI Models and Encoding

We compare a selection of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) models trained for NLI classification. Firstly, we include a pair of models trained respectively on the MNLI (Williams et al., 2018) and SNLI (Bowman et al., 2015) benchmark datasets. In (Rožanova et al., 2021b) and (Rožanova et al., 2021a), it is shown that when `roberta-large-mnli` (a model which performs well on benchmarks but poorly on the targeted NLI_{XY} challenge set) receives additional training on the adversarial HELP dataset (Yanaka et al., 2019) it improves in NLI_{XY} performance and *begins to show high probing performance for the relevant intermediate features*, context monotonicity and lexical relations: this is the necessary precondition for doing interventional probing. We include two of their models with this property: `roberta-large-mnli-help` and `roberta-large-mnli-double-finetuning`, with the other models included for a contextual comparison.

We perform probing and intervention on the final representation that precedes the NLI classification head: in the case of BERT and RoBERTa, this is the [CLS] token of the final layer.

The initial input is a tokenized NLI example from the NLI_{XY} dataset. The findings in (Rožanova et al., 2021b) show that the intermediate feature labels (context monotonicity and lexical relations) are detectable in the concatenated tokens of the substituted noun phrases: however, for interventional purposes, we perform the probing and intervention steps on the [CLS] token which serves as an input to the NLI classifier head: we have found that the same features are detectable to a comparable standard, and this is the only position at which we are able to make a sensible intervention that would allow conclusions about the final classifier head only.

3.3 Evaluation

The significant metrics for these interventional probing paradims are the *probing accuracy* before

and after the iterative nullspace projection steps (a decline to random performance indicates the feature is being “removed” from the representation in the sense that it is no longer detectable by linear probes) and the *downstream classification accuracy* on the NLI task the model’s were trained for (in our case, we report the accuracy on the NLI_XY task).

For amnesic probing, we report the performance deltas for both the probing and downstream tasks. However, for mnesic probing, a slightly more nuanced and qualitative view is helpful: it can be assumed that eventually mnesic probing will reach comparable performance to the untouched vector representations, but we are interested in the comparative rates at which this happens. As the interventions are iterative, we may feed the intervened representations into the classifier head at *each step* of the intervention process - we use this to provide a step-wise presentation of results in linear plots in figure 5.

While the tabulated deltas in table 1 results are sufficient to present our observations on amnesic probing, for comparison we also include the step-wise graphical presentations in the appendix.

4 Results and Discussion

4.1 Single Feature Amnesic Probing

The results for the standard amnesic probing procedure are in table 1. In particular, the single feature results are in the rows with features labelled *insertion relation* and *context monotonicity*. The amnesic operation is successful - the respective probing accuracies approach and reach the majority class baseline.

We also include the step-wise plots of both probing performance and downstream NLI task performance: we single out the case of the insertion relation label in figures 3 and 4, but include the full suite of expanded plots for each feature in the appendix. The length of the iterative amnesic probing process is indicative of the number of dimensions removed to reach this baseline: it can also be considered a proxy for the strength of the feature presence in the representations, or rather, the dimension of the semantic subspace corresponding to the target features.

The second phase of this process, i.e. the resubstitution of the modified representations as inputs to the NLI classifier head, can be seen in the right hand portion of table 1, labelled *NLI-XY Perform-*

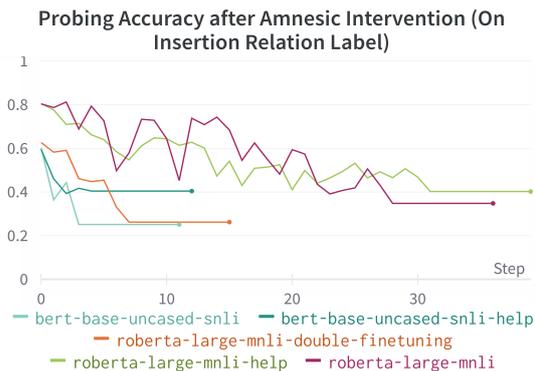


Figure 3: Step-wise probing performance throughout the amnesic probing process: a decrease towards the random baseline accuracy (roughly 0.3 for this 3-class task) indicates the feature is less and less extractable from the remaining representations as the iterative process continues.

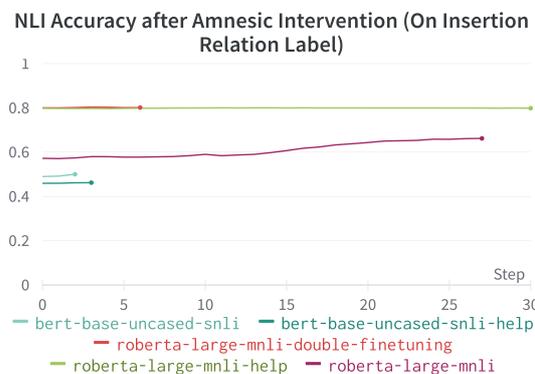


Figure 4: Downstream performance on NLI_XY after amnesic intervention (removing lexical relation information). For such an important feature to the end-task, we would expect to see a drop: but we don’t!

mance. The result is unexpected: for each of these features, *the downstream task performance appears to be unaffected after their removal*. This is surprising when the dataset is explicitly controlled to rely only on these two features.

4.2 Multi Feature Amnesic Probing

The results for the amnesic probing procedure utilizing *both* auxiliary feature label sets and the entailment gold label are in the rows of table 1 with labels *composite* and *entailment label* respectively. We observe that once again, the downstream task performance is mostly unaffected. Unlike the unexpected result in the previous section, it’s difficult to argue away the fact that this is somewhat contradictory: while single feature removal may be subject to some confounding bias, the removal of

Model	Feature	Probing Performance		NLI-XY Performance	
		Start	Intervention Δ	Start	Intervention Δ
roberta-large-mnli-help	insertion relation	80.58	-40.35	79.79	0.06
	context monotonicity	87.65	-46.22	79.79	-0.09
	composite	64.48	-43.95	79.79	0.32
	entailment label	78.05	-37.49	79.79	-1.57
roberta-large-mnli-double-finetuning	insertion relation	62.7	-36.49	80.04	0.11
	context monotonicity	89.79	-43.28	80.19	0
	composite	57.64	-49.56	80.08	-1.67
	entailment label	82.8	-24.94	80.19	-16.53
roberta-large-mnli	insertion relation	80.39	-45.59	57.22	8.99
	context monotonicity	75.44	-27.49	57.37	-0.43
	composite	72.35	-53.51	57.24	-2.27
	entailment label	73.6	-15.31	57.37	0.1
bert-base-uncased-snli-help	insertion relation	59.53	-19.1	45.95	0.28
	context monotonicity	82.72	-33.94	45.52	-2.35
	composite	37.19	-17.08	45.76	13.68
	entailment label	47.05	0.38	45.91	0
bert-base-uncased-snli	insertion relation	60.26	-35.14	48.99	1.05
	context monotonicity	81.09	-30.77	49.42	-6.25
	composite	35.37	-17.83	50.73	7.45
	entailment label	42.44	-0.24	49.42	0

Table 1: Amnesic probing performance deltas across models and target feature labels: first listed is the performance on the probing task with respect to the indicated feature, and then the accuracy on the downstream NLI-XY task. We note the results pre-intervention and the ensuing change in accuracy.

both features exhausts the variables on which this classification depends. This is highly unexpected, and suggests a point of failure for the amnesic probing process. Naturally, we cannot be without doubt that despite all our best efforts to work with a controlled dataset that relies only on these two know (but still complex) features, a model may yet find unrelated heuristics to exploit that may correlate so strongly with the downstream task label that it may perform well without representing and using these intermediate features. However, we imagine this to be a rather low probability scenario to be that the model simultaneously learns such heuristics but simultaneously learn representations that create strong clusters for the known intermediate features *without using them at all*. The models which we have observed to perform more less well on NLI-XY (such as roberta-large-mnli) are indeed estimated to be using sub-par heuristics, but this also comes with poor probing results for the intermediate features - naturally, this in itself does not imply anything conclusive, but certainly adds to our convictions.

On a separate note, it is noted in Elazar et al. (2020) that there is no control for the number of dimensions removed, while there is a clear correlation between downstream task performance and the number of label classes (and thus removed probe

directions) are in play. Our feature sets have only 2 and 3 classes respectively. In the most analagous result in (Elazar et al., 2020) where the auxiliary features had very few classes and no change on the downstream performance was observed, it was concluded that the features must have no effect on the outcome. It is very likely that *too little information* is being removed in this process to observe any impact on the downstream task performance. This could potentially be pointing to high redundancy in the representations which the amnesic intervention may struggle to remove appropriately.

4.3 Mnestic Probing

Given the possible dimensionality problem, the alternative method of *mnemonic* probing seems promising: after the mnemonic intervention, many dimensions are removed and few remain, so it appears to be a ripe setting for observing and comparing effects on downstream NLI accuracy at a finer granularity. The results for NLI-XY task accuracy after the *mnemonic* probing procedure are presented as step-wise plots in figure 5. There is a clear increase in NLI performance with subsequent addition of probe-chosen directions to the representations, especially viewed in the context of section 4.4, where we compare the performance to random choices of included directions. In the latter, performance

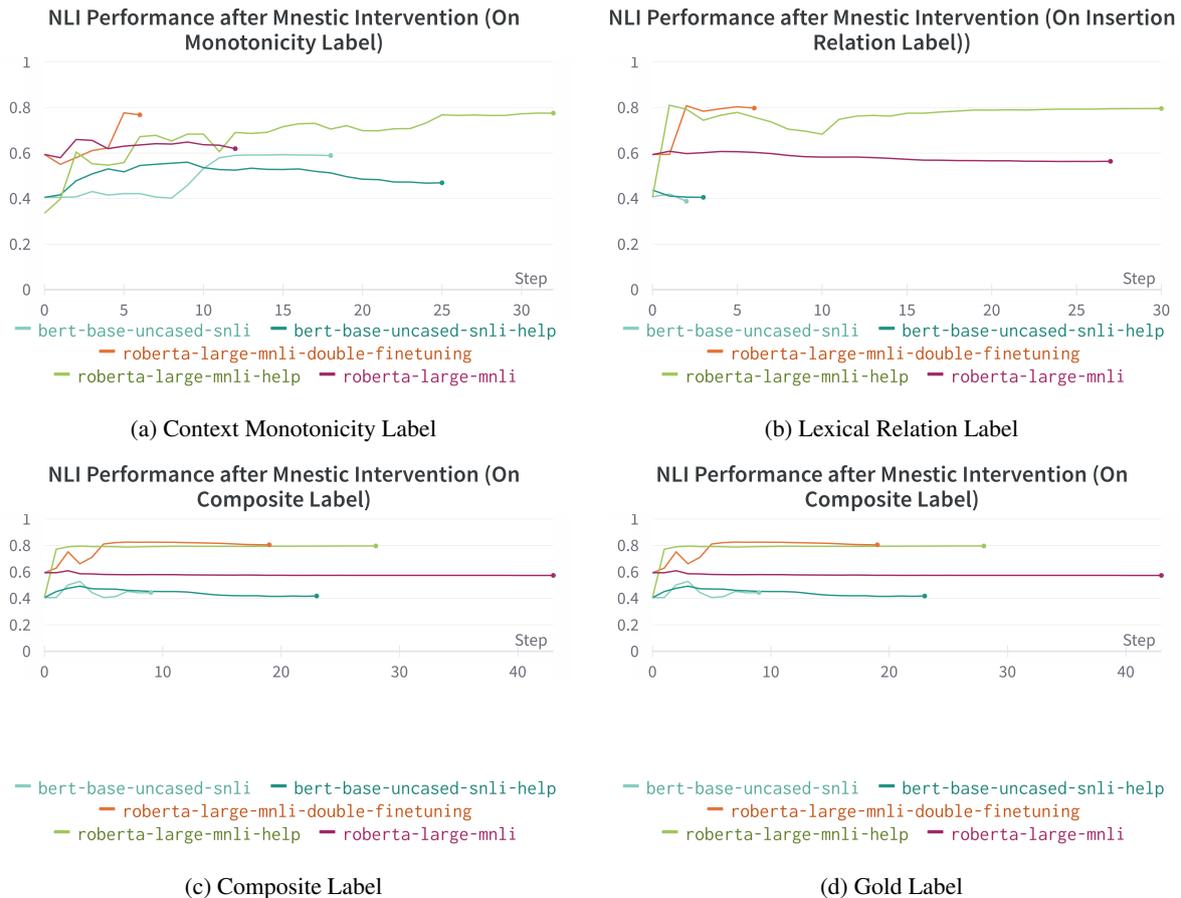


Figure 5: Downstream NLI Task Performance After Mnestic Interventions

varies randomly rather than presenting a structured increase as seen here.

We observe that the *composite* label and the gold *entailment* label are reflected in line with expectations in the mnestic probing experiments: the inclusion of the probe-selected dimensions with respect to these labels introduces a sharp and immediate increase in the NLI classifier performance. This is significantly steeper than the baseline increase observed in random addition of representation directions. Similarly, the increase is nearly as sharp for the lexical relation label. However, although an increase is observed during the iterative mnestic probing intervention for context monotonicity, this increase is not at a dramatically higher rate than adding subsequently more directions from the original representation. For monotonicity specifically, this is not enough to conclude that the feature (or at least, the corresponding probe-selected dimensions) are critical to the final classifier.

Nevertheless, we have been able to make clearer observations than were possible in the amnesic probing setting.

4.4 Control Comparison

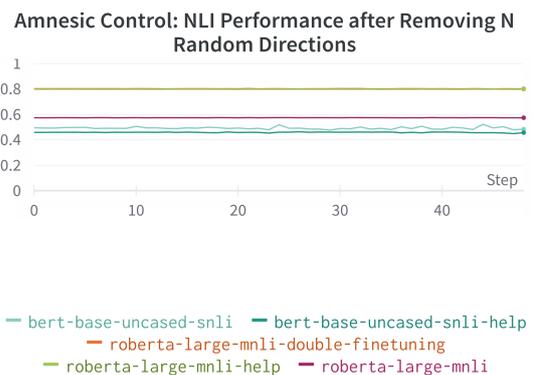


Figure 6: Amnesic control experiment: Downstream NLI accuracy upon the *removal* of n random directions of the original representation.

We contextualise all the preceding results with a set of control experiments both for amnesic (figure 6) and mnestic (figure 7) probing. Note in particular that even with very few random dimensions kept, downstream performance starts approaching comparable levels to the full representations. As

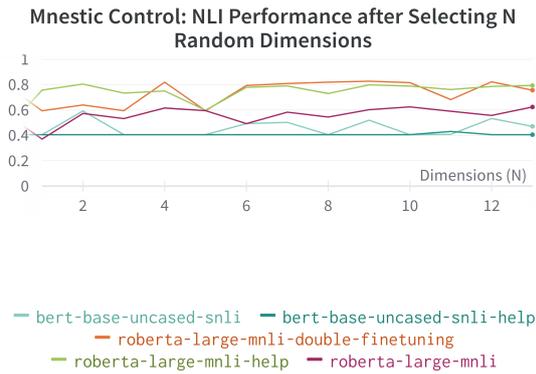


Figure 7: Mnestic control experiment: downstream NLI accuracy upon the *selection* of n random directions of the original representation.

such, a single random baseline as in Elazar et al. (2020) can be misleading: there is enough variability in the random direction results so as to allow for a false claim of feature irrelevance by simply getting lucky; as few as 3 dimensions can perform at the original model’s performance level or arbitrarily lower.

Lastly, we compare to the mnestic probing results in figure 5: with the probe-selected mnestic dimension choices, the increase in downstream performance does seem to happen faster and in a more consistent fashion, while the selection of n randomly chosen directions introduces very haphazard performance spikes. This suggests the probe-selected dimensions are consistently adding to the model’s access to the relevant information, and this may be stronger evidence for the usefulness of the examined features for the final classification.

5 Related Work

The use of probing as an interpretability strategy dates back as far as works such as Alain and Bengio (2018) and (Conneau et al., 2018), but a core set of work on the detailed development of the methodology includes Hewitt and Liang (2019); Belinkov and Glass (2019); Voita and Titov (2020); Pimentel et al. (2020). For a full survey, see Belinkov (2022).

The application of probing strategies to natural logic components has been explored in Rozanova et al. (2021b) and Geiger et al. (2020). In Rozanova et al. (2021b), probing experiments have proven effective in detecting the presence or absence of features such as *context monotonicity* and *phrase-pair relations* in the internal representations of NLI models.

Regarding interventions as interpretability tools for machine learning classifiers, there are two broad categories: those that modify the raw input (such as image or text) in a controlled way, and those that modify the hidden/latent vector representations of the data at various stages of the models’ input processing. While input-level interventions are more common as they are usually easier to control and are strongly interpretable, they don’t allow us to explore and conjecture about exact high-level representational mechanisms in the latent space. We tabulate a few relevant interventional interpretability methods in table 2. Note in particular the variation in the *generation* step for the intervened input; some use generative modelling for counterfactual examples, while we use cheaper linear probes.

The only other work in which interventional methods have been applied to natural logic is Geiger et al. (2021): a similar problem setting is considered, but at a finer granularity. Our work focuses more on the summarised abstract notion of context monotonicity as a single feature, rather than the intermediate tree nodes that determine its final monotonicity profile. The interventions used in this work are vector *interchange* interventions; partial representations from transformed inputs are used, as opposed to direct manipulations of the encoded vectors.

6 Conclusion and Future Work

Our experimental setting has shown significant limitations of amnesic probing in high-dimensional settings where there are few label classes (and consequently fewer dimension modified), even if these classes are strongly detectable. Our results point out that it is misguided to conclude that a given feature is not used when post-amnesic-intervention downstream performance fails to drop, especially in our example amnesic probing studies of a) the gold downstream feature label and b) the composite of two labels that jointly determine the entailment label. This may be due to a dimension/rank confounder variable and high redundancy of information in the representations. It remains to be checked whether high performance in the random control directions corresponds to strong alignment with these probe-selected directions: we propose an analysis of the *dot products* with the fixed set of probe-selected dimensions, which indicates a shared directionality measure (0 for orthogonal vectors and 1 for codirectional ones).

	Intervention	Tested Effect	Feature Characterisation	Requires Intermediate Labels	Intervention Linked to Concept Interpretation	Domain
Amnesic Probing / INLP (Elazar et al., 2020)	Debiasing / Feature Removal	Downstream Classifier Accuracy	Linear Classifier	Yes	No	Language Modelling
CausaLM: Causal Model Explanation Through Counterfactual Language Models (Feder et al., 2021)	Re-Training Model Copy For Counterfactual Representation	Text representation-based individual treatment effect (TREITE)	Retrained Base Model	Yes	Yes	Sentiment Analysis
Explaining Classifiers with Causal Concept Effect (Goyal et al., 2019)	Generative Modeling	Average Causal Effect Measure	VAE	Yes	Yes	Vision Classification
Concept Activation Vectors (TCAV) (Kim et al., 2018)	Value Shift in Vector Direction	Custom Gradient Sensitivity Measure	Linear Classifier	Yes	Yes	Vision Classification
Latent Space Explanation (Gat et al., 2021) by Intervention	VAE Input Discretization and Reconstruction	Reconstruction Quality	VAE	No	Qualitative Judgement (Vision Only)	Vision Classification
Meaningfully Debugging Model Mistakes Using Conceptual Counterfactuals (Abid et al., 2022)	Weighted Combination of Concept Vectors	Difference Between Concept Addition and Removal Effect	Linear Classifier	Yes	Yes	Vision Classification

Table 2: Related Work on Latent Concept Interventions

In summary: we have introduced a modification of the amnesic probing paradigm which we call *mnesic* probing which uses the same INLP process but considers the opposite intervention: using the union of projection rowspaces to keep *only* the directions the probes have identified to be mod-elling the target information. This strategy presents results that are more aligned with theoretical expectations (in the NLI case), possibly because we are now able to make comparisons in a lower rank setting.

7 Limitations

A key limitation of the *mnesic* probing strategy is that as one reconstructs the original representation one dimension at a time, information content is naturally due to increase: as such, no *mnesic* probing result can be viewed in isolation, but should be used as a comparative study. Preferably, various randomized selections of linear subspaces with the same number of dimensions should be included as baselines input representations. Furthermore, we mention two some additional caveats: firstly, the probing strategies used here to identify the informative semantic subspaces in question are always linear; relevant information may be present non-linearly. However, as with amnesic probing, we discount any non-linearly encoded information as the final model classification layer is linear and thus cannot exploit this information. Lastly, probing for subspaces which are informative of target auxiliary features may always include correlated features in the resulting subspaces; this must always be taken into account when drawing conclusions from *mnesic*/amnesic probing.

References

Abubakar Abid, Mert Yuksekgonul, and James Zou. 2022. [Meaningfully debugging model mistakes using conceptual counterfactual explanations](#). In *Pro-*

ceedings of the 39th International Conference on Machine Learning, volume 162 of *Proceedings of Machine Learning Research*. PMLR.

Guillaume Alain and Yoshua Bengio. 2018. [Understanding intermediate layers using linear classifier probes](#).

Yonatan Belinkov. 2022. [Probing classifiers: Promises, shortcomings, and advances](#). *Computational Linguistics*, 48(1):207–219.

Yonatan Belinkov and James Glass. 2019. [Analysis methods in neural language processing: A survey](#). *Transactions of the Association for Computational Linguistics*, 7:49–72.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. [What you can cram into a single \$\&\!#\&\$ vector: Probing sentence embeddings for linguistic properties](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

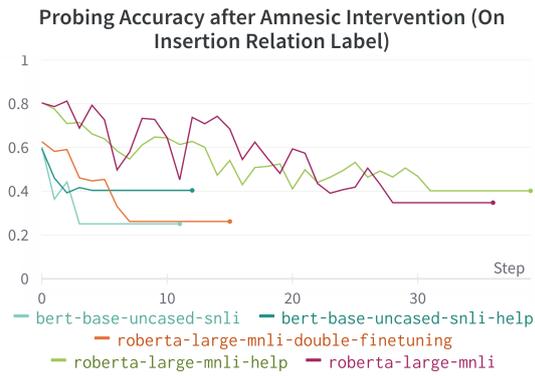
Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. [Amnesic probing: Behavioral explanation with amnesic counterfactuals](#).

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. [CausaLM: Causal model explanation through counterfactual language models](#). *Computational Linguistics*, 47(2):333–386.

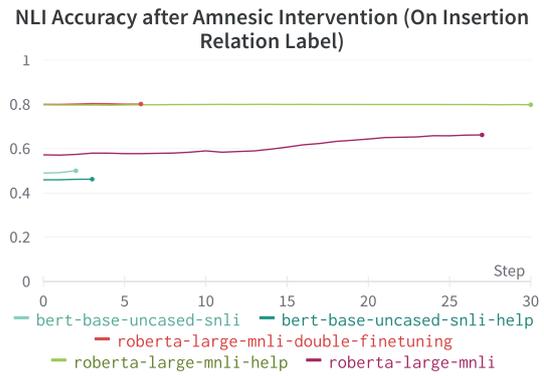
Itai Gat, Guy Lorberbom, Idan Schwartz, and Tamir Hazan. 2021. Latent space explanation by intervention.

- Atticus Geiger, Hanson Lu, Thomas F Icard, and Christopher Potts. 2021. [Causal abstractions of neural networks](#). In *Advances in Neural Information Processing Systems*.
- Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. [Neural natural language inference models partially embed theories of lexical entailment and negation](#). In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.
- Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. [Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.
- Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. 2019. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*.
- John Hewitt and Percy Liang. 2019. [Designing and interpreting probes with control tasks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.
- Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.
- Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.
- Bill MacCartney and Christopher D. Manning. 2007. [Natural logic for textual inference](#). In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague. Association for Computational Linguistics.
- Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. [Information-theoretic probing for linguistic structure](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.
- Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. [Null it out: Guarding protected attributes by iterative nullspace projection](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7237–7256. Association for Computational Linguistics.
- Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. [Probing the probing paradigm: Does probing accuracy entail task relevance?](#) In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online. Association for Computational Linguistics.
- Julia Rozanova, Deborah Ferreira, Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021a. [Supporting context monotonicity abstractions in neural nli models](#).
- Julia Rozanova, Deborah Ferreira, Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2021b. [Decomposing natural logic inferences in neural NLI](#). *CoRR*, abs/2112.08289.
- Elena Voita and Ivan Titov. 2020. [Information-theoretic probing with minimum description length](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.
- Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. [HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning](#). In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 250–255, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zining Zhu and Frank Rudzicz. 2020. [An information theoretic view on selecting linguistic probes](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9251–9262, Online. Association for Computational Linguistics.

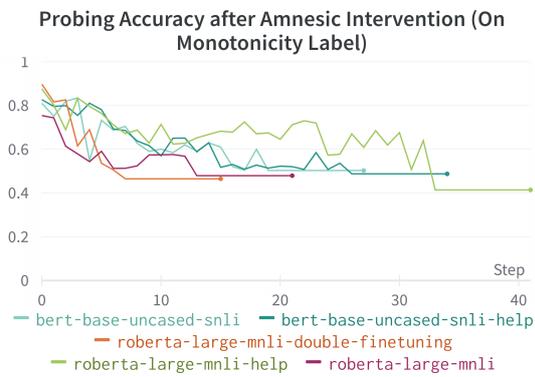
A Expanded Amnesic Intervention Results



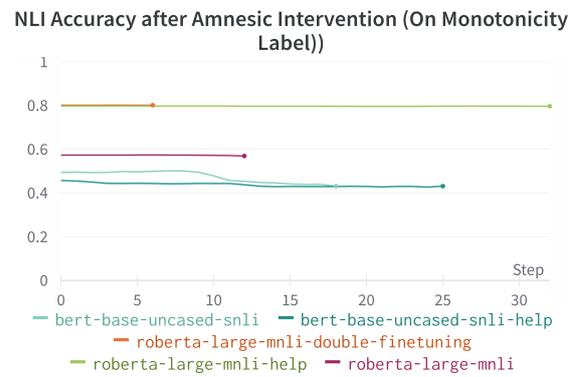
(a) Lexical Relation Probing Performance During Iterative Amnesic Intervention Process



(b) Downstream Performance On NLI_XY After Amnesic Intervention (Removing Lexical Relation Information)

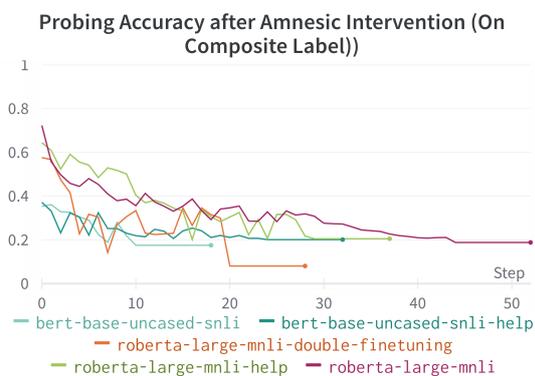


(c) Context Monotonicity Probing Performance During Iterative Amnesic Intervention Process

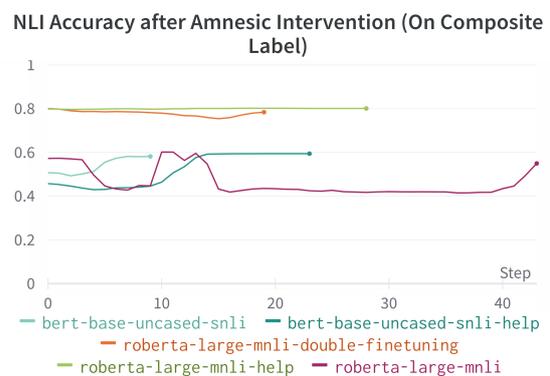


(d) Downstream Performance On NLI_XY After Amnesic Intervention (Removing Context Monotonicity Information)

Figure 8: Single Feature Amnesic Probing

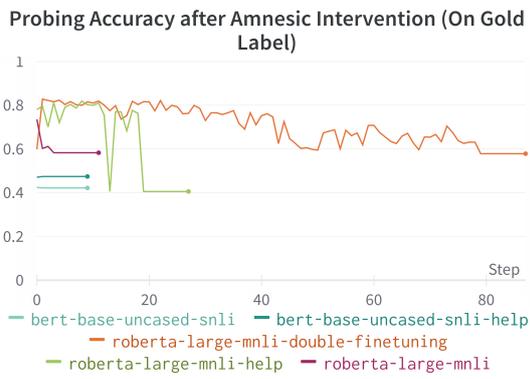


(a) Probing Performance On NLI_XY After Composite Label Amnesic Intervention

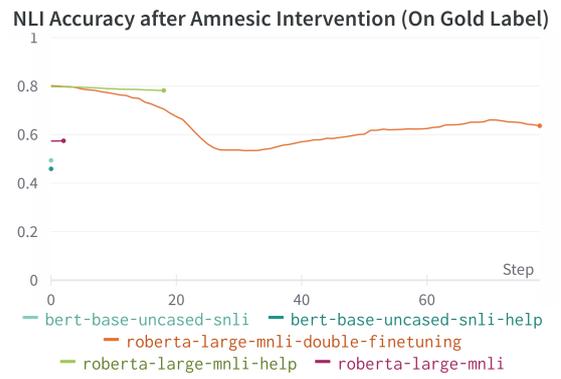


(b) Downstream Performance On NLI_XY After Composite Label Amnesic Intervention

Figure 9: Composite Feature Label Amnesic Probing



(a) Probing performance On NLI_XY after entailment label amnesic intervention.



(b) Downstream performance on NLI_XY after entailment label amnesic intervention.

Figure 10: Sanity Check: Entailment Gold Label Amnesic Probing

Program Synthesis for Complex QA on Charts via Probabilistic Grammar Based Filtered Iterative Back-Translation

Shabbirhussain Bhaisaheb, Shubham Paliwal, Rajaswa Patil,
Manasi Patwardhan, Lovekesh Vig, Gautam Shroff

TCS Research, India

{shabbirhussain.b, shubham.p3, patil.rajaswa,
manasi.patwardhan, lovekesh.vig, gautam.shroff}@tcs.com

Abstract

Answering complex reasoning questions from chart images is a challenging problem requiring a combination of natural language understanding, fine-grained perception, and analytical reasoning. Current chart based Question Answering (QA) approaches largely address structural, visual or simple data retrieval type questions with fixed-vocabulary answers and perform poorly on reasoning queries. We focus on answering realistic, complex, reasoning-based questions where the answer needs to be computed and not selected from a fixed set of choices. Our approach employs a neural semantic parser to transform Natural Language (NL) questions into SQL programs and execute them on a standardized schema populated from the extracted chart contents. In the absence of program annotations, i.e., in a weak supervision setting, we obtain initial SQL predictions from a pre-trained CodeT5 semantic parser and employ Filtered Iterative Back-Translation (FIBT) for iteratively augmenting our NL-SQL training set. The forward (neural semantic parser) and backward (language model) models are initially trained with an external NL-SQL bootstrapping data. We iteratively move towards the required NL query distribution by generating NL questions from the synthesized SQL programs using a Probabilistic Context-Free Grammar (PCFG) where the production rule probabilities are induced to be inversely proportional to the probabilities in the training data. We filter out the generated NL queries with mismatched structure and compositions. Our FIBT approach achieves State-of-the-Art (SOTA) results on reasoning-based queries in the PlotQA dataset yielding a test accuracy of 60.44%, superseding the previous baselines by a large margin.

1 Introduction

Charts and plots are compact visualization techniques capturing illustrated facts that are frequently used in scientific and financial documents for sum-

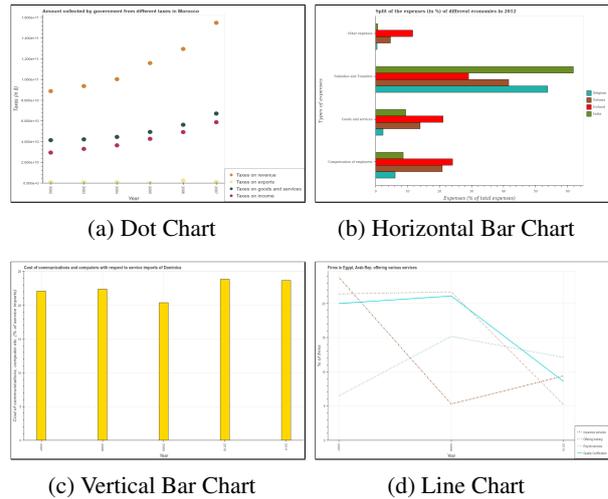


Figure 1: PlotQA Chart Types

marizing observations and drawing conclusions about the underlying data. Inferring relevant conclusions from charts entails answering complex reasoning style queries, a task which has so far proved challenging to automate. Most existing approaches and datasets for automatic QA over charts specifically focus on structural, visual, relational or simple data retrieval type queries (Chaudhry et al., 2020; Siegel et al., 2016; Kim et al., 2020). Also, as depicted in Table 1 many approaches assume either binary answers or assume the answer belongs to a fixed vocabulary.

On the other hand, for real world applications, more complex reasoning based questions have to be answered which involves a combination of perception, language understanding, and reasoning. For example, to answer the question depicted in Table 1, for the chart (d) in Figure 1, the following steps have to be followed: (i) Refer to the legend to infer that the ‘percentage of firms offering quality certification’ are depicted by the solid sky blue line, (ii) For each year value on the X-axis, retrieve the corresponding Y-values depicted by the solid sky blue line, (iii) If the value of corresponding year y is $V(y)$, then find if $\forall y_i$ and y_j , where $i \neq j$,

if $y_i > y_j$ then $V(y_i) \geq V(y_j)$, to determine if the values are monotonically increasing. In this work, we address such complex reasoning style questions on charts. These questions may also involve nested arithmetic and aggregation operations over the chart data and thus the answer is not necessarily derived from a fixed vocabulary, or extracted from the chart text. We evaluate our approach against the reasoning questions provided in the PlotQA V2 dataset (Methani et al., 2020).

We define a common schema for the data across all chart types and employ a state-of-the-art chart visual extractor to populate the schema with chart data. The PlotQA dataset does not provide any SQL program annotations for the Natural Language (NL) questions (only answers). We automatically generate SQL programs for these NL questions by using a Filtered Iterative Back-Translation (FIBT) approach (Hoang et al., 2018) and execute these programs on the extracted schema to compute the answers. We use the SPIDER (NL to SQL) dataset to train both the forward and backward models for FIBT. The NL query distribution of this dataset is different from the required PlotQA query distribution, in terms of query composition, schema (database) structure and chart domains. We build on the observations of (Guo et al., 2020), who empirically show that Iterative Back Translation (IBT) improves the performance of compositional generalization while generating logical forms from NL questions by correcting errors in the pseudo-parallel data at each iteration.

We define a Probabilistic Context-Free Grammar (PCFG), as a subset of the SQL grammar, to sample SQL programs executable on the extracted chart schema. Existing PCFG based data augmentation approaches for semantic parsing (Wang et al., 2021b), synthesize SQLs following similar compositions to that of a given query set by inducing proportional grammar probabilities. Our approach differs from these as we set the probabilities of our PCFG to be inversely proportional to the set-of programs in the training data. This results in synthesis of SQL programs that (i) were not present in the current training data and (ii) follow the program distribution of the SQL programs needed to compute the answers for the PlotQA questions. We iteratively augment the training data by generating NL questions from SQL programs. We further use denotations and a novel compositional similarity based filtration strategy for removing noisy

NL-SQL pairs. We observe that this data augmentation and filtration strategy results in improvement in the PlotQA execution accuracy for every iteration of FIBT, finally achieving State-of-the-Art performance on the reasoning-based queries for the PlotQA dataset with a 60.44% test set accuracy, superseding the previous baseline (14.82%) (Methani et al., 2020) by a large margin and even surpassing human performance. The ChartQA model (Masry et al., 2022) showed that the T5 language model offers the best overall performance for the questions in PlotQA. We demonstrate that our approach surpasses T5 for complex reasoning type of questions in PlotQA with OOV answers. The main contributions of this work are:

- To the best of our knowledge, ours is the first approach to effectively address reasoning style NL questions over charts whose answers are computed and not restricted to a fixed vocabulary.
- In absence of program annotations, we propose a weakly supervised FIBT approach for SQL synthesis with novel data augmentation and filtration strategies to adapt the Neural Parser to more closely follow target NL-Question distribution.
- Our approach allows us to achieve State-of-the-Art results on PlotQA reasoning-based questions with a 60.44% test set accuracy, superseding the previous baseline (14.82%) by a large margin and even surpassing human performance (58.70%).
- As opposed to existing end-to-end approaches (Singh and Shekhar, 2020; Kafle et al., 2020; Chaudhry et al., 2020), our approach is more interpretable as we can track reasoning patterns in the synthesized programs via the generated programs.

2 Related Work

2.1 Datasets for Chart Q&A

Chart QA datasets such as DVQA (Kafle et al., 2018) or FVQA (Kahou et al., 2017) are synthetically generated with limited variations, containing simple binary or fixed-vocabulary questions. To avoid these biases, Leaf-QA (Chaudhry et al., 2020) and PlotQA (Methani et al., 2020) datasets are constructed from open real-world sources from World Bank, Government, Global Terrorism Database, etc. Questions in these datasets are in English and are templated but paraphrased to prevent models

NL Questions	(a) Dot Chart	(b) Horizontal Bar	(c) Vertical Bar	(d) Line Chart
Structural	Is the number of dotlines equal to the number of legend labels? (Y/N)	How many groups of bars are there? (Fixed)	Does the graph contain any zero values? (Y/N)	Does the graph contain grids? (Y/N)
Retrieval	What is the amount collected as tax on revenue in 2005? (Open)	What is the label of the 4th group of bars from the top? (Chart)	What is the label or title of the Y-axis? (Chart)	What is the title of the graph? (Chart)
Reasoning	Is the difference between the amount collected as tax on goods in 2003 and 2007 greater than the difference between the amount collected as tax on exports in 2003 and 2007? (Y/N)	What is the difference between the highest and the second highest percentage of amount spent on other expenses? (Open)	Do a majority of the years between 2008 and 2011 (inclusive) have cost of communications and computer greater than 10? (Y/N)	Does the percentage of firms offering quality certification monotonically increase over the years? (Y/N)

Table 1: PlotQA Questions for Charts in Figure 1, Answer Types: Yes/No, Fixed, Chart or Open Vocabulary

Data	Images		Questions		
	Total	R*	Total	Reasoning	R*
Train	157,070	12,934	20,249,479	16,593,656	69,000
Valid	33,650	3,110	4,360,648	3,574,081	13,740
Test	33,657	-	4,342,514	3,559,392	-
Total	224,377	16,044	28,952,641	23,727,129	82,740

Table 2: PlotQA V2 Dataset Statistics. R*: Representative images (c_{rep}) and questions (q_{rep}) used for FIBT

from memorizing the templates. Both datasets have a significant proportion of analytical reasoning queries, however the PlotQA dataset has 81.95% complex value-based queries, requiring stronger numerical and analytical reasoning capabilities (Chaudhry et al., 2020). Also 80.84% PlotQA queries have answers from an open vocabulary. Moreover, $\sim 82\%$ of questions in PlotQA are reasoning based, as opposed to the recently introduced CharQA dataset (Masry et al., 2022), which has only 43% (compositional) reasoning based questions. Since our main focus is on answering complex questions requiring numerical and analytical reasoning, we use the reasoning based questions in the extended version (V2) of the publicly available PlotQA dataset (Table 2) to evaluate our approach.

2.2 Chart Q&A Approaches

Existing end-to-end approaches use deep models to combine image and question features at various levels of granularity (Kafle et al., 2020). A recent approach (Chaudhry et al., 2020) fuses the chart entities extracted using a Masked RCNN and the NL question using spatial attention to predict the answer. (Singh and Shekhar, 2020) use a structural transformer-based learning that takes the question encoding as input and uses the feature maps of the chart’s visual elements, with its localization information used as positional encodings. These approaches provide results on previously mentioned FVQA, DVQA, and LeafQA datasets on relatively simpler queries. Recently, (Masry et al., 2022; Masry and Hoque, 2021) published benchmarks for chart QA using several end-to-end approaches including, VL-T5 (Cho et al., 2021), TAPAS (Herzig

et al., 2020) and VisionTaPas, which is an extension of TAPAS (Masry et al., 2022) and T5 (Raffel et al., 2019). TAPAS is able to address very simple aggregation type queries and cannot handle complex queries with nested aggregation and arithmetic operations and thus provides poor results on PlotQA (12.90%). T5 provides the best reported results for PlotQA V2 (56.22% test accuracy for all queries). Our proposed approach, designed for complex reasoning type of queries, surpasses their results. Unlike prior end-to-end approaches, we adopt a two staged approach, which not only provides us SOTA results, but allows for more interpretability. Along similar lines, (Methani et al., 2020; Kim et al., 2020) propose a multi-stage solution, where the chart extractions are stored in a semi-structured form, and pre-defined rule-based semantic parsing (Pasupat and Liang, 2015) converts the queries into a logical form. However, these approaches do not generalize to queries not expressible by the grammar rules defined for other datasets, leading to very low test accuracy especially for complex reasoning type of queries (14.82% for PlotQA). An elaborate listing of prior work on Table Q&A and Semantic Parsing and the comparison with our approach highlighting our novelty is in Appendix A.

3 Problem Definition

Our task is defined as follows: given a chart c and a question q on the chart, output a value a that answers the question according to the information represented in the chart. The system has access to a training set $D_{chart} = (q_{chart}, c_{chart}, a_{chart})_1^N$ of questions, charts, and answers. The charts and corresponding questions in test data do not appear during training. We assume availability of a bootstrapping dataset of $D_{tr} = (s_{tr}, q_{tr}, p_{tr})_1^M$, where s are the database schema, q are the Natural Language (NL) queries posed on the schema and p are the SQL programs corresponding to the NL queries. The domain of the charts c_{chart} can be distinct from the domain of the schema s_{tr} .

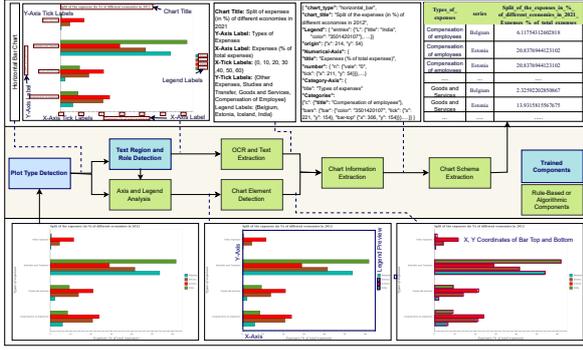


Figure 2: Chart Schema Extraction Vision Pipeline

We assume that the SQL programs p_{tr} share the primitive arithmetic, aggregation and logical operations (SUM, DIFFERENCE, RATIO, AVERAGE, MEDIAN, MAXIMUM, MINIMUM, GREATER THAN, LESS THAN) with the SQL programs required to answer the NL questions q_{chart} . However, distributions over the primitive operations and their compositions can differ.

4 Approach

We follow a two stage approach. In the first stage, we use a computer vision pipeline to extract chart and store the chart data in a format (schema) s_{chart} , common across all the chart types. In the second stage, we synthesize SQL programs for the NL questions in the dataset by using Filtered Iterative Back-Translation (FIBT). We use SQL as the target logical form for program synthesis because it’s grammar (i) is well suited to the tabular structure of our extracted data and (ii) includes the primitive operations required to be handled for the NL queries in PlotQA. Also, SQL is close to NL and easy to understand, allowing for interpretability.

4.1 Chart Schema Extraction

In the first stage, we extract chart information using the following pipeline: (i) Chart Type Detection (Trained) (ii) Text Region and Role Detection (Trained) (iii) OCR and Text Extraction (Algorithm) (iv) Axis and Legend Extraction (Rule-Based) (v) OCR and Text Extraction (Algorithm) (vi) Chart Information extraction (Algorithm).

Since the chart types have distinct visual features we fine-tune a Resnet-34 model pretrained on ImageNet to detect chart types, using chart type labels provided by PlotQA. The text present in the image is detected by employing the CRAFT model (Baek et al., 2019). However, CRAFT frequently

misses isolated characters and often yields partial detection of text regions. We propose an approach which corrects partially detected text, segments out the corrected text region and identifies text-role labels (such as chart title, legend labels, X/Y-axis, and X/Y-tick labels) for the text regions (Appendix B). We use Tesseract 4.0.0 (Smith, 2007) to extract text from the detected regions and tag them to the corresponding roles. The results of the chart element extractions and text role region extractions are described in Appendix F. We define rules to identify (i) the origin, axes and chart region from the detected chart lines by a line detection algorithm (Paliwal et al., 2021), (ii) location of the legend previews and their styles (color and pattern) using the detected legend-labels, and (iii) chart elements (bars, dots, lines) which are regions matching with each legend preview style. We extract a schema (table) from the above available chart information, by filtering noise (Appendix C) and extracting the data series elements (Appendix E).

Henceforth, we use the following nomenclature. For the horizontal-bar charts illustrated in Figure 1 (b), we refer to the X-axis as the Numerical-axis and Y-axis as the Categorical-axis. For the remaining chart types in Figure 1 viz. dot, vertical bar, and line chart, the nomenclature is reversed. We call each legend label as a series. Thus, the extracted chart information is in the form of a set-of tuples $\langle category_label, series_label, numerical_value \rangle$, with the schema (table) header being category axis label, series, and a string formed by concatenating the chart title with the numerical axis label. We store these tables s_{chart} in the SQLite3 database to facilitate the execution of synthesized SQL programs (Section 4.2) on the schema. As ‘Median’ is not an in-built aggregation operation for SQLite3, we define a stored procedure for the same.

4.2 SQL Program Synthesis

As the part of the second stage, we execute Filtered Iterative Back Translation (FIBT) (Algorithm 1), to train the neural semantic parser $M_{NL \rightarrow P}$, which is used to synthesize SQL programs for the reasoning questions in the PlotQA test set. The generated SQLs are executed on the test set chart schema s_{chart} , extracted from the corresponding chart image c_{chart} , to compute the final answer. This answer is compared with the ground truth answer a_{chart} to calculate the test accuracy.

4.2.1 Bootstrapping Data

We use SPIDER (Yu et al., 2018) augmented with a few (359) NL-SQL query pairs as the bootstrapping dataset D_{tr} to initialize the parameters of the forward $M_{NL \rightarrow P}$ and the backward $M_{P \rightarrow NL}$ models of FIBT. The augmented query pairs are defined to include the primitive operations (DIFFERENCE, RATIO, LESS THAN and MEDIAN), required by the PlotQA NL questions q_{chart} but missing in the SPIDER SQLs, leading to the bootstrapping data q_{tr} and the PlotQA questions q_{chart} sharing the same set-of primitive operations. Inclusion of such query pairs allows the models to learn these primitive operations (followed by their compositions in the subsequent FIBT iterations), which otherwise would not be possible. The query pairs are synthesized using templates. For example, the templates used to synthesize NL-SQL pairs for RATIO operation is: NL: "What is the ratio of the *numerical column name* having *categorical column* value a x to that of *categorical column* value of y ?", SQL:

```
"SELECT T1.numerical_column_name /
T2.numerical_column_name FROM
table_name T1, table_name T2 WHERE
T1.categorical_column_name =
'categorical_column_value_x' AND
T2.categorical_column_name =
'categorical_column_value_y' "
```

These queries are synthesized on a subset of SPIDER schema tables, whose structure match with our extracted chart schema. Thus in the above template, the 'numerical_column_name' is the name of a column of a table belonging to one of the SPIDER database schemas, having a numerical datatype. and 'categorical_column_name' is the column name of the same table, with text datatype with values x and y as its entries.

4.2.2 Probabilistic Context Free Grammar

We define Probabilistic Context Free Grammar (PCFG) depicted in Table 7 in the Appendix, as a subset of the SQL grammar to synthesize SQL programs: (i) to address possible compositions of primitive operations required for the PlotQA NL questions q_{chart} and (ii) executable on the schema s_{chart} . To synthesize SQL programs whose distribution match with the programs for the NL questions in the PlotQA, but are not covered by the current training data D_{tr} , we induce the probability (P_{inv}) for each of the production rules R in the

PCFG with the heuristics depicted in Equation 1.

$$P_{inv}(R) = \frac{Wt(R)}{\sum_{RHS(r)=RHS(R)} Wt(r)} \quad (1)$$

$$Wt(R) = \left(\frac{MAX_{RHS(r)=RHS(R)}(P(r))}{P(R)} \right)^{1-\alpha} \quad (2)$$

$P(R)$ gives the probability with which a rule R is triggered by the set-of SQL queries existing in the training data D_{tr} . $RHS(R)$ is the Right Hand Side of the production rule R. Thus, $RHS(r) = RHS(R)$ provides the set of all production rules which share the source node (RHS) with the rule R. α is the hyper-parameter controlling the skewedness of the distribution over the production rules. Lower the value more skewed is the distribution. For our experiments $\alpha = 0.8$.

Algorithm 1: FIBT

```
Input :  $D_{chart} = \{q_{chart}, s_{chart}, a_{chart}\}_1^N$ ,
        Defined PCFG
Output : Trained Semantic Parser  $M_{NL \rightarrow P}$ 
Initial Stage : Bootstrapping data
                 $D_{tr} = \{s_{tr}, q_{tr}, p_{tr}\}_1^M$ 
                 $q_{rep} = \text{sample}(\text{cluster}(\text{generalize}(q_{chart})))$  where  $q_{rep} \subset q_{chart}$ 
                 $D_{rep} = (q_{rep}, s_{rep}, a_{rep})_1^n, n \ll N$ 
                 $p_{filter} = \Phi$ 
1 while  $M_{NL \rightarrow P}$  and  $M_{P \rightarrow NL}$  have not converged
  do
2   Train  $M_{NL \rightarrow P}$  on  $D_{tr}$  // Forward Pass
3   Feed  $q_{rep}$  to  $M_{NL \rightarrow P}$  to generate  $p_{rep}$ 
4   Execute  $p_{rep}$  on schema  $s_{rep}$  to compute  $a_c$ 
5   if  $a_c == a_{rep}$  // Filter
6     then
7       Add  $(s_{rep}, q_{rep}, p_{rep})$  to  $D_{tr}$  Remove
           $(s_{rep}, q_{rep}, p_{rep})$  from  $D_{rep}$ 
8     end if
9     Train  $M_{P \rightarrow NL}$  on  $D_{tr}$  // Backward
        Pass
10    induce( PCFG ,  $(p_{tr} + p_{filter})$  ) (Equation 1)
11    Sample SQL  $p_{syn}$  on  $s_{chart}$  from PCFG.
12    Feed  $p_{syn}$  to  $M_{P \rightarrow NL}$  to generate  $q_{synth}$ 
13    filter_flag = 1
14    if  $\max\_sim(\text{generalize}(q_{synth}),$ 
         $\text{generalize}(q_{rep})) > \text{threshold}$ 
        // Filter1
15    then
16      if  $\text{look\_up}(q_{synth}, p_{synth})$  // Filter2
17      then
18        Add  $(s_{chart}, q_{synth}, p_{synth})$  to  $D_{tr}$ 
          // Augment
19        filter_flag = 0
20      end if
21    end if
22    if filter_flag == 1 then
23      Add  $p_{synth}$  to  $p_{filter}$ 
24    end if
25 end while
```

4.2.3 Sampling Representative Questions

As depicted in Table 2, PlotQA has ~ 16.6 Million reasoning based NL questions as the part of the training data. For a more compute efficient solution, we identify representative NL questions from PlotQA for training. We randomly sample 200K NL questions from the PlotQA training set and perform the *generalize* operation to replace schema specific information in each NL question with generalized tokens and to highlight its composition or structure. We replace the schema related entity (column headings) and values (column values) in the NL questions with more generic $\langle \text{entity} \rangle$ and $\langle \text{value} \rangle$ tags using substring matching. For example, the reasoning based question, depicted in Table 1, is modified to: ‘Is the difference between $\langle \text{entity_num} \rangle$ on $\langle \text{value_series} \rangle$ in $\langle \text{value_category} \rangle$ and $\langle \text{value_category} \rangle$ greater than the difference between $\langle \text{entity_num} \rangle$ on $\langle \text{value_series} \rangle$ in $\langle \text{value_category} \rangle$ and $\langle \text{value_category} \rangle$?’, where ‘the amount collected as tax’ being a sub-string of the numerical column name of the schema, gets replaced with the generic token $\langle \text{entity_num} \rangle$ and the values of the category column, viz. ‘2003’, ‘2007’ and the series column ‘goods’ and ‘exports’ get replaced with $\langle \text{value_category} \rangle$ and $\langle \text{value_series} \rangle$, respectively. We further get the representations of these generalized NL questions using sentence-BERT (Reimers and Gurevych, 2019) and *cluster* them via DBSCAN¹ with *cosine similarity* as the similarity metric. As DBSCAN allows us to cluster data without specifying the number of clusters in advance, with *minpoints* = 15 and $\epsilon = 0.25$, we get 345 clusters for the 200K generalized NL questions. We then randomly *sample* 200 questions from each cluster to get 69K representative generalized NL questions. We fetch the corresponding original NL questions q_{rep} for these 69K questions along with their corresponding schema c_{rep} and answers a_{rep} to form a dataset of representative queries D_{rep} . A similar sampling strategy is applied on the validation split. Table 2 illustrates the statistics of the representative dataset.

4.2.4 FIBT Forward Pass

We train the forward model $M_{NL \rightarrow P}$ with the training data D_{tr} by feeding the flattened schema,

¹<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html>

the table contents, the NL query with a separator token to the encoder and generate the SQL tokens at the output of the decoder in an auto-regressive fashion. The model is trained using cross entropy loss. We feed the NL queries q_{rep} from D_{rep} to $M_{NL \rightarrow P}$ to generate the corresponding SQL programs (p_{rep}). We execute these SQL programs on the corresponding extracted chart schema s_{rep} . The programs which do not execute to the ground truth denotations are filtered, and the training data D_{tr} is augmented by the remaining pairs.

4.2.5 FIBT Backward Pass

We train the backward model $M_{P \rightarrow NL}$ with the training data D_{tr} by feeding the flattened schema and the contents followed by the SQL program with a separator token to the encoder and generating the NL tokens at the output of the decoder in an auto-regressive fashion. We use p_{tr} in the training set D_{tr} , along with the SQL programs p_{filter} filtered in the prior iteration to *induce* the inverse probabilities of PCFG as explained in section 4.2.2. Here, the filtered programs are the ones whose equivalent NL questions do not match with PlotQA representative questions (explained later in this section). We sample SQL programs (p_{synth}) from the PCFG to be executed on s_{chart} and feed these synthesized SQL programs to the backward model $M_{P \rightarrow NL}$ to generate the corresponding NL questions q_{synth} . We (i) transform q_{synth} by using the *generalize* operation, explained in section 4.2.3, (ii) extract the representation for q_{synth} using sentence-BERT (Reimers and Gurevych, 2019) and (iii) compare them with the representations of generalized representative queries (q_{rep}) using *cosine similarity*. NL questions having their maximum similarity score (*max_sim*) below a *threshold* are filtered. The SQL programs corresponding to the filtered NL questions are added to p_{filter} , representing queries not matching the PlotQA questions. With this filtering, we still observe some synthetic questions with semantic noise, meaning the semantics of the NL questions q_{synth} and the corresponding SQL programs p_{synth} do not match. (Shen et al., 2019) uses phrases of NL questions to estimate the operator candidates in the corresponding programs and thus reduce the search space of the semantic parser. We use a similar technique of phrase-operator *look-up* to further filter the synthetic query pairs. Given a q_{synth}, p_{synth} pair the *look-up* operation returns ‘False’ if the pre-defined (set-of) phrase(s) in NL query (q_{synth})

NL Phrase		SQL Operator
ratio	↔	/
difference	↔	-
greater than	↔	>
less than	↔	<
total OR sum	↔	+ OR SUM
maximum OR highest	↔	MAX
minimum OR lowest	↔	MIN
average	↔	AVG
how many	↔	COUNT
median	↔	MEDIAN

Table 3: Mapping of NL Phrases and SQL Operators

do(es) not match with the (set-of) operator(s) in the corresponding SQL programs (p_{synth}) and returns ‘True’ otherwise. This matching is done following the pre-defined look-up dictionary with the phrase-operator mappings between the NL questions and the SQL programs (Table 3). This filtering helps to remove the semantically incorrect NL questions (q_{synth}), which have been generated by the backward model $M_{P \rightarrow NL}$ for the synthetic SQL programs (p_{synth}). With this two level filtering, the training data D_{tr} is augmented with the remaining synthetic tuples $\langle s_{chart}, q_{synth}, p_{synth} \rangle$ and is further used to train the models in the next iteration. These added synthetic queries are closer to the PlotQA queries and thus help in adapting the models to answer PlotQA questions. For every iteration, the above defined *threshold* for similarity based query filtering is automatically set to a value for which the KL-divergence between the operator distributions of (i) PlotQA questions q_{chart} and (ii) the synthetic questions getting augmented to the training set ($q_{synth} - q_{filter}$), after filtering with the phrase-operator mappings (Table 3) is minimum. This ensures that the augmented synthetic query pairs, after filtering with the *threshold*, are closer to the required PlotQA questions.

5 Results and Discussion

CodeT5 (Wang et al., 2021c) provides the best results on the Spider dataset ² in terms of execution accuracy. For chart QA, we are more interested in correctly computing the final answer (execution accuracy) than the intermediate logical form (exact match accuracy). Thus we choose CodeT5 based neural semantic parser as our forward model ($M_{NL \rightarrow SQL}$) and CodeT5 based code summarization model ³ as the backward model ($M_{SQL \rightarrow NL}$). The number of trainable param-

²Spider Leaderboard Dated: August 2022 <https://yalelily.github.io/spider>

³<https://huggingface.co/Salesforce/codet5-base-multi-sum>

ANS Type	Test Queries	Human	Plot QA	T5	Ours ES	Ours OS
YN	72,968	76.51	62.75	62.38	43.21	44.13
FV	566,655	59.97	7.95	2.41	63.91	67.70
OV	2,919,769	58.01	14.95	0.003	60.42	85.35
Total	3,559,392	58.70	14.82	1.17	60.44	84.49

Table 4: Results on PlotQA V2 Reasoning Queries (% Test Accuracy), ANS: Answer, YN: Yes/No, FV: Fixed Vocabulary, OV: Open Vocabulary, ES: Extracted Schema, OS: Oracle Schema

eters in the CodeT5 base model are 220M. We fine-tune the models with a batch size of 48 and a learning rate of 0.0001 and gradient accumulation step of 4 using the Adam optimizer on an NVIDIA Tesla V100 32GB GPU. The average run-time for training the model differs in each iteration as the number of training samples increase with the data we augment in each iteration. However, for inference on the PlotQA V2 Test Split, with the given hyper-parameter configuration, it takes ~ 360 hours. For sampling the representative questions and SQL queries from the dataset and PCFG respectively, we use the random seed of 7. The training details of the chart extraction modules are provided in Appendix D. We use test accuracy as the evaluation metric, where for numeric answers with floating point values, we consider an answer to be correct if it is within the 5% range of the correct answer as followed by (Methani et al., 2020).

Table 4 illustrates the results of the Q&A task over reasoning based queries in the PlotQA V2 test set (~ 3.56 M NL queries on ~ 33.6 K charts). Following the benchmark approaches (Methani et al., 2020; Masry et al., 2022), we report results on PlotQA V2 test set and not the validation set. As mentioned in (Methani et al., 2020) most human errors are due to numerical precision as it is difficult to visually identify the exact value from the chart even within a 5% margin. Our weakly supervised approach surpasses the baselines (PlotQA (Methani et al., 2020) and T5 (Masry et al., 2022)) by a large margin even exceeding the human baseline. We observe improvement in test accuracy results from 39.69% to 57.45% to 60.44% in the 1st, 2nd and 3rd iterations of FIBT, respectively. This demonstrates the utility of the FIBT approach with the filtering and augmentation mechanisms used for capturing the relevant query compositions.

As per the definition provided by the authors of PlotQA, the fixed vocabulary comprises of the set of top 1000 frequently occurring answer words. Our approach yields superior performance for fixed

vocabulary (FV) and open vocabulary (OV) answers. Both FV and OV answers are numerical. Prior approach of end-to-end model predicting the final answer directly (T5 (Masry et al., 2022)) and approach which address queries with distinct answer types distinctly (PlotQA(Methani et al., 2020)), learn to distinguish the queries having YES/NO (binary) type of answers from other queries. Once this is learnt, any random guess of YES/NO as an answer to these queries would lead to a performance of 50%. On the other hand, our approach trains a single model to generate SQL programs for all queries with distinct answer types. For some cases the generated SQL program for questions with binary answers does not yield a binary result. Thus, for our approach, the random accuracy for Yes/No queries is not 50%. Moreover, the SQL programs for questions with YES/NO answers are more complex as compared to the SQL programs which leads to numerical answers (FV and OV questions) in terms of entailed compositions of primitive operations involving nesting, leading to harder synthesis. Also, our approach does not yield good performance for some of these questions with binary answers as there is no explicit mapping between the phrases in the NL query and the primitive operations involved in the SQL program. For example, for the questions: ‘Do a majority of the years between 2013 and 2010 (inclusive) have a number of secure internet servers greater than 1.16?’ or ‘Do the payments made towards primary income monotonically increase over the years?’ or ‘Is the payments made towards goods and services strictly less than the payments made towards primary income over the years?’ , the model finds it harder to learn to map the abstract phrases ‘majority of’ or ‘monotonically increasing’ or ‘strictly less than’ to a composition of primitive operations in the corresponding SQL programs as this knowledge is not explicitly provided. These are the reasons that the performance of our approach for queries with YES/NO answer type is inferior as compared to the other reasoning queries.

ChartQA (Masry et al., 2022) provides results on the complete PlotQA V2 test split (~ 4.34 M questions) for all question types including structural, data retrieval and reasoning. The test accuracy of their best performing model (T5), trained on the complete PlotQA train set (~ 20.25 M questions), end-to-end is 56.22%. For fair comparison with ChartQA (Masry et al., 2022), we train the

T5 model in an end-to-end fashion (direct answer based supervision), with 69K representative questions (0.04%) of the PlotQA training set following the same input format as in ChartQA. We test the model on reasoning based questions in PlotQA V2 test data (~ 3.56 M) to obtain the results depicted in Table 4. The T5 model can address the yes/no type of binary answers but struggles on questions with numerical answers (FV and OV). Effectively, as discussed earlier, once T5 has learnt to identify questions having binary answers, a random guess would lead to 50% accuracy. For T5, low performance on non-binary reasoning questions is expected because the end-to-end training in general struggles to perform complex reasoning in the latent space, and this is compounded by the very small amount of training data used for fair comparison. On the other hand, better performance of FV and OV answer type questions, underscores the efficacy of our approach to better handle complex numerical reasoning questions. Moreover, as we generate SQLs for NL, our approach is more interpretable, allowing users to understand the reasoning steps to get the final answer.

Apart from PlotQA (Methani et al., 2020), CharQA (Masry et al., 2022) is the only other dataset available. After thorough analysis of this dataset, we observed that the dataset contains samples with incorrect ground truth labels, spurious questions and Gold data tables with incorrect information. The details of our analysis is provided in Appendix G. Hence, we have not used CharQA (Masry et al., 2022) for benchmarking our approach.

To understand the impact of errors from vision based chart schema extraction on the downstream reasoning task, we perform an ablation to calculate the test accuracy of the reasoning task using the schema constructed with oracle extractions provided by PlotQA. We observe a 24.05% lower test accuracy with the extracted schema as compared to the oracle. We further analyze the results based on operators involved in the query. We observe that our approach works well not only for reasoning questions involving one primitive operators, but also for more complex questions involving composition of numerical operators such as (i) ‘COUNT VALUES GREATER THAN AVERAGE’ (For example, *For how many years, is the payments made towards primary income greater than the average payments made towards primary income over all*

years ?) or (ii) ‘SUM GREATER THAN MAX’ (For example, *Is the sum of the payments made towards goods and services in 2008 and 2010 greater than the maximum payments made towards primary income across all years ?*) or (iii) DIFFERENCE GREATER THAN DIFFERENCE’ (For example, reasoning type of query mentioned in Table 1 for dot charts).

We further observe that our approach does not yield good results for NL questions involving nesting in the SQL program. For example, (i) queries computing a ‘DIFFERENCE’ between the ‘MAXIMUM’ and the ‘SECOND MAXIMUM’ values of the numerical column (For example, the reasoning type of query mentioned in Table 1 for (b) Horizontal bars). In the corresponding SQL program for such questions, to compute the ‘SECOND MAXIMUM’ value requires nesting, (ii) the queries which try to find if the ‘SUM’ of two numerical values is ‘GREATER THAN’ the ‘SUM’ of other two numerical values, for ‘EVERY’ value of a non-numerical column. These NL questions demand nested SQL programs which ensure the ‘SUM GREATER THAN SUM’ criteria is true for ‘ALL DISTINCT’ values of the non-numerical column. Table 7 shows that the defined PCFG allows synthesis of nested SQL programs. We observe that as such nested SQL programs are not present in the initial training set, the strategy of inducing inverse probabilities for the PCFG facilitates synthesis of nested SQL programs in the later iterations of FIBT. However, for most of such nested SQL programs the backward model fails to generate semantically meaningful NL questions. Such noisy synthetically generated NL questions are filtered by our filtration strategy in the backward pass leading to fewer nested query samples in the training data which in turn cause low test accuracy for such questions.

6 Conclusion

We present an approach for QA on charts which addresses complex reasoning based questions that require a combination of natural language understanding, fine-grained perception, and analytical reasoning. We employ a pretrained semantic parser and FIBT we generate SQL programs for the NL questions without any program annotations. Our novel PCFG based approach helps the model to adapt to the given dataset’s query compositions and domains, unseen in the bootstrapping data. As the future work we plan to extend our approach fur-

ther to handle complex questions requiring nesting by using a hierarchical or grammar based program search technique.

7 Limitations

The focus of this paper is on complex reasoning type of questions. Our approach is not designed for structural (Table 1) or visual types of questions, pertaining to attributes of the visual elements of the plot such as color, size, spatial location. These questions are not useful for real-life applications, which require analysis on chart data to draw meaningful conclusions. Our existing approach is not designed to address such questions as the extracted schema only captures the underlying data of the chart, and not the visual entities present therein.

To have a good kick-start for the FIBT pipeline, we assume that the bootstrap SPIDER data covers the primitive SQL clauses, operators and functions required for the questions in the target dataset and there is some minimal overlap between the compositions of the SQL queries in the bootstrap SPIDER data and SQL queries required for the natural language questions in the target dataset.

In the current phrase-operator based filtering strategy only limited phrases are manually designed for the mapping of NL phrases and SQL operators (Table 3). We plan to make this approach more robust by using paraphrases or semantically similar phrases to manually designed phrases for mapping.

8 Ethical Considerations

PlotQA charts contain only factual information which is openly available in public domain and is not (i) specific to any individual or (ii) offensive. The solution provided in the paper is agnostic to the domain of the data. Like any QA task, to avoid the risks involved in critical domains such as finance, healthcare or medicine, we would have to calibrate the model or need human intervention, such that the errors are not propagated to the downstream tasks.

References

Rishabh Agarwal, Chen Liang, Dale Schuurmans, and Mohammad Norouzi. 2019. [Learning to generalize from sparse and underspecified rewards](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 130–140. PMLR.

- Priyanka Agrawal, Ayushi Dalmia, Parag Jain, Abhishek Bansal, Ashish Mittal, and Karthik Sankaranarayanan. 2019. [Unified semantic parsing with weak supervision](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4801–4810, Florence, Italy. Association for Computational Linguistics.
- Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoon Yun, and Hwalsuk Lee. 2019. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374.
- Ritwick Chaudhry, Sumit Shekhar, Utkarsh Gupta, Pranav Maneriker, Prann Bansal, and Ajay Joshi. 2020. Leaf-qa: Locate, encode & attend for figure question answering. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3512–3521.
- Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR.
- DongHyun Choi, Myeong Cheol Shin, EungGyun Kim, and Dong Ryeol Shin. 2021. Ryansql: Recursively applying sketch-based slot fillings for complex text-to-sql in cross-domain databases. *Computational Linguistics*, 47(2):309–332.
- Pritha Ganguly, Nitesh Methani, Mitesh M Khapra, and Pratyush Kumar. 2020. A systematic evaluation of object detection networks for scientific plots. *arXiv preprint arXiv:2007.02240*.
- Michael Glass, Mustafa Canim, Alfio Gliozzo, Saneem Chemmengath, Vishwajeet Kumar, Rishav Chakravarti, Avi Sil, Feifei Pan, Samarth Bhadraraj, and Nicolas Rodolfo Fauceglia. 2021. Capturing row and column semantics in transformer based question answering over tables. *arXiv preprint arXiv:2104.08303*.
- Jiaqi Guo, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2021. Weakly supervised semantic parsing by learning from mistakes. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2603–2617.
- Tong Guo and Huilin Gao. 2019. [Using database rule for weak supervised text-to-sql generation](#).
- Yinuo Guo, Hualei Zhu, Zeqi Lin, Bei Chen, Jian-Guang Lou, and Dongmei Zhang. 2020. Revisiting iterative back-translation from the perspective of compositional generalization. *arXiv preprint arXiv:2012.04276*.
- Kaylin Hagopian, Qing Wang, Tengfei Ma, Yupeng Gao, and Lingfei Wu. 2019. Learning logical representations from natural languages with weak supervision and back-translation. In *Knowledge Representation & Reasoning Meets Machine Learning Workshop (KR2ML)*.
- Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisen-schlos. 2020. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.
- Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. [Iterative back-translation for neural machine translation](#). In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.
- Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. 2016. Dynamic filter networks. *Advances in neural information processing systems*, 29:667–675.
- Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656.
- Kushal Kafle, Robik Shrestha, Scott Cohen, Brian Price, and Christopher Kanan. 2020. Answering questions about data visualizations using efficient bimodal fusion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1498–1507.
- Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.
- Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. 2020. Answering questions about charts and generating visual explanations. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–13.
- Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, page 10015–10027, Red Hook, NY, USA. Curran Associates Inc.
- Qian Liu, Bei Chen, Jiaqi Guo, Zeqi Lin, and Jian-Guang Lou. 2021. Tapex: Table pre-training via learning a neural sql executor. *ArXiv*, abs/2107.07653.
- Ahmed Masry and Enamul Hoque. 2021. Integrating image data extraction and table parsing methods for chart question answering. In *Chart Question Answering Workshop, in conjunction with the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–5.
- Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

- Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1527–1536.
- Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2851–2864, Hong Kong, China. Association for Computational Linguistics.
- Shubham Paliwal, Arushi Jain, Monika Sharma, and Lovekesh Vig. 2021. Digitize-pid: Automatic digitization of piping and instrumentation diagrams. *CoRR*, abs/2109.03794.
- Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *ArXiv*, abs/1908.10084.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- Tao Shen, Xiubo Geng, Tao Qin, Guodong Long, Jing Jiang, and Daxin Jiang. 2019. Effective search of logical forms for weakly supervised knowledge-based question answering. *arXiv preprint arXiv:1909.02762*.
- Noah Siegel, Zachary Horvitz, Roie Levin, Santosh Divvala, and Ali Farhadi. 2016. Figureseer: Parsing result-figures in research papers. In *European Conference on Computer Vision*, pages 664–680. Springer.
- Hrituraj Singh and Sumit Shekhar. 2020. Stl-cqa: Structure-based transformers with localization and encoding for chart question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3275–3284.
- Ray Smith. 2007. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.
- Bailin Wang, Mirella Lapata, and Ivan Titov. 2021a. Learning from executions for semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2747–2759, Online. Association for Computational Linguistics.
- Bailin Wang, Ivan Titov, and Mirella Lapata. 2019. Learning semantic parsers from denotations with latent structured alignments and abstract programs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3774–3785, Hong Kong, China. Association for Computational Linguistics.
- Bailin Wang, Wenpeng Yin, Xi Victoria Lin, and Caiming Xiong. 2021b. Learning to synthesize data for semantic parsing. *arXiv preprint arXiv:2104.05827*.
- Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. 2021c. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. *arXiv preprint arXiv:2109.00859*.
- Tomer Wolfson, Daniel Deutch, and Jonathan Berant. 2021. Weakly supervised text-to-sql parsing through question decomposition.
- Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.
- Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: Grammar-augmented pre-training for table semantic parsing. *ArXiv*, abs/2009.13845.
- Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*.
- Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

Appendices

A Prior work on Table Q&A and Semantic Parsing

Current approaches for table QA use an end-to-end modeling approach for either: (i) directly generating the answer (TAPAS (Herzig et al., 2020)), (ii) generate a program which produces the answer

upon execution of the generated SQL (TABERT (Yin et al., 2020) and RCI (Glass et al., 2021)) or (iii) using a Language Model pre-training strategy (neural query execution engine in TAPEX (Liu et al., 2021) using synthetic SQL programs or semantic parser trained on synthetic NL-SQL pairs generated using a Synchronous Context Free Grammar (SCFG) in GraPPa (Yu et al., 2021). These approaches are mainly designed for handling data retrieval or simple aggregation type of queries defined in WikiSQL (Zhong et al., 2017) or Wiki Table (Pasupat and Liang, 2015) datasets. The complex reasoning type queries which are part of the PlotQA dataset involve nested arithmetic operations with self-joins as well as nesting in the conditional (WHERE) clauses. (Yin et al., 2020) applied their approach on the benchmark Spider Text-to-SQL dataset (Yu et al., 2018) but their results have been eclipsed by RYANSQL (Choi et al., 2021), whose decoder is specifically designed to address nested complex queries. GraPPa (Yu et al., 2021) achieves state of the art performance on the complex spider dataset after fine-tuning with spider program (SQL) annotations.

In our scenario there are no SQL program annotations for PlotQA queries, and only the output denotations are available. Since annotating a sufficiently large number of SQL programs is a resource heavy task, semantic parsers can be trained with execution output denotations under a weak-supervision setting. These execution denotations can be used to model a reward signal in order to train the underlying semantic parser (Zhong et al., 2017; Liang et al., 2018; Hagopian et al., 2019; Agrawal et al., 2019; Agarwal et al., 2019). They can also be used to train the semantic parser with a log Multiple log Marginal Likelihood (MML) objective by using a limited number of SQL programs as latent logical forms (Wang et al., 2019; Min et al., 2019; Wang et al., 2021a). While synthesizing SQL programs, the denotations can also be used to filter candidate programs with rule-based synthesis systems (Guo and Gao, 2019; Guo et al., 2021; Wolfson et al., 2021). The reward-based approaches face issues with a large program search space and possible spurious programs. Unlike the MML approaches, we do not use any gold annotated SQL programs from the dataset under consideration (PlotQA). Most of the above weak-supervision approaches do not explicitly handle modeling or synthesizing the specific complex program compositions pertinent to

the dataset under consideration. On the other hand, our approach uses a publicly available semantic parsing dataset (Spider) as the bootstrapping data to initialize the parameters of the semantic parser, and then defines a novel PCFG-based strategy to adapt the models through FIBT to answer unseen complex reasoning queries in the PlotQA by effectively capturing the relevant query compositions.

B Text Role Region Extraction

The architecture consists a *Encoder-Decoder Module*, which the U-net (Ronneberger et al., 2015). The encoder filters out irrelevant information by squeezing the feature map to a latent space. The output of the decoder is appended along the channel dimension with the trigger patch of a text region detected by CRAFT, with highlighted patch contours and provided as an input to the *Trigger-Controller Module*, extracts features using a convolutional feature extractor followed by a Global Average Pooling (GAP) layer. The features of the trigger patch are concatenated with the extracted encoder output features and are fed to the controller module to generate dynamic kernels, which are used to generate the segmentation map (Jia et al., 2016). The dynamic kernel output is also given to a fully connected linear layer to determine the text role of the region. Thus, the trigger-controller module exploits the spatial relationships between text-roles to generate dynamic kernels and obtain text-role specific segmentation maps from the decoded image. The whole network is trained with cross entropy loss on text-role class labels and text-role segmentation map (all ground truth text-regions corresponding to that text-role). During inference, given a trigger patch for an image of a detected text-region belonging to an unknown text-role, the model provides the actual text-role classification output and the segmentation map of the text-region of that text-role. Trigger patches overlapping with detected text-role regions are removed before repeating the process for the remaining trigger patches. This process may lead to multiple segmentation maps for each text-role, over which a union operation is performed.

C Noise Correction

To handle false positive and false negative detection of Numerical-axis ticks, we find the mode (M) of differences between the (X/Y) coordinates for the consecutive ticks. We remove or add ticks where

IOU	@0.90											@0.75	@0.50
	Bar	Dot-line	Leg Lbl	Leg PV	Title	X-Lbl	X-Ticks	Y-Lbl	Y-Ticks	mAP	mAP	mAP	
Existing Models (Ganguly et al., 2020)													
FiCNN (FPN+RA)	87.59	31.62	79.05	66.39	0.22	69.78	88.29	46.63	84.60	61.57	69.82	72.18	
FiCNN (RA)	63.86	14.79	70.95	60.61	0.18	83.89	60.76	93.47	50.87	55.49	89.14	96.80	
FiCNN (FPN+RA)	85.54	27.86	93.68	96.30	0.22	99.09	96.04	99.46	96.80	77.22	94.58	97.76	
PlotNet	92.80	70.11	98.47	96.33	99.52	97.31	94.29	97.66	94.48	93.44	97.93	98.32	
Ours (Train: All)	89.67	69.13	99.89	98.67	99.99	99.90	99.45	99.89	97.69	95.80	96.70		
Ours (Train: 4K)	89.67	69.13	96.31	96.31	99.63	96.35	96.84	99.48	96.58	92.86	93.66	94.50	

Table 5: Chart Extractions on the PlotQA dataset with mAP scores (in %). Leg: Legend, Lbl: Labels, PV: Preview

Method	PlotQA	Ours
Title	94.60	99.69
X-axis Label	95.50	99.73
Y-axis Label	97.07	99.59
Legend Labels	91.11	98.13
X-tick Labels	91.38	97.62
Y-tick Labels	88.07	95.94

Table 6: OCR Module Accuracy

the difference in the consecutive ticks is more or less than the mode, respectively. We add a dummy value ‘x’ for the newly added tick, if any, which is handled during correction. We replace a non-numerical tick-value detection, if any, by using M as an offset to the the neighboring tick value. To correct the tick-values not adhering to a progression followed by the majority values, we consider a tick-value as an ‘anchor’ (correct value) and calculate the other values adding and/or subtracting M from this anchor. We compute the ‘gain’ with respect to this anchor to be the intersection of the extracted values and calculated values. We repeat this process by considering distinct numerical-axis values as anchors. For the ‘anchor’ giving us the maximum ‘gain’, the corresponding calculated values are considered to be the correct set of numerical-axis tick values. Some charts (Figure 1 (a)) use scientific notation for denoting large numerical values which are converted to float values.

D Training Details for Chart Extraction Models

For the chart type classification ResNet-34 is fine-tuned for 2000 steps. We use the Adam optimizer, a learning rate of 0.0005, and a batch size of 8. The model yields 99.91% test accuracy. For text region and role detection, we use the pre-trained VGG19 and train with a batch size of 8, for 1 epoch, using the Adam optimizer with an initial learning rate of 0.0005. As the data is skewed for axes-labels, while creating training tuples, we under-sample for this text-role to avoid class imbalance.

E Data Extraction

We use interpolation and extrapolation to calculate a numerical value associated with every pixel on the numerical-axis by using numerical-axis tick coordinate pixels as reference. For every data point (pivot-point in case of dots and line charts, bar-tops in case of bars) detected we assign it: (i) a series by matching its style with the style of a legend label, (ii) a category by matching its co-

ordinate with the category tick, and (iii) a value of the pixel on the numerical axis whose coordinate matches with it. Thus, we extract the data in the form of a set-of tuples $\langle category_label, series_label, numerical_value \rangle$. For the category for which no pivot point or bar is detected for a series due to VED errors, we consider its value to be zero. We finally define a tabular schema with the column names as category axis label (category column), series (series column), and a string formed by concatenating the chart title with the numerical axis label (numerical column). All the spaces in the column names are replaced by underscores to make them SQL compatible. We insert the extracted tuples as rows in the tabular schema. The generic schema obtained for the chart in Figure 1(b) is shown in Figure 2. Charts containing only one series have a schema with only category and numerical columns.

Improvements over FIBT Iterations The generated SQL programs for the NL queries requiring certain compositions of primitive operations got corrected after the iterations of back-translation. For example, the required composition of the NL query ‘In how many years, is the amount spent in making social contributions greater than the average amount spent in making social contributions taken over all years?’ is ‘Count Greater Than Average’. After the first iteration the generated SQL program for this query is “Select t1.social_contributions > t2.social_contributions from table_data t1, table_data t2 where t1.year = ‘2010’ and t2.year = ‘2010’ “, which got corrected to “select count(*) from table_data where social_contributions > (select avg (social_contributions) from table_data)”, after the second iteration. This improvement is because of the data augmented by PCFG-based synthetic queries, which contained these SQL queries having compositions not present in the original bootstrapping data (details in Section 4.3). Some additional improvements over the iterations are because of the coverage of the additional (chart) domains, which

sql	→	sel_num_col sel_col sel_arth
sel_num_col	→	"SELECT" agg "(" "num_col_name" ")" from "WHERE" cond_series cond_cat
sel_col	→	sel_cat_col sel_series_col "WHERE" cond_num ("AND" cond_series cond_cat) ⁰⁻¹
sel_col	→	sel_cat_col sel_series_col "ORDER" "BY" "num_col_name" "DESC" "ASC" "LIMIT" "1"
sel_cat_col	→	"SELECT" "DISTINCT" ("COUNT") ⁰⁻¹ "(" "cat_col_name" ")" from
sel_series_col	→	"SELECT" "DISTINCT"("COUNT") ⁰⁻¹ "(" "series_col_name" ")" from
sel_arth	→	"SELECT" agg "(" "num_col_name" ")" arth agg "(" "num_col_name" ")" from ("WHERE" cond_series cond_cat) ⁰⁻¹
sel_arth	→	"SELECT" "DISTINCT" "(" "t1." "num_col_name" arth "t2." "num_col_name" ")" from ₂ "WHERE" "t1." cond_cat cond_series "AND" "t2." cond_cat cond_series ("AND" "t1." cond_series cond_cat "AND" "t2." cond_series cond_cat) ⁰⁻¹
sel_arth	→	"SELECT" "DISTINCT" "(" "t1." "num_col_name" arth "t2." "num_col_name" ")" arth agg "(" "t3." "num_col_name" ")" from ₂ ", " table_data "t3" "WHERE" "t1." cond_cat "AND" "t2." cond_cat ("AND" "t1." cond_series "AND" "t2." cond_series "AND" "t3." cond_series) ⁰⁻¹
sel_arth	→	"SELECT" "DISTINCT" "(" "t1." "num_col_name" arth "t2." "num_col_name" ")" arth "(" "t3." "num_col_name" arth "t4." "num_col_name" ")" from ₂ from ₄ "WHERE" "t1." cond_series cond_cat "AND" "t2." cond_series cond_cat "AND" "t3." cond_series cond_cat "AND" "t4." cond_series cond_cat
from	→	"FROM" "table_data"
from ₂	→	"FROM" "table_data" "AS" "t1" ", " "table_data" "AS" "t2"
from ₂	→	"FROM" "table_data" "AS" "t1" "JOIN" "table_data" "AS" "t2" "ON" "t1." "cat_col_name" "series" "=" "t2." "cat_col_name" "series"
from ₄	→	", " "table_data" "t3" ", " "table_data" "t4"
cond_num	→	"num_col_name" op "num_val" op "(" sel_num_col ")" "NOT" "IN" (" sel_num_col ")
cond_series	→	"series" "=" "" "series_col_val" "" "cat_col_name" "IN" "(" sel_cat_col ")"
cond_cat	→	"cat_col_name" "=" "" "cat_col_val" "" "series" "IN" "(" sel_series_col ")"
agg	→	"SUM" "MIN" "MAX" "AVG" "MEDIAN"
arth	→	"<" ">" "/" "-" "+"
op	→	"=" "<" ">"

Table 7: Probabilistic Context Free Grammar (PCFG)

are not present in the initial bootstrapping data (having queries covering the SPIDER database domains), but got covered in the PCFG based synthetically generated queries on the PlotQA chart schema, addressing the domain shift.

F Results of Plot Extraction

For plot extractions we prefer mAP @0.90 IOU over mAP @0.75 and mAP@0.50 IOU as the evaluation metric as we require precise fine-grained extractions else the resulting data errors will propagate to the downstream QA task. For OCR we use accuracy (1 - Word Error Rate (WER)) as the evaluation metric. Table 5 illustrates the SOA results on plot extraction by our approach. We have state-of-the-art results with for Chart Extraction yielding 94.92% mAP @0.90 IOU when trained with all PlotQA (157K) images, beating the baseline (Ganguly et al., 2020) by 1.48%. mAP @0.90 IOU. The extraction of dot/line regions is challenging because of their small size, sparse distribution and

eclipsed or intersected dots/lines of distinct series. Table 6 depicts the SOA results of the OCR module. The PlotQA Oracle refers to the results with the OCR model applied to the ground truth text-regions. Our predicted text-detections followed by OCR outperform both the baselines from PlotQA (Methani et al., 2020), yielding State-of-the-Art results.

G ChartQA Dataset Analysis

After thorough analysis of CharQA (Masry et al., 2022) dataset, because of the following observations we have not used it for benchmarking: (i) Samples with incorrect Ground Truth (GT) labels: Few examples: (a) for the question ‘*In what year did Nicaragua have the highest risk score of money laundering and terrorist financing?*’ on the chart ‘two_col_102453.png’, the actual answer is ‘2020’, and the provided GT is ‘2015’, (b) for the question ‘*What’s the ratio of the lowest value of green bars and blue bars?*’ on the chart ‘1392.png’, the

actual answer is '2.41' and the GT is '1.216', (c) For the question 'In republicans what is the difference between the more likely and less likely?' on the chart '9987.png' the ground truth label is '21', where as based on the question the ground truth label should be '-21'. (ii) Samples with spurious questions: The contents of the question are not relevant to the data present in the chart. Few examples: (a) For the question '*What was the third most popular brand on Foursquare?*' on the chart 'two_col_80744.png', 'Foursquare' is itself an individual brand depicted in the chart and no other information about Foursquare is provided. and the GT is 'MTV' which in itself is a separate brand and not related to Foursquare. (b) the question '*How many people checked in to New Delhi on Facebook between June and August 2017?*' posed on chart 'two_col_5556.png' has some terms such as Facebook or specified Month/Year which are not present in the chart, (iii) Gold data tables⁴ with incorrect information: Gold data tables crawled through web source 'PEW'⁵ have incorrect information, which leads to predicted answers. In the test set 310 (20.54%) samples have all the data values to be zeros and others ('4931.png', '2721.png', '11627839005738.png') have floating point errors. For example, the Gold data tables for charts '4931.png', '2721.png', '11627839005738.png' are spurious. Due to this incorrect Gold Table contents even though the GT labels corresponding to the question to these charts is correct, with incorrect data points, the resulting answer does not match the label. Due to the above listed observations, we have not used this dataset for the benchmarking purpose.

⁴https://drive.google.com/file/d/17-aqtiq_KJ16PIGOp30W0y6OJNax6SVT/view

⁵<https://github.com/vis-nlp/ChartQA/issues/8>

Exploiting Language Characteristics for Legal Domain-Specific Language Model Pretraining

Inderjeet Nair and Natwar Modani

Adobe Research, India

{inair, nmodani}@adobe.com

Abstract

Pretraining large language models has resulted in tremendous performance improvement for many natural language processing (NLP) tasks. While for non-domain specific tasks, such models can be used directly, a common strategy to achieve better performance for specific domains involves pretraining these language models over domain specific data using objectives like Masked Language Modelling (MLM), Autoregressive Language Modelling, etc. While such pretraining addresses the change in vocabulary and style of language for the domain, it is otherwise a domain agnostic approach. In this work, we investigate the effect of incorporating pretraining objectives that explicitly tries to exploit the domain specific language characteristics in addition to such MLM based pretraining. Particularly, we examine two distinct characteristics associated with the legal domain and propose pretraining objectives modelling these characteristics. The proposed objectives target improvement of token-level feature representation, as well as aim to incorporate sentence level semantics. We demonstrate superiority in the performance of the models pretrained using our objectives against those trained using domain-agnostic objectives over several legal downstream tasks.

1 Introduction

Pre-trained language models exhibit superior performance in several NLP tasks. Most of the prominent language models optimized over Masked language modelling with BERT-like (Devlin et al., 2019; Liu et al., 2019b; He et al., 2020) architecture using large unlabelled corpus to achieve state of the art results across many NLP tasks. While the sentence-level tasks like paraphrase detection (El Desouki and Gomaa, 2019) and sentiment analysis (Zhang et al., 2018) benchmarks the capability of the model in effectively modeling the holistic representation of the sentence(s), the token-level tasks like named entity recognition (Li et al.,

2020) attempted to assess the quality of contextualized token embeddings furnished by the models. However, direct application of these models to domain-specific downstream tasks yields sub-optimal performance (Lee et al., 2020), perhaps due to change in vocabulary and style of language.

To overcome this limitation, most commonly used approach involves pre-training a language model over domain specific corpora. For instance, PubMedBERT (Gu et al., 2021) and LEGALBERT (Chalkidis et al., 2020) achieved state-of-the-art results for the biomedical and legal domain specific tasks respectively by pre-training over domain specific corpus using a domain agnostic objective. In this paper, we argue that the performance of these models can be further improved by employing pre-training objectives that exploit the language characteristics of the domain. We examine two distinct language characteristics of the legal domain, propose pre-training objectives and finally demonstrate superior performance over domain-specific NLP tasks. Legal domain departs from the generic corpora in terms of specialized vocabulary, particularly formal syntax, domain-specific knowledge semantics, etc. to the extent that it can be classified as a distinct "sub-language" (Tiersma, 1999; Williams, 2007), which may be addressed by MLM based pretraining. In this paper, we study the following additional domain characteristics and formulate closely aligned objectives in addition to domain agnostic objectives like MLM:

1. **Templatized language:** Legal documents consist of clauses that are often derived from reusable text fragments with placeholders. The placeholders are substituted with appropriate replacements for specific documents. We include a pre-training objective for this characteristic by optimizing the model to distinguish the substitutions from the rest of the text. Since, there is no labelled dataset that provides such information, we also outline the

process to approximately label the data-points with placeholder spans.

2. **Availability of Soft Labels:** Contracts and legally enforceable documents can be segmented into clauses which are sections defining terms and conditions and important provisions. Clauses can be categorized into distinct types based on the aspect they address, which (the categorization) may sometimes be available as a heading/title associated with the clauses. This categorization enables us to define semantic relations between clauses. For instance, clauses having same type are closer in meaning as compared to different typed clauses. This fact is instrumental in formulating an objective to obtain semantically aware holistic representation.

We leverage these two characteristics to design a pre-training strategy, and experimentally show that a language model pretrained using our strategy outperforms domain-specific language model which is trained only on domain-agnostic objectives, such as Masked Language Modelling.

The rest of the paper is organized as follows. In Section 2, we discuss some prominent frameworks that provisions domain specific pre-trained models and survey important works in the legal AI. In Section 3, we elucidate the details for the aforementioned legal domain characteristics and describe the objective formulation and dataset curation strategy. In Section 4 discusses the training. In Section 5, we briefly describe the baseline models used to compare the performance with our pre-trained model for several legal domain tasks, and discuss the results. Finally, in Section 6, we conclude by explaining the implications of our work and discuss its natural extensions.

2 Related Works

2.1 Prominent domain adaptation pretraining approaches:

Pretrained language models trained over non-domain specific data such as transformers (Vaswani et al., 2017), BERT (Devlin et al., 2019) and its variants (Liu et al., 2019b; He et al., 2020) has resulted in state-of-the-arts results for several non-domain specific natural language processing downstream tasks. Owing to their success, a prominent approach to achieve superior results in domain-specific NLP tasks involves training these models

over domain-specific corpora. For instance, to improve the performance of the models in biomedical downstream tasks, BIOBERT (Lee et al., 2020), Clinical BERT (Alsentzer et al., 2019), Clinical BIOBERT (Alsentzer et al., 2019) and PubMedBERT (Gu et al., 2021) were pretrained over speciality corpora closely associated with the biomedical domain using the MLM objective. Recently, (Chalkidis et al., 2020) proposed LEGAL-BERT, a language model pretrained using MLM over domain specific corpora, to achieve state-of-the-art performance for several legal downstream tasks. Most of these methods focus on choosing appropriate corpora for MLM pretraining and the selection of optimal hyperparameters in contrast to the approach taken in this work. Here, we propose a new direction to adapt a pretrained language model by utilizing language characteristics. In particular, by studying the language characteristics of the legal domain, we propose pretraining objectives that explicitly tries to learn these characteristics. While there are other approaches that adapts the language model to domain-specific tasks (Rietzler et al., 2020; Han and Eisenstein, 2019; Gururangan et al., 2020), our work mainly tries to address the problem of pretraining a language model for a particular domain.

2.2 Legal Artificial Intelligence (AI)

Legal AI refers to the application of AI/NLP techniques to solve several tasks in the legal domain (Zhong et al., 2020). Due to the distinct language characteristics of the legal domain, many legal domain-specific tasks requires the expertise of legal practitioners for solving them. Furthermore, the complexity of the associated tasks requires a significant time commitment even for experienced legal professionals. Thus, this motivated the development of legal AI to reduce the tedium in understanding and solving these legal tasks.

In the legal AI, task-specific methods and datasets were proposed for the following tasks: Legal Judgement Prediction (Aletras et al., 2016), Legal Entity Recognition and Classification (Cardellino et al., 2017), Legal Question Answering (Kim and Goebel, 2017), Automated Legal Review (Hendrycks et al., 2021), Legal Text Classification (Chalkidis et al., 2021), etc. Instead on improving task-specific solution approaches, our objective is to make improvements for several downstream tasks. The objective of this work in very

similar to that of (Chalkidis et al., 2020), however, our solution approach is very different.

3 Domain Specific Objectives

We now describe the legal domain characteristics which we will use for formulating the objectives. For each of the two objectives, we also describe the associated dataset used for training. We get different pretrained language model variants by incorporating various subsets of the following objectives while pretraining.

The process of coming up with the right set of domain specific language characteristics requires significant exposure to the domain. The authors have been investigating several legal domain natural language processing tasks, and have been interviewing several practitioners for an extended period of time. The insights are a result of reading many legal domain documents and the interactions with domain experts. For one to extend our ideas in other domains, we expect them to require similar long exposure to the domain in question and opportunities to interact with domain experts. While we believe that the two characteristics identified in this work are not unique only to the legal domain, one will need to carefully evaluate whether the same characteristics apply to their chosen domain as well.

3.1 Legal Domain as a Templating Language

Contracts include clauses which often use a standardized language with some placeholders which are substituted with appropriate values (e.g., names, dates, amounts, locations, etc) for specific contracts (Figure 1). These standardized fragments with placeholders are referred to as templates (Niemeyer and Knudsen, 2005) in software engineering parlance. We refer to the tokens in the template-generated clauses that remain common across contract documents as **static** tokens and the values filled into the placeholders as **dynamic** tokens.

We propose a pre-training objective that aims to detect the **dynamic** tokens/spans from text fragments in the legal documents. Using this objective, the language model can generate holistic representation for a text-fragment cognizant of the tokens forming the dynamic part and the tokens forming the static part. This can also result in better contextualized token representation for the task of named-entity recognition or other entity level tasks.

In these Terms the following words shall have the following meanings:
"Goods" means those goods, products and/or services to be supplied and delivered by Vendor to Purchaser as described in the relevant Order.
"Purchaser" The person, company, firm, partnership or such other legal entity that places an order for Goods with Vendor and includes Purchaser's divisions, subsidiaries and affiliates.
"Vendor" means **Russel Metals Inc.** and its divisions, subsidiaries and affiliates.

In these Terms the following words shall have the following meanings:
"Goods" means those goods, products and/or services to be supplied and delivered by Vendor to Purchaser as described in the relevant Order.
"Purchaser" The person, company, firm, partnership or such other legal entity that places an order for Goods with Vendor and includes Purchaser's divisions, subsidiaries and affiliates.
"Vendor" means **AJ Forsyth** and its divisions, subsidiaries and affiliates.

Figure 1: Clauses generated from same template: The above example is believed to be generated from a standardized clause template with a placeholder in place of the text in yellow highlight. Moreover the substituted text is observed to have close correspondence with organization named-entity.

3.1.1 Dataset

One of the challenges in utilizing this characteristic in the pre-training objective is the lack of any labelled dataset with such kind of information. To overcome this limitation, we propose a dataset curation strategy that provides data points with dynamic spans. The corpus to be labelled was formed by collecting all the clauses present in the LEDGAR dataset (Tuggener et al., 2020), which consists of over 700,000 provisions in contracts.

The data curation strategy mainly consists of two steps: a) **Grouping** clauses that have very high lexical similarity which are believed to be generated from a single underlying template, b) **Contrasting** data points in a pairwise fashion for every group to differentiate the dynamic part from the static using *google-diff-match-patch*¹. Figure 3 illustrates the pipeline employed for annotating the dataset. Note that, while the contrasting tokens belong to *the dynamic part of the underlying text* (if the grouping was correct), there is inconclusive evidence for the rest of the tokens for considering them as *static* (For instance, Fig 2). This is due to the fact that some values can coincidentally be same for different instances of same clause, e.g.,

¹<https://github.com/google/diff-match-patch>

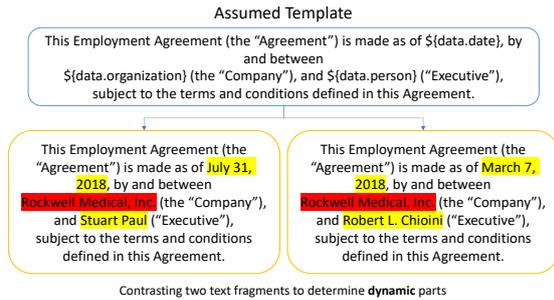


Figure 2: Limitation of the contrasting step: The two text fragments below belong to the same cluster and are believed to be generated from the template shown in the left. However, the process of contrasting annotates only some of the dynamic parts (highlighted in yellow) and misses out some (highlighted in red). Thus, the rest of the text should not be regarded as static in its entirety.

hiring date for two individuals can be the same, and therefore would not be marked as dynamic token by this strategy.

3.1.2 Objective Formulation

After applying the labelling strategy explained in the previous section, we obtain a token-wise labelled dataset $\mathcal{L} = \{(X_i, Y_i)\}_{i=1}^M$. A datapoint in \mathcal{L} is a tuple (X_i, Y_i) , where X_i represents a text-fragment as a sequence of tokens it contains ($X_i = [x_{ik}]_{k=1}^{|X_i|}$) and Y_i is the corresponding sequence of binary labels assigned to each token in X_i in the same order ($Y_i = [y_{ik}]_{k=1}^{|Y_i|}$ where $y_{ik} \in \{0, 1\}$), i.e. $y_{ik} = 1$ implies that x_{ik} belongs to the dynamic part and $y_{ik} = 0$ implies that the corresponding token can belong to any part.

Given such labelled dataset, we wish to train the language model \mathcal{M} such that $\mathcal{M}_{\text{dyn}}(X_i, x_{ik})$ provides us with the likelihood of x_{ik} being dynamic. The subscript ‘dyn’ denotes the addition of task-specific overhead architecture for detecting **dynamic** spans. We cannot directly apply binary cross entropy objective over the token-level predictions as the negative labels in our case does not imply that the corresponding tokens *are* static. To overcome this obstacle, we use the framework of positive-unlabelled (PU) (Peng et al., 2019) learning where all the tokens associated with a positive label are regarded as dynamic and rest, associated with a negative label, are regarded as unlabelled. Under this framework, all the positively labelled tokens are collected with their parent text-fragment to form the set $\mathcal{X}_p = \{(X_i, x_i)\}_{i=1}^{n_p}$ where x_i represents a positively labelled token present in the text fragment X_i . This is also repeated for the neg-

atively labelled datapoints to form the unlabelled set $\mathcal{X}_u = \{(X_i, x_i)\}_{i=1}^{n_u}$. PU learning optimizes the model parameters for the detection of dynamic parts by minimizing the following objective:

$$\begin{aligned} \mathcal{L}_{PU}(\mathcal{M}_{\text{dyn}}, \mathcal{X}_p, \mathcal{X}_u) = & \frac{1}{n_u} \sum_{(X_u, x_u) \in \mathcal{X}_u} l(\mathcal{M}_{\text{dyn}}(X_u, x_u), 0) \\ & + \frac{\pi_p}{n_p} \sum_{(X_p, x_p) \in \mathcal{X}_p} (l(\mathcal{M}_{\text{dyn}}(X_p, x_p), 1) - l(\mathcal{M}_{\text{dyn}}(X_p, x_p), 0)) \end{aligned} \quad (1)$$

where l is a positive-valued loss function that penalizes the distance between its arguments and $\pi_p \in [0, 1]$ is a hyperparameter. The above objective is derived from the following two terms: a term that incentivizes positively labeled instances to be classified as dynamic and a term that penalizes the unlabeled instances based on the assumption that the probability of being dynamic is equal to π_p Peng et al.. This implicitly assumes that the positive and unlabeled datapoints are sampled from the same distribution and the probability of a positive datapoint being labeled is independent of its input features. In contrast to the binary cross entropy objective, PU learning accounts for the possibility that some of the elements of \mathcal{X}_u can be *dynamic*.

3.2 Soft Semantic Labels for Clauses

The legal essence of many contractual documents and agreements is formed by concatenating clauses which are crucial for defining terms and conditions and important provisions. These clauses can often be categorized which can be used to optimize the model to provide semantic-aware representation scheme, and sometimes, such categorization is available as a label/title with the clause text. Formally, we want to train the language model to learn a representation scheme that maps same category clauses from the data manifold onto metrically closer points in the mapped space. We believe that by infusing the ability to generate semantic-aware representation within model, the language model may offer better performance on sentence-level tasks.

3.2.1 Dataset

We used the LEDGAR Corpus (Tuggener et al., 2020) which is a collection of labelled legal clauses and provisions. This corpus was crawled from the contracts present in the website of U.S. Securities and Exchange Commission (SEC)². While this dataset contains many clause instances with multiple labels, we retain only those clauses from this

²<https://www.sec.gov/>

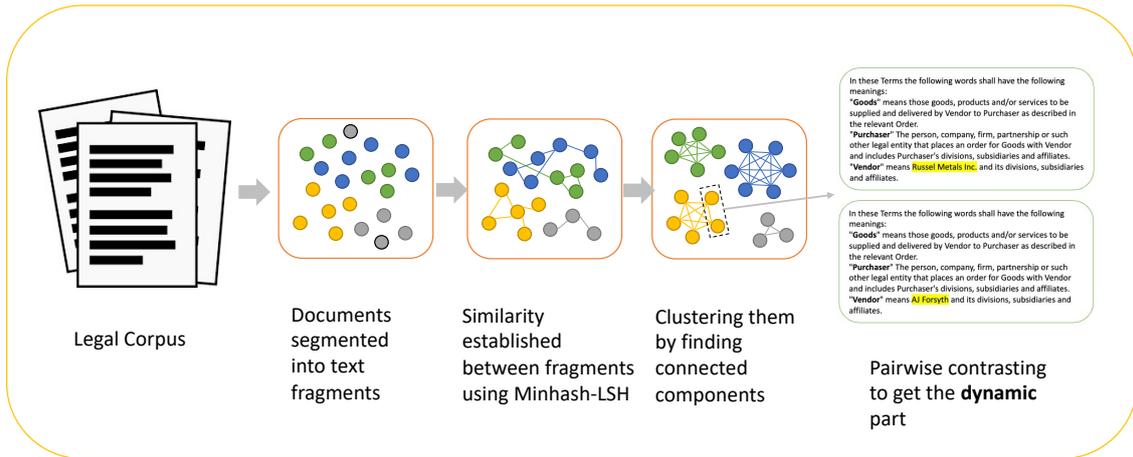


Figure 3: **Pipeline for dataset creation for dynamic part identification:** The clauses extracted from the LEDGAR corpus were originally obtained by segmenting legal documents into fragments. As clauses having fairly repetitive lexical structure are believed to be generated from the same template, the fragments are clustered using Minhash-LSH (Broder, 1997; Indyk et al., 1997), followed by finding the connected components. Finally, each pair in a cluster is contrasted to annotate what is dynamic among them.

corpus which are associated with a single label (roughly 83% of the dataset).

3.2.2 Objective Formulation

Given a language model \mathcal{M} , \mathcal{M}_{rep} denotes task specific adaptation of the original language model to generate representation for a given sentence. We formulate our requirement as a task of metric learning where the goal is to learn a function $\mathcal{M}_{\text{rep}}(\cdot) : \mathcal{X} \rightarrow \mathbb{R}^d$ that maps semantically closer input datapoints onto metrically closer points in \mathbb{R}^d . Here, \mathcal{X} denotes the domain of input clauses / provisions. Under the triplet-loss formulation, every instance in the training dataset is a triplet (x_a, x_p, x_n) where the model tries to make the distance between the representations of x_a (*anchor*) and x_p (*positive*) smaller than that between x_a and x_n (*negative*) by atleast a margin m . Mathematically, the loss function l_{tri} is defined as follows:

$$l_{\text{tri}}(x_a, x_p, x_n) = [m + D(\mathcal{M}_{\text{rep}}(x_a), \mathcal{M}_{\text{rep}}(x_p)) - D(\mathcal{M}_{\text{rep}}(x_a), \mathcal{M}_{\text{rep}}(x_n))]_+ \quad (2)$$

In the above equation, $D(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ denotes a metric function measuring distances in the mapped space.

4 Training Details

We tune the parameters of our model using the algorithm employed for multi-task learning (Liu et al., 2019a). This framework optimizes the language model over multiple tasks. The language model is

shared across different tasks by employing same encoder with shared parameters for all the task-specific overhead architectures. In each iteration of mini-batch gradient descent optimization, a task is randomly selected and corresponding task-specific mini-batch of data is sampled to apply single step of gradient descent using the task-specific objective. We curated the dataset for MLM pretraining by extracting text fragments from the SEC corpus as curated by Chalkidis et al., utilizing newline character (`\n`) as the delimiter. In our ablation studies to understand the impact of various terms in the pretraining objective on downstream performance, we utilized a randomly selected subset of 40,000 text fragments to quickly assess the importance of each of the terms. Thereafter, we also evaluate the performance of our model when a significantly larger corpora is provided for MLM.

In this paper, the parameters of the shared language model are initialized using the weights of LEGAL-BERT (12-layer, 768-hidden, 12-heads, 110M parameters)³, a domain-specific language model pre-trained using MLM. Thereafter, we investigate the performance of the model variants listed in Table 1 by comparing against LEGAL-BERT. We do not assess the performance of non-domain specific models such as BERT (Devlin et al., 2019) as the superiority of LEGAL-BERT over BERT was demonstrated in (Chalkidis et al., 2020) for some of the legal downstream tasks.

³Distributed under CC BY-SA 4.0

Table 1: Model variants to be assessed in various legal downstream tasks (on top of LEGAL-BERT). Legal Corpus for MLM was collected by randomly sampling 40,000 text fragments from the SEC corpus.

Model name	Description of Additional Pre-training
LB-PU	Dynamic span recognition using PU
LB-BC	Binary classification to identify dynamic tokens
LB-MLM	MLM over legal corpus
LB-PU-MLM	Multi-task training for PU and MLM over legal corpus
LB-TRI	Representation learning task using triplet margin loss
LB-TRI-MLM	Multi-task training for triplet margin loss and MLM over legal corpus
LB-PU-TRI	Multi-task training for PU and triplet margin loss
LB-PU-TRI-MLM	Multi-task training for PU, triplet margin loss and MLM over legal corpus

Table 2: Comparison between PU learning and Binary Classification for token-level tasks in terms of F_1 -Scores (DPI: Dynamic Part Identification)

Model name	CUAD-NER	DPI
LEGAL-BERT	0.7040	0.7107
LB-PU	0.7355	0.7507
LB-BC	0.7221	0.6835

We used a 8 GPU A10G instance for training the models. While it took 32 hours to pretrain the model with best hyperparameter settings when only 40,000 datapoints for MLM is used, the model instance pretrained over the total SEC corpus (Chalkidis et al., 2020) consumed 800 hours. HuggingFace Transformers (Wolf et al., 2020) was used for both pretraining and experimental analysis.

5 Results and Discussion

We begin this section by validating the choice of using PU learning for dynamic part detection instead of binary token classification objective. In the subsequent subsection, we describe various legal downstream tasks and their associated data to be used in comparing the performance of the models in Table 1. As our models are derived from LEGAL-BERT, it is used as a baseline in our empirical analysis and we demonstrate the improvement of our model over it for several downstream tasks.

5.1 PU learning Versus Binary classification

5.1.1 Impact on downstream performance

In this subsection, we compare the performance of the model additionally pretrained using PU learning (LB-PU) and binary classification (LB-BC) for named entity recognition (NER) and dynamic part identification (DPI).

We use the NER adaptation of the Contract Un-

derstanding Atticus Dataset (CUAD) (Hendrycks et al., 2021). CUAD labels the *contracting-party* associated with each contract. This is used for constructing a NER dataset with *contracting-party* span annotations for each datapoint. This dataset consists of 16,636 training, 2,000 validation and a 10,000 testing samples.

As the dataset curated for pretraining the language model for dynamic part identification was approximately labeled, we manually annotated few text fragments by specifying the dynamic spans using the definition in section 3.1. This manual annotation furnished 132 training instances, 32 development instances and 50 testing instances. The performance was reported by computing the F_1 -Score between the inferred spans and the ground truth dynamic spans.

The results shown in Table 2 justifies the utilization of PU learning objective. Our hypothesis that training the model to identify dynamic spans will improve its ability in recognizing named entities has been validated by the improvement in NER performance achieved through the use of the PU learning objective. This is further validated in the subsequent section through an examination of the feature representations generated by the model trained with/without PU learning. For the subsequent analysis, we disregard any models trained using binary classification objective owing to the results shown in the table. The decrease in the performance from LEGAL-BERT to LB-BC for NER and DPI stems from the fact that a subset of negatively labelled tokens in some instances are labelled as dynamic for other instances. This confuses the model in learning correct characteristics associated with these tokens, resulting in poor token-level representation.

Table 3: Performance for various legal domain task given in terms of F_1 -Scores for CUAD-NER and DPI tasks, mean of F_1 -Scores for MULTI-EURLEX tasks for Level 1, 2 and 3, and soft F_1 -Score for Contract-Discovery task (Averaged for 5 runs).

Model name	CUAD-NER	DPI	MULTI-EURLEX	Contract-Discovery
LEGAL-BERT	0.7040	0.7107	0.7535	0.4591
LB-MLM	0.7344	0.7098	0.7525	0.4367
LB-PU	0.7355	0.7507	0.7488	0.0394
LB-PU-MLM	0.7427	0.7509	0.7451	0.1701
LB-TRI	0.7325	0.7380	0.7566	0.4979
LB-TRI-MLM	0.7462	0.7091	0.7567	0.5051
LB-PU-TRI	0.7320	0.7454	0.7513	0.5032
LB-PU-TRI-MLM	0.7479	0.7628	0.7574	0.5119

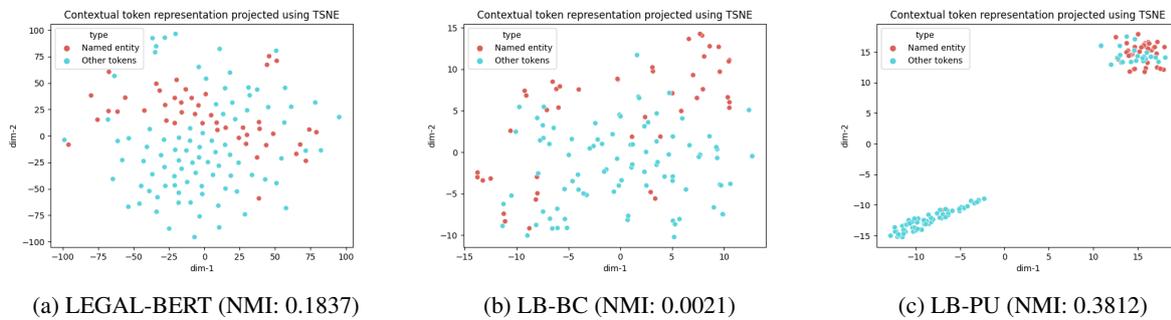


Figure 4: t-SNE projections of the contextualized embeddings obtained from different representation schemes. LB-PU visually performs the best in terms of segregating the named entities from the rest of the tokens.

5.1.2 Better feature representation for extracting named entities

We provide a qualitative justification for PU learning leading to better representation for extracting named entities in this subsection. In this assessment, we extract 30 text sentences from the **CUAD-NER** dataset that contain at least one named entity within it and compute the contextualized embeddings for the tokens in it using LEGAL-BERT, LB-BC and LB-PU. Thereafter, these embeddings are mapped to two dimensional manifold using t-SNE (Van der Maaten and Hinton, 2008) algorithm. Note that, we compute the embeddings using different representation schemes without fine-tuning on **CUAD-NER** to understand the impact of our token-level objective for distinguishing named entities from the rest of the tokens.

From Figure 4, we observe that the embeddings of the named entities and other tokens are not very well separated for LEGAL-BERT and LB-BC. On the other hand, LB-PU leads to much better segre-

gation despite not being explicitly trained for the task of named entity recognition. This can be attributed to the observation that the dynamic part of a legal text fragment corresponds to a named entity most of the times. Since, LB-PU is explicitly pretrained for the task of dynamic part detection, it furnishes suitable representation scheme for segregating named entities. While LB-BC is trained for this task, it yields suboptimal representation scheme as it does not consider the possibility that some of the unlabelled tokens may be dynamic.

5.2 Results in various downstream tasks

Apart from **CUAD-NER** and **DPI** introduced in sec 5.1.1, we consider following additional tasks to compare the performance of different models:

1. **MULTI-EURLEX** (Chalkidis et al., 2021): This dataset is meant to assess the performance in the task of Large-Scale Multi-Label Text Classification (LSMTC). The datapoints in this dataset are curated from European leg-

islative documents (EUR-LEX) and the labels derived from EUROVOC, a set of 4.3K European vocabulary labels. This dataset includes a total of 65K datapoints with the train-test-validation split of 55K-5K-5K respectively and involves fine-grained categorization of the label-set into 8 levels based on their hierarchy. We compute the performance of the model variants for 'level 1' (21 labels), 'level 2' (127 labels) and 'level 3' (567) (but report only the mean of these due to space constraint) as the other levels are not publicly available.

2. **Contract-Discovery** (Borchmann et al., 2020): This dataset is used to measure the performance of a model in semantic retrieval, where the task is to retrieve a span from a target document given a few examples (1 to 5) of similar clauses. The dataset uses about 600 target documents and is divided into 2 splits: development and test. Each of these splits consists of 5000 datapoints. The performance is evaluated by computing soft F_1 metric (Graliński et al., 2019) on the character-level inferred spans, which rewards proportionally to the extent of overlap between predicted and ground truth character spans. To solve this problem, we use the unsupervised method proposed by the authors of this task (Borchmann et al., 2020).

We can see from the Table 3 that the model pretrained using domain specific objectives achieves better performance than LEGAL-BERT for all the tasks. The models pretrained using only PU (LB-PU and LB-PU-MLM) only improves the performance for token-level tasks like CUAD-NER and DPI and achieves poor performance for other tasks. As these models only involve objectives at the token level, they offer inferior representations at the level of sentences / text-fragments as compared to other models which explains the poor performance in tasks like MULTI-EURLEX and Contract-Discovery. A similar effect is also observed for LB-MLM, where the model exhibits superior performance for some of the token-level tasks but exhibits poor performance for sentence level objective when compared against LEGAL-BERT as it does not involve any objective at the level of sentences. The models trained using triplet objective only (LB-TRI and LB-TRI-MLM) achieves better performance than LEGAL-BERT for all the tasks. This justifies the inclusion of the

objective for learning semantic-aware representation scheme. We also observe that, inclusion of MLM for the model variants almost always improves the downstream performance. This indicates the usefulness of having domain-agnostic objective like MLM in the overall objective. The model pretrained using all the objectives (LB-PU-TRI-MLM) achieves best / competitive performance for most of the tasks. It is noteworthy that even though the objective of PU learning has no direct relation to tasks such as Contract-discovery and MULTI-EURLEX, the inclusion of PU learning in combination with Triplet loss and MLM leads to further improvement in the model's effectiveness in those tasks.

These results also emphasize the importance of MLM apart from the domain-specific objectives. Here, the pretraining over MLM was performed over a dataset with about 40,000 text fragments. We believe that the performance of these models can be significantly improved by including a sufficiently larger dataset for MLM pretraining which is validated in the next subsection.

5.3 Performance when the size of MLM corpora is varied

In this section, we assess the performance of our model trained using the three objectives (PU + TRI + MLM) when the number of datapoints in the MLM corpus is varied. While the experiment performed in the previous subsection comprised of only 40,000 text fragments, this analysis assesses the model performance when the number of text fragments is varied from 1% to 100% of the total SEC corpus (Chalkidis et al., 2020).

The results shown in Table 4 clearly demonstrate that the downstream performance improves with the number of datapoints in the MLM corpus. Note that, **the pretraining corpus for LEGAL-BERT already comprises of the SEC corpus used in our analysis**. This fact also confirms the importance of involving the two objectives along with MLM for getting improved performance.

6 Conclusion

In this paper, we demonstrated a novel approach to enhance the performance of domain-specific language model across several specialty downstream tasks by exploiting the language characteristics. The objectives presented in this paper may not be applicable to all domains, which is a limitation

Table 4: Performance for various legal domain task given in terms of F_1 -Scores for CUAD-NER and DPI tasks, mean of F_1 -Scores for MULTI-EURLEX tasks for Level 1, 2 and 3, and soft F_1 -Score for Contract-Discovery task when the number of datapoints in the MLM corpus is varied.

Number of training datapoints for MLM	Fraction of the overall SEC corpus	CUAD-NER	DPI	MULTI-EURLEX	Contract-Discovery
40,000	5.56×10^{-4}	0.7479	0.7628	0.7574	0.5119
720,000	0.01	0.7483	0.7651	0.7546	0.5210
7,200,000	0.10	0.7518	0.7662	0.7547	0.5145
18,000,000	0.25	0.7457	0.7636	0.7471	0.5158
72,000,000	1.00	0.7523	0.7721	0.7577	0.5216

of our work, but the idea of formulating objectives for learning domain-specific characteristics can be applied to other specialty domains (biomedical, programming languages, etc.). Future work might involve studying other characteristics of the legal domain and understanding their impact in downstream performance. We justified the positive impact of such pretraining across several downstream tasks by conducting extensive quantitative analysis.

We conclude this section by enumerating the natural extensions of this work for future:

1. In this work, we emphasized on two characteristics in the legal domain. However, the legal domain consists of several other domain-specific characteristics. For instance, the content in a legal agreement can be structured into different parts (preamble, recitals, list of clauses, etc) and the impact of involving a pre-training objective to infer the structure of a legal document on several tasks is yet to be understood. Thus, one line of future work may involve exhaustive study of language characteristics and understanding their influence in downstream tasks.
2. In the future, we plan to study the applicability of the introduced characteristics in other domains, such as programming languages where text fragments can be classified into categories like function blocks, variable declaration, etc. and contain both static and dynamic elements that can be templated. This study may provide a thorough evaluation of the cross-domain applicability of these characteristics, including the assessment of their impact on downstream performance and the ease of curating relevant data. We would like to also

motivate the researchers in applying the principle introduced in this paper for other domains (biomedical, finance, etc.). This necessitates careful investigation in order to extract domain-specific characteristics, as well as a mechanism for training the language model to understand these characteristics.

7 Limitations

We now discuss the limitations of our work. The first limitation (or requirement) is need for significant computational power. As we showed in Section 5.3 of our paper, when the corpus size for MLM training is increased from 0.0556% to 100% of the SEC corpus, while the performance improved by about 1% on multiple tasks, the computational requirement went up from 32 hours (on a 8 GPU A10G instance) to 800 hours.

Secondly, we had built our model on top of a domain specific pre-trained language model (which had used only MLM objective on a domain specific corpus). In theory, since we do include MLM as one of the objectives, we should be able to get comparable performance with or without domain specific pretrained language model. However, due to significant cost involved, we did not train a model starting from general domain language model (e.g., BERT or RoBERTa) to compare its performance against model built on top of domain specific pre-trained language model. Therefore, we cannot make a claim if our proposed method would result in comparable performance improvement for the domains where such pre-trained models are not available.

Third, our method relies on identifying the domain specific characteristics and building objective functions suitable to exploit them. This requires building domain expertise and/or collaborat-

ing with domain experts. Since this process cannot be automated, it requires additional cost and human effort. Also, good automated data curation strategies may or may not be feasible for other domain specific characteristics, limiting using usefulness for training large language models.

Finally, we have only experimented with English language corpus. While the data curation strategy we used should be applicable in most other languages also for legal domain, the static/dynamic token classification task particularly may depend on grammatical rules for sentence construction, which may not be similar in all languages.

However, we believe that despite these limitations, our work points to possibility of improved performance of language models by using domain specific characteristics (beyond MLM based pre-training), which should open doors for more such explorations and significant advances in the state of art.

References

- Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoȃuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93.
- Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. [Publicly available clinical BERT embeddings](#). In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Łukasz Borchmann, Dawid Wisniewski, Andrzej Gretkowski, Izabela Kosmala, Dawid Jurkiewicz, Łukasz Szalkiewicz, Gabriela Pałka, Karol Kaczmarek, Agnieszka Kaliska, and Filip Galiński. 2020. [Contract discovery: Dataset and a few-shot semantic retrieval challenge with competitive baselines](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4254–4268, Online. Association for Computational Linguistics.
- Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE.
- Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. 2017. A low-cost, high-coverage legal named entity recognizer, classifier and linker. In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*, pages 9–18.
- Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. [MultiEURLEX - a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. [LEGAL-BERT: The muppets straight out of law school](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Mohamed I El Desouki and Wael H Gomaa. 2019. Exploring the recent trends of paraphrase detection. *International Journal of Computer Applications*, 975(S 8887).
- Filip Galiński, Anna Wróblewska, Tomasz Stanisławek, Kamil Grabowski, and Tomasz Górecki. 2019. [GEval: Tool for debugging NLP datasets and models](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 254–262, Florence, Italy. Association for Computational Linguistics.
- Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don't stop pretraining: Adapt language models to domains and tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Xiaochuang Han and Jacob Eisenstein. 2019. [Unsupervised domain adaptation of contextualized embeddings for sequence labeling](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China. Association for Computational Linguistics.

- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*.
- Piotr Indyk, Rajeev Motwani, Prabhakar Raghavan, and Santosh Vempala. 1997. Locality-preserving hashing in multidimensional spaces. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 618–625.
- Mi-Young Kim and Randy Goebel. 2017. Two-step cascaded textual entailment for legal bar exam question answering. In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*, pages 283–290.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.
- Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.
- Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. [Multi-task deep neural networks for natural language understanding](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. [Roberta: A robustly optimized bert pretraining approach](#).
- Patrick Niemeyer and Jonathan Knudsen. 2005. *Learning java*. "O'Reilly Media, Inc."
- Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuan-Jing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419.
- Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2020. [Adapt or get left behind: Domain adaptation through BERT language model fine-tuning for aspect-target sentiment classification](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4933–4941, Marseille, France. European Language Resources Association.
- Peter M Tiersma. 1999. *Legal language*. University of Chicago Press.
- Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. 2020. [LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1235–1241, Marseille, France. European Language Resources Association.
- Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Christopher Williams. 2007. *Tradition and change in legal English: Verbal constructions in prescriptive texts*, volume 20. Peter Lang.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.
- Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. [How does NLP benefit legal system: A summary of legal artificial intelligence](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, Online. Association for Computational Linguistics.

Global Constraints with Prompting for Zero-Shot Event Argument Classification

Zizheng Lin¹, Hongming Zhang² & Yangqiu Song¹

¹Department of Computer Science and Engineering, HKUST

²Tencent AI Lab, Seattle

{zlinai, yqsong}@cse.ust.hk; hongmzhang@global.tencent.com

Abstract

Determining the role of event arguments is a crucial subtask of event extraction. Most previous supervised models leverage costly annotations, which is not practical for open-domain applications. In this work, we propose to use global constraints with prompting to effectively tackle event argument classification without any annotation and task-specific training. Specifically, given an event and its associated passage, the model first creates several new passages by prefix prompts and cloze prompts, where prefix prompts indicate event type and trigger span, and cloze prompts connect each candidate role with the target argument span. Then, a pre-trained language model scores the new passages, making the initial prediction. Our novel prompt templates can easily adapt to all events and argument types without manual effort. Next, the model regularizes the prediction by global constraints exploiting cross-task, cross-argument, and cross-event relations. Extensive experiments demonstrate our model’s effectiveness: it outperforms the best zero-shot baselines by 12.5% and 10.9% F1 on ACE and ERE with given argument spans and by 4.3% and 3.3% F1, respectively, without given argument spans. We have made our code publicly available.¹

1 Introduction

Event Argument Classification² (EAC), finding the roles of event arguments, is an important and challenging event extraction sub-task. As shown in Figure 1, a “Transfer-Money” event whose trigger is “acquiring” has several argument spans (e.g., “Daily Planet”). By determining the role of these arguments (e.g., “Daily Planet” as “Beneficiary”), we

¹<https://github.com/HKUST-KnowComp/Constraints-with-Prompting-for-Zero-Shot-EAC>

²We focus on event argument because existing zero-shot trigger extraction models like Zhang et al. (2021) are already strong enough, but the arguments remain a challenge. Our argument identification approach is described in Section 3.1.

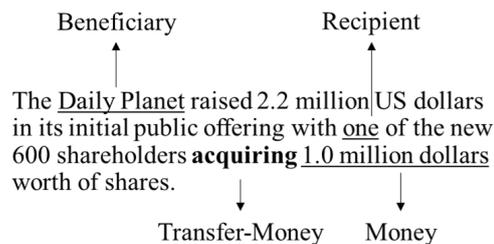


Figure 1: An example of EAC. The trigger is in **bold face**. Arguments are underlined and connected to their roles by arrows.

can obtain a better understanding of the event, thus benefiting related applications like stock price prediction (Ding et al., 2015) and biomedical research (Zhao et al., 2021).

Many previous EAC works require numerous annotations to train their models (Lin et al., 2020; Hsu et al., 2022; Liu et al., 2022), which is not only costly as the annotations are labor-intensive but also difficult to be generalized to datasets of novel domains. Accordingly, some EAC models adopt a few-shot learning paradigm (Ma et al., 2022; Hsu et al., 2022). However, they are sensitive to the few-shot example selection and they still require costly task-specific training, which hinders their real-life deployment. There have been some zero-shot EAC models based on transfer learning (Huang et al., 2018), or label semantics (Zhang et al., 2021; Wang et al., 2022), or prompt learning (Liu et al., 2020; Lyu et al., 2021; Huang et al., 2022; Mehta et al., 2022). However, these models’ corresponding limitations impede their real-life deployment. The model based on transfer learning can be ineffective when new event types are very different from the observed one. As for models using label semantics, they require a laborious preparation process and have unsatisfactory performance. Regarding models adopting prompt learning, they need tedious prompt design customized to every new type of events and arguments, and their performance is

also limited.

To address the aforementioned issues, we propose an approach using global constraints with prompting to tackle zero-shot EAC. Global constraints can be viewed as a type of supervision signal from domain knowledge, which is crucial for zero-shot EAC since supervision from annotations is inaccessible. Moreover, our model’s constraints module provides abundant global insights across tasks, arguments, and events. Prompting can also be regarded as a supervision signal as it induces abundant knowledge from Pre-Trained Language Models (PTLM). Unlike previous zero-shot EAC works, which need a tedious prompt design for every new type of events and arguments, the novel prompt templates of our model’s prompting module can be easily adapted to all possible types of events and arguments in a fully automatic way. Specifically, given an event and its passage, our model first adds prefix prompt, cloze prompt, and candidate roles into the passage, which creates a set of new passages. The Prefix prompt describes the event type and trigger span. Cloze prompt connects each candidate to the target argument span. Afterwards, our model adopts a PTLM to compute the language modeling loss for each of the new passages, whose negative value would be the respective prompting score. The role with the highest prompting score is the initial prediction. Then, our model uses global constraints to regularize the initial prediction. The global constraints are based on the domain knowledge of the following relations: (1) cross-task relation, where our model additionally performs another one or more classification task on target argument span, and our model’s predictions on EAC and other task(s) should be consistent; (2) cross-argument relation, where arguments of one event should collectively abide by certain constraint(s); (3) cross-event relation, where some argument playing a certain role in one event should play a typical role in another related event.

We conduct comprehensive experiments to demonstrate the effectiveness of our model. Particularly, our approach surpasses all zero-shot baselines by at least 12.5% and 10.9% F1 on ACE and ERE, respectively. When argument spans are not given, our model outperforms the best zero-shot baseline by 4.3% and 3.3% F1 on ACE and ERE, respectively. Besides that, we also conduct experiments to show that both the prompting and constraints modules contribute to the final success.

2 Methodology

We first present an overview of our approach. Then we introduce the details by describing its prompting module and global constraints regularization module. We follow (Liu et al., 2021) to name a prompt inserted before input text as *prefix prompt*, and a prompt with slot(s) to fill in and insert in the middle of input text as *cloze prompt*.

2.1 Overview

As shown in Figure 2, given a passage with a target argument span, our model infers the target’s role without annotation and task-specific training. Our model has two modules. The first module is the prompting module that creates and scores several new passages. During creation, the model adds prefix prompt, cloze prompt, and candidate roles into the passage, where the prefix prompt contains information about event type and trigger, and the cloze prompt joins each candidate with a target argument span.³ Afterwards, the model uses a PTLM to score the new passages. Our novel prompt templates can easily adapt to all possible events and arguments without manual work. Initial prediction is the role with the best prompting scores. The second module is the global constraints regularization module, where the model regularizes the prediction by three types of global constraints: cross-task constraint, cross-argument constraint, and cross-event constraint. All global constraints are based on event-related domain knowledge about inter-task, inter-argument, and inter-event relations.

2.2 Prompting Module

In this section, we describe the prompting module in detail. Given a passage, we first add a prefix prompt containing information about the event type and trigger span to the beginning. Such a prompt can guide a PTLM to: (1) accurately capture the input text’s perspective related to the event; (2) have a clear awareness of the trigger. Based on the definitions of events and triggers (Grishman et al., 2005), we create the following prefix prompt: “**This is a [] event whose occurrence is most clearly expressed by [].**” where the first and second pairs of square brackets are the placeholders of event type and trigger span respectively. We also con-

³Since we focus on event argument classification, we assume that the event types and trigger spans are given. The settings without given argument spans will be discussed in Section 3.1.

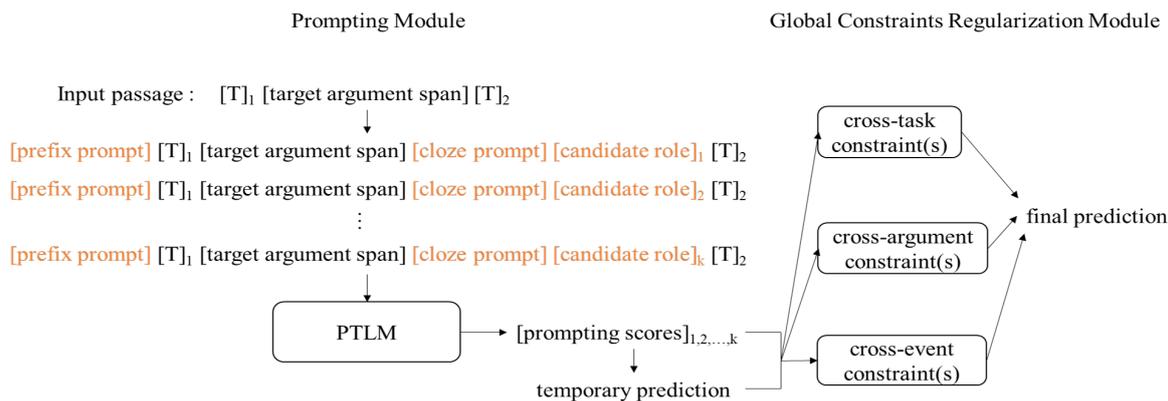


Figure 2: Model overview using prediction for one argument span as an example. $[T]_1$ and $[T]_2$ are the parts of the input passage before and after the span, respectively. k is the number of candidate roles of the event type.

ducted some experiments comparing different prefix prompts in Section A, and the results showed that the prefix above is the most effective.

Second, for each candidate role, the module inserts the cloze prompt behind the target argument span, and the role fills the prompt’s slot. The cloze prompt adopts the hypernym extraction pattern “**M and any other []**” (Dai et al., 2021), where “M” denotes the argument span and the square bracket is the placeholder of the candidate role. We did not try other hypernym extraction patterns as (Dai et al., 2021) had shown that our pattern is the most effective. The motivation for adopting the hypernym extraction pattern for cloze prompt is that, to some extent a role can be regarded as a context-specific hypernym of the respective argument span of the associated event (e.g., “Beneficiary” can be seen as a context-specific hypernym of “Daily Planet” of the Transfer-Money event described by the example in Figure 1). Hence, such a prompt induces the linguistic and commonsense knowledge stored in PTLM to help identify which candidate role is the most reasonable.

After adding the previous two types of prompts, we get several new passages. For instance, suppose the passage is “In Baghdad, a bomb was fired at 17 people.” whose event type is “Conflict:Attack”, trigger is “fired”, target argument span is “bomb”, and candidate roles are {“Attacker”, “Instrument”, “Place”, “Time”, “Target”}. The created passages would be: (1) “*This is a Attack event whose occurrence is most clearly expressed by “fired.”* In Baghdad, a bomb *and any other attacker* was fired at 17 people.”; (2) “*This is a Attack ... “fired.” ... bomb and any other instrument* was ...”; and simi-

lar text for other roles.⁴

For each new passage, we apply a PTLM to compute the language modeling loss. The negative value of the loss would be the prompting score of the respective passage, where a higher value indicates higher plausibility according to the PTLM. **Since our model’s prompt templates are independent of event type and argument role, their adaptation to any new type of events and arguments is trivial and fully automatic.** Hence, our prompting method is more scalable and generalizable than those of previous zero-shot EAC models, since, for every new type of events and arguments they need to design a customized prompt. For instance, for every type of events/arguments, Lyu et al. (2021) manually design a unique prompt as text entailment/question answering template. The initial prediction would be the role with the highest prompting score. Since the steps of obtaining scores for each candidate role are independent of other candidate roles, we implement the steps of different candidate roles in parallel. Such a parallel implementation significantly improves our model’s efficiency.

2.3 Global Constraints Regularization Module

This module regularizes the prediction by the following three types of global constraints.⁵

Cross-task constraint exploits the label dependency between EAC and auxiliary task(s) so that

⁴We only use the subtype of all events following the pre-processing done by (Lin et al., 2020)

⁵We designed 14 global constraints in total and we used preliminary experiments to choose the three most effective ones. In the preliminary experiments, we randomly sample 1k instances covering all trigger and argument types. We then evaluate each constraint on the sampled subset.

our model can get global information from the auxiliary task(s) about event arguments. We use **Event Argument Entity Typing (EAET)** as the auxiliary task. The task aims to classify an argument into its context-dependent entity type (e.g., PER). As specified in ACE2005 ontology, an argument of a certain role in an event can only be one of several respective entity types (e.g., an argument of “Attack” role in a Conflict:Attack event can only be “ORG,” “PER,” or “GPE”). Based on this domain knowledge, we design the cross-task constraint as follows: (1) For each input passage, our model performs prompting for EAET, where the prompting is the same as in Section 2.2 except that candidate entity types replace the candidate roles in cloze prompt.; (2) After obtaining the scores and prediction of EAET, the model check the consistency between the predictions of EAC and EAET; (3) If the consistency is violated and the score of EAC’s predicted role is lower, then discard the current role, use the role with the highest score in the remaining ones, and check the consistency again; (4) The constraint ends when the labels of two tasks are consistent. An example illustrating this type of constraint is shown in Figure 3.

Cross-argument constraint is based on domain knowledge about relationships between arguments within an event. Specifically, our model constrains the number of particular arguments for some or all events. For instance, it is very unlikely that an event mentioned is associated with multiple “Time” arguments. Such constraints offer a global understanding of event arguments to our model. The cross-argument constraint we adopt is “A Personnel:End-POSITION event has at most one Position argument.” Given a Personnel:End-POSITION event, our model first checks the number of “Position” argument. If the number is more than one, then our model will first collect the arguments whose roles are “Position” and remove the one with the highest score among these arguments. Then for each remaining argument, our model would change the role to its candidate with the second highest score. An example illustrating this type of constraint is shown in Figure 4.

Cross-event constraint regularizes predicted roles of arguments shared by related events. A model with such a constraint can have global insights into event arguments, because while they are making inferences for the arguments of one event, they are aware of the information of other

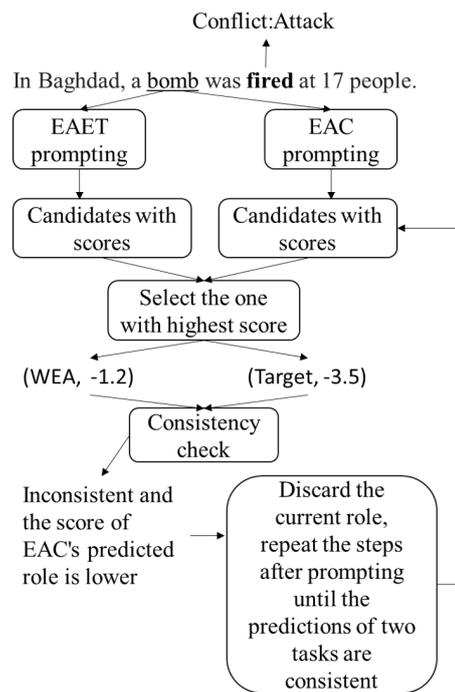


Figure 3: An Example of cross-task constraint. The text in **bold face** is the trigger, underlined text is target argument span, and a tuple denotes a predicted label with its prompting score (e.g., “(Target, -3.5)” denotes the predicted label “Target” with its prompting score“-3.5”). Similar notations are adopted in all remaining figures.

related event(s) and cross-event relations. The cross-event constraint we adopt is “**If a Life:Injure event and a Conflict:Attack event share arguments, then Injure.Place is the same as Attack.Place, Injure.Victim is the same as Attack.Target, Injure.Instrument is the same as Attack.Instrument, Injure.Time is the same as Attack.Time, Injure.Agent is the same as Attack.Attacker**”. Given a passage containing an Injure and an Attack event sharing arguments, the model imposes the constraint by checking the consistency between the respective roles of each shared argument as specified in the constraint. Any inconsistency would be fixed by changing the role with a lower prompting score to the new one satisfying the consistency. An example illustrating this type of constraint is shown in Figure 5.

Our constraint modeling method can be easily generalized to other datasets/ontologies by simply using the knowledge about corresponding cross-task, cross-argument, and cross-event relations to design new constraints. The design processes are not costly as we could easily find such knowledge

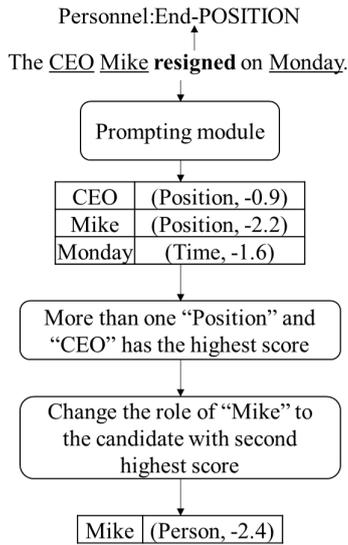


Figure 4: An Example of cross-argument constraint.

from the guidelines of the target dataset.

3 Experiments

We first present the experimental settings, baselines used for comparison, and some implementation details. Next, we show and analyze the experiment results. Then we present a detailed analysis of the prompting module and global constraints regularization module. Finally, we conduct an error analysis.

3.1 Settings

We use ACE (2005-E⁺)⁶ (Doddington et al., 2004; Lin et al., 2020) and ERE(-EN) (Song et al., 2015) as datasets. In total, ACE has 33 event types and 22 roles, whereas ERE has 38 event types and 21 roles. We pre-process all events to keep only the event subtypes whenever applicable, as done in (Lin et al., 2020). Following the pre-processing in (Zhang et al., 2021), for each dataset, we merge all splits into one test set since our approach is zero-shot. When argument spans are not given, we pipeline our model with an argument identification module adapted from (Lyu et al., 2021). Specifically, we replace the QA model in (Lyu et al., 2021) with a more powerful PTLM with a span classification head on top, and the whole model has been fined-tuned for extractive QA tasks. Then for a passage, we prompt each role using the new QA model as in (Lyu et al., 2021). We collect the prompt results for all roles (ignoring the “None” result) as

⁶<https://www ldc.upenn.edu/collaborations/past-projects/ace>

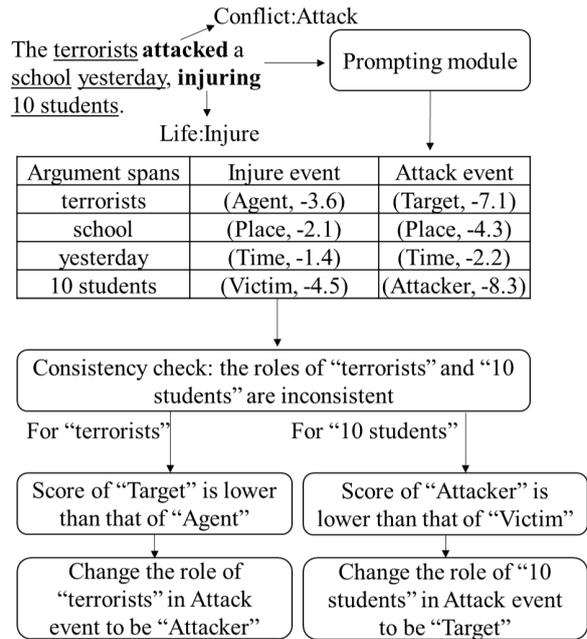


Figure 5: An Example of cross-event constraint.

candidate spans for the passage. We use the F1 score for evaluation following (Ji and Grishman, 2008), where argument spans are evaluated on the head level when not given. Regarding PTLMs, We use GPT-J (6B) (Wang and Komatsuzaki, 2021) instances from Huggingface (Wolf et al., 2020), where an instance for causal language modeling is used for prompting, and an instance for QA is used for argument identification. In all the following sections except Section 3.2, we conduct experiments on ACE, assuming that argument spans are given.

3.2 Main Results

We report the main results comparing our models with three previous powerful zero-shot models (Liu et al., 2020; Lyu et al., 2021; Zhang et al., 2021). Moreover, we also report the results of a SOTA supervised model (Hsu et al., 2022). We obtain the results of all compared methods from our own experiments to ensure a fair comparison on the same datasets and same settings. From Table 1, we have the following observations:

- Our model achieves superior performance on both datasets under both settings compared with all zero-shot baselines. Specifically, our model surpasses the best zero-shot baselines (Zhang et al., 2021) by 12.5% and 10.9% on ACE and ERE, respectively. Without argument spans, our model outperforms the respective best zero-shot baselines (Lyu et al.,

Model	ACE		ERE	
	argument span given	argument span not given	argument span given	argument span not given
(Hsu et al., 2022) (supervised)	79.3	71.8	79.8	72.5
(Liu et al., 2020)	46.1	24.2	40.9	22.8
(Lyu et al., 2021)	47.8	26.9	44.5	26.3
(Zhang et al., 2021)	53.6	23.5	51.9	20.2
Ours	66.1	31.2	62.8	29.6

Table 1: Performance of supervised model, zero-shot baselines, and our model. The best scores among the ones of zero-shot methods are in bold font.

2021) by 4.3% and 3.3% on ACE and ERE, respectively, which is also a noticeable gap. Such large performance improvements can be attributed to the following: (1) the prefix prompt guides the PTLM to effectively capture input’s event-related perspective and trigger; (2) the cloze prompt leverages linguistic and commonsense knowledge stored in PTLM to improve its contextual understanding of event arguments; (3) the global constraints regularization incorporate global information and domain knowledge in inference. In Section 3.3, we compare the effects of using different PTLMs like BERT in the prompting module, and the results show that our model consistently outperforms previous zero-shot models, as shown in Table 1 and Figure 6.

- Compared with the supervised SOTA model (Hsu et al., 2022), there is still a significant gap between our model’s performance and that it. Specifically, (Hsu et al., 2022) outperforms our model by 13.2% and 17.0% on ACE and ERE, respectively. When argument spans are not provided, (Hsu et al., 2022) outruns our model by 40.6% and 42.9% on ACE and ERE, respectively. We can see that the advantage of supervised SOTA over our zero-shot method is much more distinct when argument spans are not given in advance. This is probably because our zero-shot argument identification module described in Section 3.1 is not powerful enough, which causes severe error propagation to our EAC model.

3.3 Analysis of Prompting Module

We conduct experiments to examine the effects of different configurations of prefix prompt templates. Specifically, we compare our model’s complete prefix prompt with the following configurations: (1) removing event type information from the prefix; (2) removing trigger information from the prefix;

Configurations	F1	Δ
complete prefix prompt	66.1	-
w/o event type	64.4	-1.7
w/o trigger	64.9	-1.2
w/o prefix prompt	62.8	-3.3

Table 2: Results of using different configurations of prefix prompt.

(3) removing the whole prefix. For instance, suppose the passage is “In Baghdad, a bomb was fired at 17 people.” mentioned in Section 2.2, the prefix in configuration (1) would be “**This event’s occurrence is most clearly expressed by ‘fired’.**”, the prefix in configuration (2) would be “**This is a Attack event.**”, and in configuration (3) there would be no prefix. The corresponding results are shown in Table 2, where we have the following observations. First, removing either event type or trigger from the prefix prompt will cause a performance drop, which indicates that both kinds of information have contributions to the prompting process. Second, event type plays a more significant role than trigger does in prefix prompt, and the joint effect of them is greater than the sum of their respective effects.

In addition, we examine the effects of using different PTLMs in the prompting module. We compare the following PTLMs with GPT-J (6B): BERT (large, uncased) (Devlin et al., 2019), RoBERTa (large) (Liu et al., 2019), BART (large) (Lewis et al., 2020), GPT-2 (xl) (Radford et al., 2019), T5 (11B) (Raffel et al., 2020). The results are shown in Figure 6, where we have the following observations. First, the instance using GPT-J has the best performance, surpassing other instances by 4.2% to 7.9%. This shows that GPT-J has a better ability to understand events and their associated arguments compared to other PTLMs. Second, as PTLMs are listed in ascending order based on their numbers of

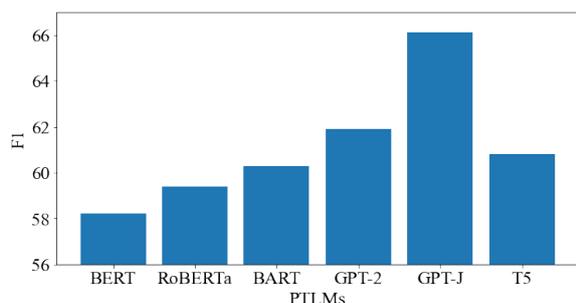


Figure 6: Comparison between the performance of using different PTLMs in prompting module.

parameters, we can see that for the first five models, the performance increases as the sizes of PTLMs become larger, which is consistent with the widely accepted notion that the larger model has a better capability of solving language tasks. However, the instance using the largest PTLM, T5 (11B), has a worse performance than GPT-2 and GPT-J. This is probably because autoregressive language modeling is more suitable for capturing information related to event arguments than mask language modeling is.

3.4 Analysis of Global Constraints Regularization Module

We conduct experiments to study the individual effect of each global constraint on the overall performance. The results are shown in Table 3, where we have the following observations. First, every

Model	F1	Δ
Full model	66.1	-
w/o cross-task constraint	60.5	-5.6
w/o cross-argument constraint	64.8	-1.3
w/o cross-event constraint	63.6	-2.5

Table 3: Results of using different configurations of global constraints.

global constraint used by our model is beneficial to overall performance, which demonstrates that exploiting the domain knowledge about cross-task, cross-argument, and cross-event relations indeed provides our model with global understanding of event arguments. Second, the contribution of cross-task constraint is the most significant, which suggests that the global insights from the entity typing tasks are more effective in improving our model’s reasoning ability about event arguments. Third, the cross-argument constraint is less effective than the

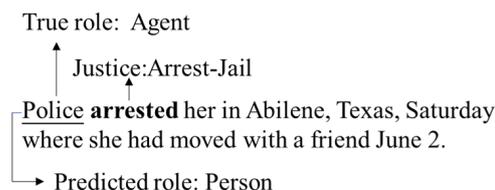


Figure 7: An Example of the wrong prediction caused by too general argument roles. The text in **bold face** denotes trigger and the underlined text denotes target argument span.

other constraints, which shows that the global insights provided by the cross-argument constraint is less informative than those provided by the other constraints.

Apart from the three global constraints described above, we have designed another 11 global constraints, which rely on cross-argument or cross-event relations. We add each of them into our model to check their respective effects on the overall performance. The results of three of them are in Table 4, whereas the results of all of them are in Section B. From the results, we can find that each of these constraints either brings minor improvement or even has a negative influence on the overall performance. Hence, we do not incorporate these constraints in our model to maintain our model’s efficiency and effectiveness.

3.5 Error Analysis

We manually checked 100 wrong predictions of our model and found that most of the errors are caused by too general roles of some event types. Specifically, some roles’ linguistic meanings are so general that a model, not knowing their detailed event-type-dependent semantics, tends to assign them to some arguments which should have been assigned other roles. An example is shown in Figure 7. The example describes a Justice:Arrest-Jail event, which is associated with the following roles: “Person,” “Agent,” “Crime,” “Time,” and “Place.” “Person” refers to the person who is jailed or arrested, whereas “Agent” refers to the jailer or the arresting agent. In the example, the argument span’s true role should be “Agent” according to the detailed event-type-dependent semantics of “Person” and “Agent.” However, our approach is zero-shot and directly models all role labels as natural language words, without incorporating the detailed event-type-dependent semantics of those roles, which are too general (e.g., “Person”). Therefore, our model assigns “Person” to “Police” since it is reasonable

Global constraint	Effect on overall performance
There is at most one Time-Arg in each event.	0.4
A TRANSPORT event has at most one ORIGIN argument.	-0.1
If an Arrest-Jail event and a Charge-Indict event share arguments, Arrest-Jail.Person is the same as Charge-Indict.Defendant, they share the same Crime argument.	0.3

Table 4: Results of three other global constraints. Results of all other global constraints are in Section B

from the perspectives of linguistic and common-sense knowledge, and “Person” is much more common than “Agent” in the pre-training corpus of the PTLM in the prompting module, which makes it have much higher likelihood in the language modeling process. Incorporating event-type-dependent semantics of the roles which are too general into our model is left as future work.

4 Related Work

In this section, we introduce related works about constraint modeling, event extractions, and prompt-based Information Extraction (IE).

4.1 Constraint Modeling

Constraint modeling, as an important technique in machine learning and NLP, aims to improve a model’s performance by incorporating domain knowledge as constraints (Ganchev et al., 2010; Chang et al., 2012, 2013; Deutsch et al., 2019; Chang et al., 2008, 2010; Graça et al., 2010). One of the most significant advantages of constrained modeling is that it enables a model to capture the expressive and complex dependency structure in structured prediction problems like EAC (Chang et al., 2012). Especially in zero-shot scenarios, constrained modeling can provide useful indirect supervision to a model, which further boosts performance (Ganchev et al., 2010). Some previous works have adopted constraints based on event-related domain knowledge to classify event arguments (Lin et al., 2020; Zhang et al., 2021). However, their constraints either require labor-intensive annotations (Lin et al., 2020) or consider limited global information (e.g., cross-event relations) (Zhang et al., 2021). In this paper, our model uses global constraints to regularize prediction by incorporating global insights from cross-task, cross-argument, and cross-event relations.

4.2 Event Extraction

Event extraction is a fundamental information extraction task (Sundheim, 1992; Grishman and Sundheim, 1996; Riloff, 1996; Grishman et al., 2005; Chen et al., 2021; Du and Cardie, 2020; Liu et al., 2020), which can be further divided into four sub-tasks: trigger identification, trigger classification, argument identification, and argument classification. Traditional efforts mostly focus on the supervised setting (Ji and Grishman, 2008; Liao and Grishman, 2010; Liu et al., 2016; Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018; Zhang et al., 2019; Wadden et al., 2019; Lin et al., 2020). However, these works could suffer from the huge burden of human annotation. In this work, we focus on the argument classification task and propose a model using prompting and global constraints, without annotation and task-specific training.

4.3 Prompt-based IE

With the fast development of large PTLMs like T5 (Raffel et al., 2020), GPT-3 (Brown et al., 2020), and Pathway Language models (Chowdhery et al., 2022), the prompt-based method has been an efficient tool of applying those giant models into downstream NLP tasks (Liu et al., 2021). IE is not an exception. People have been using leverage prompts and giant models to solve IE tasks like named entity recognition (Cui et al., 2021), semantic parsing (Shin et al., 2021), and relations extraction (Chen et al., 2022; Han et al., 2021) in a zero-shot or few-shot way. However, previous prompting methods for IE need a tedious prompt design for every new type of events and arguments. In contrast, our model’s prompt templates can be adapted to all possible types of events and arguments in a fully automatic way.

5 Conclusion

We propose a zero-shot EAC model using global constraints with prompting. Compared with previ-

ous works, our model does not require any annotation or manual prompt design, and our constraint modeling method can be easily adapted to any other datasets. Hence, our model can be easily generalized to any open-world event ontologies. Experiments on two standard event extraction datasets demonstrate our model’s effectiveness.

6 Limitations

Our work has the following limitations. One limitation is that our model is not aware of the detailed event-type-dependent semantics of those roles which are too general, as discussed in Section 3.5. In the future, we will work on enabling our model to capture the event-type-dependent semantics of the roles which are too general. Another limitation is that our model’s performance is still unsatisfactory compared with SOTA supervised model when argument spans are not given, as discussed in Section 3.2. In the future, we will work on designing a more powerful zero-shot event argument identification module for our model, so that we can obtain satisfactory zero-shot EAC performance even when argument spans are not given.

7 Acknowledgement

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20) and the GRF (16211520 and 16205322) from RGC of Hong Kong, the MHKJFS (MHP/001/19) from ITC of Hong Kong and the National Key R&D Program of China (2019YFE0198200) with special thanks to HKMAAC and CUSBLT, and the Jiangsu Province Science and Technology Collaboration Fund (BZ2021065). We also thank the support from the UGC Research Matching Grants (RMGS20EG01-D, RMGS20CR11, RMGS20CR12, RMGS20EG19, RMGS20EG21, RMGS23CR05, RMGS23EG08).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei.

2020. Language models are few-shot learners. In *NeurIPS*.

Kai-Wei Chang, Rajhans Samdani, and Dan Roth. 2013. A constrained latent variable model for coreference resolution. In *EMNLP*, pages 601–612. ACL.

Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative Learning over Constrained Latent Representations. In *NAACL*. ACL.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2012. Structured learning with constrained conditional models. *Machine learning*, 88(3):399–431.

Ming-Wei Chang, Lev-Arie Ratinov, Nicholas Rizzolo, and Dan Roth. 2008. Learning and inference with constraints. In *AAAI*, pages 1513–1518. AAAI Press.

Muhao Chen, Hongming Zhang, Qiang Ning, Manling Li, Heng Ji, Kathleen McKeown, and Dan Roth. 2021. Event-centric natural language processing. In *ACL Tutorial*, pages 6–14.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*, pages 167–176. ACL.

Yulong Chen, Yang Liu, Li Dong, Shuohang Wang, Chenguang Zhu, Michael Zeng, and Yue Zhang. 2022. Adaprompt: Adaptive model training for prompt-based NLP. *CoRR*, abs/2202.04824.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pilla, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of ACL/IJCNLP*, pages 1835–1845. ACL.

- Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. Ultra-fine entity typing with weak supervision from a masked language model. In *ACL/IJCNLP*, pages 1790–1799. ACL.
- Daniel Deutsch, Shyam Upadhyay, and Dan Roth. 2019. A general-purpose algorithm for constrained sequential inference. In *CoNLL*, pages 482–492. ACL.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186. ACL.
- Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *IJCAI*, pages 2327–2333. AAAI Press.
- George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. Weischedel. 2004. The automatic content extraction (ACE) program - tasks, data, and evaluation. In *LREC*. ELRA.
- Xinya Du and Claire Cardie. 2020. [Event extraction by answering \(almost\) natural questions](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 671–683. Association for Computational Linguistics.
- Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(67):2001–2049.
- João Graça, Kuzman Ganchev, and Ben Taskar. 2010. Learning tractable word alignment models with complex constraints. *Comput. Linguistics*, 36(3):481–504.
- Ralph Grishman and Beth Sundheim. 1996. Message understanding conference- 6: A brief history. In *COLING*, pages 466–471.
- Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyu’s english ace 2005 system description. *ACE*, 5.
- Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. *CoRR*, abs/2105.11259.
- I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Degree: A data-efficient generative event extraction model. In *NAACL*. ACL.
- Kuan-Hao Huang, I-Hung Hsu, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. Multilingual generative language models for zero-shot cross-lingual event argument extraction. In *ACL*, pages 4633–4646. ACL.
- Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *ACL*, pages 2160–2170. ACL.
- Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*, pages 254–262. ACL.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880. ACL.
- Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *ACL*, pages 789–797. ACL.
- Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *ACL*, pages 7999–8009. ACL.
- Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *EMNLP*, pages 1641–1651. ACL.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.
- Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging framenet to improve automatic event detection. In *ACL*. ACL.
- Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. [Dynamic prefix-tuning for generative template-based event extraction](#). In *ACL*, pages 5216–5228. ACL.
- Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *EMNLP*, pages 1247–1256.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.
- Qing Lyu, Hongming Zhang, Elicor Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *ACL/IJCNLP (Short Papers)*, pages 322–332. ACL.
- Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: prompting argument interaction for event argument extraction. In *ACL*, pages 6759–6774. ACL.
- Sneha Mehta, Huzefa Rangwala, and Naren Ramakrishnan. 2022. [Improving zero-shot event extraction via sentence simplification](#). *CoRR*, abs/2204.02531.

- Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *NAACL-HLT*, pages 300–309. ACL.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *AAAI*, pages 1044–1049. AAAI Press.
- Richard Shin, Christopher H. Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained language models yield few-shot semantic parsers. In *EMNLP*, pages 7699–7715. ACL.
- Zhiyi Song, Ann Bies, Stephanie M. Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: annotation of entities, relations, and events. In *The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, EVENTS@HLP-NAACL*, pages 89–98. ACL.
- Beth Sundheim. 1992. Overview of the fourth message understanding evaluation and conference. In *MUC*, pages 3–21.
- David Wadden, Ulme Wennberg, Yi Luan, and Hananeh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *EMNLP-IJCNLP*, pages 5783–5788. ACL.
- Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. <https://github.com/kingoflolz/mesh-transformer-jax>.
- Sijia Wang, Mo Yu, Shiyu Chang, Lichao Sun, and Lifu Huang. 2022. Query and extract: Refining event extraction as type-oriented binary decoding. In *Findings of ACL*, pages 169–182. ACL.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP (Demos)*, pages 38–45. ACL.
- Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot label-aware event trigger and argument classification. In *Findings of ACL/IJCNLP*, pages 1331–1340. ACL.
- Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intell.*, 1(2):99–120.
- Weizhong Zhao, Jinyong Zhang, Jincai Yang, Tingting He, Huifang Ma, and Zhixin Li. 2021. A novel joint biomedical event extraction framework via two-level modeling of documents. *Inf. Sci.*, 550:27–40.

A Comparison between Different Prefix Prompts

In this section, we conduct experiments on ACE-2005 dataset to compare the effectiveness of using different prefix prompts in our models. We compare the following prefix prompts with the one discussed in Section 2.2: (1) “**This is a [] event whose trigger is "[]".**”; (2) “**The event type is [], and its occurrence is most clearly expressed by "[]".**”; (3) “**The event type is [] and the trigger is "[]".**”. The results are shown in Table 5, where “Prefix (0)” refers to the prefix prompt discussed in Section 2.2, whereas “Prefix (1)” refers to the first prefix prompt described in this section, and so on. From the table we can see that the prefix

Prefix Prompt	F1
Prefix (0)	66.1
Prefix (1)	65.2
Prefix (2)	65.6
Prefix (3)	63.0

Table 5: Performance of different prefix prompts.

prompt described in Section 2.2 is the most effective one, which might be due to the fact that the prefix prompt not only is based on the definitions of events and triggers (Grishman et al., 2005), but also has a natural and smooth expression.

B Results of all Other Global Constraints

In this section, we present the results of all other global constraints. The results are shown in Table 6.

Global constraint	Effect on overall performance
There is at most one Time-Arg in each event.	0.4
There is at most one Place-Arg in each event.	0.1
A TRANSPORT event has at most one Destination argument.	-0.2
A TRANSPORT event has at most one ORIGIN argument.	-0.1
A START-POSITION event has at most one Person argument.	0.2
A START-POSITION event has at most one Entity argument.	-0.1
A START-POSITION event has at most one Position argument.	0.1
A End-POSITION event has at most one Person argument.	-0.2
If a Start-Position event and an End-Position event share arguments, then Start-Position.Person is the same as End-Position.Person, and Start-Position.Entity is the same as End-Position.Entity, Start-Position.Position is the same as End-Position.Position.	0.1
If an Arrest-Jail event and a Charge-Indict event share arguments, Arrest-Jail.Person is the same as Charge-Indict.Defendant, they share the same Crime argument.	0.3
If a Die event and an Attack event share arguments, then Die.Place is the same as Attack.Place, Die.Victim is the same as Attack.Target, Die.Instrument is the same as Attack.Instrument, Die.Time is the same as Attack.Time, Die.Agent is the same as Attack.Attacker.	-0.2

Table 6: Other global constraints and corresponding effects on overall performance.

Distillation of encoder-decoder transformers for sequence labelling

Marco Farina^{*1} Duccio Pappadopulo^{*1} Anant Gupta¹

Leslie Huang¹ Ozan İrsoy¹ Thamar Solorio^{†1,2}

¹Bloomberg ²Department of Computer Science, University of Houston
{mfarina19, dpappadopulo, agupta968, lhuang328, oirsoy}@bloomberg.net
tsolorio@uh.edu

Abstract

Driven by encouraging results on a wide range of tasks, the field of NLP is experiencing an accelerated race to develop bigger language models. This race for bigger models has also underscored the need to continue the pursuit of practical distillation approaches that can leverage the knowledge acquired by these big models in a compute-efficient manner. Having this goal in mind, we build on recent work to propose a hallucination-free framework for sequence tagging that is especially suited for distillation. We show empirical results of new state-of-the-art performance across multiple sequence labelling datasets and validate the usefulness of this framework for distilling a large model in a few-shot learning scenario.

1 Introduction

Sequence labelling (SL) can be defined as the task of assigning a label to a span in the input text. Some examples of SL tasks are: i) named entity recognition (NER), where these labelled spans refer to people, places, or organizations, and ii) slot-filling, where these spans or slots of interest refer to attributes relevant to complete a user command, such as *song name* and *playlist* in a dialogue system. In general, these spans vary semantically depending on the domain of the task.

Despite the strong trend in NLP to explore the use of large language models (LLMs) there is still limited work evaluating prompting and decoding mechanisms for SL tasks. In this paper we propose and evaluate a new inference approach for SL that addresses two practical constraints:

- **Data scarcity:** The lack of vast amounts of annotated, and sometimes even the lack of unlabelled data, in the domain/language of interest.

- **Restricted computing resources at inference time:** LLMs are very effective, but deploying them to production-level environments is expensive, especially in contexts with latency constraints, such as in a live dialogue system.

Data scarcity leads us to consider high-performing encoder-decoder based LLMs. We address deployment concerns by considering distillation of such models into much smaller SL architectures, for instance Bi-Directional Long Short Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997) units, through the use of both labelled and unlabelled data.

A standard distillation approach, knowledge distillation (KD) (Hinton et al., 2015), requires access to the probability that the teacher network assigns to each of the possible output tags. This probability distribution is typically unavailable at inference time for LLMs; thus, distillation of encoder-decoder models needs to resort to pseudo-labels:¹ the student is trained on the one-hot labels that the teacher assigns to examples in an unlabelled dataset. This prevents the student model from learning those relationships among the probabilities of the incorrect classes that the teacher has learned. Similar arguments apply to decoder-only models.

In this paper, we propose **SenT'**, a simple modification of the *Simplified Inside Sentinel+Tag* (**SenT**) format by Raman et al. (2022). We combine our target sequence format with a scoring mechanism for decoding, which we collectively call **SenTScore**. This combination results in an effective framework that allows us to employ a language model to perform sequence labelling and knowledge distillation. We show that **SenTScore** is an hallucination-free

¹In this paper, we refer to distillation with pseudo-labels as the process by which a student model is trained on the one-hot labels (and only those labels) generated by a teacher model on an unlabeled dataset. We wish to distinguish this from KD, in which the probability distribution over labels is also used. See also Shleifer and Rush (2020).

*Equal contribution

† Research completed during sabbatical at Bloomberg

Original text										
play	wow	by	jon	theodore						
Encoder input for SenT' format										
<extra_id_0>	play	<extra_id_1>	wow	<extra_id_2>	by	<extra_id_3>	jon	<extra_id_4>	theodore	<extra_id_5>
Expected decoder output for SenT' format										
<extra_id_0>	O	<extra_id_1>	TRACK	<extra_id_2>	O	<extra_id_3>	ARTIST	<extra_id_4>	I	<extra_id_5>

Table 1: An example of how an original input text (from the SNIPS dataset) is transformed into the SenT' input for the model, and the format for the expected output. We use the explicit form of the special token strings used by T5. The addition of the extra token at the end of the input differentiates SenT' from SenT. Notice the modified BIO scheme (sBIO) that we use in our experiments: a unique I tag is used for each of the output tags; so if the original tag set is T , the tags generated by the model are $\bar{T} \equiv T \cup \{I, O\}$.

decoding scheme, and that even with smaller models it outperforms the original **SenT** format across a variety of standard SL datasets.

Our proposed **SenTScore** method defines a sequence of scores over the output tags that can be aligned with those generated by the sequence tagging student network, making KD possible. We find an advantage in terms of performance in using KD as opposed to just pseudo-labels as a distillation objective, especially for smaller distillation datasets.

In sum, our contributions are:

- A new, hallucination-free, inference algorithm for sequence labelling with encoder-decoder (and possibly decoder only) transformer models, **SenTScore**, that achieves new state-of-the-art results on multiple English datasets.
- Empirical evidence showing an advantage of **SenTScore** when distilling into a smaller student model. This approach is particularly promising in the few-shot setting, which makes it even more appealing and practical.

2 Related work

Using LLMs to perform sequence tagging is discussed by Athiwaratkun et al. (2020); Yan et al. (2021); Paolini et al. (2021); Qin and Joty (2021); Xue et al. (2022) and Raman et al. (2022). While these previous works have minor differences in the prompting format of the models, all but the last one include input tokens as part of the target sequence. Different from our work, all previous models are prone to hallucinate.

Distillation refers to training a small student model from scratch using supervision from a large pretrained model (Bucilua et al., 2006; Hinton et al., 2015). Distillation of transformer-based models for different NLP tasks is typically discussed in the context of encoder-only models (e.g. Tang et al.,

2019; Mukherjee and Hassan Awadallah, 2020; Jiao et al., 2020), with a few exceptions looking at distillation of decoder-only models (e.g. Artetxe et al., 2021).

In this paper we will discuss two approaches to distillation: *pseudo-labels* and *knowledge distillation* (KD). In the first approach the student model is trained on the hard labels generated by the teacher on some (unlabelled) dataset. In the second approach additional soft information provided by the teacher is used: typically the probability distribution the teacher assigns to the labels.

In the context of sequence labelling, using pseudo-labels allows us to perform distillation on any teacher-student architecture pair. KD, on the other hand, requires access to the teacher's probability distribution over the output tags. These are not usually available in language models for which the output distribution is over the whole vocabulary of tokens. We are not aware of other works which modify the decoder inference algorithm to generate such probabilities. However, there is recent work distilling internal representations of the teacher model, with the most closely related work to us being Mukherjee and Hassan Awadallah (2020). In that work the authors distill a multilingual encoder-only model into a BiLSTM architecture using a two-stage training process. This two-stage process, however, assumes a large unlabelled set for distilling internal model representations, embedding space, and teacher logits, and another significant amount of labelled data for directly training the student model using cross-entropy loss.

3 Datasets

We select seven English datasets that have been used in recent work on slot labelling: ATIS (Hemphill et al., 1990), SNIPS (Coucke et al., 2018), MIT corpora (Movie, MovieTrivia, and

Restaurant)², and the English parts of mTOP (Li et al., 2021) and of mTOD (Schuster et al., 2019). Some statistics about the datasets are shown in Table 2. Some of these datasets (ATIS, SNIPS, mTOP, and mTOD) come from dialogue-related tasks, while the MIT ones have been used for NER.

We use the original training, development, and test sets of the SNIPS, mTOP, and mTOD datasets. For the ATIS dataset we use the splits established in the literature by Goo et al. (2018), in which a part of the original training set is used as the dev set. Similarly, we follow Raman et al. (2022)³ to obtain a dev set out of the original training set for each of the MIT datasets.

We notice that all datasets, with the exception of MovieTrivia, contain some duplicates. Among these, all apart from Restaurant contain examples in the test set that are also duplicated in the train and dev sets. This happens for fewer than 30 instances, with the exception of mTOD, where more than 20% of the test set examples are also found in the train and dev sets. How these duplicates are handled varies across the literature; we do not remove duplicates from the datasets used in our main results.

However, for mTOD, we also obtained results on a version of the dataset that was deduplicated as follows: If an example is duplicated, we retain it in the highest priority (defined below) split and removed from the others. To ensure the test set is as close as possible to the original test set, we order the splits in ascending order of priority as follows: test, dev, and train. We found that the F1 scores on the deduped mTOD dataset are within 0.5 points those on the original mTOD dataset across all experiments; as such, we do not report the deduped results in the following sections.

In addition to covering different domains, there are noticeable differences across the datasets in terms of the number of tags and the number of labelled examples for evaluation and testing, as can be seen in Table 2. This set of seven datasets allows us to gather robust empirical evidence for the proposed work that we present in what follows.

4 Score-based sequence labelling

Using LLMs for sequence tagging requires reframing the problem as a sequence-to-sequence task. In

²The MIT datasets were downloaded from: <https://groups.csail.mit.edu/sls/>

³Private communication with authors

Datasets	# tags	# train	# dev	# test
ATIS	83	4478	500	893
SNIPS	39	13084	700	700
MovieTrivia	12	7005	811	1953
Movie	12	8722	1053	2443
Restaurant	8	6845	815	1521
mTOP (en)	75	15667	2235	4386
mTOD (en)	16	30521	4181	8621

Table 2: Number of examples per partition and number of unique tags in the SL datasets we used.

Raman et al. (2022), the strategy that proved the most effective, at least when applied to the mT5 encoder-decoder architecture, was the *Simplified Inside Sentinel+Tag* (**SenT** in this paper). In this format (see Table 1), the original text is first tokenized according to some pretokenization strategy (whitespace splitting for all the datasets considered), and each of the tokens is prepended with one of the extra token strings provided by mT5 (the *sentinel* tokens). The resulting concatenation is then tokenized using the mT5 tokenizer and fed to the encoder-decoder model. The output that the decoder is expected to generate is the same input sequence of special token strings, which are now alternated with the tags corresponding to the original tokens.

Given the set T of string labels to be used to annotate a span of text, the scheme used to associate tags across tokens is a modification of the standard BIO scheme: we use $t \in T$ for any token that starts a labelled span, a single tag I for each token that *continues* a labelled span, and O to tag tokens that do not belong to labelled spans. We refer to this scheme as *Simplified Inside BIO* (sBIO), and we indicate with $\bar{T} \equiv T \cup \{I, O\}$ the tag set associated to it.

Raman et al. (2022) argue that the success of SenT can be attributed to two factors: 1) on the one hand, the use of sentinel tokens mimics the denoising objective that is used to pretrain mT5; 2) on the other hand, when compared to other decoding strategies, SenT does not require the decoder to copy parts of the input sentences and also produces shorter outputs. Both these facts supposedly make the task easier to learn and reduce the number of errors from the decoder (*hallucinations*, as they are often referred to in the literature).

We remark however that any output format among those described in the literature can be made completely free of hallucinations by constraining

decoding (either greedy or beam search based) through a finite state machine enforcing the desired output format (see for instance De Cao et al., 2020). In what follows we describe our proposed decoding approach that builds on this previous work.

4.1 SenTScore

Regardless of possible constraints imposed during generation, both **SenT** and the other algorithms described in Raman et al. (2022) use the decoder autoregressively at inference time to generate the target sequence. Since generation proceeds token by token and the textual representation of a tag is a variable length sequence of tokens, it is nontrivial to extract the scores and probabilities that the model assigns to individual tags.

We propose a different approach to inference, one in which the decoder is used to score sequences of tags. For this purpose, we consider a sequence tagging task with a label set T , and the associated sBIO tag set \bar{T} . Given an input sentence S , we use a pre-tokenizer (such as whitespace splitting) to turn S into a sequence of token strings $x_1 \dots x_L$, of size L . The SenT format is obtained by interleaving these tokens with special token strings to obtain the input string $S_{\text{in}} = s_0 x_1 s_1 \dots x_L$. We use juxtaposition to indicate string concatenation. In what follows, we will work with **SenT'**, a modification of SenT in which an additional special token is appended at the end, $S_{\text{in}} \leftarrow S_{\text{in}} s_L$. The reason for doing this will become clear in what follows.

The valid output strings that can be generated by the decoder are the $|\bar{T}|^L$ sequences of the form $S_{\text{out}} = s_0 t_1 s_1 \dots t_L s_L \in \mathcal{O}$ where $t \in \bar{T} \equiv T \cup \{I, O\}$ consistent with the sBIO scheme convention. The encoder-decoder model can be used to calculate the log-likelihood of each of such strings $\log \mathcal{L}_\theta(S_{\text{out}}; S_{\text{in}})$, where θ represents the model parameters, and the best output will be:

$$S_{\text{out}}^* = \arg \max_{S \in \mathcal{O}} \log \mathcal{L}(S; S_{\text{in}})$$

Exact inference is infeasible but can be approximated using beam search as described in Algorithm 1. The outputs of the algorithm are the top-K output strings and the score distribution associated with each of the output tags. As is evident from Table 1, it is simple to map back the final output string S^* to the sequence of output tags and labelled spans.

At decoding time the output string is initialized with the first sentinel token s_0 . At the i -th step,

Algorithm 1 SenTScore beam search

Require: Encoder-decoder parameters θ , input S_{in} with L tokens, sBIO tag set \bar{T} , beam size K

Ensure: Approximate top- K output sequences $\mathcal{B}_{\text{text}}$ and their sBIO tag scores, $\mathcal{B}_{\text{scores}}$

```

 $\mathcal{B}_{\text{text}} \leftarrow [s_0]_{i=1 \dots K}$ 
 $\mathcal{B}_{\text{scores}} \leftarrow [[]]_{i=1 \dots K}$ 
for  $i = 1$  to  $L$  do
   $\mathcal{H} \leftarrow [z t s_i]_{z \in \mathcal{B}_{\text{text}}, t \in \bar{T}}$  ▷ Generate hypotheses
   $\mathcal{S} \leftarrow [\log \mathcal{L}_\theta(h; S_{\text{in}})]_{h \in \mathcal{H}}$  ▷ Score hypotheses
   $\Pi \leftarrow \text{K-argsort } \mathcal{S}$  ▷ top-K args
   $\mathcal{B}_{\text{text}} \leftarrow \text{TAKE}(\mathcal{H}; \Pi)$  ▷ Update text beam
   $\tilde{\mathcal{S}} \leftarrow \text{RESHAPE}(\mathcal{S}; K, |\bar{T}|)$  ▷ Reshape scores
  for  $k = 0$  to  $K - 1$  do ▷ Update score beam
     $\tilde{k} \leftarrow \Pi[k] \bmod K$ 
     $\mathcal{B}_{\text{scores}}[k] \leftarrow \text{APPEND}(\mathcal{B}_{\text{scores}}[\tilde{k}]; \mathcal{S}[\tilde{k}])$ 
  end for
end for
return  $\mathcal{B}_{\text{text}}, \mathcal{B}_{\text{scores}}$ 

```

SenTScore uses the model likelihood to score each of the $|\bar{T}|$ possible continuations of the output sequence

$$t s_i \text{ with } t \in \bar{T}, \quad (1)$$

picks the highest scoring one, and keeps track of the score distribution. s_i in Eq. 1, the *next* sentinel token, plays the crucial role of an EOS token at each step. This is needed to normalize the probability distribution: the likelihood of the string $s_0 t_1 \dots s_{k-1} t'_k$ is always bounded by that of the string $s_0 t_1 \dots s_{k-1} t_k$ if t is a prefix of t' , and we would never predict t' as a continuation of $s_0 t_1 \dots s_{k-1}$. This explains why we prefer using **SenT'** over **SenT**.

Finally, while **SenTScore** changes the inference algorithm, the finetuning objective we use throughout is still the original language modelling one.

4.2 Distillation

The main advantage of **SenTScore** is in the distillation setting. At each inference step, the algorithm assigns a likelihood to each sBIO tag. This distribution can be used to train the student network by aligning it to the teacher's pre-softmax logits, in a standard knowledge distillation setup.

In detail, given an input sequence S_{in} , let $(\mathbf{y}_i^*)_{i=1 \dots L}$ be the sequence of sBIO output tags (as $|\bar{T}|$ -dimensional one-hot vectors) as inferred by the teacher model, and let $(\mathbf{u}_i^*)_{i=1 \dots L}$ (also $|\bar{T}|$ -dim. vectors) be the associated sequence of log-likelihoods. We indicate with \mathbf{p}_i^* the probability obtained by softmaxing \mathbf{u}_i^* and by \mathbf{q}_i the output of the softmax layer from the student. The contribution of each of the tags to the distillation objective

that we use to train the student sequence tagger is

$$-\sum_k (y_i^*)_k \log (p_i^*)_k + \lambda_{KL} KL(\mathbf{p}_i^* || \mathbf{q}_i). \quad (2)$$

The first term is the standard cross-entropy contribution from the pseudo-labels, while the second is the knowledge distillation term, implemented with a KL divergence with λ_{KL} its associated positive weight.

We stress that we are allowed to write the second term only because **SentScore** provides us with the tag scores. This is not the case for any of the formats proposed in [Raman et al. \(2022\)](#) or, as far as we know, elsewhere.⁴

5 Experimental settings

We evaluate the models by computing the **F1** score on the test set of each dataset. **F1** is calculated following the CoNLL convention as defined by [Tjong Kim Sang and De Meulder \(2003\)](#), where an entity is considered correct iff the entity is predicted exactly as it appears in the gold data. We show micro-averaged **F1** scores.

The first set of experiments we performed are intended to investigate whether our proposed **SentScore** approach is competitive with respect to recent results on the same datasets (Table 3). Our **SentScore** model is a pretrained T5-base model (220M parameters) finetuned on each of the datasets.⁵ We trained each model for 20 epochs, with patience 5, learning rate of 10^{-3} , and batch size 32. We also want to know how the proposed framework compares against the following strong baselines:

BiLSTM: Our first baseline is a BiLSTM tagger ([Lample et al., 2016](#)).⁶ The BiLSTM has a hidden dimension of size 200. Its input is the concatenation of 100d pretrained GloVe6B embeddings ([Pennington et al., 2014](#)) from [StanfordNLP](#) with the 50d hidden state of a custom character BiLSTM. We trained each model for 100 epochs, with patience 25, learning rate of 10^{-3} , and batch size 16.

BERT: We finetune a pretrained BERT-base cased

⁴Strictly speaking the student defines $p(\cdot | t_1^* \dots t_{i-1}^*; S_{in})$ (star means predicted) while \mathbf{q}_i corresponds to $p(s_0 t_1^* \dots t_{i-1}^* s_{i-1} \cdot | S_{in})$. This discrepancy is resolved by the invariance of the softmax under constant shifts of its arguments.

⁵All our results are in the greedy setting. We find very small differences in performance by using beam search, while inference time grows considerably.

⁶We do not include a CRF layer.

model ([Devlin et al., 2019](#)) (110M parameters) for the SL task and report results for each of the seven datasets. While we consider BERT a baseline model, we note that this pretrained architecture continues to show good performance across a wide range of NLP tasks, and for models in this size range BERT is still a reasonable choice. In preliminary experiments we compared results from the case and uncased versions of BERT and we found negligible differences. We decided to use the cased version for all experiments reported here. We trained each model for 30 epochs, with patience 10, learning rate of 5×10^{-5} , and batch size 64.

SentT': The pretrained model is the same as that used for **SentScore**. The goal of this baseline is to assess improvements attributed to our proposed decoding mechanism. This model is also the closest model to prior SOTA. The main difference between our results and those in [Raman et al. \(2022\)](#) is the pretrained model. They used a multilingual T5 model ([Xue et al., 2021](#)) with 580M parameters, whereas we use a smaller monolingual version ([Raffel et al., 2020](#)).

All the above models were trained with the AdamW optimizer ([Loshchilov and Hutter, 2017](#)). The best checkpoint for each training job was selected based on highest micro-F1 score on the validation set. All pretrained transformer models are downloaded from [Huggingface](#).

5.1 Distillation experiment

We apply **SentScore** and the loss function described in Section 4.2, to distill a finetuned T5 model into a BiLSTM architecture to perform sequence tagging. To mimic a low-resource setting, we randomly downsample the train/dev splits of all the datasets. We consider two sets of sizes for these gold train/dev splits: a 100/50 split and a 300/150 one. In both settings the remainder of the original training set is used for the distillation component using pseudo-labels.

We then finetune T5 using the **SentT'** format on each of these two gold splits. The resulting model is used as the teacher in a distillation setting in which the student is a BiLSTM. The BiLSTM student is trained on the full training set by using the downsampled gold labels, but pseudo-labels and scores generated by the T5 teacher using **SentScore** with $K = 1$ in the rest of the training data. We use a temperature parameter τ to rescale the distribution **SentScore** defines over \bar{T} . We use $\tau = 10$ in all the distillation experiments.

Dataset	BiLSTM [1M]		BERT [110M]		T5 [220M] (SenT')		T5 (SenTScore)		mT5 [580M] (SenT)	
	Perfect	F1	Perfect	F1	Perfect	F1	Perfect	F1	Perfect	F1
ATIS	89.06	95.56	88.57	95.27	86.56	94.77	89.81	95.99	<u>90.07</u>	95.96
SNIPS	87.24	95.02	89.71	95.47	89.86	95.43	91.00	96.07	89.81	95.53
MovieTrivia	32.41	69.81	36.2	69.15	36.35	70.76	39.58	71.99	<u>39.85</u>	<u>73.01</u>
Movie	69.79	86.72	69.46	85.83	71.88	87.53	74.29	88.35	72.74	87.56
Restaurant	58.32	77.39	58.97	77.69	58.65	78.77	63.77	80.91	62.93	80.39
mTOP (en)	81.10	88.94	84.4	90.98	84.18	90.64	86.66	92.29	86.56	92.28
mTOD (en)	91.70	95.62	92.35	95.83	92.24	96.04	92.94	96.24	<u>93.19</u>	<u>96.42</u>

Table 3: Our results comparing BERT-base and a BiLSTM against a T5-base model using SenT' and **SenTScore** on different SL datasets are shown in the first 4 columns. Number in square brackets are model sizes in terms of number of parameters. Results from [Raman et al. \(2022\)](#) are copied in the last column. Bold scores represent our best results; underlined scores in the last column highlight those cases in which [Raman et al. \(2022\)](#) outperforms us.

Dataset - F1	BiLSTM	BERT	T5	BiLSTM (distilled)	Dataset - F1	BiLSTM	BERT	T5	BiLSTM (distilled)
ATIS	79.93	79.43	85.01	86.75	ATIS	86.43	84.95	89.33	90.25
SNIPS	51.63	52.16	54.33	57.18	SNIPS	69.19	72.77	76.34	79.84
MovieTrivia	48.26	50.26	55.74	57.85	MovieTrivia	57.64	58.41	63.60	65.34
Movie	60.82	61.80	67.09	70.51	Movie	73.54	73.76	77.39	79.20
Restaurant	47.26	53.17	56.87	61.13	Restaurant	61.62	62.97	68.52	68.62
mTOP (en)	43.12	46.08	51.94	54.77	mTOP (en)	57.22	63.28	67.73	69.62
mTOD (en)	68.68	76.95	79.43	82.26	mTOD (en)	83.46	85.51	88.68	89.82

(a) Gold train/dev split of size 100/50

(b) Gold train/dev split of size 300/150

Table 4: Distillation results and comparisons with baselines. The distillation results use the full objective function in Eq. 2 with $\lambda_{KL} = 1$.

The training schedule we follow is the same we use to train the BiLSTM baseline model, with the only exception that the best checkpoint is selected on the reduced dev set.

6 Results

The comparisons between baselines, SenT', and **SenTScore** are shown in Table 3. **SenTScore** is used with a $K = 1$ beam size. Larger beams result in very similar performance and a considerable slowdown of inference time. **SenTScore** consistently outperforms SenT' with constrained decoding, and all other baselines. Our intuition is that one advantage of **SenTScore** comes from the fact that decoding happens tag-wise as opposed to token-wise (as in pure beam search). The last column of Table 3 shows the performance of the SenT implementation of [Raman et al. \(2022\)](#). **Perfect** scores are also reported for completeness. They are evaluated at the sentence level and correspond to the fraction of perfectly predicted examples. However these results are not directly comparable: [Raman et al. \(2022\)](#) use a different and larger model (mT5-base with 580M parameters) and different optimization details. Nevertheless **SenTScore** achieves bet-

ter performance in a majority of cases.

6.1 Distillation results

Tables 4a and 4b show the result of the distillation experiments with 100/50 and 300/150 train/dev gold splits, respectively. While a BiLSTM tagger trained on the gold data significantly underperforms a finetuned T5-base model, once the BiLSTM is distilled on the silver data generated using SenTScore, it outperforms even the original teacher model. We notice that the difference between student and teacher decreases for larger gold set size, suggesting that the effect is related to regularization properties of the distillation process. A similar phenomenon has been observed elsewhere, for instance in [Furlanello et al. \(2018\)](#) albeit with teacher and student sharing the same architecture.

In order to isolate the benefits of training the teacher model using KD as opposed to just pseudo-labels, we perform a set of ablation studies. For each dataset, we distill a BiLSTM student on a training set $\mathcal{T} = \mathcal{G} \cup \mathcal{S}$, where \mathcal{G} is the original gold set and \mathcal{S} is a random sample from the complement of \mathcal{G} . We choose $|\mathcal{S}| = 0, 250, 500$. The student is distilled using Eq. 2 with two choices of the loss

Dataset - F1	No silver		250 silver		500 silver	
	$\lambda_{KL} = 0$	$\lambda_{KL} = 1$	$\lambda_{KL} = 0$	$\lambda_{KL} = 1$	$\lambda_{KL} = 0$	$\lambda_{KL} = 1$
ATIS	79.93 (0.85)	82.35 (0.44)	83.09 (1.49)	84.42 (1.42)	83.75 (1.74)	85.10 (1.54)
SNIPS	51.63 (1.25)	55.65 (1.38)	54.34 (0.71)	56.02 (1.71)	55.66 (1.21)	57.00 (1.17)
MovieTrivia	48.26 (0.95)	51.86 (1.09)	53.11 (1.26)	55.19 (0.50)	53.97 (1.55)	56.00 (0.38)
Movie	60.82 (0.67)	64.20 (1.07)	67.04 (0.66)	69.41 (0.66)	67.73 (1.28)	70.12 (0.60)
Restaurant	47.26 (0.83)	50.19 (0.81)	54.24 (1.17)	56.20 (0.94)	56.29 (1.11)	57.95 (0.88)
mTOP (en)	43.12 (1.84)	46.43 (1.01)	46.68 (2.31)	49.08 (1.65)	49.57 (0.47)	50.33 (2.17)
mTOD (en)	68.68 (2.68)	70.40 (0.97)	76.12 (1.07)	77.86 (1.45)	77.86 (0.85)	79.77 (0.82)

Table 5: Distillation experiments with varying silver dataset size and ablation of the KD term in Eq. 2. The gold data split is the same as in Table 4a, with train/dev sizes of 100/50. The numbers in parentheses represent the standard deviation of the scores obtained by varying all the random seeds that appear at training time: BiLSTM weight initialization, batch scheduling, and the choice of the silver data set.

Dataset - F1	No silver		250 silver		500 silver	
	$\lambda_{KL} = 0$	$\lambda_{KL} = 1$	$\lambda_{KL} = 0$	$\lambda_{KL} = 1$	$\lambda_{KL} = 0$	$\lambda_{KL} = 1$
ATIS	86.43 (1.09)	88.42 (0.48)	89.15 (0.65)	89.39 (0.49)	89.73 (0.66)	89.98 (0.31)
SNIPS	69.19 (0.74)	73.06 (0.54)	72.11 (1.11)	75.02 (1.16)	73.99 (0.81)	75.73 (1.47)
MovieTrivia	57.64 (0.45)	60.30 (0.34)	60.25 (0.37)	62.11 (0.54)	61.38 (0.46)	62.89 (0.52)
Movie	73.54 (0.40)	76.30 (0.33)	75.88 (0.44)	76.96 (0.44)	76.52 (0.58)	77.58 (0.26)
Restaurant	61.62 (0.43)	63.78 (0.27)	64.33 (0.90)	65.33 (0.66)	65.20 (0.80)	66.24 (0.63)
mTOP (en)	57.22 (0.73)	60.36 (0.50)	60.81 (0.92)	62.70 (0.69)	62.02 (0.94)	64.37 (0.84)
mTOD (en)	83.46 (0.59)	85.52 (0.20)	86.35 (0.40)	87.08 (0.50)	86.82 (0.33)	87.89 (0.40)

Table 6: Distillation experiments with varying silver dataset size and ablation of the KD term in Eq. 2. The gold data split is the same as in Table 4b, with a train/dev size given by 300/150. All experimental details are common with Table 5.

multipliers: $\lambda_{KL} = 1$ and $\lambda_{KL} = 0$. The first setting is the same used in Tables 4a and 4b, while the second drops the KD loss and only keeps the pseudo-labels for distillation. Whenever pseudo-labels and scores are used, they are generated by the SenTScore algorithm.

The results are shown in Tables 5 and 6. We see a consistent trend in which KD outperforms training the student using only pseudo-labels. This in particular motivates SenTScore as an inference algorithm. The results also show that for our choice of teacher and student architectures, and datasets, the gap between KD and pseudo-labels is reduced when more silver data are used. Figure 1 further explores the relationship between amount of pseudo-labeled data and gains from KD with $|\mathcal{S}| = 0, 250, 500, 2000$. The trend with more pseudo-labeled data remains unchanged.

7 Limitations and future work

A reasonable critique to our focus on real-world constraints is the simple fact the datasets we are using are not real-world ones. From noise to tokenization choices, many issues arise when considering datasets outside of the academic domain. However,

we believe our methods are simple enough to be applicable to real-world scenarios and our results to be independent of these various subtleties.

Some issues that could be addressed in future work have to do with the exploration of even larger models and different architectures such as decoder-only ones (Radford et al., 2018, 2019; Brown et al., 2020; Zhang et al., 2022; Chowdhery et al., 2022; Black et al., 2021). We note, however, that in all our experiments we finetune all the weights of the pretrained models we use. When using extremely large models this becomes impractical. Recent work (Turc et al., 2019) suggests that KD with compact encoder-only student models, such as BERT, is a promising avenue for further research. Exploring the pure few-shot scenario, or only finetuning a subnetwork, for instance by using adapters à la Hously et al., 2019, would be also interesting.

8 Conclusion

Real-time systems need to find a trade-off between performances and computing resources, the latter constraint coming either from budget or some other service requirement. Such trade-offs become particularly evident with large pretrained transformer

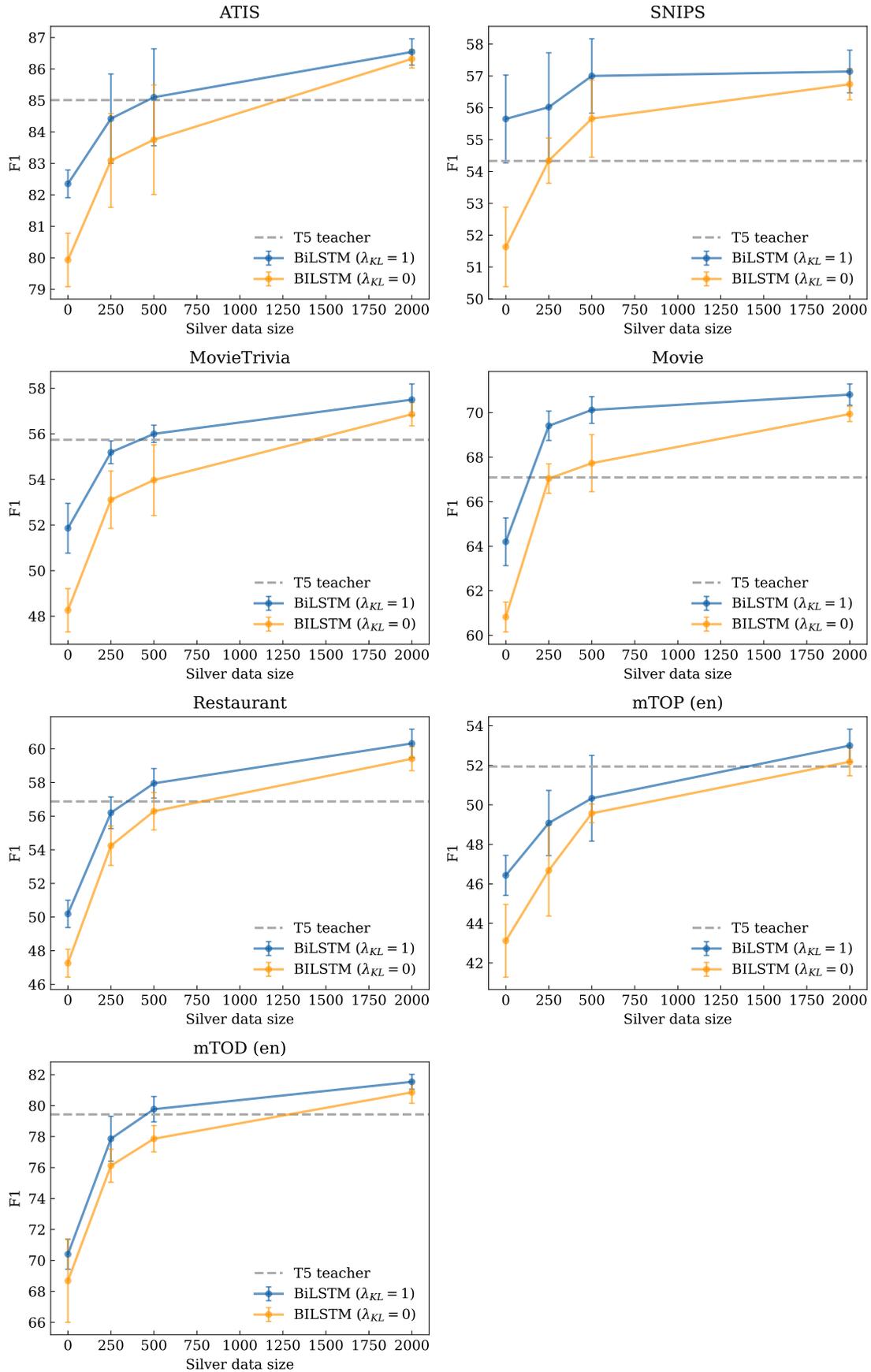


Figure 1: A graphical representation of the distillation results in Table 4a (100/50 gold train/dev split) as a function of the size of the silver dataset. Knowledge distillation using SenTScore generated scores outperforms pseudo-labels.

models, which achieve SOTA results on many NLP tasks at the cost of being extremely hard and expensive to deploy in a real-world setting.

The standard solution for this is distillation. In this paper we have revisited these issues for the SL task, which is often the first crucial step in many real-world NLP pipelines. We propose a new inference algorithm, SenTScore, that allows us to leverage the performance of arbitrarily large encoder-decoder transformer architectures by distilling them into simpler sequence taggers using KD as opposed to just pseudo-labelling.

Ethical considerations

The intended use of our proposed approach is related to sequence labelling tasks where there are latency constraints and limited labelled data available. While it is not impossible to identify potential misuses of this technology, it is not immediately clear what those malicious uses would be. On the contrary, this paper contributes to the body of work investigating efficient solutions for deployment of live systems.

Computing infrastructure and computational budget

All of our experiments were run on single V100 GPU machines with 32GB. The most expensive experiments relate to finetuning a model, including best checkpoint selection. In this case, the running time is directly related to the dataset size. For the experiments using the full train/dev set, running time varies from 45 minutes (mATIS corpus) to a few hours (mTOD corpus) for a T5-base model. Training a model takes, on average, around 4 iterations per second with batch size 32. For the generation of pseudo-labels, we did not implement batch processing and it takes around 0.15 seconds to annotate each sample.

References

- SLS corpora. <https://groups.csail.mit.edu/sls/downloads/>. Accessed: 2022-09-09.
- Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giri Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O’Horo, Jeff Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Ves Stoyanov. 2021. [Efficient large scale language modeling with mixtures of experts](#).
- Ben Athiwaratkun, Cicero Nogueira dos Santos, Jason Krone, and Bing Xiang. 2020. [Augmented natural language for generative sequence labeling](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 375–385, Online. Association for Computational Linguistics.
- Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. [GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow](#). If you use this software, please cite it using these metadata.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Matheus Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. [Model compression](#). In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’06*, page 535–541, New York, NY, USA. Association for Computing Machinery.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. [Palm: Scaling language modeling with pathways](#). *arXiv preprint arXiv:2204.02311*.
- Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. [Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces](#).
- Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. [Autoregressive entity retrieval](#).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

- Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. [Born again neural networks](#).
- Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. [Slot-gated modeling for joint slot filling and intent prediction](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.
- Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. [The ATIS spoken language systems pilot corpus](#). In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the knowledge in a neural network](#).
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#).
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for NLP](#). In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Huggingface. [Models - Hugging Face](#).
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. [TinyBERT: Distilling BERT for natural language understanding](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [Neural architectures for named entity recognition](#).
- Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. [MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2017. [Fixing weight decay regularization in adam](#). *ArXiv*, abs/1711.05101.
- Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. [XtremeDistil: Multi-stage distillation for massive multilingual models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2221–2234, Online. Association for Computational Linguistics.
- Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. [Structured prediction as translation between augmented natural languages](#).
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Chengwei Qin and Shafiq Joty. 2021. [Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5](#).
- Alec Radford, Karthik Narasimhan, and Tim Salimans and Ilya Sutskever. 2018. [Improving language understanding by generative pre-training](#). <https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf>.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). Technical report, OpenAI.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Karthik Raman, Iftexhar Naim, Jiecao Chen, Kazuma Hashimoto, Kiran Yalasang, and Krishna Srinivasan. 2022. [Transforming sequence tagging into a seq2seq task](#).
- Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. [Cross-lingual transfer learning for multilingual task oriented dialog](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sam Shleifer and Alexander M. Rush. 2020. [Pre-trained summarization distillation](#).
- StanfordNLP. [GloVe: Global Vectors for Word Representation](#).
- Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. [Distilling task-specific knowledge from bert into simple neural networks](#).
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition](#). In

Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, pages 142–147.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Well-read students learn better: On the importance of pre-training compact models](#).

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. [Byt5: Towards a token-free future with pre-trained byte-to-byte models](#). *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. [A unified generative framework for various NER subtasks](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5808–5822, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [Opt: Open pre-trained transformer language models](#).

Predicting Desirable Revisions of Evidence and Reasoning in Argumentative Writing

Tazin Afrin and Diane Litman
University of Pittsburgh
Pittsburgh, Pennsylvania 15260
{taa74, dlitman}@pitt.edu

Abstract

We develop models to classify desirable evidence and desirable reasoning revisions in student argumentative writing. We explore two ways to improve classifier performance – using the essay context of the revision, and using the feedback students received before the revision. We perform both intrinsic and extrinsic evaluation for each of our models and report a qualitative analysis. Our results show that while a model using feedback information improves over a baseline model, models utilizing context - either alone or with feedback - are the most successful in identifying desirable revisions.

1 Introduction

Successful essay writing by students typically involves multiple rounds of revision and assistance from teachers, peers, or automated writing evaluation (AWE) systems. Natural language processing (NLP) has become a key component of AWE systems, with NLP being used to assess the content and structure of student writing and to automatically provide formative feedback (Beigman Klebanov and Madnani, 2020; Zhang et al., 2016; Writing Mentor, 2016; Wang et al., 2020). While some students produce revised texts that are in line with the feedback automatically generated by a system or provided by other humans, other students either ignore the feedback or are unsuccessful in their feedback implementation attempts (Wang et al., 2020). Hence, analyzing student revisions in terms of their *desirability for improving essay quality* is important. The development of AWE systems that leverage NLP to analyze a revision’s alignment to feedback messages is one approach to convey to students a sense of a good revision direction.

Our research focuses on the *automatic classification of desirable and undesirable revisions of evidence use and reasoning*¹ in argumentative writ-

¹Such revisions of text *content* are generally considered most important in revising (Faigley and Witte, 1981).

ing. Argumentative writing is a skill that students need to develop to be strong writers and learners. By evidence use, we refer to examples and details that students use to support an argument. By reasoning, we refer to how evidence is explained and linked to an overall argument. Desirable revisions (e.g., add relevant evidence) are student revisions that have hypothesized utility in improving an essay in response to feedback (e.g., add more evidence), while undesirable revisions (e.g., add irrelevant evidence) do not have such hypothesized utility.

Table 1 shows example desirable and undesirable revisions of evidence and reasoning from original to revised drafts of an essay aligned at the sentence-level. In response to the feedback shown at the top of Table 1, the student adds both reasoning and evidence. Sentences 3, 5, and 9 are added desirable reasoning, desirable evidence, and undesirable reasoning respectively. The student also modified fluency in other sentences which is not shown here. Sentences 1, 4, and 7 are identical in both drafts.

In this paper, we first describe the labeling of desirable and undesirable revisions in three existing corpora of evidence and reasoning revisions. We then describe a baseline model and enhanced models using context and feedback information to predict revision desirability. Finally, we present results from intrinsic and extrinsic evaluations to demonstrate the utility of our enhanced models.

2 Related Work

NLP research on revision analysis primarily focuses on two domains: Wikipedia and academic writing. Studies in Wikipedia revisions focused on error correction, paraphrase or vandalism detection (Daxenberger and Gurevych, 2012), factual versus fluency edits (Bronner and Monz, 2012), semantic edit intention (Yang et al., 2017), etc. In academic writing, revision studies have instead focused on defining revisions purpose tailored to argumentative writing (Zhang and Litman, 2015;

Feedback message: "...Explain how the evidence helps to make your point... ... Tie the evidence not only to the point you are making within a paragraph, but to your overall argument..."			
	Original Draft	Revised Draft	Revision
1.	The author convinced me by saying in the passage that, "The plan is to get people out of poverty, assure them access to health care and help them stabilize the economy and quality of life in their communities."	The author convinced me by saying in the passage that, "The plan is to get people out of poverty, assure them access to health care and help them stabilize the economy and quality of life in their communities."	No-change
2.		...	
3.		They can do that by assuring that the people of Sauri, Kenya have food, water, liter, and a place to stay.	Added Desirable Reasoning
4.	Also, in paragraph 3 it says, "The goals are supposed to be met by 2025; some other targets are set for 2035."	Also, in paragraph 3 it says, "The goals are supposed to be met by 2025; some other targets are set for 2035."	No-change
5.		If the plans are going to be achieved in 2025 than their plans will be achieved in only 7 more years which would be in our life time.	Added Desirable Evidence
6.	
7.	Since so many people weren't fighting against poverty in 2010 people were being sent to the hospital and not even being treated cause they didn't have the money so, so many people died.	Since so many people weren't fighting against poverty in 2010 people were being sent to the hospital and not even being treated cause they didn't have the money so, so many people died.	No-change
8.	
9.		The kids and their families didn't have the money but but this supports my evidence by talking about how the kids don't go to school it's because them and their family are in poverty.	Added Undesirable Reasoning

Table 1: Example of revisions extracted from an essay from our elementary-school dataset.

Kashefi et al., 2022) and understanding the pattern of revisions (Afrin and Litman, 2019; Shibani et al., 2018). Exploring the pattern of iterative revision have also been studied in scientific writing (Du et al., 2022). While there have been some attempts at defining revisions in terms of their *quality* (e.g., vagueness of Wikipedia edits (Debnath and Roth, 2021), statement strength in scientific writing (Tan and Lee, 2014), quality of claims in online debate (Skitalinskaya et al., 2021), and improvement in argumentative writing (Afrin and Litman, 2018)), they fail to incorporate feedback students were provided. Afrin et al. (2020) is the first study that touched on student revisions in terms of their utility in improving the essay with respect to automated feedback messages. However,

their framework was applied to one dataset and they did not investigate state-of-the-art models for automatic classification. In this work, we focus on a simplified binary classification task to distinguish between desirable and undesirable revisions in student argumentative writing, and particularly explore the utility of two predictors of revision desirability - *context and feedback*. We also apply our model on *multiple student corpora*.

Previous revision classification approaches either do not create contextual features (Daxenberger and Gurevych, 2013; Zhang and Litman, 2015), or the context features represent only shallow information such as 'location' (Zhang and Litman, 2015). Zhang and Litman (2016) incorporated context by using cohesion blocks focusing on adjacent sen-

Datasets	#Students	Grade Level	Feedback Source	Essay Drafts Used	Essay Score Range	Improvement Score Range
Elementary	143	5 th & 6 th	AWE	1 and 2	[1, 4]	[0, 3]
High-school	47	12 th	peer	1 and 2	[0, 5]	[-2, +3]
College	60	college	X	2 and 3	[15, 33]	-1, +1

Table 2: Comparison of datasets used in this study (X = Not available).

Data	Example Feedback
Elementary (AWE generated)	Explain the evidence: Tell your reader why you included each piece of evidence. Explain how the evidence helps to make your point. Explain how the evidence connects to the main idea & elaborate: Tie the evidence not only to the point you are making within a paragraph, but to your overall argument. Elaborate. Give a detailed and clear explanation of how the evidence supports your argument.
High-school (peer feedback)	for the spendthrifts and the hoarders, you used a good example for spendthrifts but im confused on where you example for hoarding is. if it is mike tyson, i think you should include more detail about that. your fifth circle could use more detail as to what exactly made him hate man, because im confused about the story.

Table 3: Examples of feedback messages from elementary and high-school data.

tences of the target revision, and sequence labeling to utilize the interdependent revisions. Inspired by this work, we propose a new approach to extract longer context information.

Prior studies of revision quality in writing have not considered feedback students receive before revision when defining an annotation scheme (Tan and Lee, 2014; Afrin and Litman, 2018), or have not explored the benefit of using feedback during classification (Afrin et al., 2020). We leverage both pre-defined AWE feedback messages and free form peer feedback in identifying desirable revisions.

Previous studies have explored revision generation for argument writing task (Ito et al., 2019) and paraphrase generation tasks (Mu and Lim, 2022). However, state-of-the-art language models are not leveraged for revision classification task. The pre-trained Bidirectional Encoder Representations from Transformer (BERT) (Devlin et al., 2019) model has shown to be effective in various NLP models including sentence classification and sentence-pair classification. BERT has also produced excellent results in various argument mining tasks (Chakrabarty et al., 2019; Reimers et al., 2019; Ghosh et al., 2021). In this work, we leverage the standard pre-trained BERT model (bert-based-uncased) (Devlin et al., 2019) to create the model for our revision classification task.

3 Data and Resources

Our data consists of three corpora of paired drafts of argumentative essays, written in response to a prompt and revised in response to feedback. A comparison of the data is shown in Table 2. The diversity of the corpora along multiple dimensions helps ensure the utility of our proposed models.

The *elementary* school students wrote Draft1 about an article on a project in Kenya, then received AWE system feedback focused on students’ use of text evidence and reasoning (selected based on automatic scoring). An example of the feedback messages is shown in Table 3. All essay pairs were later graded on a scale from 0 to 3 to indicate improvement from Draft1 to Draft2 in line with the feedback (kappa = 0.77) (Wang et al., 2020).

The *high-school* students wrote Draft1 in response to a prompt about Dante’s Inferno (Zhang and Litman, 2015), then received peer feedback along 6 rubric dimensions (e.g., evidence, organization, etc.). We only utilize feedback about evidence in this work (shown in Table 3), because it is closely related to the revisions we are considering. Drafts 1 and 2 of each high-school essay were separately graded by expert graders. We create an improvement score for each essay pair, calculated as the difference of the holistic score between drafts.

The *college* essays were written by 60 students on technology proliferation (Zhang et al., 2017). Students received general feedback after Draft1,

	Desirable Evidence	Undesirable Evidence	Desirable Reasoning	Undesirable Reasoning
Elementary	Relevant	Irrelevant+Repeat +Non-Text-Based + Minimal	LCE + Para- phrase	Not-LCE + Generic + Commentary + Minimal
High-school College			LCE	Paraphrase+ Not-LCE+ Generic + Commentary+ Minimal

Table 4: Desirable and Undesirable revision mapping.

then revised to create Draft2, then revised again without any further textual² feedback to create Draft3. Drafts 2 and 3 were later graded by experts based on a rubric. We create a binary improvement score for each essay pair, calculated as 1 if Draft3 improved compared to Draft2, -1 otherwise.

For all corpora, sentences from the two drafts were aligned manually based on semantic similarity. Aligned sentences represent one of four operations between drafts – no change, modification, sentence deleted from Draft1, sentence added to Draft2. Each pair of changed aligned sentences was then extracted as a *revision* (rows 3, 5 and 9 in Table 1) and annotated for its *purpose* (revise reasoning, evidence, and reasoning in rows 3, 5 and 9, respectively). Kappa of the purpose annotation was 0.753 (Afrin et al., 2020). From among the full set of annotations, we only use evidence and reasoning revisions for the current study because they are the most frequent for elementary and high-school data³. Due to low frequency of evidence revisions, we only use reasoning revisions for college data.

Finally, to understand how students revise evidence and reasoning, whether their revisions were desirable, and whether desirable revisions relate to measures of essay improvement, we then applied the evidence and reasoning revision categorization scheme developed in (Afrin et al., 2020). In this scheme, revisions related to evidence are characterized by five codes – Relevant, Irrelevant, Repeat evidence, Non-text based, and Minimal. Reasoning revisions are characterized by six codes – Linked claim-evidence (LCE), Not LCE, Paraphrase evidence, Generic, Commentary, and Minimal. The annotation was done by an expert familiar with the coding scheme (Cohen’s kappa in a previous study was 0.833 for evidence and 0.719 for reasoning).

²Feedback was given using AWE interface visualizations.

³1475 revisions were extracted from elementary-school data. Other 700 revisions (claim, word-usage, grammar mistakes, etc.) are not considered due to low frequency. 1269 revisions were extracted from high-school data. Other 772 revisions are not considered due to low frequency.

Labeling Desirable Revisions. In this paper, we abstract the evidence and reasoning revision annotations described above into two new categories – *desirable* revision and *undesirable* revision. The mapping is shown in Table 4. Desirable revisions are those that have hypothesized utility in improving the essay after revision, and are encouraged by the writing task. Given a different writing task with different feedback messages, different categories may be desirable in improving the essay quality. For our corpus, relevant evidences are desirable because they support a claim in the essay. All the other categories of evidence revisions are combined as undesirable. For reasoning revisions, LCE and paraphrase reasoning are combined as desirable for the elementary-school data⁴. On the other hand, only LCE is a desirable reasoning revision for the high-school and college data. The rest of the reasoning revisions are combined as undesirable. Table 5 shows the number of desirable and undesirable revisions for each corpus⁵. We did not combine evidence and reasoning revisions, because the schema to label each is different.

Extracting Context. We use two methods to extract context of the target revision, *simple context* (SC) and *longer context* (LC). Following Zhang and Litman (2016), we only focus on the sentences before and after the target revision to extract simple context. For example, simple context for the 3rd revision in Table 1 consists of sentence 2 and 4 from the revised draft. For longer context, we introduce a new method that considers all the sentences that are revised around the target sentence until we find a sentence that is not changed. This makes sure that the context window will have text extracted from both drafts. For example in Table 1, sentence 3 will not have any context from Draft1 using the simple context method. But with longer context, sentences 1 to 4 from the original draft will be considered as context1 from Draft1; sentences 1 to 4

⁴Paraphrase is encouraged by the writing task.

⁵See Appendix A for more data distributions.

		Before Augmentation				After Augmentation		
		N	Desirable	Undesirable	Total	Desirable	Undesirable	Total
Evidence	Elementary	143	239	147	386	4658	2946	7604
	High-school	47	80	30	110	1168	511	1679
Reasoning	Elementary	143	186	203	389	3881	3844	7725
	High-school	47	202	185	387	2963	2817	5780
	College	60	114	93	207	3186	2329	5515

Table 5: Statistics for number of revisions in each corpus. Average number of revisions over 10-fold cross-validation is shown after data augmentation (N = #Student).

from revised draft will be considered as context2 from Draft2. The length of the context will vary depending on the number of revisions within the window. For example, context1 for sentence 3 consists of 2 sentences from Draft1 (1 and 4, 2 was added) while sentence 5 had 3 (4, 6, and 7).

4 Predicting Revision Desirability

In this section, we describe the models for automatically classifying desirable revisions. First, we describe a data augmentation process to increase the training data. Then we describe a model to identify revision desirability, and extend it to use context and the feedback information. We setup our models to answer the following research questions:

RQ1: Is the context of the revision predictive of revision desirability?

RQ2: Is the feedback received before revising the essay predictive of revision desirability?

RQ3: Do the context and feedback together boost the identification of desirable revision?

4.1 Data Augmentation

Our limited amount of revision data is not suitable to experiment with various state-of-the-art machine learning and deep learning models. To generate more training examples, we use a customized version of the synonym replacement (SR) data augmentation strategy – randomly pick a word from the sentence and replace it with a synonym (Wei and Zou, 2019). For each sentence, we replaced one random word with its synonyms but did not consider multiple words at the same time to preserve the hand-annotated revision categories. We ignored stop words, selected words that are more than length of 5 characters, and used maximum 5 synonyms per word to limit the number of data generated. The synonyms are extracted from the Synset from WordNet lexical database from Natural Language Toolkit (NLTK) in Python (Bird et al.,

2009), e.g., the word ‘achieve’ in sentence 5 of Table 1 can be replaced by ‘accomplish’. Then the augmented new revision is added as a training instance. The last three columns in Table 5 show the average number of revisions after augmentation.

4.2 Models

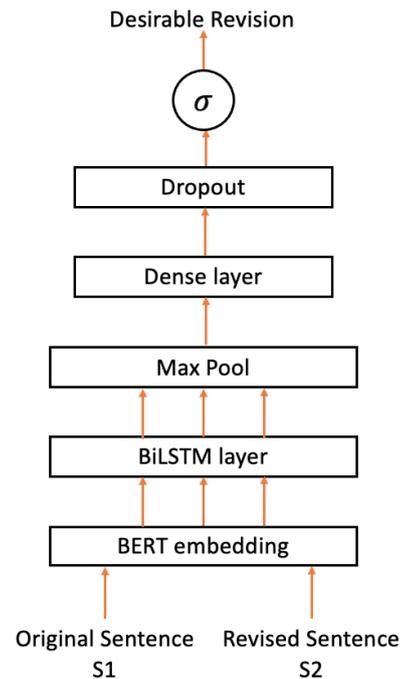


Figure 1: Our model M architecture.

Figure 1 shows the neural network model used in this study (**Model M**). We used the pre-trained ‘bert-based-uncased’ from Keras Huggingface Library (Devlin et al., 2019; Wolf et al., 2020) and encode our revision sentence pair using BERT encoder. After encoding, we use a BiLSTM layer and a Dense layer to build our neural network model using the Keras library (Chollet et al., 2015). This architecture allows easy incorporation of context and feedback as direct inputs, as discussed below.

Bidirectional Long Short Term Memory networks (BiLSTM) has been used in revision classification (Anthonio and Roth, 2020) in addition to various sentence-pair modeling and sentence classification tasks (Vlad et al., 2019; He and Lin, 2016) etc. Vlad et al. (2019) used a BERT-BiLSTM capsule model with additional dense layers with dropout. Following these works, we add a BiLSTM layer after extracting the embedding from BERT to process the input sequences.⁶ We used a dropout and recurrent dropout rate of 0.1. To down-sample the output representation from the BiLSTM, we take the maximum value over the time dimension using the GlobalMaxPool1D (Chollet et al., 2015).

To improve performance while still keeping the model simple, we add a dense layer after BiLSTM with ‘relu’ as the activation function (Javid et al., 2021). In order to make the model robust to overfitting, we add a dropout layer with rate 0.2. The output is then passed to the final output dense layer with 1 neuron. Since this is a binary classification task, we use ‘Sigmoid’ as the activation function.

We tune the model using Adam optimizer with learning rate $\{1e^{-3}, 1e^{-4}, 1e^{-5}\}$ and batch size $\{16, 32, 64\}$ using a validation set of 2000 instances extracted from the elementary evidence augmented data. Finally, we select the learning rate at $1e^{-3}$ and batch size 16, and apply the same to all data. The hidden layer size is set to 64. There were 434,817 trainable parameters in the model.

Context Model. In this model, in addition to the revision we also provide the context1 from Draft1 and context2 from Draft2 as input to the model to answer **RQ1**. Since BERT cannot handle more than 512 tokens and our context can be long in some cases, we did not concatenate contexts from two drafts before encoding. First, we encode each context from each draft using the BERT encoder and extract the embedding. Then the context1 and context2 embeddings are concatenated with the revision input in the order of [revision pair, context1, context2]. Then the concatenated embedding is sent to the BiLSTM layer. There is no change in the following layers. When the context is longer than 512 tokens, it is truncated from the end.⁷

Feedback Model. To answer **RQ2**, we use feed-

back information to predict revision desirability. We first concatenate all the sentences from the feedback messages. Then we encode the whole feedback message using BERT encoder and extract the embedding. The embedding is then concatenated with the input revision from the baseline model in the order of [revision pair, feedback] and sent to the BiLSTM layer. Feedback messages longer than 512 tokens are truncated from the end.⁸

Context & Feedback Model. We also experiment with context and feedback together to answer **RQ3**. We encode context and feedback as we did in the previous models. The embeddings are then concatenated in the order [revision pair, context1, context2, feedback] and sent to the BiLSTM layer.

Baseline Model. We compare our models with a simple model used in prior work that uses logistic regression (LogR) (Afrin et al., 2020) using GloVe word2vec (Pennington et al., 2014) features for revision classification.

5 Results and Evaluation

5.1 Intrinsic Evaluation

In our intrinsic evaluation (see Table 6), we compare whether context and/or feedback model performance improves over the proposed model **M** in terms of average unweighted F1-score⁹, over 10-folds of cross-validation. Without augmentation, our model does not learn at all from the very small amount of data, hence we only report results using augmented data. Augmentation is done at each fold on the training instances. Test instances are kept original, no augmentation applied. We ran the model 10 epochs for each fold.

First, we compare model **M** and its extensions with the LogR baseline. We see that **M** improved over LogR for all cases except high-school evidence classification. Similarly, **M** plus context and/or feedback improved over LogR in all cases except with feedback for high-school evidence.

To answer **RQ1**, we look at the results of the context model and see that our proposed longer context representation (**LC**) always improved over **M** (no context), which is not true for simple context (**SC**). For elementary data, **LC** performed better than **SC**, while for high-school data, **SC** performed better than **LC**. Recall that for high-school data, we did not truncate any context, which means students did

⁶We also experimented with simpler neural nets (e.g., no BiLSTM layer) as our core proposed model, but they did not perform better than model **M**.

⁷No truncation was needed for high-school data. For elementary school, about 9% and 4% contexts were deleted for evidence and reasoning, respectively.

⁸No truncation was needed for elementary data. For high-school, feedback messages were truncated for 55% of students.

⁹See Appendix A for more results.

	Elementary		High-school		College
Model	Evidence	Reasoning	Evidence	Reasoning	Reasoning
LogR	0.469	0.537	0.470	0.495	0.462
M	0.569	0.597	0.446	0.649	0.613
+SC	0.548	0.611	0.489	0.679	0.545
+LC	0.574	0.627	0.474	0.665	0.634
+F	0.570	0.639	0.452	0.652	–
+LC&F	0.587	0.649	0.521	0.664	–

Table 6: Intrinsic evaluation: average unweighted f1-score over 10-fold cross-validation. Best are marked bold.

not make multiple consecutive revisions frequently. This could explain why **SC** was better for high-school data. For college data, **SC** did not improve over **M**, but **LC** showed the best performance.

To answer **RQ2**, the results of the feedback model (**F**) in Table 6 show that while **F** did improve over **M** for each task, in most cases the increase is low. Desirable reasoning classification for elementary-school data had the most benefit using the feedback. This could be because every elementary-school student was specifically asked to provide more details or explain their evidence. For high-school data, although **F** improved over **M**, it did not improve over LogR for evidence.¹⁰

To answer **RQ3**, we only consider longer context and feedback messages (**LC&F**). As shown in Table 6, the **LC&F** model always improved model **M**'s performance and has the best performance except high-school reasoning revision. This indicates that feedback messages were most helpful when combined with the context, especially for elementary-school reasoning revisions where the performance increased more than 0.05 points. This could be because students did not receive feedback at sentence-level; instead, the feedback is usually about specific areas of the essay or about the argumentative structure of the essay. Hence, when combined with the context, it helps the model to capture a better picture.

5.2 Extrinsic Evaluation

To confirm that revision desirability is indeed related to the essay improvement scores described in Section 3, we calculated the Pearson correlation between the frequency of desirable and undesirable revisions (gold annotations) to improvement score. For extrinsic evaluation, we then replicate the correlation calculation for the predicted labels to see if the frequency of predicted desirable revisions are

¹⁰No feedback available for college data Draft2 and Draft3.

still correlated to the essay improvement. Table 7 shows the gold and predicted correlations.

Model **M** showed to be consistent with Gold annotations for elementary reasoning and high-school evidence prediction. **M** also showed higher correlation than LogR when it is consistent with Gold.

Overall, the number of desirable revisions predicted by **LC** showed the highest R values. While we do not expect the models to have higher correlations than the gold annotations, **LC** did in one case (desirable reasoning prediction for high-school data). Gold annotations did not show significant negative correlations to undesirable revisions. This is because the scoring rubrics typically did not penalize for revisions that did not improve the essay, as long as revising didn't make the essay worse. **LC** also did not show any significant correlation to undesirable revisions. Unexpectedly, **SC** did in one case (undesirable reasoning for high-school).

Model **F** similarly yielded significant positive correlation with desirable revisions and had higher correlations than model **M**. In most cases Model **F** is consistent with Gold annotations, except for undesirable reasoning revisions for high-school data.

Model **LC&F** also showed higher significant correlation for the predicted labels compared to Model **M**. However, unlike the intrinsic evaluation it does not show us the best performance.

Unfortunately, we did not see any significant correlation for the college data. But in most cases, desirable revisions showed positive sign, while undesirable revisions showed negative sign.

6 Qualitative Analysis

In order to better understand the model predictions, in Table 8 we compare gold and predicted labels for a few example revisions. The first example (taken from Table 1) is predicted as desirable whenever longer context information was available. Otherwise, it is wrongly predicted as undesirable. Look-

	Elementary (N=143)				High-school (N=47)				College (N=60)	
	Evidence		Reasoning		Evidence		Reasoning		Reasoning	
	D	U	D	U	D	U	D	U	D	U
Gold	0.200*	0.039	0.450*	-0.022	0.391*	0.040	0.351*	0.272	0.029	-0.131
LogR	0.112	0.182*	0.231*	0.226*	0.229	0.240	0.371*	0.207	0.030	-0.095
M	0.156	0.106	0.339*	0.114	0.321*	0.156	0.249	0.396*	0.039	-0.181
+SC	0.137	0.137	0.321*	0.093	0.350*	0.025	0.335*	0.307*	-0.016	-0.123
+LC	0.152	0.084	0.422*	-0.039	0.366*	-0.030	0.407*	0.257	0.083	-0.246
+F	0.125	0.162	0.360*	0.080	0.323*	0.090	0.327*	0.322*	-	-
+LC&F	0.139	0.117	0.381*	0.041	0.354*	-0.064	0.406*	0.239	-	-

Table 7: Extrinsic evaluation: significant correlations using predicted desirability that are consistent with using gold labels are marked bold (* $p < .05$, $N = \#Students$, D: Desirable, U: Undesirable).

Original Draft	Revised Draft	Gold	M	+SC	+LC	+F	+LC&F
	They can do that by assuring that the people of Sauri, Kenya have food, water, liter, and a place to stay.	D R	U	U	D	U	D
We think \$5 dollars isn't that much money but they live in poverty.	We think \$5 dollars isn't that much money but they live in situations where \$5 is a weeks worth of money.	D E	D	U	U	D	U
	They had water, food, electricity, supplies, medicine, and simple things.	U E	D	U	U	D	U

Table 8: Revision examples with gold and predicted labels. D: Desirable, U: Undesirable, E: Evidence, R: Reasoning

ing at this revision (sentence 3) and its context from Table 1, we can see that sentence 3 mentions about the ‘people’, ‘food, water, liter, and a place to stay’. The context mention ‘people’, ‘health care’ and ‘quality of life’. We think those phrases helped the context model to identify this example as desirable. However, although feedback messages asked to ‘explain the evidence’, the feedback model was not successful in identifying this as desirable.

The second example is a desirable evidence predicted as undesirable by context and desirable by the feedback model. The AWE feedback asked the student to use more evidence and add details. We think the feedback model tied the extra information in the modified sentence to what was asked for.

The last example is an undesirable evidence predicted correctly only by the models using context information. Although the example text resembles a desirable evidence, it is actually undesirable because it was repeated. Obviously, the model needed context to identify that it is a repeated evidence.

7 Conclusion

In this study, we presented new models for the automatic identification of desirable revisions in three corpora of argumentative writing varying in writer’s level of expertise, source of feedback, and grading rubrics. We presented a new method of extracting context from essay revisions. Using intrinsic and extrinsic evaluation we showed that models using the context information performed best in identifying desirable revisions. We also studied the use of feedback messages received by students to predict desirable revisions. To the best of our knowledge this is the first model to use feedback information to analyze student revision. Our experiments showed that feedback information also helped improve classifier performance, particularly when used with context. We have released the college data annotated with revision desirability. It can be downloaded from this link: <https://petal-cs-pitt.github.io/data.html>. The code is also available from here: <https://github.com/tazin-afrin/desirable-revision-classification>

Acknowledgements

The research reported here was supported, in whole or in part, by the National Science Foundation (NSF) grant 1735752 and 2202347 to the University of Pittsburgh. We would like to thank the anonymous reviewers for taking the time to review our paper and provide us with detailed feedback. We would also like to thank the members of the PETAL lab for their valuable feedback. The opinions expressed are those of the authors and do not represent the views of the Institute.

Discussion of Limitations

Our use of both context and feedback could be enhanced in future work. First, we sometimes needed to truncate context or feedback from the end, which may remove useful information. In the future, we plan to use other transformer architectures capable of handling longer sequences (e.g., Longformer (Beltagy et al., 2020)). Second, while our proposed method of extracting longer context enables the use of variable length context windows, our method does not guarantee that the context will include the major claim. Since evidence and reasoning are most effective when used to support a claim, their revision desirability might depend on the essay's claim. Third, since the feedback received by students was largely framed at the essay-level, we did not attempt to connect the messages with specific sentence revisions. Such modeling could potentially improve feedback performance.

Additional limitations include that our classifier input was based on perfect alignment of the sentences in the essay drafts and used gold evidence and reasoning revision purpose labels. An end-to-end system would have lower performance due to errors propagated from alignment and purpose classification. Our data is also limited in that essays are all of an argumentative writing style and annotated for only two types of content revisions. Also, the corpus is small. Although, we used simple augmentation to generate enough data to experiment with complex learning models, in the future we plan to explore other options for data augmentation. We also would like to use similar argumentative essays to fine-tune the BERT architecture.

Ethical Considerations

All corpora were collected under protocols approved by an institutional review board, including

that the data is not publicly available, except the college data. While the breach of private student information from the elementary and high school data will thus not pose any ethical concern, other researchers can not replicate our results for those data. However, since the college data with its purpose annotations was already made available by the original researchers, our new desirability annotations can be released upon acceptance of this study. The claims of the paper match the experimental results and the results can be hypothesized to generalize. In the future, the proposed models may be incorporated into AWE systems for student writers. While identifying and providing feedback on revision desirability will be helpful to students in improving their writing, there is the risk that the system might sometimes provide poor advice based on incorrect model classifications. Since the dataset is still fairly small after data augmentation, it is possible that the model may learn biased representation of the revisions (e.g., always predict longer revisions with more information as desirable).

References

- Tazin Afrin and Diane Litman. 2018. Annotation and classification of sentence-level revision improvement. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 240–246, New Orleans, Louisiana.
- Tazin Afrin and Diane J. Litman. 2019. Identifying editor roles in argumentative writing from student revision histories. In *Artificial Intelligence in Education - 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part II*, volume 11626 of *Lecture Notes in Computer Science*, pages 9–13. Springer.
- Tazin Afrin, Elaine Lin Wang, Diane Litman, Lindsay Clare Matsumura, and Richard Correnti. 2020. Annotation and classification of evidence and reasoning revisions in argumentative writing. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, Seattle, Washington, USA (Remote).
- Talita Anthonio and Michael Roth. 2020. [What can we learn from noun substitutions in revision histories?](#) In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1359–1370, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Beata Beigman Klebanov and Nitin Madnani. 2020. [Automated evaluation of writing – 50 years and counting.](#) In *Proceedings of the 58th Annual Meeting of*

- the Association for Computational Linguistics*, pages 7796–7810, Online. Association for Computational Linguistics.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc."
- Amit Bronner and Christof Monz. 2012. **User edits classification using document revision histories**. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 356–366, Avignon, France. Association for Computational Linguistics.
- Tuhin Chakrabarty, Christopher Hidey, Smaranda Muresan, Kathy McKeown, and Alyssa Hwang. 2019. AMPERSAND: Argument mining for PERSuAsive oNline discussions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2933–2943, Hong Kong, China. Association for Computational Linguistics.
- Francois Chollet et al. 2015. Keras, <https://github.com/fchollet/keras>, [online; accessed 10-10-2021].
- Johannes Daxenberger and Iryna Gurevych. 2012. A corpus-based study of edit categories in featured and non-featured wikipedia articles. In *Proceedings of the 24th International Conference on Computational Linguistics*, COLING '12, pages 711–726, Mumbai, India.
- Johannes Daxenberger and Iryna Gurevych. 2013. Automatically classifying edit categories in wikipedia revisions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 578–589, Seattle, Washington, USA. Association for Computational Linguistics.
- Alok Debnath and Michael Roth. 2021. **A computational analysis of vagueness in revisions of instructional texts**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 30–35, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022. **Understanding iterative revision from human-written text**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590, Dublin, Ireland. Association for Computational Linguistics.
- Lester Faigley and Stephen Witte. 1981. Analyzing revision. *College Composition and Communication*, 32(4):400–414.
- Debanjan Ghosh, Ritvik Shrivastava, and Smaranda Muresan. 2021. **“laughing at you or with you”: The role of sarcasm in shaping the disagreement space**. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1998–2010, Online. Association for Computational Linguistics.
- Hua He and Jimmy Lin. 2016. **Pairwise word interaction modeling with deep neural networks for semantic similarity measurement**. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948, San Diego, California. Association for Computational Linguistics.
- Takumi Ito, Tatsuki Kuribayashi, Hayato Kobayashi, Ana Brassard, Masato Hagiwara, Jun Suzuki, and Kentaro Inui. 2019. **Diamonds in the rough: Generating fluent sentences from early-stage drafts for academic writing assistance**. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 40–53, Tokyo, Japan. Association for Computational Linguistics.
- Alireza M. Javid, Sandipan Das, Mikael Skoglund, and Saikat Chatterjee. 2021. A relu dense layer to improve the performance of neural networks. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2810–2814.
- Omid Kashefi, Tazin Afrin, Meghan Dale, Christopher Olshefski, Amanda Godley, Diane Litman, and Rebecca Hwa. 2022. **Argrewrite v.2: an annotated argumentative revisions corpus**. *Language Resources and Evaluation*, pages 1574–0218.
- Wenchuan Mu and Kwan Hui Lim. 2022. **Revision for concision: A constrained paraphrase generation task**. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 57–76, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna

- Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy. Association for Computational Linguistics.
- Antonette Shibani, Simon Knight, and Simon Buckingham Shum. 2018. Understanding revisions in student writing through revision graphs. In *International Conference on Artificial Intelligence in Education*, pages 332–336, Cham. Springer International Publishing.
- Gabriella Skitalinskaya, Jonas Klaff, and Henning Wachsmuth. 2021. [Learning from revisions: Quality assessment of claims in argumentation at scale](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1718–1729, Online. Association for Computational Linguistics.
- Chenhao Tan and Lillian Lee. 2014. A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2: Short Papers, pages 403–408, Baltimore, MD, USA.
- George-Alexandru Vlad, Mircea-Adrian Tanase, Cristian Onose, and Dumitru-Clementin Cercel. 2019. [Sentence-level propaganda detection in news articles with transfer learning and BERT-BiLSTM-capsule model](#). In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 148–154, Hong Kong, China. Association for Computational Linguistics.
- Elaine Lin Wang, Lindsay Clare Matsumura, Richard Correnti, Diane Litman, Haoran Zhang, Emily Howe, Ahmed Magooda, and Rafael Quintana. 2020. [erevis\(ing\): Students’ revision of text evidence use in an automated writing evaluation system](#). *Assessing Writing*, 44:100449.
- Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*, pages 6382–6388.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- The Writing Mentor. 2016. ETS writing mentor, <https://mentormywriting.org/>, [online; accessed 02-06-2019].
- Diyi Yang, Aaron Halfaker, Robert E. Kraut, and Eduard H. Hovy. 2017. Identifying semantic edit intentions from revisions in wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP’17*, pages 9–11, Copenhagen, Denmark. Association for Computational Linguistics.
- Fan Zhang, Homa Hashemi, Rebecca Hwa, and Diane Litman. 2017. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578. Association for Computational Linguistics.
- Fan Zhang, Rebecca Hwa, Diane Litman, and Homa B. Hashemi. 2016. Argrewrite: A web-based revision assistant for argumentative writings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 37–41, San Diego, California. Association for Computational Linguistics.
- Fan Zhang and Diane Litman. 2015. Annotation and classification of argumentative writing revisions. In *Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 133–143, Denver, Colorado. Association for Computational Linguistics.
- Fan Zhang and Diane Litman. 2016. Using context to predict the purpose of argumentative writing revisions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1424–1430, San Diego, California. Association for Computational Linguistics.

A Appendix A: Additional Results

Data	Revision	Add	Delete	Modify	Total
Elementary (N=143)	Total Evidence	265	63	58	386
	Desirable Evidence	159	50	30	239
	Undesirable Evidence	106	13	28	147
	Total Reasoning	270	59	60	389
	Desirable Reasoning	140	28	18	186
	Undesirable Reasoning	130	31	42	203
High-school (N=47)	Total Evidence	93	10	7	110
	Desirable Evidence	73	7	0	80
	Undesirable Evidence	20	3	7	30
	Total Reasoning	324	40	23	387
	Desirable Reasoning	184	13	5	202
	Undesirable Reasoning	140	27	18	185
College (N=60)	Total Evidence	25	1	0	26
	Desirable Evidence	23	1	0	24
	Undesirable Evidence	2	0	0	2
	Total Reasoning	191	13	3	207
	Desirable Reasoning	104	7	3	114
	Undesirable Reasoning	87	6	0	93

Table 9: Detailed data distribution.

		Evidence			Reasoning		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Elementary	LogR	0.510	0.519	0.469	0.572	0.573	0.537
	M	0.587	0.587	0.569	0.613	0.609	0.597
	+SC	0.587	0.575	0.548	0.624	0.626	0.611
	+LC	0.640	0.594	0.574	0.644	0.638	0.627
	+F	0.592	0.595	0.570	0.675	0.658	0.639
	+LC&F	0.636	0.605	0.587	0.681	0.664	0.649
High-school	LogR	0.493	0.535	0.470	0.600	0.555	0.495
	M	0.434	0.476	0.446	0.668	0.662	0.649
	+SC	0.489	0.535	0.489	0.701	0.690	0.679
	+LC	0.480	0.502	0.474	0.681	0.673	0.665
	+F	0.469	0.480	0.452	0.668	0.663	0.652
	+LC&F	0.554	0.549	0.521	0.683	0.679	0.664
College	LogR				0.507*	0.514	0.462*
	M				0.667	0.653	0.613
	+SC				0.593	0.593	0.545
	+LC				0.703	0.670	0.634

Table 10: 10-fold cross-validation result for classifying desirable evidence and reasoning, more metrics.

Discourse Structure Extraction from Pre-Trained and Fine-Tuned Language Models in Dialogues

Chuyuan Li¹, Patrick Huber², Wen Xiao²

Maxime Amblard¹, Chloé Braud³, Giuseppe Carenini²

¹ Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

² University of British Columbia, V6T 1Z4, Vancouver, BC, Canada

³ IRIT, Université de Toulouse, CNRS, ANITI, Toulouse, France

¹{firstname.name}@loria.fr, ³chloe.braud@irit.fr,

²{huberpat, xiaowen3, carenini}@cs.ubc.ca

Abstract

Discourse processing suffers from data sparsity, especially for dialogues. As a result, we explore approaches to build discourse structures for dialogues, based on attention matrices from Pre-trained Language Models (PLMs). We investigate multiple tasks for fine-tuning and show that the dialogue-tailored Sentence Ordering task performs best. To locate and exploit discourse information in PLMs, we propose an unsupervised and a semi-supervised method. Our proposals thereby achieve encouraging results on the STAC corpus, with F_1 scores of 57.2 and 59.3 for the unsupervised and semi-supervised methods, respectively. When restricted to projective trees, our scores improved to 63.3 and 68.1.

1 Introduction

In recent years, the availability of accurate transcription methods and the increase in online communication have led to a vast rise in dialogue data, necessitating the development of automatic analysis systems. For example, summarization of meetings or exchanges with customer service agents could be used to enhance collaborations or analyze customers issues (Li et al., 2019; Feng et al., 2021); machine reading comprehension in the form of question-answering could improve dialogue agents' performance and help knowledge graph construction (He et al., 2021; Li et al., 2021). However, simple surface-level features are oftentimes not sufficient to extract valuable information from conversations (Qin et al., 2017). Rather, we need to understand the semantic and pragmatic relationships organizing the dialogue, for example through the use of discourse information.

Along this line, several discourse frameworks have been proposed, underlying a variety of annotation projects. For dialogues, data has been primarily annotated within the Segmented Discourse Representation Theory (SDRT) (Asher et al., 2003).

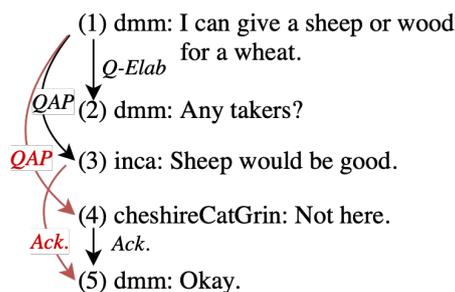


Figure 1: Excerpt of dependency structures in file *s2-leagueM-game4*, STAC. Red links are non-projective.

Discourse structures are thereby represented as dependency graphs with arcs linking spans of text and labeled with semantico-pragmatic relations (e.g. *Acknowledgment* (Ack) or *Question-Answer Pair* (QAP)). Figure 1 shows an example from the Strategic Conversations corpus (STAC) (Asher et al., 2016). Discourse processing refers to the retrieval of the inherit structure of coherent text, and is often separated into three tasks: EDU segmentation, structure building (or attachment), and relation prediction. In this work, we focus on the automatic extraction of (naked) structures without discourse relations. This serves as a first critical step in creating a full discourse parser. It is important to note that naked structures have already been shown to be valuable features for specific tasks. Louis et al. (2010) mentioned that they are the most reliable indicator of importance in content selection. Xu et al. (2020); Xiao et al. (2020) on summarization, and Jia et al. (2020) on thread extraction, also demonstrated the advantages of naked structures.

Data sparsity has always been an issue for discourse parsing both in monologues and dialogues: the largest and most commonly used corpus annotated under the Rhetorical Structure Theory, the RST-DT (Carlson et al., 2001) contains 21, 789 discourse units. In comparison, the largest dialogue discourse dataset (STAC) only contains 10, 678

units. Restricted to domain and size, the performance of supervised discourse parsers is still low, especially for dialogues, with at best 73.8% F_1 for the naked structure on STAC (Wang et al., 2021). As a result, several transfer learning approaches have been proposed, mainly focused on monologues. Previous work demonstrate that discourse information can be extracted from auxiliary tasks like sentiment analysis (Huber and Carenini, 2020) and summarization (Xiao et al., 2021), or represented in language models (Koto et al., 2021) and further enhanced by fine-tuning tasks (Huber and Carenini, 2022). Inspired by the latter approaches, we are pioneering in addressing this issue for dialogues and introducing effective semi-supervised and unsupervised strategies to uncover discourse information in large pre-trained language models (PLMs). We find, however, that the monologue-inspired fine-tuning tasks are not performing well when applied to dialogues. Dialogues are generally less structured, interspersed with more informal linguistic usage (Sacks et al., 1978), and have structural particularities (Asher et al., 2016). Thus, we propose a new Sentence Ordering (SO) fine-tuning task tailored to dialogues. Building on the proposal in Barzilay and Lapata (2008), we propose crucial, dialogue-specific extensions with several novel shuffling strategies to enhance the pair-wise, inter-speech block, and inter-speaker discourse information in PLMs, and demonstrate its effectiveness over other fine-tuning tasks.

In addition, a key issue in using PLMs to extract document-level discourse information is how to choose the best attention head. We hypothesize that the location of discourse information in the network may vary, possibly influenced by the length and complexity of the dialogues. Therefore, we investigate methods that enables us to evaluate each attention head individually, in both unsupervised and semi-supervised settings. We introduce a new metric called “Dependency Attention Support” (DAS), which measures the level of support for the dependency trees generated by a specific self-attention head, allowing us to select the optimal head without any need for supervision. We also propose a semi-supervised approach where a small validation set is used to choose the best head.

Experimental results on the STAC dataset reveal that our unsupervised and semi-supervised methods outperform the strong LAST baseline (F_1 56.8%, Sec. 4), delivering substantial gains on the com-

plete STAC dataset (F_1 59.3%, Sec. 5.2) and show further improvements on the tree-structured subset (F_1 68.1%, Sec. 6.3).

To summarize, our contributions in this work are: (1) Discourse information detection in pre-trained and sentence ordering fine-tuned LMs; (2) Unsupervised and semi-supervised methods for discourse structure extraction from the attention matrices in PLMs; (3) Detailed quantitative and qualitative analysis of the extracted discourse structures.

2 Related Work

Discourse structures for complete documents have been mainly annotated within the Segmented Discourse Representation Theory (SDRT) (Asher et al., 2003) or the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988), with the latter leading to the largest corpora and many discourse parsers for monologues, while SDRT is the main theory for dialogue corpora, i.e., STAC (Asher et al., 2016) and Molweni (Li et al., 2020). In SDRT, discourse structures are dependency graphs with possibly non-projective links (see Figure 1) compared to constituent trees structures in RST. Early approaches to discourse parsing on STAC used varied decoding strategies, such as Maximum Spanning Tree algorithm (Muller et al., 2012; Li et al., 2014; Afantenos et al., 2012) or Integer Linear Programming (Perret et al., 2016). Shi and Huang (2019) first proposed a neural architecture based on hierarchical Gated Recurrent Unit (GRU) and reported 73.2% F_1 on STAC for naked structures. Recently, Wang et al. (2021) adopted Graph Neural Networks (GNNs) and reported marginal improvements on the same test set (73.8% F_1).

Data sparsity being the issue, a new trend towards semi-supervised and unsupervised discourse parsing has emerged, almost exclusively for monologues. Huber and Carenini (2019, 2020) leveraged sentiment information and showed promising results in cross-domain setting with the annotation of a silver-standard labeled corpus. Xiao et al. (2021) extracted discourse trees from neural summarizers and confirmed the existence of discourse information in self-attention matrices. Another line of work proposed to enlarge training data with a combination of several parsing models, as done in Jiang et al. (2016); Kobayashi et al. (2021); Nishida and Matsumoto (2022). In a fully unsupervised setting, Kobayashi et al. (2019) used similarity and dissimilarity scores for discourse tree creation, a

method that can not be directly used for discourse graphs though. As for dialogues, transfer learning approaches are rare. [Badene et al. \(2019a,b\)](#) investigated a weak supervision paradigm where expert-composed heuristics, combined to a generative model, are applied to unseen data. Their method, however, requires domain-dependent annotation and a relatively large validation set for rule verification. Another study by [Liu and Chen \(2021\)](#) focused on cross-domain transfer using STAC (conversation during online game) and Molwani (Ubuntu forum chat logs). They applied simple adaptation strategies (mainly lexical information) on a SOTA discourse parser and showed improvement compared to bare transfer: trained on Molwani and tested on STAC F_1 increased from 42.5% to 50.5%. Yet, their model failed to surpass simple baselines. Very recently, [Nishida and Matsumoto \(2022\)](#) investigated bootstrapping methods to adapt BERT-based parsers to out-of-domain data with some success. In comparison to all this previous work, to the best of our knowledge, we are the first to propose a fully unsupervised method and its extension to a semi-supervised setting.

As pre-trained language models such as BERT ([Devlin et al., 2019](#)), BART ([Lewis et al., 2020](#)) or GPT-2 ([Radford et al., 2019](#)) are becoming dominant in the field, *BERTology* research has gained much attention as an attempt to understand what kind of information these models capture. Probing tasks, for instance, can provide fine-grained analysis, but most of them only focus on sentence-level syntactic tasks ([Jawahar et al., 2019](#); [Hewitt and Manning, 2019](#); [Mareček and Rosa, 2019](#); [Kim et al., 2019](#); [Jiang et al., 2020](#)). As for discourse, [Zhu et al. \(2020\)](#) and [Koto et al. \(2021\)](#) applied probing tasks and showed that BERT and BART encoders capture more discourse information than other models, like GPT-2. Very recently, [Huber and Carenini \(2022\)](#) introduced a novel way to encode long documents and explored the effect of different fine-tuning tasks on PLMs, confirming that pre-trained and fine-tuned PLMs both can capture discourse information. Inspired by these studies on monologues, we explore the use of PLMs to extract discourse structures in dialogues.

3 Method: from Attention to Discourse

3.1 Problem Formulation and Simplifications

Given a dialogue D with n *Elementary Discourse Units* (EDUs) $\{e_1, e_2, e_3, \dots, e_n\}$, which are the

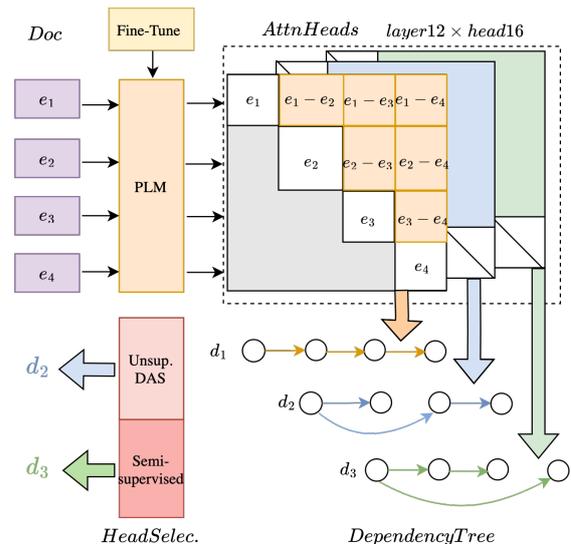


Figure 2: Pipeline for discourse structure extraction.

minimal spans of text (mostly clauses, at most a sentence) to be linked by discourse relations, the goal is to extract a Directed Acyclic Graph (DAG) connecting the n EDUs that best represents its SDRT discourse structure from attention matrices in PLMs¹ (see Figure 2 for an overview of the process). In our proposal, we make a few simplifications, partially adopted from previous work. We do not deal with SDRT *Complex Discourse Units* (CDUs) following [Muller et al. \(2012\)](#) and [Afanenos et al. \(2015\)](#), and do not tackle relation type assignment. Furthermore, similar to [Shi and Huang \(2019\)](#), our solution can only generate discourse trees. Extending our algorithm to non-projective trees ($\approx 6\%$ of edges are non-projectives in tree-like examples) and graphs ($\approx 5\%$ of nodes with multiple incoming arcs) is left as future work.

3.2 Which kinds of PLMs to use?

We explore both vanilla and fine-tuned PLMs, as they were both shown to contain discourse information for monologues ([Huber and Carenini, 2022](#)).

Pre-Trained Models: We select BART ([Lewis et al., 2020](#)), not only because its encoder has been shown to effectively capture discourse information, but also because it dominated other alternatives in preliminary experiments, including DialoGPT ([Zhang et al., 2020](#)) and DialogLM ([Zhong et al., 2022](#)) - language models pre-trained with conversational data².

¹For more details on extracting discourse information from attention mechanisms see [Liu and Lapata \(2018\)](#).

²See Appendix E for additional results with other PLMs.

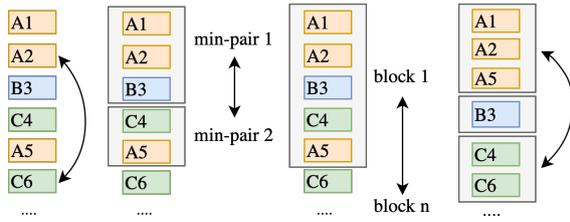


Figure 3: Shuffling strategies (left to right: partial, minimal-pair, block, speaker-turn) on a sequence of utterances 1 to 6, with A, B, C as the speakers.

Fine-Tuning Tasks: We fine-tune BART on three discourse-related tasks:

(1) **Summarization:** we use BART fine-tuned on the popular CNN-DailyMail (CNN-DM) news corpus (Nallapati et al., 2016), as well as on the SAMSum dialogue corpus (Gliwa et al., 2019).

(2) **Question Answering:** we use BART fine-tuned on the latest version of the Stanford Question Answering Dataset (SQuAD 2.0) (Rajpurkar et al., 2018).

(3) **Sentence Ordering:** we fine-tune BART on the Sentence Ordering task – reordering a set of shuffled sentences to their original order. We use an in-domain and an out-of-domain dialogue datasets (Sec. 4) for this task. Since fully random shuffling showed very limited improvements, we considered additional strategies to support a more gradual training tailored to dialogues. Specifically, as shown in Figure 3, we explore: (a) *partial-shuf*: randomly picking 3 utterances in a dialogue (or 2 utterances if the dialogue is shorter than 4) and shuffling them while maintaining the surrounding context. (b) *minimal-pair-shuf*: shuffling minimal pairs, comprising of a pair of speech turns from 2 different speakers with at least 2 utterances. A speech turn marks the start of a new speaker’s turn in the dialogue. (c) *block-shuf*: shuffling a block containing multiple speech turns. We divide one dialogue into $[2, 5]$ blocks based on the number of utterances³ and shuffle between blocks. (d) *speaker-turn-shuf*: grouping all speech productions of one speaker together. The sorting task consists of ordering speech turns from different speakers’ production. We evenly combine all permutations mentioned above to create our **mixed-shuf** data set and conduct the SO task.

³Block size is designed to be as twice or 3 times bigger than “min-pair”, we thus set criteria aiming to have ≈ 6 EDUs per block: $|utt.| < 12 : b = 2$, $|utt.| \in [12, 22] : b = 3$, $|utt.| \in [22, 33] : b = 4$, $|utt.| \geq 33 : b = 5$.

Choice of Attention Matrix: The BART model contains three kinds of attention matrices: encoder, decoder and cross attention. We use the encoder attention in this work, since it has been shown to capture most discourse information (Koto et al., 2021) and outperformed the other alternatives in preliminary experiments on a validation set.

3.3 How to derive trees from attention heads?

Given an attention matrix $A^t \in \mathbb{R}^{k \times k}$ where k is the number of tokens in the input dialogue, we derive the matrix $A^{edu} \in \mathbb{R}^{n \times n}$, with n the number of EDUs, by computing $A^{edu}(i, j)$ as the average of the submatrix of A^t corresponding to all the tokens of EDUs e_i and e_j , respectively. As a result, A^{edu} captures how much EDU e_i depends on EDU e_j and can be used to generate a tree connecting all EDUs by maximizing their dependency strength. Concretely, we find a Maximum Spanning Tree (MST) in the fully-connected dependency graph A^{edu} using the Eisner algorithm (Eisner, 1996). Conveniently, since an utterance cannot be anaphorically and rhetorically dependent on following utterances in a dialogue, as they are previously unknown (Afantenos et al., 2012), we can further simplify the inference by applying the following hard constraint to remove all backward links from the attention matrix A^{edu} : $a_{ij} = 0$, if $i > j$.

3.4 How to find the best heads?

Xiao et al. (2021) and Huber and Carenini (2022) showed that discourse information is not evenly distributed between heads and layers. However, they do not provide a strategy to select the head(s) containing most discourse information. Here, we propose two effective selection methods: fully unsupervised or semi-supervised.

3.4.1 Unsupervised Best Head(s) Selection

Dependency Attention Support Measure (DAS): Loosely inspired by the confidence measure in Nishida and Matsumoto (2022), where the authors define the confidence of a teacher model based on predictive probabilities of the decisions made, we propose a DAS metric measuring the degree of support for the maximum spanning (dependency) tree (MST) from the attention matrix. Formally, given a dialogue g with n EDUs, we first derive the EDU matrix A^{edu} from its attention matrix A^g (see Sec. 3.3). We then build the MST T^g by selecting $n - 1$ attention links l_{ij} from A^{edu} based on the tree generation algorithm. DAS measures the strength

of all those connections by computing the average score of all the selected links:

$$DAS(T^g) = \frac{1}{n-1} \sum_{i=1}^n \sum_{j=1}^n Sel(A^g, i, j) \quad (1)$$

with $Sel(A^g, i, j) = A_{ij}^g$, if $l_{ij} \in T^g$, 0 otherwise. Note that DAS can be easily adapted for a general graph by removing the restriction to $n - 1$ arcs.

Selection Strategy: With DAS, we can now compute the degree of support from each attention head h on each single example g for the generated tree $DAS(T_h^g)$. We therefore propose two strategies to select attention heads based on the DAS measure, leveraging either global or local support. The **global** support strategy selects the head with highest averaged DAS score over all the data examples:

$$H_{global} = \arg \max_h \sum_{g=1}^M DAS(T_h^g) \quad (2)$$

where M is the number of examples. In this way, we select the head that has a generally good performance on the target dataset.

The second strategy is more adaptive to each document, by only focusing on the **local** support. It does not select one specific head for the whole dataset, but instead selects the head/tree with the highest support for each single example g , i.e.,

$$H_{local}^g = \arg \max_h DAS(T_h^g) \quad (3)$$

3.4.2 Semi-Supervised Best Head(s) Selection

We also propose best heads selection using a few annotated examples. In conformity with real-world situations where labeled data is scarce, we sample three small subsets with $\{10, 30, 50\}$ data points (i.e., dialogues) from the validation set. We examine every attention matrix individually, resulting in $12 \text{ layers} \times 16 \text{ heads}$ candidate matrices for each dialogue. Then, the head with the highest micro- F_1 score on the validation set is selected to derive trees in the test set. We also consider layer-wise aggregation, with details in Appendix A.

4 Experimental Setup

Datasets: We evaluate our approach on predicting discourse dependency structures using the STAC corpus (Asher et al., 2016), a multi-party dialogue dataset annotated in the SDRT framework. For the summarization and question-answering

Dataset	#Doc	#Utt/doc	#Tok/doc	#Spk/doc	Domain
DailyDialog	13,118	13	119	2	Daily
STAC	1,161	11	50	3	Game

Table 1: Key statistics of datasets. Utt = sentences in DD or EDUs in STAC; Tok = tokens; Spk = speakers.

fine-tuning tasks, we use publicly available HuggingFace models (Wolf et al., 2020) (see Appendix F). For the novel sentence ordering task, we train BART model on the STAC corpus and the DailyDialog corpus (Li et al., 2017). The key statistics for STAC and DailyDialog can be found in Table 1. These datasets are split into train, validation, and test sets at 82%, 9%, 9% and 85%, 8%, 8% respectively. The Molweni corpus (Li et al., 2020) is not included in our experiments due to quality issues, as detailed in Appendix B.

Baselines: We compare against the simple yet strong unsupervised LAST baseline (Schegloff, 2007), attaching every EDU to the previous one. Furthermore, to assess the gap between our approach and supervised dialogue discourse parsers, we compare with the Deep Sequential model by Shi and Huang (2019) and the Structure Self-Aware (SSA) model by Wang et al. (2021).

Metrics: We report the micro- F_1 and the Unlabeled Attachment Score (UAS) for the generated naked dependency structures.

Implementation Details: We base our work on the transformer implementations from the HuggingFace library (Wolf et al., 2020) and follow the *text-to-marker* framework proposed in Chowdhury et al. (2021) for the SO fine-tuning procedure. We use the original separation of train, validation, and test sets; set the learning rate to $5e - 6$; use a batch size of 2 for DailyDialog and 4 for STAC, and train for 7 epochs. All other hyper-parameters are set following Chowdhury et al. (2021). We do not do any hyper-parameter tuning. We omit 5 documents in DailyDialog during training since the documents lengths exceed the token limit. We replace speaker names with markers (e.g. Sam \rightarrow "spk1"), following the preprocessing pipeline for dialogue utterances in PLMs.

5 Results

5.1 Results with Unsupervised Head Selection

Results using our novel unsupervised DAS method on STAC are shown in Table 2 for both the global

(H_g) and local (H_l) head selection strategies. These are compared to: (1) the unsupervised LAST baseline (at the top), which only predicts local attachments between adjacent EDUs. LAST is considered a strong baseline in discourse parsing (Muller et al., 2012), but has the obvious disadvantage of completely missing long-distance dependencies which may be critical in downstream tasks. (2) The supervised Deep Sequential parser by Shi and Huang (2019) and Structure Self-Aware model by Wang et al. (2021) (center of the table), both trained on STAC, reaching resp. 71.4%⁴ and 73.8% in F_1 .

In the last sub-table we show unsupervised scores from pre-trained and fine-tuned LMs on three auxiliary tasks: summarization, question-answering and sentence ordering (SO) with the mixed shuffling strategy. We present the global head (H_g) and local heads (H_l) performances selected by the DAS score (see section 3.4.1). The best possible scores using an oracle head selector (H_{ora}) are presented for reference.

Comparing the values in the bottom sub-table, we find that the pre-trained BART model underperforms LAST (56.8), with global head and local heads achieving similar performance (56.6 and 56.4 resp.). Noticeably, models fine-tuned on the summarization task (“+CNN”, “+SAMSum”) and question-answering (“+SQuAD2”) only add marginal improvements compared to BART. In the last two lines of the sub-table, we explore our novel sentence ordering fine-tuned BART models. We find that the BART+SO approach surpasses LAST when using local heads (57.1 and 57.2 for DailyDialog and STAC resp.). As commonly the case, the intra-domain training performs best, which is further strengthened in this case due to the special vocabulary in STAC. Importantly, our PLM-based unsupervised parser can capture some long-distance dependencies compared to LAST (Section 6.2). Additional analysis regarding the chosen heads is in Section 6.1.

5.2 Results with Semi-Sup. Head Selection

While the unsupervised strategy only delivered minimal improvements over the strong LAST baseline, Table 3 shows that if a few annotated examples are provided, it is possible to achieve substantial gains. In particular, we report results on the vanilla BART model, as well as BART model fine-

⁴We re-train the model, scores are slightly different due to different train-test splits, as in Wang et al. (2021).

Model			
<i>Unsupervised Baseline</i>			
LAST			56.8
<i>Supervised Models</i>			
Deep-Sequential (2019)			71.4
SSA-GNN (2021)			73.8
<i>Unsupervised PLMs</i>			
BART	H_g	H_l	H_{ora}
+ CNN	56.6	56.4	57.6
+ SAMSum	56.8	56.7	57.1
+ SQuAd2	56.7	56.6	57.6
+ SO-DD	55.9	56.4	57.7
+ SO-DD	56.8	57.1	58.2
+ SO-STAC	56.7	57.2	59.5

Table 2: Micro- F_1 on STAC for LAST, supervised SOTA models and unsupervised PLMs. $H_g/H_l/H_{ora}$: global/local/oracle heads. Best (non-oracle) score in the 3rd block is in bold. DD: DailyDialog.

Train on \rightarrow	BART	+ SO-DD	+ SO-STAC
Test with \downarrow	F_1	F_1	F_1
LAST BSL	56.8	56.8	56.8
H_{ora}	57.6	58.2	59.5
Unsup H_g	<u>56.6</u>	56.8	56.7
Unsup H_l	56.4	<u>57.1</u>	<u>57.2</u>
Semi-sup 10	57.0 _{0.012}	57.2 _{0.012}	57.1 _{0.026}
Semi-sup 30	57.3 _{0.005}	57.3 _{0.013}	59.2 _{0.009}
Semi-sup 50	57.4_{0.004}	57.7_{0.005}	59.3_{0.007}

Table 3: Micro- F_1 on STAC from BART and SO fine-tuned BART with unsupervised and semi-supervised approaches. Semi-supervised scores are averaged from 10 random runs. Subscription is standard deviation.

tuned on DailyDialog (“+SO-DD”) and STAC itself (“+SO-STAC”). We execute 10 runs for each semi-supervised setting ([10, 30, 50]) and report average scores and the standard deviation.

The oracle heads (i.e., H_{ora}) achieve superior performance compared to LAST. Furthermore, using a small scale validation set (50 examples) to select the best attention head remarkably improves the F_1 score from 56.8% (LAST) to 59.3% (+SO-STAC). F_1 improvements across increasingly large validation-set sizes are consistent, accompanied by smaller standard deviations, as would be expected. The semi-supervised results are very encouraging: with 30 annotated examples, we already reach a performance close to the oracle result, and with more examples we can further reduce the gap.

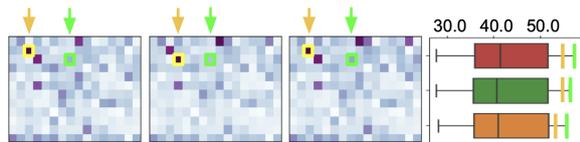


Figure 4: Heatmaps: DAS score matrices (layers: top to bottom=12 to 1, heads: left to right=1 to 16) for BART, BART+SO-DD, BART+SO-STAC. Darker purple=higher DAS score.

Boxplot: Head-aggregated UAS scores for model BART (orange), BART+SO-DD (green) and BART+SO-STAC (red). Light green=head with highest UAS. Yellow=head with highest DAS score.

6 Analysis

6.1 Effectiveness of DAS

We now take a closer look at the performance degradation of our unsupervised approach based on DAS in comparison to the upper-bound defined by the performance of the oracle-picked head. To this end, Figure 4 shows the DAS score matrices (left) for three models with the oracle heads and DAS selected heads highlighted in green and yellow, respectively. These scores correspond to the global support strategy (i.e., H_g). It becomes clear that the oracle heads do not align with the DAS selected heads. Making a comparison between models, we find that discourse information is consistently located in deeper layers, with the oracle heads (light green) consistently situated in the same head for all three models. It is important to note that this information cannot be determined beforehand and can only be uncovered through a thorough examination of all attention heads.

While not aligning with the oracle, the top performing DAS heads (in yellow) are among the top 10% best heads in all three models, as shown in the box-plot on the right. Hence, we confirm that the DAS method is a reasonable approximation to find discourse intense self-attention heads among the 12×16 attention matrices.

6.2 Document and Arc Lengths

The inherent drawback of the simple, yet effective LAST baseline is its inability to predict indirect arcs. To test if our approach can reasonably predict distant arcs of different length in the dependency trees, we analyze our results in regards to the arc lengths. Additionally, since longer documents tend to contain more distant arcs, we also examine the performance across different document lengths.

Arc Distance: To examine the extracted discourse structures for data sub-sets with specific arc lengths, we present the UAS score plotted against different arc lengths on the left side in Figure 5. Our analysis thereby shows that direct arcs achieve high UAS score ($> 80\%$), independent of the model used. We further observe that the performance drops considerably for arcs of distance two and onwards, with almost all models failing to predict arcs longer than 6. BART+SO-STAC model correctly captures an arc of distance 13. Note that the presence for long-distance arcs (≥ 6) is limited, accounting for less than 5% of all arcs.

We further analyze the precision and recall scores when separating dependency links into *direct* (adjacent forward arcs) and *indirect* (all other non-adjacent arcs), following Xiao et al. (2021). For direct arcs, all models perform reasonably well (see Figure 6 at the bottom). The precision is higher ($\approx +6\%$ among all three BART models) and recall is lower than the baseline (100%), indicating that our models predict less direct arcs but more precisely. For indirect arcs (top in Figure 6), the best model is BART+SO-STAC (20% recall, 44% precision), closely followed by original BART (20% recall, 41% precision). In contrast, the LAST baseline model completely fails in this scenario (0 precision and recall).

Document Length: Longer documents tend to be more difficult to process because of the growing number of possible discourse parse trees. Hence, we analyze the UAS performance of documents in regards to their length, here defined as the number of EDUs. Results are presented on the right side in Figure 5, comparing the UAS scores for the three selected models and LAST for different document lengths. We split the document length range into 5 even buckets between the shortest (2 EDUs) and longest (37 EDUs) document, resulting in 60, 25, 16, 4 and 4 examples per bucket.

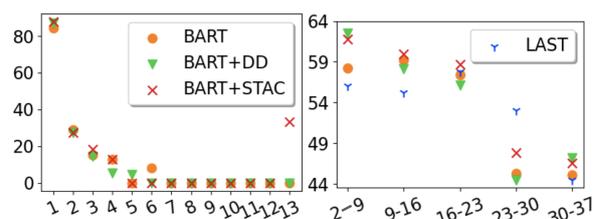


Figure 5: Left: UAS and arcs' distance. x axis: arc distance. Right: averaged UAS for different length of document. x axis: #EDUs in a document. y axis: UAS.

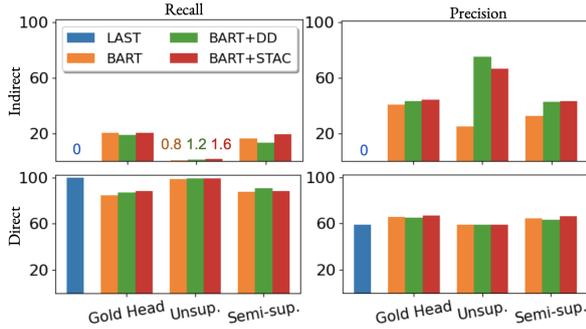


Figure 6: Comparison of recall (left) and precision (right) scores of indirect (top) and direct (bottom) links in LAST, BART, and SO fine-tuned BART models.

	#Doc	#EDUs		#Arcs	
		Single-in	Multi-in	Proj.	N-proj.
(1) Non-Tree	48	706	79	575	170
(2) Tree	61	444	0	348	35
- Proj. tree	48	314	0	266	0

Table 4: STAC test set ground-truth tree and non-tree statistics. “Single-in” and “multi-in” means EDU with single or multiple incoming arcs.

For documents with less than 23 EDUs, all fine-tuned models perform better than LAST, with BART fine-tuned on STAC reaching the best result. We note that PLMs exhibit an increased capability to predict distant arcs in longer documents. However, in the range of [23, 30], the PLMs are inclined to predict a greater number of false positive distant arcs, leading to under-performance compared to the LAST baseline. As a result, we see that longer documents (≥ 23) are indeed more difficult to predict, with even the performance of our best model (BART+STAC) strongly decreasing.

6.3 Projective Trees Examination

Given the fact that our method only extracts projective tree structures, we now conduct an additional analysis, exclusively examining the subset of STAC containing projective trees, on which our method could in theory achieve perfect accuracy.

Table 4 gives some statistics for this subset (“proj. tree”). For the 48 projective tree examples, the document length decreases from an average of 11 to 7 EDUs, however, still contains $\approx 40\%$ indirect arcs, keeping the task difficulty comparable. The scores for the extracted structures are presented in Table 5. As shown, all three unsupervised models outperform LAST. The best model is still BART fine-tuned on STAC, followed by the inter-domain

Train on \rightarrow	BART	+ SO-DD	+ SO-STAC
Test with \downarrow	F ₁	F ₁	F ₁
LAST BSL	62.0	62.0	62.0
H _{ora}	64.8	67.4	68.6
Unsup H _g	<u>62.5</u>	62.5	62.1
Unsup H _t	62.1	<u>62.9</u>	<u>63.3</u>
Semi-sup 10	54.6 _{0.058}	59.2 _{0.047}	61.6 _{0.056}
Semi-sup 30	60.3 _{0.047}	60.3 _{0.044}	65.6 _{0.043}
Semi-sup 50	64.8_{0.000}	66.3_{0.023}	68.1_{0.014}

Table 5: Micro-F₁ scores on STAC projective tree subset with BART and SO fine-tuned BART models.

	Avg.branch	Avg.height	%leaf	Norm. arc
GT	1.67	3.96	0.46	0.43
BART	1.20	5.31	0.31	0.34
+SO-DD	1.32 _{0.014}	5.31 _{0.146}	0.32 _{0.019}	0.37 _{0.003}
+SO-STAC	1.27 _{0.076}	5.28 _{0.052}	0.32 _{0.011}	0.35 _{0.015}

Table 6: Statistics for gold and extracted projective trees in BART and fine-tuned BART models.

fine-tuned +SO-DD and BART models. Using the semi-supervised approach, we see further improvement with the F₁ score reaching 68% (+6% than LAST). Degradation for direct and indirect edges’ precision and recall scores see Appendix C.

Following Ferracane et al. (2019), we analyze key properties of the 48 gold trees compared to our extracted structures using the semi-supervised method. To test the stability of the derived trees, we use three different seeds to generate the shuffled datasets to fine-tune BART. Table 6 presents the averaged scores and the standard deviation of the trees. In essence, while the extracted trees are generally “thinner” and “taller” than gold trees and contain slightly less branches, they are well aligned with gold discourse structures and do not contain “vacuous” trees, where all nodes are linked to one of the first two EDUs. Further qualitative analysis of inferred structures is presented in Appendix D. Tellingly, on two STAC examples our model succeeds in predicting $> 82\%$ of projective arcs, some of which span across 4 EDUs. This is encouraging, providing anecdotal evidence that our method is suitable to extract reasonable discourse structures.

6.4 Performance with Predicted EDUs

Following previous work, all our experiments have started with gold-standard EDU annotations. However, this would not be possible in a deployed dis-

Gold #	Predicted #	Precision %	Recall %	F ₁ %
1155	1081	96.0	93.4	94.8

Table 7: EDU segmentation results on STAC test set using DisCoDisCo model (Gessler et al., 2021), which is re-trained on 50 random dialogues from the validation set. Scores are averaged over three runs.

	LAST	Unsupervised			Semi-supervised		
		H _g	H _l	H _{ora}	semi-10	semi-30	semi-50
Gold	56.8	56.7	57.2	59.5	57.4 _{0.004}	57.7 _{0.005}	59.3_{0.007}
Pred	48.9	50.8	51.1	52.6	50.6 _{0.020}	52.1 _{0.007}	52.2_{0.004}

Table 8: Gold EDUs and predicted EDUs parsing results with BART+SO-STAC model. Scores for predicted EDUs are averaged over three runs.

course parser for dialogues. To assess the performance of such system, we conduct additional experiments in which we first perform EDU segmentation and then feed the predicted EDUs to our methods.

To perform EDU segmentation, we employ the DisCoDisCo model (Gessler et al., 2021), pre-trained on a random sample of 50 dialogues from the STAC validation set. We repeat this process three times to accommodate instability. Our results, as shown in Table 7, align with those previously reported in Gessler et al. (2021) (94.9), with an F-score of 94.8. In the pre-training phase, we utilize all 12 hand-crafted features⁵, and opt for treebanked data for enhanced performance (94.9 compared to 91.9 for plain text data). The treebanked data is obtained using the Stanza Toolkit (Qi et al., 2020).

For evaluation, we adapt the discourse analysis pipeline proposed by Joty et al. (2015). The results are shown in Table 8, comparing the predicted and gold EDUs. The best head (i.e., H_{ora}) performance decreases by ≈ 7 points, from 59.5 to 52.6, as well as unsupervised and semi-supervised results. Despite the drop, our unsupervised and semi-supervised models still outperform the LAST baseline. A similar loss of ≈ 6 points is also observed for RST-style parsing in monologues, as reported in Nguyen et al. (2021).

7 Conclusion

In this study, we explore approaches to build naked discourse structures from PLMs attention matrices to tackle the extreme data sparsity issue in dialogues. We show sentence ordering to be the best

⁵Such as POS tag, UD deprel, sentence length, etc..

fine-tuning task and our unsupervised and semi-supervised methods for selecting the best attention head outperform a strong baseline, delivering substantial gains especially on tree structures. Interestingly, discourse is consistently captured in deeper PLMs layers, and more accurately for shorter links.

In the near future, we intend to explore graph-like structures from attention matrices, for instance, by extending treelike structures with additional arcs of high DAS score and applying linguistically motivated constraints, as in Perret et al. (2016). We would also like to expand shuffling strategies for sentence ordering and to explore other auxiliary tasks. In the long term, our goal is to infer full discourse structures by incorporating the prediction of rhetorical relation types, all while remaining within unsupervised or semi-supervised settings.

Limitations

Similarly to previous work, we have focused on generating only projective tree structures. This not only covers the large majority of the links ($\approx 94\%$), but it can also provide the backbone for accurately inferring the remaining non-projective links in future work. We focus on the naked structure, as it is a significant first step and a requirement to further predict relations for discourse parsing.

We decided to run our experiments on the only existing high quality corpus, i.e., STAC. In essence, we traded-off generalizability for soundness of the results. A second corpus we considered, Molweni, had to be excluded due to serious quality issues.

Lastly, since we work with large language models and investigate every single attention head, computational efficiency is a concern. We used a 4-core GPU machine with the highest VRAM at 11MiB. The calculation for one discourse tree on one head was approximately 0.75 seconds (in STAC the averaged dialogue length is 11 EDUs), which quickly summed up to 4.5 hours with only 100 data points for 192 candidate trees in one LM. When dealing with much longer documents, for example AMI and conversational section in GUM (in average > 200 utterances/dialogue), our estimation shows that one dialogue takes up to ≈ 2 minutes, which means 6.5 hours for 192 candidate trees. Even though we use parallel computation, the exhaustive “head” computation results in a tremendous increase in time and running storage. One possibility is to investigate only those “discourse-rich” heads, mainly in the deeper layers, for future work.

Ethical Considerations

We carefully select the dialogue corpora used in this paper to control for potential biases, hate-speech and inappropriate language by using human annotated corpora and professionally curated resources. Further, we consider the privacy of dialogue partners in the selected datasets by replacing names with generic user tokens.

Since we are investigating the nature of the discourse structures captured in large PLMs, our work can be seen as making these models more transparent. This will hopefully contribute to avoid unintended negative effects, when the growing number of NLP applications relying on PLMs are deployed in practical settings.

In terms of environmental cost, the experiments described in the paper make use of RTX 2080 Ti GPUs for tree extraction and A100 GPUs for BART fine-tuning. We used up to 4 GPUs for the parallel computation. The experiments on corpus STAC took up to 1.2 hours for one language model, and we tested a dozen models. We note that while our work is based on exhaustive research on all the attention heads in PLMs to obtain valuable insights, future work will be able to focus more on discourse-rich heads, which can help to avoid the quadratic growth of computation time for longer documents.

References

- Stergos Afantenos, Nicholas Asher, Farah Benamara, Anais Cadilhac, Cedric Dégremont, Pascal Denis, Markus Guhe, Simon Keizer, Alex Lascarides, Oliver Lemon, et al. 2012. [Modelling strategic conversation: model, annotation design and corpus](#). In *Proceedings of the 16th Workshop on the Semantics and Pragmatics of Dialogue (Seinedial)*, Paris.
- Stergos Afantenos, Eric Kow, Nicholas Asher, and Jérémy Perret. 2015. [Discourse parsing for multi-party chat dialogues](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 928–937, Lisbon, Portugal. Association for Computational Linguistics.
- Nicholas Asher, Nicholas Michael Asher, and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press.
- Nicholas Asher, Julie Hunter, Mathieu Morey, Benamara Farah, and Stergos Afantenos. 2016. [Discourse structure and dialogue acts in multiparty dialogue: the STAC corpus](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2721–2727, Portorož, Slovenia. European Language Resources Association (ELRA).
- Sonia Badene, Kate Thompson, Jean-Pierre Lorré, and Nicholas Asher. 2019a. [Data programming for learning discourse structure](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 640–645, Florence, Italy. Association for Computational Linguistics.
- Sonia Badene, Kate Thompson, Jean-Pierre Lorré, and Nicholas Asher. 2019b. [Weak supervision for learning discourse structure](#). In *EMNLP*.
- Regina Barzilay and Mirella Lapata. 2008. [Modeling local coherence: An entity-based approach](#). *Computational Linguistics*, 34(1):1–34.
- Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. [Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory](#). In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.
- Somnath Basu Roy Chowdhury, Faeze Brahman, and Snigdha Chaturvedi. 2021. [Is everything in order? a simple way to order sentences](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10769–10779.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jason Eisner. 1996. [Three new probabilistic models for dependency parsing: An exploration](#). In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
- Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. [A survey on dialogue summarization: Recent advances and new frontiers](#). *arXiv preprint arXiv:2107.03175*.
- Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2019. [Evaluating discourse in structured text representations](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 646–653, Florence, Italy. Association for Computational Linguistics.
- Luke Gessler, Shabnam Behzad, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. 2021. [Discodisco at the disrpt2021 shared task: A system for discourse segmentation, classification, and connective detection](#). In *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 51–62.
- Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. [SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization](#). In *Proceedings of the 2nd Workshop on*

- New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.
- Yuchen He, Zhuosheng Zhang, and Hai Zhao. 2021. [Multi-tasking dialogue comprehension with discourse parsing](#). In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 551–561, Shanghai, China. Association for Computational Linguistics.
- John Hewitt and Christopher D. Manning. 2019. [A structural probe for finding syntax in word representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.
- Patrick Huber and Giuseppe Carenini. 2019. [Predicting discourse structure using distant supervision from sentiment](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2306–2316, Hong Kong, China. Association for Computational Linguistics.
- Patrick Huber and Giuseppe Carenini. 2020. [MEGA RST discourse treebanks with structure and nuclearity from scalable distant sentiment supervision](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7442–7457, Online. Association for Computational Linguistics.
- Patrick Huber and Giuseppe Carenini. 2022. [Towards understanding large-scale discourse structures in pre-trained and fine-tuned language models](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*.
- Ganesh Jawahar, Benoît Sagot, and Djamel Seddah. 2019. [What does BERT learn about the structure of language?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- Qi Jia, Yizhu Liu, Siyu Ren, Kenny Zhu, and Haifeng Tang. 2020. [Multi-turn response selection using dialogue dependency relations](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1911–1920.
- Kailang Jiang, Giuseppe Carenini, and Raymond Ng. 2016. [Training data enrichment for infrequent discourse relations](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2603–2614, Osaka, Japan. The COLING 2016 Organizing Committee.
- Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. [How can we know what language models know?](#) *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. [Codra: A novel discriminative framework for rhetorical analysis](#). *Computational Linguistics*, 41(3):385–435.
- Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. 2019. [Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction](#). In *International Conference on Learning Representations*.
- Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2021. [Improving neural rst parsing model with silver agreement subtrees](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1600–1612.
- Naoki Kobayashi, Tsutomu Hirao, Kengo Nakamura, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2019. [Split or merge: Which is better for unsupervised RST parsing?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5797–5802, Hong Kong, China. Association for Computational Linguistics.
- Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. [Discourse probing of pretrained language models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3849–3864.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Jiaqi Li, Ming Liu, Min-Yen Kan, Zihao Zheng, Zekun Wang, Wenqiang Lei, Ting Liu, and Bing Qin. 2020. [Molwenti: A challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2642–2652, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Jiaqi Li, Ming Liu, Zihao Zheng, Heng Zhang, Bing Qin, Min-Yen Kan, and Ting Liu. 2021. [Dadgraph: A discourse-aware dialogue graph neural network for multiparty dialogue machine reading comprehension](#). *arXiv preprint arXiv:2104.12377*.

- Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. [Keep meeting summaries on topic: Abstractive multi-modal meeting summarization](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196, Florence, Italy. Association for Computational Linguistics.
- Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. [Text-level discourse dependency parsing](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35, Baltimore, Maryland. Association for Computational Linguistics.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. [DailyDialog: A manually labelled multi-turn dialogue dataset](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Yang Liu and Mirella Lapata. 2018. [Learning structured text representations](#). *Transactions of the Association for Computational Linguistics*, 6:63–75.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Zhengyuan Liu and Nancy Chen. 2021. [Improving multi-party dialogue discourse parsing via domain integration](#). In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 122–127, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.
- Annie Louis, Aravind K Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization.
- Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. [The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems](#). In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic. Association for Computational Linguistics.
- William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- David Mareček and Rudolf Rosa. 2019. [From balustrades to pierre vinken: Looking for syntax in transformer self-attentions](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275, Florence, Italy. Association for Computational Linguistics.
- Philippe Muller, Stergos Afantenos, Pascal Denis, and Nicholas Asher. 2012. [Constrained decoding for text-level discourse parsing](#). In *Proceedings of COLING 2012*, pages 1883–1900, Mumbai, India. The COLING 2012 Organizing Committee.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence RNNs and beyond](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.
- Thanh-Tung Nguyen, Xuan-Phi Nguyen, Shafiq Joty, and Xiaoli Li. 2021. [Rst parsing from scratch](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1613–1625.
- Noriki Nishida and Yuji Matsumoto. 2022. [Out-of-domain discourse dependency parsing via bootstrapping: An empirical analysis on its effectiveness and limitation](#). *Transactions of the Association for Computational Linguistics*, 10:127–144.
- Jérémy Perret, Stergos Afantenos, Nicholas Asher, and Mathieu Morey. 2016. [Integer linear programming for discourse parsing](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 99–109, San Diego, California. Association for Computational Linguistics.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.
- Kechen Qin, Lu Wang, and Joseph Kim. 2017. [Joint modeling of content and discourse relations in dialogues](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 974–984, Vancouver, Canada. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. [Language models are unsupervised multitask learners](#). *OpenAI blog*, 1(8):9.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. [Know what you don’t know: Unanswerable questions for SQuAD](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.
- Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1978. [A simplest systematics for the organization](#)

- of turn taking for conversation. In *Studies in the organization of conversational interaction*, pages 7–55. Elsevier.
- Emanuel A Schegloff. 2007. *Sequence organization in interaction: A primer in conversation analysis I*, volume 1. Cambridge university press.
- Zhouxing Shi and Minlie Huang. 2019. [A deep sequential model for discourse parsing on multi-party dialogues](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7007–7014.
- Ante Wang, Linfeng Song, Hui Jiang, Shaopeng Lai, Junfeng Yao, Min Zhang, and Jinsong Su. 2021. [A structure self-aware model for discourse parsing on multi-party dialogues](#). In *Proceedings of the Thirtieth International Conference on International Joint Conferences on Artificial Intelligence*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2020. [Do we really need that many parameters in transformer for extractive summarization? discourse can help !](#) In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 124–134, Online. Association for Computational Linguistics.
- Wen Xiao, Patrick Huber, and Giuseppe Carenini. 2021. [Predicting discourse trees from transformer-based neural summarizers](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4139–4152, Online. Association for Computational Linguistics.
- Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. [Discourse-aware neural extractive text summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. [DIALOGPT : Large-scale generative pre-training for conversational response generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.
- Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. [Dialoglm: Pre-trained model for long dialogue understanding and summarization](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11765–11773.
- Zining Zhu, Chuer Pan, Mohamed Abdalla, and Frank Rudzicz. 2020. [Examining the rhetorical capacities of neural language models](#). In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 16–32, Online. Association for Computational Linguistics.

A Semi-sup. Layer-Wise Results

We consider both **layer-wise** attention matrices - averaging 16 attention heads for every layer which gives 12 candidate layers -, and **head-wise** attention matrices - taking each attention matrix individually which results in 192 candidate matrices. Here we show results completed with layer-wise matrices for the whole test set and tree-like examples in Table 9 and Table 10.

B Molweni Corpus Quality Investigation

Molweni (Li et al., 2020) is a corpus derived from Ubuntu Chat Corpus (Lowe et al., 2015). It contains 10,000 short dialogues between 8 to 15 utterances, annotated in SDRT framework.

Considering the complexity of Ubuntu chat logs (multiple speakers, entangled discussion with various topics), we first conduct an examination of the corpus. Disappointingly, we found heavy repetition within sequential documents and inconsistency in discourse annotation among the same utterances. We thus decide not to include it in this work.

Train on → Test with ↓	BART F ₁	+ SO-DD F ₁	+ SO-STAC F ₁
Gold H	57.6	58.2	59.5
Semi-sup-10 1L	55.8 _{0.008}	55.7 _{0.010}	55.6 _{0.009}
Semi-sup-30 1L	55.8 _{0.006}	56.5 _{0.004}	56.3 _{0.004}
Semi-sup-50 1L	56.2 _{0.002}	56.4 _{0.007}	56.4 _{0.001}
Semi-sup-10 1H	57.0 _{0.012}	57.2 _{0.012}	57.1 _{0.026}
Semi-sup-30 1H	57.3 _{0.005}	57.3 _{0.013}	59.2 _{0.009}
Semi-sup-50 1H	57.4_{0.004}	57.7_{0.005}	59.3_{0.007}

Table 9: Micro-F₁ scores on STAC test set with BART and fine-tuned models. H = “head”, L = “layer”. Best semi-supervised score is in bold. Subscription is std. deviation.

Train on → Test with ↓	BART F ₁	+ SO-DD F ₁	+ SO-STAC F ₁
Gold H	64.8	67.4	68.6
Semi-sup-10 1L	59.4 _{0.028}	60.6 _{0.029}	58.3 _{0.018}
Semi-sup-30 1L	62.1 _{0.002}	61.8 _{0.012}	59.8 _{0.009}
Semi-sup-50 1L	62.1 _{0.000}	62.3 _{0.003}	59.9 _{0.006}
Semi-sup-10 1H	54.6 _{0.058}	59.2 _{0.047}	61.6 _{0.056}
Semi-sup-30 1H	60.3 _{0.047}	60.3 _{0.044}	65.6 _{0.043}
Semi-sup-50 1H	64.8_{0.000}	66.3_{0.023}	68.1_{0.014}

Table 10: Micro-F₁ scores on STAC projective tree subset with BART and SO fine-tuned BART models.

Clus ID	Doc ID	#Theor =arc	#Err arc	#Theor =rel	#Err rel
1	{1, 2, 3}	18	2	16	2
2	{7, 8, 9}	18	0	18	7
3	{10, 11, 12, 13, 14}	80	4	76	25
...					
105	500	4787	284	4503	606
-	-	100%	5.9%	100%	13.5%

Table 11: Quantitative resume of link and relation inconsistency in Molweni test set. “Theor =arc”: number of arcs between the same utterances, *a priori* should be linked in the same way; “Theor =rel”: number of relations between the linked utterances.

Clusters: Among 500 dialogues in discourse augmented test set, we found 105 “clusters”. One cluster groups all the documents with only one or two different utterances. For instance, document id 10 and 11 are in the same cluster since only the second utterance is different (Figure 10). A similar situation is attested in the documents {1, 2, 3}, {7, 8, 9}, {19, 20, 21}, to name a few.

Annotation Inconsistency: A closer examination of the annotation in similar examples reveals inconsistency for both discourse links and rhetorical relations. Precisely, we investigate every *document pair* (two documents in the same cluster) in all 105 clusters in the test set. A visualization of inconsistency for documents 10 and 11 is shown in Figure 10: apart from EDU₂, we expect the same links and relations among other EDUs. However, we observe one link inconsistency (in red) and two relation inconsistencies (in blue). In total, we find 6% of link errors (#Err arc) within the same EDUs and 14% of relation errors (#Err rel) in the test set⁶. The scores are shown in Table 11.

The Ubuntu Chat Corpus contains long dialogues with entangled discussion. A pre-processing had been made to generate shorter dialogues. While these slightly different short dialogues could be interesting for other dialogue studies in the field. Our focus on the discourse structure request more various data points and most importantly, the coherent discourse annotation.

C Precision and Recall Scores for Direct and Indirect Arcs in STAC Tree Set

To compare the performance of the whole test set and tree-structured subset, we present the recall and

⁶For validation and train sets we find similar error rates.

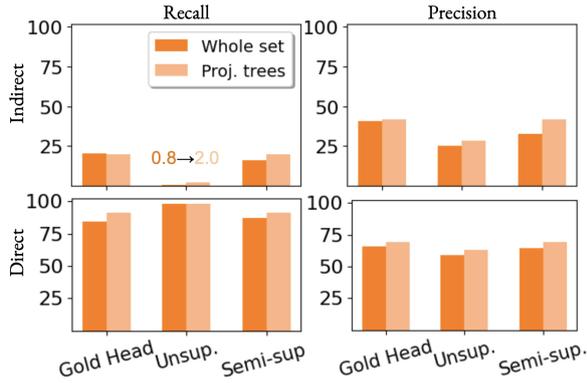


Figure 7: Recall and precision metrics in whole test set (darker color) vs. projective tree subset (brighter color), with BART model.

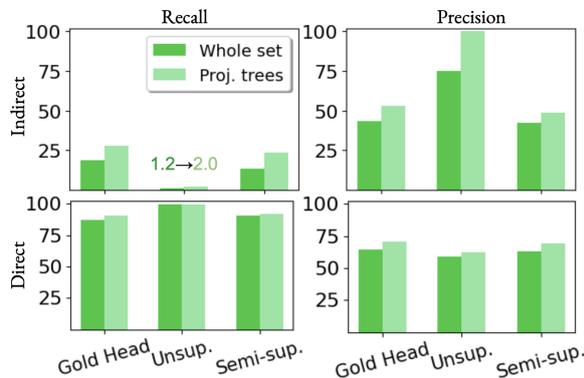


Figure 8: Recall and precision metrics in whole test set (darker color) vs. projective tree subset (brighter color), with BART+SO-DD model.

precision scores of BART (Fig. 7), BART+SO-DD (Fig. 8), and BART+SO-STAC (Fig. 9) separately.

D Qualitative Analysis in STAC

We show a few concrete tree examples: 3 well predicted (Figure 11, 12, 13), 3 badly predicted (Figure 14, 15, 16), and 2 random examples (Figure 17, 18). Some patterns observed from badly predicted structures: (1) chain-style prediction: as shown in Figure 15 and 18 where only adjacent EDUs are linked together; (2) inaccurate indirect arc prediction: especially for long documents such as the one in Figure 16.

E Results with other PLMs

We test with RoBERTa (Liu et al., 2019), DialoGPT (Zhang et al., 2020), and DialogLED (DialogLM with Longformer) (Zhong et al., 2022) to see how different language models encode discourse information. As shown in Table 12, the most discourse-rich head in RoBERTa slightly under-

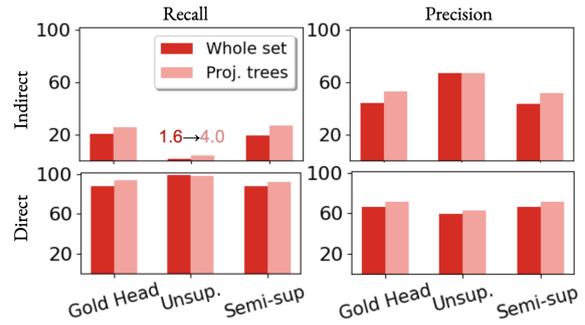


Figure 9: Recall and precision metrics in whole test set (darker color) vs. projective tree subset (brighter color), with model BART+SO-STAC.

Model	H_{ora}	Unsup			Semi-sup	
		H_g	H_l	Semi10	Semi30	Semi50
BART	57.6	56.6	56.4	57.0 _{0.012}	57.3 _{0.005}	57.4 _{0.004}
+ SO-DD	58.2	56.8	57.1	57.2 _{0.012}	57.3 _{0.013}	57.7 _{0.005}
+ SO-STAC	59.5	56.7	57.2	57.1 _{0.026}	59.2 _{0.009}	<u>59.3</u> _{0.007}
RoBERTa	57.4	56.8	56.8	55.6 _{0.013}	56.8 _{0.002}	<u>56.9</u> _{0.003}
DialoGPT	56.2	42.7	36.2	52.9 _{0.043}	55.1 _{0.017}	<u>56.2</u> _{0.000}
DialogLED	57.2	56.8	56.7	54.6 _{0.026}	54.7 _{0.061}	<u>56.6</u> _{0.019}
+ SO-DD	57.7	56.4	56.6	55.0 _{0.028}	56.1 _{0.024}	57.3 _{0.009}
+ SO-STAC	58.4	56.8	57.1	57.7 _{0.001}	<u>58.2</u> _{0.005}	57.7 _{0.001}

Table 12: Micro- F_1 on STAC with other PLMs. Best score (except H_{ora}) in each row is underlined.

perform BART (-0.2%), so does the DialogLED (-0.4%) and DialoGPT (-1.4%). Sentence ordering fine-tuned DialogLED model outperforms the original one, proving that our proposed SO task can help encoding the discourse information.

F Huggingface Models

Table 13 shows the models and the sources we obtained from Huggingface library (Wolf et al., 2020).

Model
BART-large
https://huggingface.co/facebook/bart-large
BART-large-cnn
https://huggingface.co/facebook/bart-large-cnn
BART-large-samsum
https://huggingface.co/linydub/bart-large-samsum
BART-large-finetuned-squad2
https://huggingface.co/phiodyr/bart-large-finetuned-squad2
RoBERTa-large
https://huggingface.co/roberta-large
DialoGPT-small
https://huggingface.co/microsoft/DialoGPT-small
DialogLED-large-5120
https://huggingface.co/MingZhong/DialogLED-large-5120

Table 13: Huggingface models and URLs.

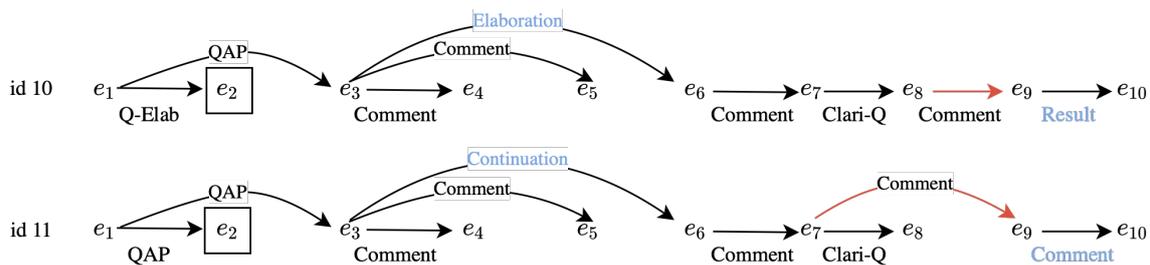


Figure 10: Similar documents in the same cluster. Circled EDUs are different. In red: inconsistent discourse arcs; in blue: inconsistent rhetorical relation.

test id 10:

[e₁] matthew99857: so do i need additional hardware to fix it ?
 [e₂] vocx: ca n't you disable the raid from the bios ? check your motherboard manual .
 [e₃] ikonkia: just use the disk as an individual disk
 [e₄] sugi: vocxi : oh i am sorry . i misunderstood you . thank i will try it now
 [e₅] vocx: you need to word better your answers , seems like nobody in getting you today .
 [e₆] sugi: vocx : iso 9660 cd-rom filesystem data udf filesystem data (unknown version , id 'nsr01 ')
 [e₇] ikonkia: looks like that should work as a loop back file system
 [e₈] sugi: -mount -o loop but instead of .iso .mdf ? or the .mds file ?
 [e₉] ikonkia: try it , linux see 's it as a " image " so it may work
 [e₁₀] sugi: vocx : wow it worked , i feel retard for nto

test id: 11

[e₁] matthew99857: so do i need additional hardware to fix it ?
 [e₂] ikonkia: no you need to stop using raid
 [e₃] ikonkia: just use the disk as an individual disk
 [e₄] sugi: vocxi : oh i am sorry . i misunderstood you . thank i will try it now
 [e₅] vocx: you need to word better your answers , seems like nobody in getting you today .
 [e₆] sugi: vocx : iso 9660 cd-rom filesystem data udf filesystem data (unknown version , id 'nsr01 ')
 [e₇] ikonkia: looks like that should work as a loop back file system
 [e₈] sugi: -mount -o loop but instead of .iso .mdf ? or the .mds file ?
 [e₉] ikonkia: try it , linux see 's it as a " image " so it may work
 [e₁₀] sugi: vocx : wow it worked , i feel retard for nto

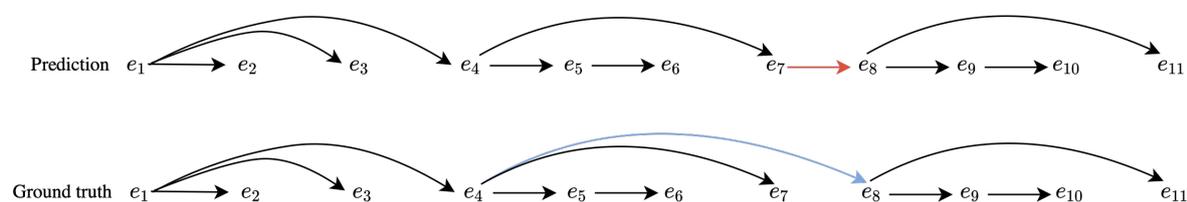


Figure 11: Well predicted example: *pilot02-4*. #EDUs: 11. UAS: 90%. In red: FP arcs; in blue: FN arcs.

[e₁] Cat: anyone would give me clay? [e₂] Thomas: none here [e₃] william: no [e₄] Cat: I have one wood to exchange [e₅] Cat: any takers? [e₆] william: no [e₇] Cat: for sheep, wheat or clary [e₈] Thomas: can I buy a sheep for two ore? [e₉] william: have none [e₁₀] Thomas: kk [e₁₁] Cat: no sheep

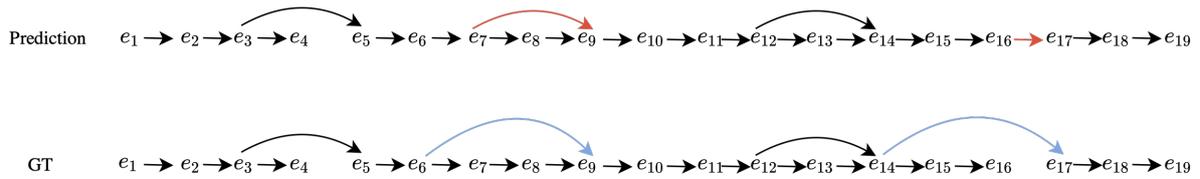


Figure 12: Well predicted example: *pilot02-18*. #EDUs: 19. UAS: 88.9%. In red: FP arcs; in blue: FN arcs.

[e₁] william: hi markus. [e₂] william: how many people are we waiting for? [e₃] Thomas: think it's 1 more [e₄] william: ok [e₅] Markus: yes, one more [e₆] Markus: seems there's a hiccup logging into the game ... [e₇] Thomas: that's ok, I not on a schedule [e₈] Thomas: *I'm [e₉] Markus: I guess you two had no problems joining the game? [e₁₀] william: nope [e₁₁] Markus: Ah great! [e₁₂] Markus: So, one of you can now start the game. [e₁₃] Markus: Have fun! [e₁₄] william: the arrow is pointing at me [e₁₅] william: but i cant press roll [e₁₆] william: oh sorry [e₁₇] Thomas: u can place a settlement [e₁₈] Thomas: first [e₁₉] Thomas: u roll later

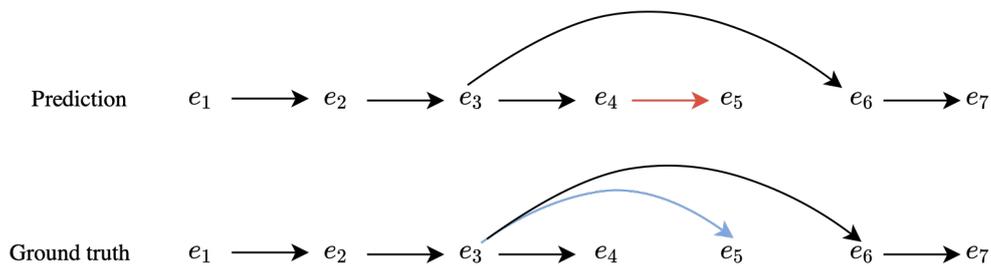


Figure 13: Well predicted example: *s1-league3-game3*. #EDUs: 7. UAS: 83.3%. In red: FP arcs; in blue: FN arcs.

[e₁] Gaeilgeoir: ? [e₂] yiin: build road [e₃] inca: think we're meant to negotiate trades in the chat before offering [e₄] yiin: oop [e₅] yiin: ok then [e₆] inca: part of the guys' experiment [e₇] yiin: oh i see

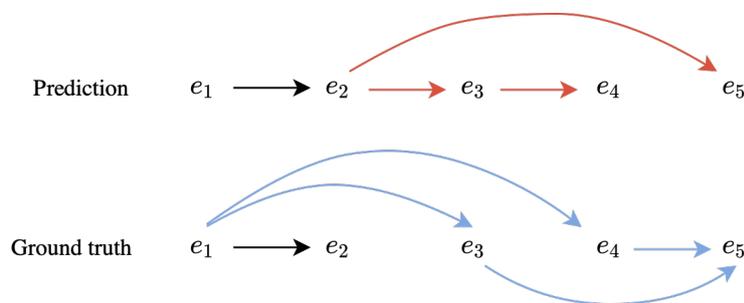


Figure 14: Badly predicted example: *s2-leagueM-game4*. #EDUs: 5. UAS: 20%. In red: FP arcs; in blue: FN arcs.

[e₁] dmm: i can give a sheep or wood for a wheat. [e₂] dmm: any takers? [e₃] inca: sheep would be good. [e₄] CheshireCatGrin: Not here. [e₅] dmm: okay.

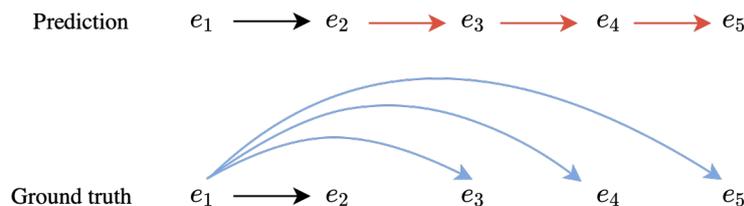


Figure 15: Badly predicted example: *s1-league3-game3*. #EDUs: 5. UAS: 25%. In red: FP arcs; in blue: FN arcs.

[e₁] nareik15: anyone have ore. [e₂] nareik15: I have some wood to trade. [e₃] yiin: no sorry. [e₄] inca: nope, sorry. [e₅] Gaeilgeoir: no, sorry.

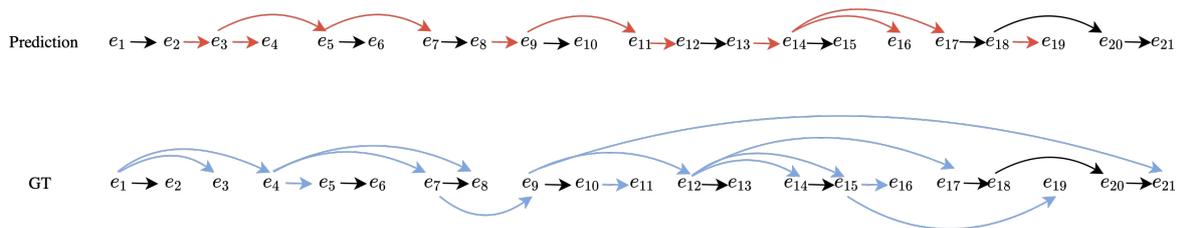


Figure 16: Badly predicted example: *s1-league4-game2*. #EDUs: 21. UAS: 30%. In red: FP arcs; in blue: FN arcs.
 [e₁] Shawnus: need wheat [e₂] Shawnus: want..clay? [e₃] ztime: you odo? [e₄] ztime: yer.. [e₅] ztime: I need clay..
 [e₆] ztime: can give wheat [e₇] Shawnus: k [e₈] Shawnus: this might be where i lose my road card a? [e₉] ztime:
 er.. [e₁₀] ztime: I think the trade is wrong? [e₁₁] ztime: did you want wheat? [e₁₂] Shawnus: yes [e₁₃] Shawnus:
 for clay [e₁₄] ztime: it said you wanted clay... [e₁₅] somdechn: We all want wheat man [e₁₆] somdechn: and clay..
 [e₁₇] ztime: ok [e₁₈] ztime: thanks.. [e₁₉] Shawnus: haha [e₂₀] Shawnus: thanks [e₂₁] somdechn: That happens in
 the real game as well.

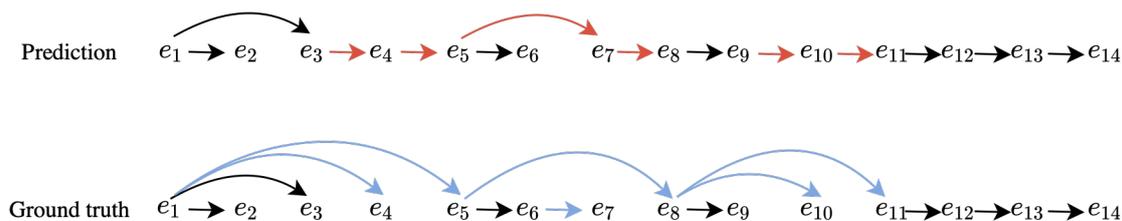


Figure 17: Random example: *s2-league4-game2*. #EDUs: 14. UAS: 53.9%. In red: FP arcs; in blue: FN arcs.
 [e₁] ztime: 7!!!! [e₂] somdechn: Yeah right... [e₃] ztime: what... is this a fix? [e₄] Shawnus: hahaha [e₅] ztime: ok
 anyone want wheat? [e₆] Shawnus: nope [e₇] Shawnus: just someone to roll 9's.. [e₈] somdechn: Yes [e₉] somdechn:
 I can give you wood. [e₁₀] ztime: was that yes to a trade somdech? [e₁₁] ztime: OK.. cool.. for 1 wheat?
 [e₁₂] somdechn: and an ore.. :) [e₁₃] ztime: err.. don't have ore.. [e₁₄] ztime: thanks..

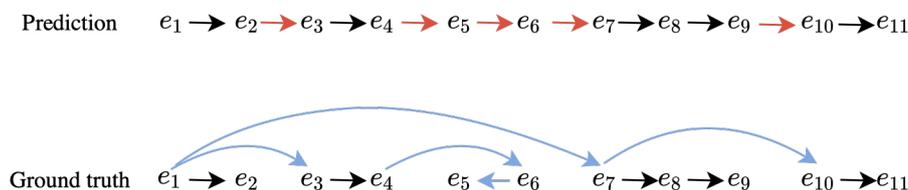


Figure 18: Random example: *s1-league3-game3*. #EDUs: 11. UAS: 50%. In red: FP arcs; in blue: FN arcs.
 [e₁] nareik15: anyone have wood to trade. I have sheep [e₁] yiin: no [e₁] Gaeilgeoir: Sorry, [e₁] Gaeilgeoir: I need
 wood too [e₁] Gaeilgeoir: I have wheat [e₁] Gaeilgeoir: if you want [e₁] inca: do you have wheat kieran? [e₁] inca:
 if so [e₁] inca: i can trade wood [e₁] nareik15: sorry, [e₁] nareik15: plenty of sheep though :)

Relation Extraction with Weighted Contrastive Pre-training on Distant Supervision

Zhen Wan Fei Cheng Qianying Liu
Zhuoyuan Mao Haiyue Song Sadao Kurohashi

Kyoto University, Japan

{zhenwan, feicheng, ying, zhuoyuanmao, song, kuro}
@nlp.ist.i.kyoto-u.ac.jp

Abstract

Contrastive pre-training on distant supervision has shown remarkable effectiveness in improving supervised relation extraction tasks. However, the existing methods ignore the intrinsic noise of distant supervision during the pre-training stage. In this paper, we propose a weighted contrastive learning method by leveraging the supervised data to estimate the reliability of pre-training instances and explicitly reduce the effect of noise. Experimental results on three supervised datasets demonstrate the advantages of our proposed weighted contrastive learning approach compared to two state-of-the-art non-weighted baselines. Our code and models are available at: <https://github.com/YukinoWan/WCL>.

1 Introduction

Relation extraction (RE) is the task of identifying the relationship between entities mentioned in the text, which can benefit many downstream tasks such as question answering and knowledge base population. Since most of the existing RE models (Zhang et al., 2020; Zeng et al., 2020; Lin et al., 2020; Wang and Lu, 2020; Zhong and Chen, 2021) are trained on the labeled data, the amount of training data limits the performance of supervised RE systems. To tackle this problem, recent work leverage semi-supervised distant supervision (DS) (Mintz et al., 2009; Lin et al., 2016; Vashishth et al., 2018; Chen et al., 2021) approach to generate abundant training data by aligning knowledge bases (KBs) and raw corpora. However, distantly supervised relation extraction (DSRE) inevitably suffers from wrong labeling noise. Introducing a robust framework that utilizes both the abundant but noisy data from DS and the scarce but accurate data from human annotations becomes a new research line to improve RE systems.

Recent works (Baldini Soares et al., 2019; Ormándi et al., 2021; Peng et al., 2020) propose a

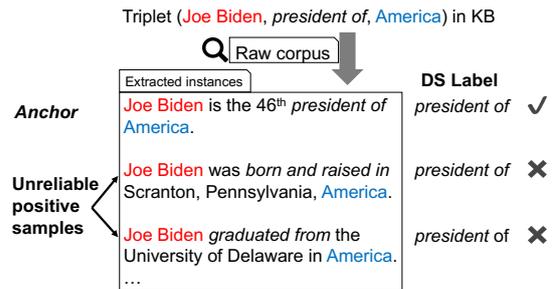


Figure 1: An example of unreliable positive samples caused by DS noise.

two-stage RE framework that they first design a RE-oriented task to pre-train BERT on DS data and then fine-tune on human-annotated (HA) datasets. Peng et al. (2020) use Wikipedia articles as the corpus and Wikidata as the KB in the pre-training stage to construct the DS data, and they introduce a contrastive learning-based method to pre-train BERT on the generated DS data. Given an anchor instance with a specific relation in the DS data, their contrastive learning method randomly selects one positive sample holding the same relation and maximizes the similarity between the anchor and positive sample. Meanwhile, their method randomly selects multiple negative samples holding different relations from the anchor and minimizes the similarity between the anchor and negative samples. The results show that their RE-oriented pre-training can effectively improve the final performance of the RE task on various target datasets.

However, in their pre-training stage, they ignore the intrinsic wrong labeling noise in the generated DS data. Since their method relies on the DS-labeled relation types to sample positive and negative instances, the noisy labeling problem leads to unreliable samples in Figure 1, potentially limiting the pre-training stage’s effectiveness. To better utilize DS data, we propose a novel weighted contrastive learning framework to both use the abundant DS data and tackle the inevitable DS noise.

First, we train a relation classifier on the HA dataset and leverage the classifier to predict the relation type of instances in the DS data. Then for each DS instance, based on the output of the classifier, we can compute the confidence score to measure the reliability of its labeled relation type. Finally, we introduce weights based on computed confidence scores into the contrastive learning loss to focus more on reliable instances while less on noisy ones.

Besides, distant supervision relies on the existing KBs to align raw corpora. To alleviate the need for KBs, we propose a new strategy to extract a triplet set from the HA dataset for generating DS data. We also include a KB-derived DS dataset in our experiments to show that our proposal can still work well for regular DS.

In conclusion, we propose a weighted contrastive pre-training approach for supervised relation extraction and introduce its details in Section 2. Then we perform the experiments on three datasets to compare our proposed method with existing baselines in Section 3.

2 Proposed Method

2.1 Overview

We show the overview of our proposal in Figure 2. We start by generating the DS data relying on the HA dataset. Then in the first stage, we introduce a weighted contrastive learning method by leveraging the HA data to estimate the reliability of DS instances for contrastive pre-training. In the second stage, we further fine-tune our pre-trained model on the HA dataset.

2.2 Distantly Supervised Dataset Construction

Since DS uses existing knowledge bases to generate training data, in the case that we have no proper existing KBs in some domains but only the annotated dataset, we first extract all entities based on each sentence, and if any two of them are labeled a relation type, they will generate a triplet with a particular relation. Otherwise, they will still generate a triplet but labeled NA (no relation). After constructing the KB, we can extract sentences containing two entities of each triplet from raw corpora. To balance the number of sentences extracted by each triplet, we also add an upper bound 100 to the number of extracted sentences.

2.3 Two-stage RE Framework

Instance representation In our pre-training stage, we use BERT to obtain the representation for each input instance. For the input format, we follow PURE (Zhong and Chen, 2021) by adding extra special markers to mark the beginning and the end of two entities. For example, given an instance x : “*Joe Biden is the president of America.*”, the input sequence is “[CLS] [H_CLS] *Joe Biden* [H_SEP] *is the president of* [T_CLS] *America* [T_SEP]. [SEP]”. Denote the k -th output vector of the BERT encoder as h_k . Assuming i and j are the indices of two beginning entity markers [H_CLS] and [T_CLS], we define the instance representation as:

$$\mathbf{x} = h_i \oplus h_j \quad (1)$$

where \oplus stands for concatenation. Then we use the instance representation for further reliability estimation and the weighted contrastive learning in the pre-training stage.

Reliability estimation With the instance representation, we first fine-tune BERT on the HA dataset as a supervised RE task. Then with the trained relation classifier \mathcal{F} , we can make predictions on each instance in the DS data. Given an input instance x with DS labeled relation r , we can derive the confidence score c to estimate its reliability by:

$$c = \frac{\exp(\mathcal{F}(\mathbf{x}, r))}{\sum_{r' \in R} \exp(\mathcal{F}(\mathbf{x}, r'))} \quad (2)$$

where R is the set of all relation classes, and $\mathcal{F}(\mathbf{x}, r)$ computes the output of our relation classifier on the labeled class r . Through this approach, we can estimate the reliability of the labeled relation for each DS instance by its corresponding confidence score.

Stage 1: DS weighted contrastive pre-training

Contrastive learning aims at maximizing the similarity between a given instance and its positive samples while minimizing the similarity between the given instance and its negative samples. As for existing work, Peng et al. (2020) focuses on the relationship level that DS instances labeled the same relation are positive samples while DS instances labeled different relations are negative samples. The latest DSRE work (Chen et al., 2021) augments the anchor as a positive sample to avoid the effect of DS noise. Both works do not explicitly address the problem of unreliable positive and negative samples.

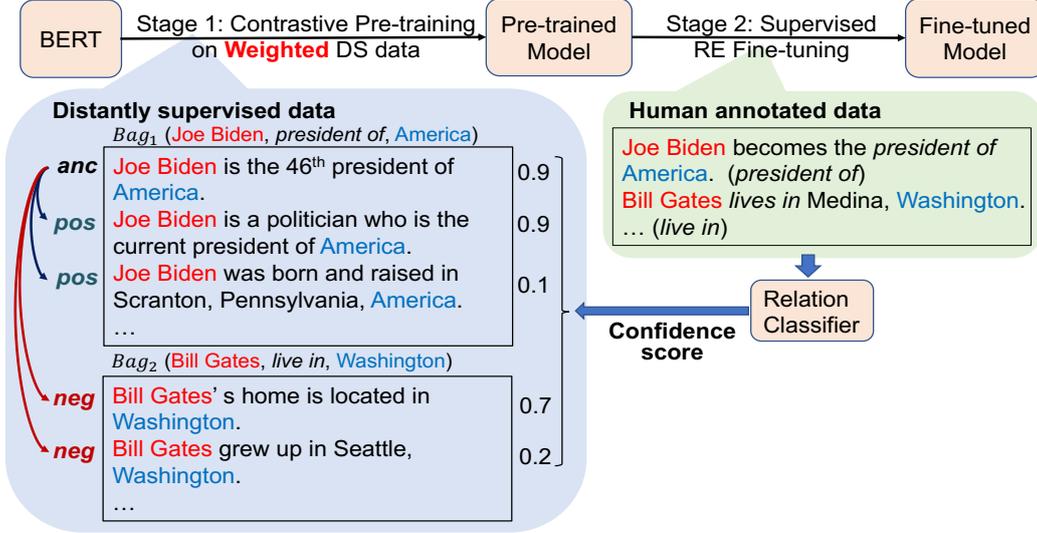


Figure 2: Overview of our proposed method.

In our work, we introduce a robust weighted contrastive learning (WCL) method with the help of reliability estimations for each instance. Given a batch input with multiple bags: $(Batch = \{B_i\}_{i=1}^G)$ where G is the number of bags in one batch, and the labeled relational triplets are different from each other. Each bag B is constructed by a certain relational triplet (e_1, r, e_2) with all instances x inside satisfying this triplet. Moreover, each instance comes along with a confidence score c estimating its reliability: $B_i = \{x_j, c_j\}_{j=1}^{N_i}$, where N_i denotes the size of bag B_i . Then the weighted contrastive learning loss of an anchor instance x_j in the bag B_i is:

$$\mathcal{L}_{WCL}^{(ij)} = -\log \left\{ \sum_{k=1}^{N_i} c_j c_k e^{\cos(\mathbf{x}_j, \mathbf{x}_k)/T} / \left(\sum_{k=1}^{N_i} c_j c_k e^{\cos(\mathbf{x}_j, \mathbf{x}_k)/T} + \sum_{r_m \neq r_j} c_j c_m e^{\cos(\mathbf{x}_j, \mathbf{x}_m)/T} \right) \right\} \quad (3)$$

where $\cos(\cdot)$ denotes the cosine similarity between two instance representations, \mathbf{x}_k denotes the representation of a positive instance sampled from the same bag, and $r_m \neq r_j$ denotes that negative samples x_m are selected from all instances in the batch that is labeled a different relation from x_j . We follow (Khosla et al., 2020) to incorporate multiple positive instances sampled from the same bag. T denotes a scaling temperature.

With the help of confidence scores, the model will focus on more reliable instances while ignoring unreliable instances, which keep pace with our goal

to utilize reliable DS data.

Besides, to inherit the ability of language understanding from BERT and avoid catastrophic forgetting, we also adopt the masked language modeling (MLM) objective from BERT.

Eventually, we define our final pre-training loss:

$$\mathcal{L} = \mathcal{L}_{WCL} + \mathcal{L}_{MLM} \quad (4)$$

Stage 2: Supervised relation extraction We then fine-tune the pre-trained model on HA datasets with state-of-the-art (SOTA) methods. For i2b2 2010VA, we follow BLUEBERT (Peng et al., 2019) by treating the relation extraction task as a sentence classification and replacing two named entities in the sentence with predefined tags. For the other two datasets, we follow the encoding method of PURE (Zhong and Chen, 2021) as introduced at the beginning of Section 2.3.

3 Experiments

3.1 Setup

HA and DS datasets We evaluate our approach on three HA relation extraction datasets: i2b2 2010VA, ACE05, and Wiki20m. Table 2 shows the statistics of each dataset. The i2b2 2010VA is a medical domain RE dataset, while the other two datasets are collected from general domains. We generate the DS data for i2b2 2010VA and ACE05 from corresponding raw corpora. Meanwhile, Wiki20m is a regular KB-based distantly supervised RE dataset containing both DS data and

Methods	i2b2 2010VA		ACE05		Wiki20m	
	25%	100%	25%	100%	25%	100%
FT	66.86	75.22	62.81	70.41	68.87	88.54
CIL + FT	67.92	75.39	59.72	69.69	89.67	91.64
RECN + FT	67.65	75.43	60.34	69.40	89.23	91.96
WCL + FT (ours)	68.50	76.15	61.30	69.47	90.28	92.67

Table 1: **Evaluation results on various datasets.** 25% denotes the low-resource setting, and 100% denotes the full-resource setting. We compute three-run average Micro-F1 for our proposed methods in all the results.

Dataset	# Rel.	# Train	# Dev	# Test
i2b2 2010VA	8	3,120	11	6,147
ACE05	6	10,051	2,424	2,050
Wiki20m	80	8,279	4,140	28,977
ACE05 (NP)	6	3,939	922	923

Table 2: **Statistics of datasets.** Rel. denotes relation types. NP denotes removing pronoun from ACE05.

Dataset	# Triplets	Corpora	# DS Ins. (NA)
i2b2 2010VA	2,777	MIMIC-III	36K (76K)
ACE05	3,883	Gigaword5	98K (461K)
Wiki20m	-	Wiki20m	286K (698K)
ACE05 (NP)	3,218	Gigaword5	60K (273K)

Table 3: **Statistics of DS data.** Triplets are extracted from the HA dataset. DS Ins. denotes relational instances generated by DS. NA denotes the no-relation instances. NP denotes removing pronoun from ACE05.

HA data and it is worth noting that we intend to show that our method can also work well on existing DS datasets. Table 3 shows the statistics of DS data.

Baselines We have a naive baseline by directly fine-tuning (FT) on each dataset as a supervised RE task. We set two two-stage framework baselines: the first one is to use the SOTA method RE-Context-or-Names (RECN) (Peng et al., 2020) in pre-training, and the second one is to use the SOTA DSRE method Contrastive Instance Learn-

Methods	ACE05		Wiki20m	
	25%	100%	25%	100%
FT	60.45	69.82	66.58	89.38
CIL + FT	60.12	69.36	90.25	91.26
RECN + FT	58.68	68.04	90.18	91.75
WCL + FT (ours)	60.20	69.73	91.06	92.94

Table 4: **Evaluation results on the development set datasets.** 25% denotes the low-resource setting, and 100% denotes the full-resource setting.

Methods	ACE05 (no pronouns)	
	25%	100%
FT	62.22	70.29
CIL + FT	63.31	69.76
RECN + FT	62.43	70.09
WCL + FT (ours)	64.45	71.10

Table 5: **Evaluation on ACE05 after removing pronouns.**

ing (CIL) (Chen et al., 2021) in pre-training.

Implementation details To further confirm the effectiveness of our proposal, we also conduct the experiments in the low-resource setting by randomly selecting 25% of the full HA data to construct the DS data for pre-training and finally fine-tune on this 25% HA data. Refer to Appendix A for other implementation details.

3.2 Main Results

Table 1 compares our model to other baselines. From the results, we can observe that: (1) For both the i2b2 2010VA and the Wiki20m, all two-stage models outperform the FT baseline, which indicates the effectiveness of our strategy to construct DS data from HA datasets, especially in the low-resource setting. (2) For both the i2b2 2010VA and the Wiki20m, our proposed model achieves the best F1 scores over all baselines. This improvement shows that it is worth estimating the reliability of each DS instance with the help of HA datasets in our weighted contrastive pre-training. (3) For the ACE05, the pre-training methods cannot outperform the FT baseline. To analyze this problem, we perform extra experiments on ACE05 in Section 3.3.

Besides, we also compare performances on the development set of two datasets as shown in Table 4, the experiment results emphasize the consistent improvement of our proposed methods.

3.3 Further Analysis

We find that ACE05 contains many pronoun entities, for example, "*He lives in America.*". As pronoun entities such as "*He*" naturally come along with much more severe noise in DS, we also conduct extra experiments by removing sentences containing pronoun entities in ACE05 and the corresponding DS data to confirm the effect of pronouns.

After removing pronoun entities in ACE05, as shown in Table 5, our model outperforms all baselines, including FT, which indicates that pronoun entities bring mishandled noise in the pre-training stage and limit the effect of our DS data construction approach.

4 Conclusions

We introduce a weighted contrastive pre-training method by leveraging the HA dataset to estimate the reliability of instances in the abundant DS data. To alleviate the need for KBs, we also propose to construct DS data based on the triplets derived from the HA dataset for pre-training. Experimental results demonstrate that our proposed method outperforms SOTA work on target HA datasets.

Limitations

In this paper, we propose a weighted contrastive pre-training approach for supervised relation extraction.

While our approach is simple and effective, one limitation is that the reliability estimation requires a certain amount of annotated data. Under certain settings such as few-shot learning, large-scale labeled data may not be available, and the reliability of DS data could be estimated in an unsupervised manner based on similarity-based metrics, which we leave as future work.

Another limitation is that the distant supervision of ACE05 contains additional noise caused by pronoun entities. In Section 3.3, we do the investigation by temporally removing them. In future work, we assume more strict DS extraction criteria (e.g. only entity pairs located in the same clause) might reduce the production of such noise and alleviate this situation.

Acknowledgements

This work was partially supported by MHLW PRISM Grant Number 21AC5001, JSPS KAKENHI Grant Number 22J1371, JSPS KAKENHI

Grant Number 21J23124, JSPS KAKENHI Grant Number 21H00308, and JST SPRING Grant Number JPMJSP2110.

References

- Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. [Matching the blanks: Distributional similarity for relation learning](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.
- Tao Chen, Haizhou Shi, Siliang Tang, Zhigang Chen, Fei Wu, and Yueting Zhuang. 2021. [CIL: Contrastive instance learning framework for distantly supervised relation extraction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6191–6200, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. [Supervised contrastive learning](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.
- Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. [Neural relation extraction with selective attention over instances](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.
- Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. [A joint neural model for information extraction with global features](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. [Distant supervision for relation extraction without labeled data](#). In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Róbert Ormándi, Mohammad Saleh, Erin Winter, and Vinay Rao. 2021. [Webred: Effective pretraining and finetuning for relation extraction on the web](#). *CoRR*, abs/2102.09681.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. [Learning from Context or Names? An Empirical Study on Neural Relation Extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3661–3672, Online. Association for Computational Linguistics.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. [Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets](#). In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. [RESIDE: Improving distantly-supervised neural relation extraction using side information](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266, Brussels, Belgium. Association for Computational Linguistics.

Jue Wang and Wei Lu. 2020. [Two are better than one: Joint entity and relation extraction with table-sequence encoders](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. [Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9507–9514.

Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. [Minimize exposure bias of Seq2Seq models in joint entity and relation extraction](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 236–246, Online. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. [A frustratingly easy approach for entity and relation extraction](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

A Implementation Details

During the construction of DS data, we use the preprocessing tool NLTK to split raw corpora into sentences.

Hyperparameter	Range	Best
Bag size	2-8	4
Batch size	8-32	16
Temperature	0.05-1.0	0.2

Table 6: Hyperparameter optimization.

We use *bert-base-uncased* (Devlin et al., 2019) as the base encoders for ACE05, ACE05 (no pronouns), and Wiki20m, for a fair comparison with previous works. We also use *bluebert* (Peng et al., 2019) as the base encoder for i2b2 2010VA since the SOTA performance is achieved based on this effective medical domain BERT.

For baseline models, we modify their official implementations to fit our experiments and follow the model settings in their papers. For our proposed method, the primary hyperparameters in the experiments are batch size, bag size, and contrastive learning temperature that directly influence the weighted contrastive learning loss, and we show our searching ranges and best values in Table 6.

We used 8 NVIDIA A100 for pre-training and 2 NVIDIA RTX3090 for fine-tuning.

CK-Transformer: Commonsense Knowledge Enhanced Transformers for Referring Expression Comprehension

Zhi Zhang

ILLC, University of Amsterdam
z.zhang@uva.nl

Helen Yannakoudakis

Dept. of Informatics, King’s College London
helen.yannakoudakis@kcl.ac.uk

Xiantong Zhen

United Imaging
zhenxt@gmail.com

Ekaterina Shutova

ILLC, University of Amsterdam
e.shutova@uva.nl

Abstract

The task of multimodal referring expression comprehension (REC), aiming at localizing an image region described by a natural language expression, has recently received increasing attention within the research community. In this paper, we specifically focus on referring expression comprehension with commonsense knowledge (KB-Ref), a task which typically requires reasoning beyond spatial, visual or semantic information. We propose a novel framework for Commonsense Knowledge Enhanced Transformers (CK-Transformer) which effectively integrates commonsense knowledge into the representations of objects in an image, facilitating identification of the target objects referred to by the expressions. We conduct extensive experiments on several benchmarks for the task of KB-Ref. Our results show that the proposed CK-Transformer achieves a new state of the art, with an absolute improvement of 3.14% accuracy over the existing state of the art¹.

1 Introduction

Referring expression comprehension (REC) aims at locating a target object/region in an image given a natural language expression as input. The nature of the task requires multi-modal reasoning and joint visual and language understanding. In the past few years, several REC tasks and datasets have been proposed, such as RefCOCO (Yu et al., 2016), RefCOCog (Mao et al., 2016) and RefCOCO+ (Yu et al., 2016) (RefCOCOs). These ‘conventional’ REC tasks typically focus on identifying an object based on visual or spatial information of the object, such as its colour, shape, location, etc.; therefore primarily evaluating a model’s reasoning abilities over visual attributes and spatial relationships.

In practice, however, people often describe an object using non-visual or spatial information – consider, for example, the sentence (expression) “Give

¹The code will be available in <https://github.com/FightingFighting/CK-Transformer>

me something soft but rich in starch to eat” (Wang et al., 2020). Such instances require reasoning beyond spatial and visual attributes, and need to be interpreted with respect to the common sense knowledge (fact) embedded in the expressions, such as knowledge about which kind of objects are edible, soft and rich in starch in the given image. Recently, Wang et al. (2020) proposed a new dataset, KB-Ref, to evaluate the reasoning ability of a model over not only visual and spatial features but also commonsense knowledge. The dataset is devised such that at least one piece of fact from a knowledge base (KB) is required for a target object (referred to by an expression) to be identified.

Therefore, searching for appropriate facts from a KB is also crucial part in KB-Ref. In contrast to the only existing work (Wang et al., 2020), in which for each object candidate, the top-K facts with the highest cosine similarity between the averaged Word2Vec (Mikolov et al., 2013) embedding of the fact and the given expression are maintained, our framework focuses on multi-modal embedding and reasoning simultaneously over both the expression and the image to identify the top-K facts. Multi-modal features encode richer information helping to improve reasoning over varying (semantic) contexts and identification of relevant facts; for example, the above example of expression can be answered with the object “banana” in an image (or, equivalently, with the object “mashed potato” in another image).

In this paper, we propose a novel multi-modal framework for KB-Ref – Commonsense Knowledge Enhanced Transformers (CK-Transformer, CK-T for short) – that integrates (top-K) facts into all object candidates in an image for better identification of the target object. Specifically, our contributions are four-fold: 1) We propose the CK-T (see Figure 1) that effectively integrates diverse input from different modalities: vision, referring expressions and facts; 2) To the best of our knowledge,

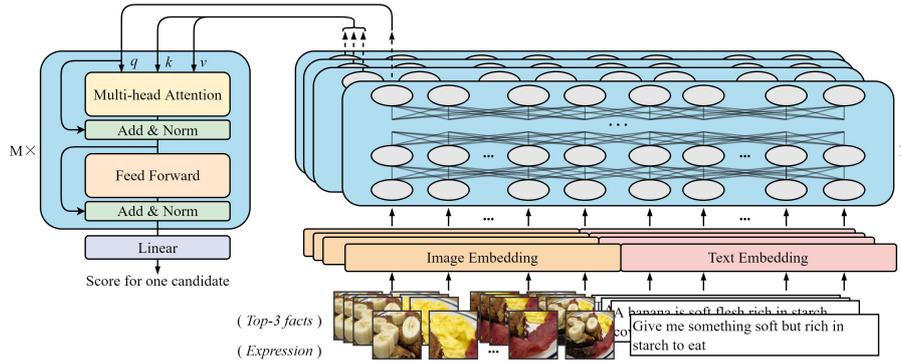


Figure 1: CK-Transformer. For each candidate (the first one in the figure), given an expression, a set of visual region candidates and top-K facts ($K=3$ in the figure), the model first encodes the expression and all top-K facts into corresponding multi-modal features, then fuses these features and maps them into a matching score for the candidate.

our approach is the first that introduces visual information into the identification of (top-K) relevant facts; 3) Our approach achieves a new state of the art using only top-3 facts per (candidate) object, which is furthermore substantially more efficient compared to existing work utilizing as much as top-50 facts; 4) We introduce facts into ‘conventional’ REC tasks, leading to improved performance.

2 Related Work

Referring expression comprehension with commonsense knowledge Different from conventional REC tasks (see Appendix A for details), KB-Ref focuses on querying objects given an expression that requires commonsense knowledge reasoning. The authors benchmarked a baseline model, ECIFA, for integration of facts, expression and image, and selects the target object by comparing the match scores between the image features and corresponding top-K fact features for all object candidates in the image (Wang et al., 2020). In our framework, we select top-K facts for each candidate by comparing the cosine similarity between the fact and expression embedding, where the embeddings are generated from a multi-modal encoder rather than a text encoder used in the ECIFA model.

Pre-trained vision–language encoders Several pre-trained multi-modal encoders (Su et al., 2019; Li et al., 2019; Chen et al., 2020; Tan and Bansal, 2019) have been proposed, achieving state-of-the-art results on vision–language tasks. Currently, UNITER (Chen et al., 2020) as one of powerful pre-trained encoders achieves the best performance on REC tasks (RefCOCOs). In this paper, we adapt UNITER such that it is used as a multi-modal en-

coder in fact search and as part of the CK-T.

3 Methodology

We formulate KB-Ref as a classification problem based on an image I consisting of a set of candidates (image regions) $I = \{c_j\}_{j=1}^n$ obtained from either ground-truth labels or predictions of a pre-trained object detector. Specifically, given an expression e , an image I and a KB, we first search for top-K facts $F_i^K = \{f_j\}_{j=1}^k$ from the KB for each candidate c_i , and then feed e , I , and F_i^K (the selected facts over I) into our CK-T simultaneously to predict the target object over all candidates.

3.1 Image-based fact search

For each candidate c_i in a given image, we retrieve all the facts from the KB (see Appendix D for details on the KB used in our framework) according to its category (e.g., a candidate object may belong to category ‘car’). Then, we calculate the cosine similarity between the facts and the given expression, where the similarities are obtained from a similarity extractor which we train by adapting UNITER. Specifically, given image–expression and image–fact pairs as input, we extract expression and fact features respectively from the position of the cross-modality output of UNITER (corresponding to the input of [CLS] token, see Appendix B for details), and then calculate the cosine similarity between the two. During training, inspired by Devlin et al. (2018), we replace 50% of ground-truth facts with random facts from the KB (with a similarity of 0), to help the model better distinguish useful facts from non-useful ones. Finally, we maintain top-K facts F_i^K with higher similarities to the expression

for each candidate c_i .

3.2 Commonsense Knowledge Enhanced Transformer

The CK-T consists of a bi-modal encoder (see 3.2.1) and a fact-aware classifier (see 3.2.2).

3.2.1 Bi-modal encoder

The bi-modal encoder (initialized by UNITER-base with N=12 layers (Chen et al., 2020)) integrates two modalities: image and text (e or f_i). Specifically, after generating the input embedding E_{Inp} consisting of image and text embedding (same with UNITER, see Appendix C for details), for each candidate c_i , we extract the expression-aware and fact-aware object features respectively (f_i^e and f_i^f) from the position of the visual output corresponding to c_i in the same encoder, based on the input of all candidates I , and e or f_i .

3.2.2 Fact-aware classifier

The fact-aware classifier is composed of multi-head attention layers and fully connected layers. For each candidate c_i , f_i^e and F_i^f (all K fact-aware object features for c_i) are fed into the integrator simultaneously (*Key* and *Value* are from F_i^f , and *Query* is from f_i^e), and fused into one three-source object features f_i^t (image, expression and top-K facts).

Finally, f_i^t is mapped into a match score s_i for c_i by a linear layer, and the optimization objective is to minimize the cross-entropy loss over all scores $\{s_j\}_{j=1}^n$ corresponding to all candidates I .

4 Results

We compare our CK-T to existing approaches on KB-Ref task without and with facts. Then we explore the importance of introducing visual information for fact search. Furthermore, we introduce facts into the traditional RefCOCO dataset, which was collected from MSCOCO (Lin et al., 2014) but differs in the types of expressions and object candidate settings. We extract image region features using an off-the-shelf detector, Faster R-CNN with ResNet-101 (Ren et al., 2015), based on bounding boxes (bbxes) (ground-truth labels or predicted results from the detector). See Appendix D and E for details about these datasets and experiment setting². Through parameter search on K and M (see Figure 3 and 4 in Appendix F), we keep M = 2 Fact-aware classifier blocks and top-3 facts for each candidate.

Model	Accuracy (%)	
	Val	Test
CMN (Hu et al., 2017)	41.28	40.03
SLR (Yu et al., 2017)	44.03	42.92
VC (Niu et al., 2019)	44.63	43.59
LGARNs (Wang et al., 2019)	45.11	44.27
MAttNet (Yu et al., 2018)	46.86	46.03
ECIFA-nf (Wang et al., 2020)	37.95	35.16
CK-T-nf (Ours)	58.02	57.53
ECIFA (Wang et al., 2020)	59.45	58.97
MAtt+E (Wang et al., 2020)	64.08	63.57
CK-T-Word2Vec	60.40	61.39
CK-T-Uw/oImage	64.44	64.78
CK-T (Ours)	65.62	66.71
Human	–	90.31

Table 1: Accuracy on KB-Ref dataset without and with facts (top and bottom part, respectively) using ground-truth bounding boxes and object categories.

Ground-truth bounding boxes and categories

By following Wang et al. (2020), we report our results on KB-Ref without and with facts. As can be seen in Table 1 (top), CK-T-nf, a version of CK-T without facts³, achieves an accuracy of 57.53% on the test set, outperforming existing approaches that do not utilize facts by approximately 11% – 22%. At the bottom part of the table we can see that our fact-enhanced CK-T model achieves the highest accuracy (66.71%) on the test set, which is 7.74% higher than that of ECIFA (a baseline model proposed by Wang et al. (2020)), and 3.14% higher than MAtt+E⁴. It is worth noting that both ECIFA and MAtt+E incorporate the top-50 facts for each candidate, which is considerably higher compared to top-3 facts in our CK-T. We surmise this is due to the fact that our fact search approach utilizes multi-modal fact and expression embeddings.

Predicted bounding boxes and categories To facilitate a fair comparison with ECIFA-d (Wang et al., 2020), we also use the maximum 10 detected bboxes for each image (CK-T-m10). As can be seen in Table 2, CK-T-m10 achieves an accuracy which is $\approx 5\%$ higher than that of ECIFA-d on the test set. CK-T-m100, a variant using at most 100 detected bboxes achieves a substantial improvement

²Including the efficiency discussion about our model

³all word tokens in fact sentences are replaced with only one [MASK] token.

⁴Wang et al. (2020) introduces their facts fusion module –Episodic Memory Module (E)–into MAttNet model (MAtt) (Yu et al., 2018) widely used for conventional REC.

Model	Accuracy (%)	
	Val	Test
ECIFA-d (Wang et al., 2020)	24.11	23.82
CK-T-m10 (Ours)	28.33	28.71
CK-T-m100 (Ours)	35.66	35.96

Table 2: Accuracy on KB-Ref using predicted bbxes and object categories.

with $\approx 7\%$, compared with CK-T-m10. This difference is primarily due to the increase in the number of correctly detected bbxes and predicted categories. Specifically, we find that with the top-100 bbxes, the number of samples containing the target bbxes rises from 18,901 to 31,653, while among these target bbxes, the number of correctly predicted categories grows from 11,324 to 15,928, out of a total of 43,284 samples in the KB-Ref dataset. This can also explain the dramatic decline on the accuracy between CK-T and CK-T-m10.

Incorporating image features into fact search

We experiment with various approaches to fact search and evaluate their effectiveness on KB-Ref (Table 1). We first utilize top-k facts searched in (Wang et al., 2020), where they use a pre-trained Word2Vec (Skip-Gram) (Mikolov et al., 2013) for searching facts (CK-T-Word2Vec). Then, we also selected facts from similarity predictors based on only text as input (CK-T-Uw/oImage)⁵, instead of image-text pairs in CK-T. As shown in Table 1, both CK-T-Uw/oImage and CK-T achieve better accuracy on the test set compared to CK-T-Word2Vec. Compared to CK-T-Uw/oImage, CK-T achieves around 2% higher accuracy. This is primarily due to the additional visual information used during fact search (see Appendix I for the examples of the selected facts by these fact search methods).

Introducing facts in traditional REC tasks We incorporate facts from the KB used in KB-Ref into the tasks of RefCOCOs using CK-T. Table 3 shows the results comparison based on the ground-truth bbxes and categories (discussion about the detected results can be seen in Appendix G). Compared with U_{REC} ⁶, the model introducing facts achieves better or equal accuracy on all RefCOCOs tasks, where RefCOCOg is improved more than RefCOCO and

⁵Inspired by Frank et al. (2021), we replace all object candidate feature with the average of all image region features.

⁶Chen et al. (2020) achieve state-of-the-art results on RefCOCOs by finetuning UNITER. (We re-finetune the model for fair comparison and conduct McNemar Test)

Task	Accuracy (%)		
	U_{REC}	Intro Facts	
Ref-COCO	Val	90.98	91.43
	Test A	91.50	92.09
	Test B	90.89	90.95
Ref-COCO+	Val	83.23	83.45
	Test A	85.09	85.49
	Test B	79.08	79.08
Ref-COCOg	Val	86.23	87.21
	Test	85.79	87.59

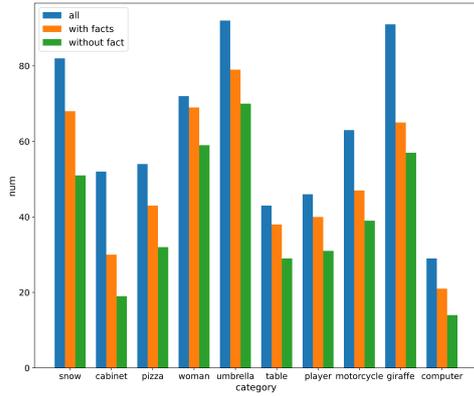
Table 3: Introducing facts into RefCOCO, RefCOCO+ and RefCOCOg. RefCOCO and RefCOCO+ have two different test sets, Test A and Test B, containing multiple persons and multiple objects in images respectively.

RefCOCO+. This is because RefCOCOg has less same-category object candidates in an image compared to RefCOCO and RefCOCO+ (an average of 1.63 and 3.9 per image, respectively) (Yu et al., 2016), and thus the retrieved facts integrated into different candidates are diversified (we first retrieve facts using the category), which contributes to distinguishing between candidates. This difference can also be proved in McNemar Test, where we find the change in the proportion of errors is statistically significant after introducing facts as compared to before on RefCOCOg ($p\text{-value} = 1.19e-08 < \alpha = 0.05$), while the similar proportions are found on RefCOCO and RefCOCO+ (see Appendix H for details about McNemar Test). The overall impact of commonsense knowledge in traditional REC is, however, not substantial. This is primarily due to much smaller number (78) of categories among the candidates in RefCOCOs, compared to 1805 in the KB-Ref (Wang et al., 2020). This limits the variety of selected facts, therefore impacting the extent to which commonsense knowledge is useful.

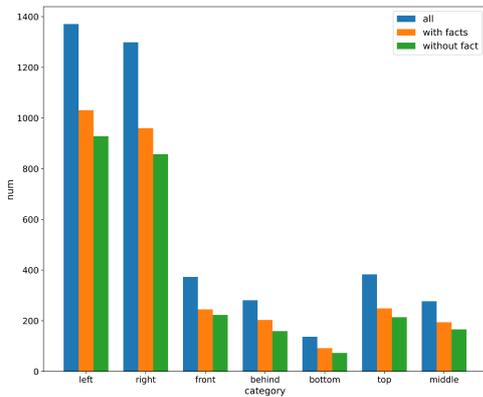
5 Analysis

To investigate in what cases commonsense knowledge helps, we conduct a fine-grained analysis of model performance on the test set of KB-Ref. Specifically, we compare the samples predicted by model with and without facts (CK-T and CK-T-nf) on three aspects: object categories, spatial relationships and the size of the bounding box.

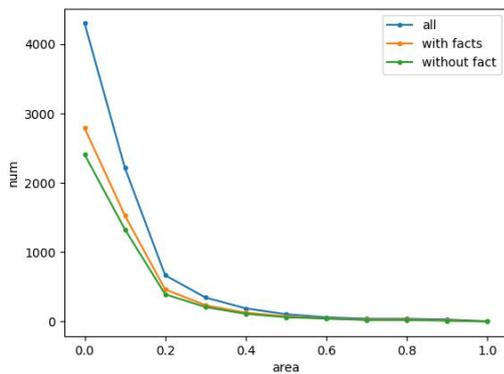
Object categories The test set contains 1502 categories and CK-T outperforms CK-T-nf on 1347 categories. Top 10 categories for which most improvement is observed are shown in Figure 2(a).



(a) Top 10 categories showing most improvement after introducing facts.



(b) The analysis of spatial relationships.



(c) The analysis of different bounding box sizes.

Figure 2: Fine-grained analysis. *all*: the total number of samples in the test set; *with fact*: the number of test samples that CK-T predicts correctly; *without fact*: the number of test samples that CK-T-nf predicts correctly.

In case of the 155 categories that do not show improvement, we find that the average number of samples per category is 6.68, making the results less reliable.

Spatial relationships We then investigate to what extent solving the REC task with and without facts relies on spatial reasoning, and whether there are particular spatial relationships between objects for which the use of facts is most crucial. Similar to the works of (Kazemzadeh et al., 2014; Johnson et al., 2017), we focus on the following spatial relationships: *left*, *right*, *front*, *behind*, *bottom*, *top*, *middle*. As shown in Figure 2(b), the model with facts (CK-T) outperforms that without facts (CK-T-nf) on all spatial relationships.

The size of the bounding box We then investigate the role of facts when identifying objects of different sizes, using the size of their bounding box as a proxy. We use the normalized area of the bounding box as the metric of bboxes size. As shown in Figure 2(c), the facts improve model performance on all bounding box sizes.

6 Conclusion

In this paper, we proposed CK-Transformer, which effectively integrates commonsense knowledge and the expression into the representations of the corresponding visual objects for multi-modal reasoning on KB-Ref. Our CK-Transformer achieves a new state-of-the-art performance on KB-Ref using only top-3 most relevant facts. We also demonstrated that visual information is beneficial for fact search. Finally, we show that commonsense knowledge improves conventional REC tasks across three different datasets.

7 Limitations

The computational requirements of our model are affected by the number of facts. Specifically, we train our CK-Transformer for 10000 steps with a batch size of 64 on one Titan RTX GPU, which takes 2.5, 3, 3.5, 7 days with the number of facts: 3, 5, 10, 20 respectively. The CK-Transformer processes 3.8, 2.8, 2.1, 0.7 samples on average per second at training time and 8.3, 7.3, 6.6, 1.1 samples per second at test time, with these amounts of facts. The computational requirements of our models are thus substantial, and future work should consider improving computational efficiency and thus reducing environmental impact.

References

- Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Stella Frank, Emanuele Bugliarello, and Desmond Elliott. 2021. Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers. *arXiv preprint arXiv:2109.04448*.
- Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. 2017. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1115–1124.
- Ronghang Hu, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. 2016. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564.
- Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.
- Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*.
- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Yulei Niu, Hanwang Zhang, Zhiwu Lu, and Shih-Fu Chang. 2019. Variational context: Exploiting visual and textual context for grounding referring expressions. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):347–359.
- Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24:1143–1151.
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypertexted, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.
- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vi-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.
- Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.
- Niket Tandon, Gerard De Melo, and Gerhard Weikum. 2017. Webchild 2.0: Fine-grained commonsense knowledge distillation. In *Proceedings of ACL 2017, System Demonstrations*, pages 115–120.

- Peng Wang, Dongyang Liu, Hui Li, and Qi Wu. 2020. Give me something to eat: Referring expression comprehension with commonsense knowledge. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 28–36.
- Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. 2019. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1960–1968.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315.
- Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer.
- Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7282–7290.
- Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. 2018. Grounding referring expressions in images by variational context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4158–4166.

A Referring expression comprehension

Early approaches to REC use joint embedding of image and language by combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), and predict the target object that has the maximum probability given an input expression and an image (Mao et al., 2016; Hu et al., 2016; Zhang et al., 2018). In order to model different types of information encoded in input expression (subject appearance, location, and relationship to other objects), subsequent work used modular (attention) networks, to “match” the input to corresponding regions in the image, predicting as the target the region with the highest matched score (Hu et al., 2017; Yu et al., 2018).

B UNITER

UNITER is trained using four pre-training tasks, Masked Language Modeling (MLM), Masked Region Modeling (MRM), Image–Text Matching (ITM), and Word–Region Alignment (WRA), on four large-scale image–text datasets, COCO (Lin et al., 2014), Visual Genome (Krishna et al., 2016), Conceptual Captions (Sharma et al., 2018), and SBU Captions (Ordonez et al., 2011). This enables UNITER to capture fine-grained alignments between images and language. The architecture of UNITER is similar to BERT (Devlin et al., 2018) apart from the input and the output. Specifically, the input consists of an image (a set of visual region candidates), a sentence and [CLS] token, and they respectively lead to different outputs, i.e. vision output, language output and cross-modality output on the top of UNITER.

C Input embedding

Same with UNITER, we extract the input embeddings E_{Inp} consisting of an image and a text embedding corresponding to the object candidate I and text (an expression e or a fact f_i) respectively.

Image embedding The image embedding E_I is computed by summing three types of embeddings: visual feature embedding, visual geometry embedding and modality segment embedding. We first extract the visual features $V = \{v_1, v_2, \dots, v_n\}$ for all candidates using Faster R-CNN (pooled RoI features), and build a geometry feature $G = \{g_1, g_2, \dots, g_n\}$ for all candidates, where g_i is a 7-dimensional vector consisting of the geometry information of the bounding box corresponding to

candidate c_i , namely normalized top, left, bottom, right coordinates, width, height, and area, denoted by $g_i = [x1, y1, x2, y2, w, h, w * h]$. Visual feature embeddings and visual geometry embeddings are generated by mapping the visual features and the geometry features into the same vector space through a fully connection layer fc :

$$E_I = LN(fc(V) + fc(G) + M_I) \quad (1)$$

where LN is the layer normalization layer and M_I is the modality segment embedding for the image input (like segment embedding for two sentence in BERT model).

Text embedding Similarly, the text embedding E_T is computed based on three different types of embeddings: token embedding, position embedding and modality embedding (Normally there is a fourth embedding, sentence segment embedding similarly to BERT, but, in our task, both expressions and facts consist of one sentence only and so only the first sentence segment embedding is used). Similar to BERT (Devlin et al., 2018), the text $W = \{w_1, w_2, \dots, w_u\}$ is first tokenized by WordPieces (Wu et al., 2016), which are then built into token embeddings $T = \{t_1, t_2, \dots, t_v\}$ and position embeddings $P = \{p_1, p_2, \dots, p_v\}$ according to their position in the text sequence.

$$E_T = LN(T + P + M_T) \quad (2)$$

where M_T is the modality segment embedding for the text input.

Input embedding The final input embedding E_{Inp} is computed by concatenating image embedding E_I and text embedding E_T :

$$E_{Inp} = [E_I, E_T] \quad (3)$$

D Datasets

We use the KB-Ref dataset (Wang et al., 2020) aiming at evaluating the task of referring expression comprehension based on commonsense knowledge. KB-Ref consists of 43,284 expressions for 1,805 object categories on 16,917 images, as well as a knowledge base of key–value (category–fact) pairs collected from three common knowledge resources: Wikipedia, ConceptNet (Speer et al., 2017) and WebChild (Tandon et al., 2017)). KB-Ref is split into a training set (31,284 expressions with 9,925 images), a validation set (4,000 expressions with

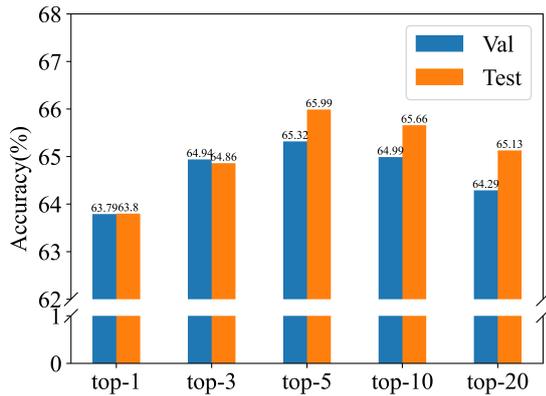


Figure 3: Accuracy across a varying number of facts (top-K).

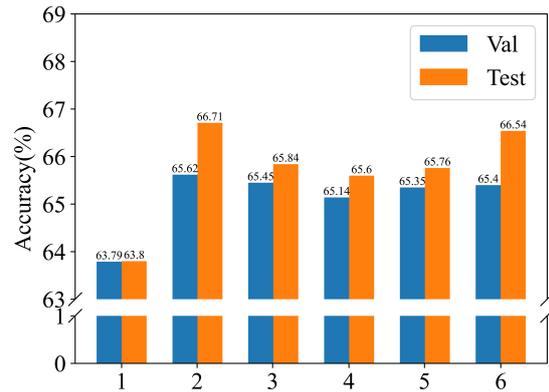


Figure 4: Accuracy across a varying number of fact-aware classifier block (M).

2,290 images) and a test set (8,000 expressions with 4,702 images).

We furthermore introduce commonsense knowledge into traditional tasks/datasets of referring expression comprehension, namely RefCOCO, RefCOCog and RefCOCO+⁷. The datasets are devised from the MSCOCO image dataset (Lin et al., 2014) but differ in the types of expressions and object candidate settings. Specifically, RefCOCO+ does not allow the use of absolute location words in the expressions, and most expressions focus on the appearance of the objects. The expressions in RefCOCog are longer and contain more descriptive words. RefCOCO and RefCOCO+ contain more objects of the same category within an image.

E Experimental settings

We extract image region features using Faster R-CNN with ResNet-101 (Ren et al., 2015) which was pre-trained on Visual Genome (Krishna et al., 2016) using object and attribute annotations (Anderson et al., 2018). For bounding box detection, we keep the bounding boxes with at least 0.2 confidence score indicating the extent of detection. In the CK-T, the hidden layer dimension is 768 and the number of multi-head attention heads is 12. The models are trained using Adamw (Loshchilov and Hutter, 2017) with a learning rate of $6e^{-5}$ and a batch size of 64 on Titan RTX GPUs. Our CK-Transformer has 120M parameters in total where fact-aware classifier has 34M and bi-modal encoder has 86M. As for UNITER model, we use same setting with *UNITER-base*, except for using Nvidia

⁷following Apache License 2.0

Apex⁸ for speeding up training. The efficiency of our model is effected by the number of facts. Specifically, we train our CK-Transformer 10000 steps and a batch per step, which takes 2.5, 3, 3.5, 7 days with the number of facts: 3, 5, 10, 20 respectively. The CK-Transformer trains 3.8, 2.8, 2.1, 0.7 sample in average per second and tests 8.3, 7.3, 6.6, 1.1 sample per second.

F Impact of CK-T structure

We explore the impact in performance on KB-Ref as we vary the number of top-K facts (K) and fact-aware classifier block (M) on the development set. We first keep the number of the fact-aware classifier block constant and set it to 1 to experiment with different values for K from 1 to 20. As shown in Figure 3, as K increases, performance starts to improve with a peak at K=5 before starting to gradually decrease performance.

In the second experiment, we keep K constant and set it to 3 and explore the effect of varying values for M. We observe that the highest accuracy is achieved with with top-3 facts and 2 integrator layers as shown in Figure 4.

G Introducing facts in traditional REC tasks based on detection

The results of introducing facts in traditional REC tasks based on detected bbxes and categories are shown in Table 4. Compared to result based on ground-truth bbxes and categories (Table 3), the improvement on models based on detection is less or even worse than the models without facts.

⁸<https://github.com/NVIDIA/apex>

Task		Accuracy (%)	
		U_{REC}	Intro Facts
Ref-COCO	Val ^d	81.15	81.06
	Test A ^d	86.85	86.87
	Test B ^d	74.48	73.97
Ref-COCO+	Val ^d	74.74	74.68
	Test A ^d	81.05	80.70
	Test B ^d	65.88	66.07
Ref-COCOG	Val ^d	74.49	74.69
	Test ^d	75.24	74.86

Table 4: Introducing facts into RefCOCO, RefCOCO+ and RefCOCOG based on detection (d).

Task		McNemar Test
		(p -value)
RefCOCO	Test A	0.049
	Test B	0.905
RefCOCO+	Test A	0.297
	Test B	0.966
RefCOCOG	Test	$1.19e-08$

Table 5: The McNemar Test between models before and after introducing facts on the tasks of RefCOCOs.

H McNemar Test

We also report the statistical significance for accuracy (shown in Table 3) on the tasks of RefCOCOs. Specifically, we conduct the McNemar Test between models before and after introducing facts, on the test set of RefCOCO, RefCOCO+ and RefCOCOG, respectively. As shown in Table 5, as for Test set on RefCOCOG and Test A on RefCOCO p -value = $1.19e-08$ and p -value = 0.049 (< 0.05) respectively, which means the proportion of errors is statistically significantly different after introducing facts as compared to before. However, the change in the proportion of errors after introducing facts on other tasks (Test B on RefCOCO, Test A and Test B on RefCOCO+) is not statistically significant. This is reasonable, as the error from detection will affect the fact search (we first retrieve facts using the category) and thus more error information is introduced into CK-Transformer, which make the performance worse.

I Example searched fact using different methods

As shown in Figure 5, there are several facts which are selected from three different fact search methods: CK-Transformer, CK-T-Uw/oImage and CK-

T-Word2Vec. As we can see in the Table, normally the facts of CK-Transformer model (green) is the best relevant with the referring expression (blue) and the facts in CK-T-Word2Vec model is the worst relevant with the expression.

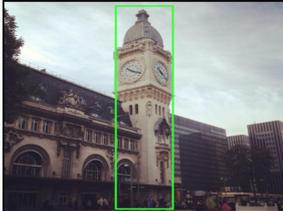
			
Exp: The tool under the mouse can improve the usability of the mouse	Exp: The tall building has instruments on the upper exterior walls that used as a reference to find out the time.	Exp: The most important part of the tree with branches and leaves	Exp: An access point as an underground public utility.
Fact: A mousepad enhances the usability of the mouse compared to using a mouse directly on a table.	Fact: Clock towers are a specific type of building which houses a turret clock and has one or more clock faces on the upper exterior walls.	Fact: The trunk is the most important part of the tree for timber production.	Fact: Manholes are often used as an access point for an underground public utility, allowing inspection, maintenance, and system upgrades.
Fact (Uw/oImage): A mousepad is a surface for placing and moving a computer mouse.	Fact (Uw/oImage): The tower has four clock faces, two of which are in diameter, at about high.	Fact (Uw/oImage): An automobile has a trunk.	Fact (Uw/oImage): A manhole is an opening to a confined space such as a shaft, utility vault, or large vessel.
Fact (Word2Vec): Mousepad on the mouse.	Fact (Word2Vec): Before the middle of the twentieth century, most people did not have watches, and prior to the 18th century even home clocks were rare.	Fact (Word2Vec): A trunk is an example of a box.	Fact (Word2Vec): These covers are traditionally made of metal, but may be constructed from precast concrete, glass reinforced plastic or other composite materials.

Figure 5: Example fact search process (using the top-1 fact) for different search methods: CK-T (green), CK-T-Uw/oImage (orange) and CK-T-Word2Vec (yellow).

Curricular Next Conversation Prediction Pretraining for Transcript Segmentation

Anvesh Rao Vijjini^{1*} Hanieh Deilamsalehy²
Franck Deroncourt² Snigdha Chaturvedi¹

¹UNC Chapel Hill

{anvesh, snigdha}@cs.unc.edu

²Adobe Research

{deilamsa, franck.deroncourt}@adobe.com

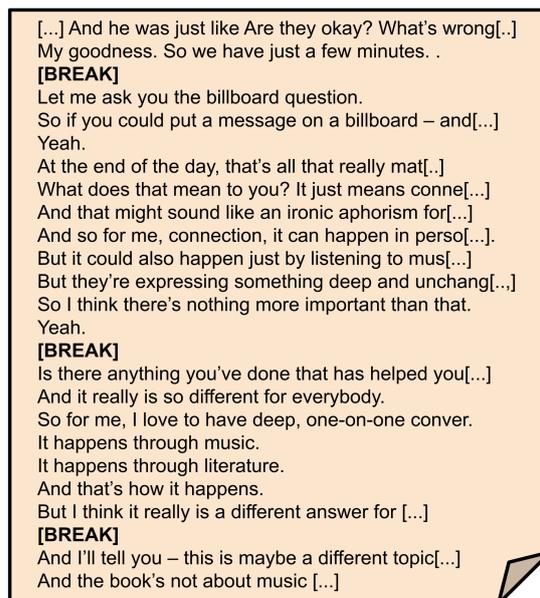
Abstract

Transcript segmentation is the task of dividing a single continuous transcript into multiple segments. While document segmentation is a popular task, transcript segmentation has significant challenges due to the relatively noisy and sporadic nature of data. We propose pre-training strategies to address these challenges. The strategies are based on “Next Conversation Prediction” (NCP) with the underlying idea of pretraining a model to identify consecutive conversations. We further introduce “Advanced NCP” to make the pretraining task more relevant to the downstream task of segmentation break prediction while being significantly easier. Finally we introduce a curriculum to Advanced NCP (Curricular NCP) based on the similarity between pretraining and downstream task samples. Curricular NCP applied to a state-of-the-art model for text segmentation outperforms prior results. We also show that our pre-training strategies make the model robust to speech recognition errors commonly found in automatically generated transcripts.

1 Introduction

Text segmentation is the task of identifying segment breaks to organize a continuous text into semantically independent segments. Prior research in text segmentation has largely focused on segmenting documents such as Wikipedia articles (document segmentation) (Lukasik et al., 2020; Zhang et al., 2019; Badjatiya et al., 2018; Koshorek et al., 2018) or dialogues such as chat or text messages (dialogue segmentation) (Hsueh et al., 2006; Arguello and Rosé, 2006; Xia et al., 2022; Xing and Carenini, 2021). In this paper we address text segmentation of transcripts (transcript segmentation). Figure 1 shows examples of segments in transcript data. Transcript segmentation can help summarize long videos, podcasts or meetings by segmenting and summarizing the transcript such as “Video

* Work done during internship at Adobe Research.



[...] And he was just like Are they okay? What's wrong[...]
My goodness. So we have just a few minutes. .
[BREAK]
Let me ask you the billboard question.
So if you could put a message on a billboard – and[...]
Yeah.
At the end of the day, that's all that really mat[...]
What does that mean to you? It just means conne[...]
And that might sound like an ironic aphorism for[...]
And so for me, connection, it can happen in perso[...]
But it could also happen just by listening to mus[...]
But they're expressing something deep and unchang[...]
So I think there's nothing more important than that.
Yeah.
[BREAK]
Is there anything you've done that has helped you[...]
And it really is so different for everybody.
So for me, I love to have deep, one-on-one conver.
It happens through music.
It happens through literature.
And that's how it happens.
But I think it really is a different answer for [...]
[BREAK]
And I'll tell you – this is maybe a different topic[...]
And the book's not about music [...]

Figure 1: Transcript Segmentation example from the SliceCast-Podcast (Midei and Mandic, 2019) dataset. Here each line indicates start of a new sentence and segment breaks are noted with “[BREAK]”.

chapters” in YouTube videos or “Outline Generation” (Zhang et al., 2019) from documents.

However, only few works have addressed segmentation of transcripts (Midei and Mandic, 2019; Jing et al., 2021; Gruenstein et al., 2008). As shown in Figure 1, transcripts consist of a mix of short sentences, utterances, interjections and long form document style answers. Unlike Wikipedia articles or chat, the sporadic and non uniform flow of text in transcripts makes annotation of segment breaks hard even for humans (Gruenstein et al., 2008). Furthermore, transcripts often involve Automatic Speech Recognition errors such as insertions, deletions, replacement and lack of proper punctuation which add to the challenges. As a result of these challenges, most labeled transcript segmentation datasets are small in size making it difficult for models to be trained on them.

To address this issue, we propose pretraining strategies that can be useful in resource constrained

settings where huge labeled datasets are not available. Our first strategy is Next Conversation Prediction (NCP). In this strategy, pairs of conversations are classified into 1 or 0 based on whether they contiguously in the transcript or not. We hypothesize that the effectiveness of this pretraining task relies on its similarity and relevance to the segmentation task. Our second strategy, Advanced Next Conversation Prediction (Advanced NCP), introduces conditions on the NCP pretraining data to increase the relevance of the pretraining strategy to the segmentation task. Our third strategy is Curricular NCP where we pretrain the model in two distinct phases based on which pretraining samples are closest to the task of transcript segmentation.

Our experiments show that the application of the proposed pretraining strategies on multiple segmentation architectures outperforms their corresponding non pretrained versions. Also, NCP does not rely on segmentation labels. We show that it is a strong unsupervised approach that outperforms state-of-the-art unsupervised model for transcript segmentation. Finally, we observe that the pretraining strategies makes the model more robust to noise and better at predicting highly segmented regions of a transcript.

Our contributions are:

- Propose a pretraining strategy based on Next Conversation Prediction for transcript segmentation. We show that it also acts as a strong unsupervised approach for this task.
- Propose Advanced NCP and Curricular NCP pretraining strategies based on similarity and relevance of pretraining samples to segmentation task.
- Provide a new state-of-the-art in transcript segmentation.
- Evaluate robustness of proposed pretraining strategies to noisy training data.
- We perform additional analysis to investigate the errors made by the pretrained models.

2 Related Work

Text segmentation has been addressed in both unsupervised (Solbiati et al., 2021; Glavaš et al., 2016) and supervised manner (Midei and Mandic, 2019; Lukasik et al., 2020; Koshorek et al., 2018; Badjatiya et al., 2018) with early works focusing on unsupervised techniques (Hearst, 1997; Choi, 2000; Utiyama and Isahara, 2001; Eisenstein, 2009). However, since the definition of a segment,

could be highly domain and data dependant, supervised learning is desirable.

Koshorek and Cohen (2017) and Koshorek et al. (2018) use LSTMs to identify if a sentence ends a segment or not. Similarly, Li et al. (2018) use GRUs and pointer-generator networks for this task. These works in segmentation propose a hierarchical approach, where the sentences are encoded into a fixed size representation followed by mapping the representations to a sequence of binary labels whether the current segment is ending at this sentence or not. Badjatiya et al. (2018) use attention based CNN-LSTMs and phrase the task differently by providing inputs of a median sentence and its right and left context to identify segment breaks. Lukasik et al. (2020) simplify the new paradigm for this task by using the left and right contexts with respect to an end of a sentence as input. They were also the first to use large pretrained language models for this task. They establish a new SOTA in text segmentation. Hence, we use their model as the base model in our pretraining experiments.

Very few works have focused on transcript segmentation. Midei and Mandic (2019) provide a podcast dataset for research in this domain and propose an LSTM and Universal Sentence Encoder (Cer et al., 2018) based sequence labeling model. Jing et al. (2021) identify introductions in podcast transcripts using BERT (Devlin et al., 2019). Solbiati et al. (2021) propose an unsupervised technique for meeting transcript segmentation. They use large language model representations including Sentence BERT (Reimers and Gurevych, 2019) and BERT to compute cosine similarity between subsequent conversations and estimate segment breaks. We present a new pretraining strategy aimed towards addressing transcript segmentation but not specific to any one model architecture. Our work is also related to Curriculum Learning (Bengio et al., 2009). It has gained popularity among NLP tasks such as Sentiment Analysis (Cirik et al., 2016) Question Answering (Sachan and Xing, 2016), NLG (Liu et al., 2018) and the GLUE benchmark (Xu et al., 2020). More recently, some works have used curriculum learning in the pretraining process of large language models. Wang et al. (2020) propose curriculum learning for pretraining the encoder of their speech translation system on multiple speech based tasks of varying difficulty. Nagatsuka et al. (2021) gradually introduce longer samples to BERT's pretraining to observe performance improvements in

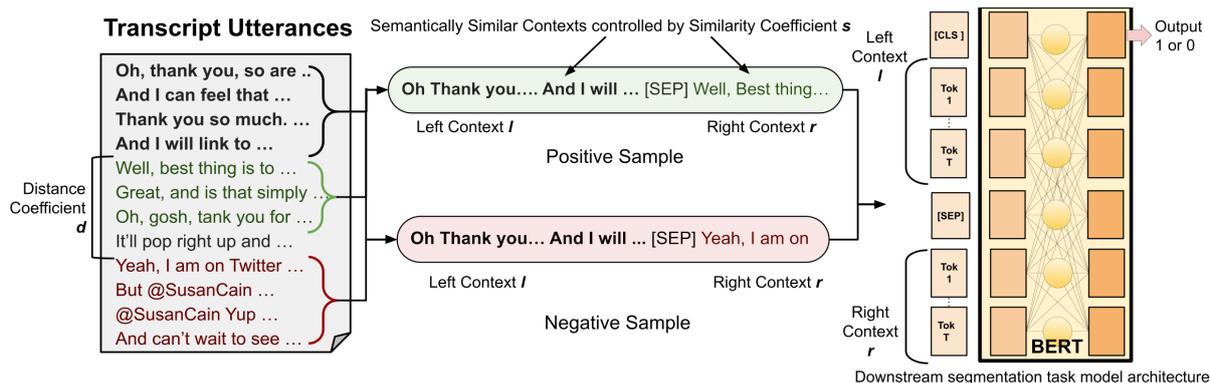


Figure 2: Advanced Next Conversation Prediction (NCP) pretraining strategy illustrated with roles of the coefficients in controlling pretraining task difficulty and similarity to the downstream task.

resource poor settings. In our curriculum learning setting, we order pretraining samples based on their similarity to the downstream segmentation task.

3 Text segmentation as a binary classification task

There are two major ways supervised text segmentation has been addressed in the past. First, by phrasing it as a sequence labeling problem, where each sentence has a label indicating if it ends the segment or not (Midei and Mandic, 2019; Lukasik et al., 2020; Koshorek et al., 2018). As a result an entire transcript forms a single training instance consisting of a sequence of binary labels.

Second, by phrasing text segmentation as a binary classification task where we provide left context and right context around a sentence end and predict if the two contexts belong to the same segment or not. In particular, the input in this task is two text segments - left context (l) and right context (r) of the end of a sentence, each T tokens in length. The output in this task is 0 - if l and r belong to two different segments (segment break) and 1 if l and r belong to the same segment (not a segment break). In this setting, the number of instances is proportional to the number of segments. Lukasik et al. (2020) note in their experiments that the second setting outperforms the first on transcript segmentation. We follow the second setting and refer to it as the segmentation task in the rest of this paper.

4 Curricular Next Conversation Prediction

In this section, we explain the proposed pretraining strategy. First, we explain our basic pretraining strategy - Next Conversation Prediction (NCP). Then, we present improvements on this strategy

to make the pretraining task easier and more relevant to the transcript segmentation task (Advanced NCP). Finally, we introduce curriculum learning to our pretraining strategy (Curricular NCP) that presents the pretraining instances in an order that is more helpful for the segmentation task.

4.1 Next Conversation Prediction

Next Conversation Prediction is the backbone of our proposed pretraining strategies. Large language models such as BERT have gained recent success on the segmentation task (Lukasik et al., 2020). One of the pretraining tasks in BERT is NSP (Next Sentence Prediction). In NSP, a sentence pair is provided as input and the model predicts if the sentences occurred consecutively in their original corpora. We hypothesize that BERT’s success in text segmentation might be attributed to the NSP pretraining’s similarity to the segmentation task.

Motivated by NSP, we propose a pretraining strategy based on Next Conversation Prediction (NCP) for the task of transcript segmentation. In NCP, we address a binary classification task. The input is a pair of transcript contexts and the output is a label indicating whether the contexts are adjacent or not. Specifically, the input consists of the left (l) and right (r) contexts of a sentence end (T tokens each). The output is 1 if the two contexts are adjacent and 0 otherwise. Note that NCP does not use any information about segment break labels and so can potentially be used on transcripts dataset without segment break annotations.

NCP has two major advantages over NSP. First, NCP has longer contexts making the model learn information from a wide range of sentences varying in content and style. Second, NCP as pretraining task makes the pretraining step similar to the seg-

mentation task in terms of the structures of input and output.

4.2 Advanced Next Conversation Prediction

The predictive difficulty of the negative samples (label 0) in NCP depends on the distance between the left and right contexts, l and r , in the transcript. A greater distance between the contexts makes the NCP task easier but more different from, and hence potentially less useful for, the segmentation task. Similarly, the difficulty of the positive samples (label 1) depends on how semantically similar the contexts are to each other. Positive samples with highly semantically similar contexts will be easy to identify. To control the difficulty of the NCP samples, we introduce the following conditions on the positive and negative samples of the pretraining data respectively.

$$Sim(l, r) \geq s \text{ for label 1} \quad (1)$$

$$Dist(l, r) = d \text{ for label 0} \quad (2)$$

where $Sim()$ is a semantic similarity function¹ quantifying similarity between l and r , and s is the similarity coefficient. $Dist()$ is the distance between l and r in terms of number of sentences between them and d is the distance coefficient. Figure 2 illustrates the Advanced NCP.

In vanilla NCP, by default, the distance between non consecutive contexts is greater than 1 and there is no semantic similarity filter ($s = 0$). Increasing s will make the pretraining task easier as the positive samples (label 1) have the additional constraint of being semantically similar. However, increasing s too much can filter out too much pretraining data. Similarly, decreasing d will make the task harder but more relevant to the segmentation task.

4.3 Curricular Next Conversation Prediction

Curriculum learning (Bengio et al., 2009) proposes that models observing training samples in an increasing order of difficulty have an advantage over models observing samples in an otherwise random order. Motivated by this, we introduce a curriculum to Advanced NCP pretraining. The pretraining samples from Advanced NCP are divided into two distinct sets - “similar” to downstream task (or “harder” since, in general, segmentation is a harder task than NCP) and “dissimilar” to the segmentation task (or “easier”). In order to estimate the

¹We use Sentence BERT (Reimers and Gurevych, 2019) to compute representations of l and r , followed by cosine similarity for $Sim()$

similarity or dissimilarity of the NCP pretraining samples to the segmentation task, we use a classification model trained for the segmentation task and use it to predict labels for the pretraining instances. We refer to this as the “Auxiliary model” and classify a pretraining sample as “similar” if the Auxiliary model correctly predicts its label and vice versa. In the spirit of curriculum learning, we divide the pretraining into two steps. First training on the “dissimilar” or “easy” (from the perspective of segmentation) samples followed by the “similar” or “hard” samples. This order makes sure that the model has smoother transition between the two tasks that are semantically close but different. Figure 6 illustrates the Curricular NCP process. Table 4 shows examples of Dissimilar NCP and Similar NCP from the SliceCast-Podcast dataset. All the examples are labeled 0 in their respective tasks.

While the Auxiliary Model can be any classification model trained on the segmentation task dataset, we use a model that is additionally pretrained on Advanced NCP data. The Auxiliary model is tested on Advanced NCP samples. While these samples were used during the pretraining of the Auxiliary model, after finetuning on the segmentation task model might not predict the same labels it observed during the pretraining. In our experiments with the SliceCast-Podcast dataset (described in Section 5.1) we indeed observe that 64.4% samples are miss-classified (hence, “dissimilar”) and 35.6% are correctly classified (hence, “similar”).

5 Experimental details

In this section, we describe the dataset, the base model upon which our pretraining is tested, the implementational details, metrics and baselines.

5.1 Dataset

We use the SliceCast-Podcast (Midei and Mandic, 2019) dataset for our experiments. This dataset has 46 podcast transcripts and a total of 643 segments. On average, each transcript has 12.4 segments, though there could be high variation in number of segments as the standard deviation is 4.1. We consider 416 segments for training and 181 segments for testing purposes. While creating training data for the pretraining and the segmentation task, positive and negative samples are sampled equally. For this, in the segmentation task we randomly down sample samples labeled 1. Figure 1 shows examples of segment breaks from this dataset.

5.2 Base Binary Classification Model

We use BERT as the base classification model in the Auxiliary model. Across all classification tasks - Advanced NCP, Curricular NCP and the Segmentation task, the input is provided by concatenating l and r contexts with the token “[SEP]”. Hence the input is “ l [SEP] r ”. T , the maximum input size of l and r individually is 150^2 . The output is taken from the first position (“[CLS]”) and a binary cross entropy layer is attached to enable binary classification.

Lukasik et al. (2020) propose Cross Segment BERT, for the transcript segmentation task. We use this as the base segmentation model for transcript segmentation after finetuning it on the data described in Section 5.1.

5.3 Coefficient details for Advanced NCP

The two coefficients explained in Section 4.2 control difficulty of the Advanced NCP task and hence its relevance to the pretraining task. We experimented with various values of s and d in the Advanced NCP task. Following which, we measure performance of the these pretrained models (with different values of s and d) on a balanced Advanced NCP test data. The accuracy results are reported in Fig. 3a. A darker shade of green indicates better performance. As we can see, in general, Advanced NCP performance (accuracy) increases as the coefficients increase, making the positive and negative samples easier to identify. However, large values of s results in filtering out too many positive samples and hence the size of the training dataset leading to a decreasing in performance.

The aforementioned Advanced NCP pretrained models are then fine-tuned on the segmentation task. All the models are finetuned on the dataset described in Section 5.1. For model comparison we use the F1 of the the segment break class (0) on a held out set of containing 61 segments. Results of the finetuned models are illustrated in Figure 3b. Comparing Figures 3a and 3b we can observe that while a low performance on the Advanced NCP task also corresponds to a low performance on the segmentation task, the converse is not true. At high coefficient values, especially the distance coefficient d , the pretraining task is too distinct from the segmentation task leading to low efficacy of the pretraining strategy.

²Original Cross Segment BERT (Lukasik et al., 2020) used 125 in most of their experiments. We follow a similar setting

Models	F1 (↑)	Pk (↓)	WDiff (↓)
S-BERT	4.1	50.5	65.1
CSB	17.5	42.5	37.3
Adv. NCP + CSB	22.1**	37.2**	36.2**
Curr. NCP + CSB	22.6*	35.6**	37.5**

Table 1: Evaluation results for S-BERT (Solbiati et al., 2021), CSB (Lukasik et al., 2020) and CSB pretrained with the proposed strategies. WDiff refers to WindowDiff. Pretrained models significantly outperform CSB in all metrics. Introduction of curriculum to Advanced NCP also shows improvement. * and ** denote the difference is significant with $p < 0.03$ and $p < 0.06$ via t-test.

Models	F1-0 (↑)	Pk (↓)	WDiff (↓)
Hier.	19.9	39.2	37.1
Adv. NCP + Hier.	20.6	38.5	36.2
Curr. NCP + Hier.	20.3	37.4	36.6

Table 2: Performance of the Hierarchical (Hier.) model (Lukasik et al., 2020) before and after pretraining with Advanced NCP (Adv. NCP + Hier.) and Curricular NCP (Curr. NCP + Hier.). Application of pretraining on Hierarchical shows improvement.

Using these two figures, we find the ideal coefficients such that the Advanced NCP strategy is sufficiently easy but relevant to the downstream task concurrently. We choose $d = 200$ and $s = 0.7$.

5.4 Metrics

In line with previous works (Midei and Mandic, 2019; Lukasik et al., 2020; Solbiati et al., 2021), we use F1 score and Pk score (Beeferman et al., 1999) for evaluating text segmentation models. The scores are calculated by using the model to predict existence of segment break after each sentence end in the test set and then comparing ground truth segment break predictions and predicted segment breaks. F1 score of the label 0 is considered. This score is a strict measure as it rewards the model only if the predicted segment breaks and ground truth segment breaks exactly align. Pk score is less harsh. Pk score is calculated by using a sliding window such that predicted segment breaks near ground truth are penalised less than predictions that are far away³. One criticism of the Pk score is that it favours models that make fewer segment break predictions. To address this Pevzner and Hearst (2002) proposed WindowDiff to account for the number of segment break predictions as well. For WindowDiff and Pk, we consider size of sliding window to be half of the average segment length in number of sentences, as is the standard practice.

³we encourage the reader to look at assemblyai.com

$s \downarrow d \rightarrow$	15	66	100	200	500
0.4	0.58	0.63	0.61	0.69	0.79
0.6	0.61	0.69	0.59	0.77	0.78
0.7	0.59	0.75	0.69	0.77	0.8
0.8	0.62	0.74	0.76	0.82	0.89
0.9	0.59	0.71	0.68	0.7	0.68

(a) Advanced NCP task performance in Accuracy. A higher number implies low task difficulty.

$s \downarrow d \rightarrow$	15	66	100	200	500
0.4	0	0.04	0.1	0.12	0.12
0.6	0.001	0.11	0.11	0.14	0.1
0.7	0.001	0	0.17	0.24	0.09
0.8	0.02	0.12	0.18	0.18	0.11
0.9	0.03	0.04	0.06	0	0.09

(b) Segmentation task performance in F1 score of the 0 label across different Advanced NCP pretraining.

Figure 3: Performance of Advanced NCP pretraining and Segmentation task with varying coefficients d and s . As we can observe higher pretraining task performance does not necessarily imply higher downstream task performance.

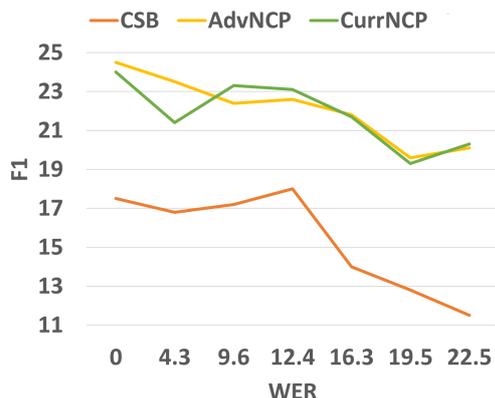


Figure 4: Performance with respect to change in transcription errors (WER). x -axis is represents different WER rates and y -axis represents the F1 scores. Proposed pretraining strategies makes the CSB model more robust to transcription errors.

For both Pk and WindowDiff, lower scores indicate better performance.

5.5 Baselines

S-BERT We compare the supervised techniques (pretrained and non pretrained) against this Sentence BERT based unsupervised transcript segmentation baseline (Solbiati et al., 2021) to show the motivation for this task to be addressed in a supervised setting.

Cross Segment BERT (CSB) This is a BERT model for the segmentation task without any proposed pretraining. The model is based on Lukasik et al. (2020), originally proposed for document segmentation where left and right contexts are concatenated with a separator and provided to the BERT model for binary classification. CSB formed a state-of-the-art in text segmentation. Hence, we apply our pretraining strategies on this model.

6 Results and Discussions

6.1 Pretraining on Cross Segment BERT

Table 1 presents the results of application of the proposed pretraining strategies - Advanced NCP and Curricular NCP on CSB (Advanced NCP + CSB and Curricular NCP + CSB respectively). We

also compare with S-BERT, the unsupervised baseline.

Challenges of an unsupervised setting The unsupervised baseline, S-BERT, vastly underperforms all other models (supervised) on all metrics (row 1 and other rows). This is because the definition of a segment could be data and domain specific. In such a case, deriving its interpretation from a supervised data becomes imperative. Hence, despite the difficulty of annotation, supervised approaches are favoured.

Improvement due to proposed pretraining By comparing pretrained models with CSB (row 2 and rows 3,4), we see that pretrained models outperform across all metrics indicating their effectiveness. We also outperform the transcript segmentation baseline proposed by Midei and Mandic (2019). However, we do not apply our pretraining strategy to it since it adopts a sequence labeling paradigm, and adapting proposed pretraining strategies for such models is left for future work.

By comparing Advanced NCP and Curricular NCP (row 3 and row 4), we see that proposing a curriculum to the pretraining leads to better F1 and Pk scores. We give two major reasons for this improvement - First, our ordering of pretraining samples in Curricular NCP is relevant to the segmentation task. Prior research in curriculum learning show such sample orderings are more effective than arbitrary sample orderings such as sentence length for sentiment analysis (Rao et al., 2020). Second, the transcript segmentation data is small in size and previous works note the efficacy of curriculum learning in resource poor settings (Cirik et al., 2016; Nagatsuka et al., 2021).

6.2 Pretraining on Sequence Labeling

To further observe efficacy of the proposed pretraining approaches, we apply them on a sequence labeling approach. We use a model based on Hierarchical BERT model from Lukasik et al. (2020) which is compatible with our pretraining task. In

this "Hierarchical" baseline- first, the left and the right context pairs are obtained by taking $T = 150$ tokens of left and right context around each sentence end. Next, CSB model is used to obtain representations for context pairs. Hence, each transcript is converted to sequence of context pair representations. Finally, this sequence of context pair representations is then input to an LSTM (50 units) in a one-to-one sequence labelling setting to output an equally long sequence of 1s and 0s. Similar to segmentation in the binary classification setting (explained in Section 3), 0 indicates a segment break and 1 indicates absence of a segment break. In this hierarchical baseline, we swap the CSB model with Advanced NCP + CSB model to obtain Advanced NCP + Hierarchical model and using a similar process we obtain Curricular NCP + Hierarchical.

The performances of all models are reported in Table 2. By comparing the pretrained models (Advanced NCP + Hierarchical and Curricular NCP + Hierarchical) to the model without pretraining (Hierarchical), we observe an improvement. The performance increment between vanilla and pretrained models, has diminished slightly in this sequence labelling setting as compared to CSB as based model setting. This is possibly because the Hierarchical model, involves more parameters (LSTM units) that have not been updated during our pretraining steps as opposed to the CSB model, where all parameters were involved in the pretraining. Regardless, pretraining leads to an improvement across all metrics. This is consistent with Table 1, showing that proposed pretraining methods have merits across the downstream model architecture (CSB or Hierarchical).

6.3 Utility in an unsupervised setting

To further understand the relationship between the pretraining and the segmentation task, we do cross domain testing. Here, we use an NCP pretrained model (prior to finetuning) to make predictions on the segmentation test data. Since NCP does not use any segmentation information, this method is unsupervised in segmentation prediction. We also make predictions on Curricular NCP pretraining test data using the segmentation model ("Pretraining Test Data").

Results of this experiment are tabulated in Table 3. For the pretraining test data, we use accuracy for performance comparison. WindowDiff is used for the segmentation test data. Comparing the perfor-

Models	Pretraining	Segmentation
	Test Data (\uparrow)	Test Data (\downarrow)
S-BERT	55.8	65.1
NCP	69.3	61.5
CSB	61.8	37.3
Curr. NCP + CSB	66.4	37.5

Table 3: Results of the cross domain testing experiment. We report accuracy for the pretraining task and WindowDiff for the segmentation task. NCP does not use any segment information and outperforms S-BERT in segmentation, thereby forming a strong unsupervised approach.

mances of S-BERT and NCP on the segmentation task, we observe that NCP outperforms S-BERT. This shows that the proposed pretraining approximates the segmentation task and gives the necessary domain knowledge to perform well even in an unsupervised setting. Next we compare the performances of the models trained for segmentation task (Curricular NCP + CSB and CSB) with the performance of NCP. We can see that Curr NCP +CSB and CSB are performing better than NCP on segmentation task but not on pretraining task. This shows that is a significant difference between the two tasks.

6.4 Robustness

Since, automatically generated transcripts tend to be noisy, in this section we measure the robustness of proposed pretraining strategies to noise in training data. In this experiment, we synthesize noise in the training samples using Easy Data Augmentation (EDA) (Wei and Zou, 2019). EDA introduces noise to transcript samples by four operations - synonym replacement, random insertion, random swap, and random deletion. EDA also provides a temperature variable to control how intensely these operations are applied. By increasing the temperature in some of these operations, we obtain six SliceCast-Podcast variants with increasing WER rates (4.33%, 9.60%, 12.42%, 16.33%, 19.51% and 22.53%) with respect to the original dataset. Only random insertion, swap, and deletion are used for introducing noise. Note that the test data is not changed across these variants. Figure 4 shows the performance (F1) of CSB, and CSB model pretrained with Advanced and Curricular NCP.

We observe that the results align with the results reported in Table 1 i.e. First, Pretrained models always outperform CSB. Second, Curricular NCP pretrained model generally outperforms Advanced

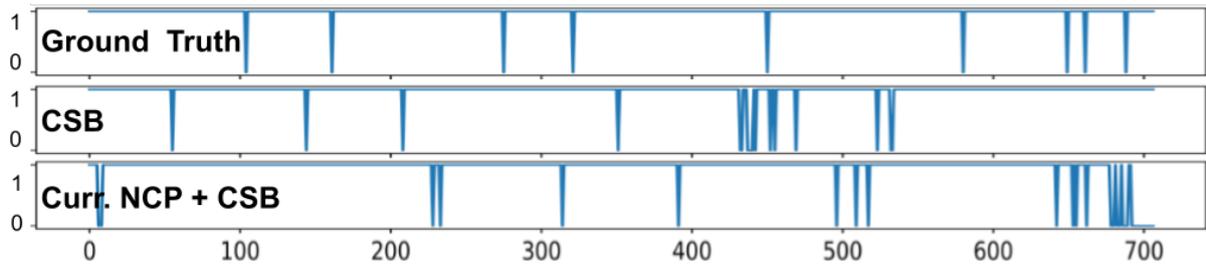


Figure 5: Segmentation break predictions across all sentences of a test transcript, illustrated in the ground truth annotation, annotation by CSB model and annotation by Curricular NCP pretrained CSB model. The pretrained model is able to catch the highly segmented area of the transcript.

NCP pretrained model. Furthermore, we observe a decreasing performance trend in all the models as WER increases. This is expected behaviour because as the data becomes more noisy, we lose valuable clues that reflect start and end of a segment. However, we observe that the pretrained models have a lower decrement in performance as compared to CSB as the WER increases. The overall performance decrement among CSB, Advanced NCP + CSB and Curricular NCP + CSB is -5.7% , -1.7% , -2.4% respectively. This shows that pretraining introduces robustness in the model.

6.5 Qualitative Analysis

To further analyze the advantages of pretraining, we visualize the segment break predictions across all sentences of a transcript from the test set. Figure 5 shows segment break annotations in the ground truth, and predictions by the CSB model and the Curricular NSP pretrained CSB model. x -axis represents the number of sentences and y -axis represents label predictions. As discussed in Section 4.2, we use the model after each sentence to predict segment breaks. Looking at the ground truth annotations, we can see that segment lengths can vary greatly within a transcript. Some segments are more dense than others. We can observe that Curricular NCP helps the model to correctly identify a region of dense segment breaks. Identifying such dense regions might require large training data to correctly understand the dynamics of segment sizes. In such cases, pretraining of NCP can make up for less labelled data.

6.6 Error Analysis

We further investigate the kind of errors the models (with and without pretraining) are making. In general, we note both CSB and pretrained CSB tend to over-predict segment breaks. Their precision and recall for label 0 are as follows - 14.6 and 22.1 for

CSB and 20.6 and 25.1 for Curricular NCP + CSB. This is consistent with Figure 5 where we observe that pretrained model is better at identifying highly segmented areas.

Next, we manually analyzed the kinds of errors the models are making. We find that both models over-rely on certain cues to over-predict segment breaks. For example, the models, with and without pretraining, were more likely to predict a segment break for samples in which the left context ended in a question but the ground truth data had no such bias. Similarly, among CSB’s segment break predictions, 8.29% had “yeah” in the beginning of the right context, whereas this number is only 6.63% in the ground truth segment breaks. Pretraining reduces this over-reliance (the corresponding number for Curricular NCP + CSB is 6.84%). Tables 5 and 6 provide more information. We leave further investigations into these errors for future work.

7 Conclusion

In this paper, we propose novel pretraining strategies for transcript segmentation. Our pretraining strategies address major challenges associated with transcript data. The pretraining strategies are based on the idea of next conversation prediction. This strategy by itself also forms a strong unsupervised baseline for segmentation. Additional improvements make NCP more relevant and useful to the segmentation task. We further introduced a curriculum in the pretraining strategies based on similarity of pretraining samples to the segmentation samples. Our results showed that our proposed pretraining strategies are robust to noise in training data and they are effective for improving performance of multiple model architectures for segmentation.

8 Limitations

NCP requires the dataset to be marked with sentence breaks. Segmentation datasets might not have this annotation. While an off-the-shelf sentence break identifier model can do this sub-task, this could introduce some noise to the training dataset.

While we have shown that NCP applies to multiple segmentation task architectures (Hierarchical and CSB in Tables 1 & 2), it might not be applicable across all segmentation architectures. Since NCP relies on its similarity to the segmentation task, pretraining on differently defined segmentation tasks might not yield benefits without alterations.

A different transcript segmentation dataset might be significantly different from NCP such that the pretraining's benefits taper off. However, it is hard to comment on this with the currently available datasets for this task.

We hope that future work explores these concerns and that our work can be a stepping stone in this exciting direction.

9 Ethical Considerations

We train our model on a publicly available podcast dataset that might contain (potentially harmful) social biases. Furthermore, since this an informal use of language, the text is rife with colloquialisms, some of which could be triggering or sexually explicit. Since, we have not employed any bias removal methods, model might predict segment breaks based on spurious correlations such as usage of specific pronouns or mention of specific genders. All the trained models are only tested on English language dataset and might not necessarily carry well to other languages.

References

- Jaime Arguello and Carolyn Rosé. 2006. Topic segmentation of dialogue. *Analyzing Conversations in Text and Speech (ACTS)*, pages 42–49.
- Pinkesh Badjatiya, Litton J Kurisinkel, Manish Gupta, and Vasudeva Varma. 2018. Attention-based neural text segmentation. In *European Conference on Information Retrieval*, pages 180–193. Springer.
- Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34(1):177–210.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.
- Freddy YY Choi. 2000. Advances in domain independent linear text segmentation. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.
- Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. 2016. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of deep bidirectional transformers for language understanding*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jacob Eisenstein. 2009. Hierarchical text segmentation from multi-scale lexical cohesion. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 353–361.
- Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2016. *Unsupervised text segmentation using semantic relatedness graphs*. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130, Berlin, Germany. Association for Computational Linguistics.
- Alexander Gruenstein, John Niekrasz, and Matthew Purver. 2008. Meeting structure annotation. In *Recent Trends in Discourse and Dialogue*, pages 247–274. Springer.
- Marti A Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64.
- Pei-Yun Hsueh, Johanna D Moore, and Steve Renals. 2006. Automatic segmentation of multiparty dialogue. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 273–280.
- Elise Jing, Kristiana Schneck, Dennis Egan, and Scott A Waterman. 2021. Identifying introductions in podcast episodes from automatically generated transcripts. *arXiv preprint arXiv:2110.07096*.
- Omri Koshorek and Adir Cohen. 2017. Learning text segmentation using deep lstm.

- Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. 2018. Text segmentation as a supervised learning task. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 469–473.
- Jing Li, Aixin Sun, and Shafiq R Joty. 2018. Segbot: A generic neural text segmentation model with pointer network. In IJCAI.
- Cao Liu, Shizhu He, Kang Liu, Jun Zhao, et al. 2018. Curriculum learning for natural answer generation. In IJCAI, pages 4223–4229.
- Michał Łukasik, Boris Dachev, Kishore Papineni, and Gonçalo Simões. 2020. Text segmentation by cross segment attention. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4707–4716, Online. Association for Computational Linguistics.
- Brian Midei and Marko Mandić. 2019. Neural text segmentation on podcast transcripts. github.com/bmmidei/SliceCast.
- Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2021. Pre-training a bert with curriculum learning by increasing block-size of input text. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), pages 989–996.
- Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics, 28(1):19–36.
- Vijjini Anvesh Rao, Kaveri Anuranjana, and Radhika Mamidi. 2020. A sentiwordnet strategy for curriculum learning in sentiment analysis. In International Conference on Applications of Natural Language to Information Systems, pages 170–178. Springer.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992.
- Mrinmaya Sachan and Eric Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 453–463.
- Alessandro Solbiati, Kevin Heffernan, Georgios Damaskinos, Shivani Poddar, Shubham Modi, and Jacques Cali. 2021. Unsupervised topic segmentation of meetings with bert embeddings. arXiv preprint arXiv:2106.12978.
- Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In Proceedings of the 39th annual meeting of the Association for Computational Linguistics, pages 499–506.
- Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. 2020. Curriculum pre-training for end-to-end speech translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3728–3738.
- Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.
- Jinxiong Xia, Cao Liu, Jiansong Chen, Yuchen Li, Fan Yang, Xunliang Cai, Guanglu Wan, and Houfeng Wang. 2022. Dialogue topic segmentation via parallel extraction network with neighbor smoothing. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2126–2131.
- Linzi Xing and Giuseppe Carenini. 2021. Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring. In Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 167–177.
- Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6095–6104.
- Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2019. Outline generation: Understanding the inherent content structure of documents. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 745–754.

A Appendix

A.1 Additional information on the Text Segmentation task

Transcript segments represent a mix of change in topics, sub-topics and/or nature of discourse. For example, new segments may start when the participants change their discussion from Health domain to Toastmasters or within health from mammograms to genetics. Other ways segments may change based on whether the discussion has changed from a short dialogue style conversation

Pretraining Example "Disimilar" to Segmentation Task	[...] get serious. so i think it's appropriate that this is the week that we're going to talk about don't just sit there. right. in this episode of the podcast. yes yes. so great." [SEP] 'tell me a little bit about how this book came to be. oh, this book was written right after move your dna. like six weeks after. and i wrote it because mark sisson who is a big paleo icon and has... primal blueprint is his big book. he wanted me to write, [...]
Pretraining Example "Similar" to Segmentation Task	[...] spinner is. only to realize that it's a thing that everyone else knows except for me. right well you weren't on social media all summer so that's how that got by you." [SEP] 'maybe but even my kids didn't know what they were and then they went to a birthday party where everyone else had them and they were like, " we have to have fidget spinners. " [...]
Downstream Task Example	[...] so thinking about writing those letters. like there's the difference in calling, maybe, there's something in it for the writer as well. yeah. you encounter yourself in a different way." [SEP] "at least that's my experience as a writer. when i am on the page with words in my hand, moving across a piece of paper, i'm writing to whoever i'm writing to. [...]

Table 4: Examples of "similar" and "dissimilar" samples to the downstream task. The ordering from top to bottom is also the order we follow for training Curricular NCP.

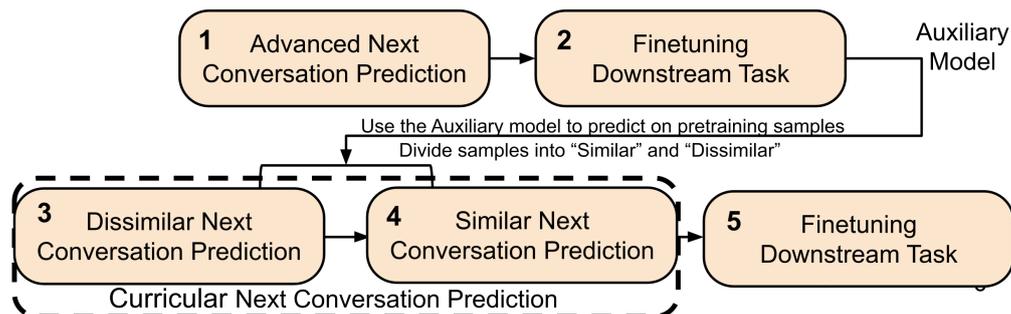


Figure 6: Proposed Curricular NCP Pretraining illustrated. First the auxiliary model is obtained to rank the pretraining samples into "similar" and "dissimilar". Following which the curriculum can be followed.

Models	SB	Not SB
CSB	5.07	1.38
Curr. NSP + CSB	4.56	1.01
Ground	3.87	3.31

Table 5: Percentage of samples which had a "?" within the last 10 characters of the left context. Here, "Segment Break" (SB) and "Not Segment Break" (Not SB) implies ground truth in "Ground" and predictions in case of the models. For example, 5.07% of samples predicted with a segment break for CSB had left context ending in "?". Both CSB and Curricular NSP + CSB tend to over predict segment breaks when the left context ends with a "?" compared to ground truth, which has no such bias.

Models	SB	Not SB
CSB	8.29	0.69
Curr. NSP + CSB	6.84	1.01
Ground	6.63	3.87

Table 6: Percentage of samples which had "yeah" within the first 5 words of the right context. Here, "Segment Break" (SB) and "Not Segment Break" (Not SB) implies ground truth in "Ground" and predictions in case of the models. For example, 6.84% of samples predicted with a segment break for the pretrained model had "yeah" within five tokens after the predicted segment break. While both CSB and Curricular NSP + CSB make incorrect predictions, the distribution is more closer to ground truth after pretraining.

to a long answer QA session. The individual segments are often too verbose and diverse (average length 206.5 words and standard deviation 500.13) to be presented unedited. Hence, we gave an idea

of what these segments look like in Figure 1, with individual sentences of a segment truncated.

A.2 Toolkits

We use NLTK toolkit Link: <https://www.nltk.org/> for computing WindowDiff https://www.nltk.org/_modules/nltk/metrics/windowdiff.html. NLTK version is 3.6.2.

A.3 Training and Inference Details

Number of parameters: BERT-base has 110 million parameters.

GPU Details: We use a NVIDIA GeForce RTX 2080 Ti machine to train and infer all our models. All experimental results except for Tables 6 and Tables 5 are reported over a mean of 3 runs.

A.4 Dataset License Details

The dataset we have used SliceCast-Podcasts, Link - <https://github.com/bmmidei/SliceCast#Small-scale-podcast-dataset> was licensed under the MIT License. Our research is consistent with the intended use.

Author Index

- Abend, Omri, 2285
Aberer, Karl, 1788
Abujabal, Abdalghani, 782
Afrin, Tazin, 2550
Agarwal, Ankur, 454
Agarwal, Oshin, 1850
Aggarwal, Tanvi, 1882
Agrawal, Aishwarya, 1201
Ahmad, Wasi Uddin, 726
Ahmadvand, Ali, 331
Ahn, Euijai, 158
Aizawa, Akiko, 1163
Ajjour, Yamen, 1411
Akhtar, Mubashara, 399
Akhter, Mahmud Elahi, 670
Aletras, Nikolaos, 736
Allein, Liesbeth, 176
Allen, James, 1837
Alnor, Anne, 1398
Amblard, Maxime, 2562
Amir, Silvio, 809
Anandkumar, Anima, 793
Anastasopoulos, Antonios, 1501
Aoki, Yoichi, 1154
Arabzadeh, Negar, 331
Arcan, Mihael, 296
Arora, Manni, 415
Asgharian, Masoud, 1912
Atluri, Sandeep, 2089
Awadallah, Ahmed Hassan, 331
Azab, Mahmoud, 47
- Bajaj, Payal, 2226
Balachandran, Vidhisha, 1107
Balalau, Oana, 1285
Balasubramanian, Niranjan, 1882
Baleato Rodríguez, Daniel, 472
Ballesteros, Miguel, 1898
Banaei, Mohammadreza, 1788
Bar-Zeev, Naor, 1358
Baral, Chitta, 1042
Barlacchi, Gianni, 1736
Barthelemy, Johan, 1058
Bałazy, Klaudia, 1788
Beinborn, Lisa, 655
Benajiba, Yassine, 1898
Benamara, Farah, 686
Berant, Jonathan, 1856
- Berger, Uri, 2285
Beuls, Katrien, 1347
Bhaisaheb, Shabbirhussain, 2501
Bhattacharjee, Abhik, 726
Bi, Wei, 376
Bi, Xiaohan, 564
Birch, Alexandra, 984
Blunsom, Phil, 1201
Bontcheva, Kalina, 736
Born, Logan, 2136
Bosco, Cristina, 686
Bourgeade, Tom, 686
Brandt, Oliver, 894
Brassard, Ana, 1154, 2165
Braud, Chloe, 2562
Bryant, Christopher, 1608
Bu, Yuheng, 2416
Bugliarello, Emanuele, 1201
Buitelaar, Paul, 296
Byrne, Bill, 149, 1443, 1736
- Callison-Burch, Chris, 415
Cao, Shuyang, 2029
Carenini, Giuseppe, 2562
Carreras, Xavier, 227
Cartuyvels, Ruben, 176
Catanzaro, Bryan, 793
Cazzaro, Francesco, 227
Cha, Meeyoung, 697
Chaitanya, G, 1870
Chandrashekar, Jaideep, 2345
Chang, Du-Seong, 158
Chang, Xiaojun, 2192
Chaturvedi, Snigdha, 2597
Chaudhary, Vishrav, 1455, 2226
Chen, Chung-Chi, 69
Chen, Chung-chi, 106
Chen, Derek, 1551
Chen, Hsin-Hsi, 106, 384
Chen, Maximillian, 844
Chen, Pinzhen, 1542
Chen, Sishuo, 564
Chen, Wenhui, 1120
Chen, Yi-Pei, 384
Cheng, Fei, 2580
Cherian, Santhosh, 1922
Chhaya, Niyati, 883
Choi, Sungchul, 551

Choo, Jaegul, 239
 Choudhury, Monojit, 2106, 2226
 Chrabrowa, Aleksandra, 925
 Chronopoulou, Alexandra, 2054
 Cignarella, Alessandra Teresa, 686
 Coca, Alexandru, 1443
 Cocarascu, Oana, 399
 Cohan, Arman, 1933
 Cohen, Roi, 1856
 Contractor, Danish, 1778
 Cook, Kate, 2226
 Cordeiro, Lucas, 2489
 Cordier, Thibault, 432
 Cuenca, Grace, 265

 D'Arcy, Mike, 1933
 Dagan, Gautier, 133
 Dai, Xiang, 894
 Dalal, Dhairya, 296
 Dandapat, Sandipan, 2226
 Danilevsky, Marina, 1806
 Dankers, Verna, 472
 Das, Subhro, 2416
 Davoodi, Elnaz, 1201
 de Gispert, Adria, 1736
 de Melo, Gerard, 1181, 1632, 2192
 Deilamsalehy, Hanieh, 2597
 del Tredici, Marco, 1736
 Deng, Lambert, 1471
 Dernoncourt, Franck, 2597
 Dingwall, Nicholas, 512
 Dixit, Kalpit, 1898
 Dodge, Jesse, 2054
 Dong, Yanfei, 1471
 Doostmohammadi, Ehsan, 1485
 Doumen, Jonas, 1347
 Downey, Doug, 1933
 Dreyer, Markus, 2089
 Dror, Rotem, 705
 Du, Haowei, 2439
 Duong Nguyen, Anh-Khoa, 1163

 Ebling, Sarah, 1706
 Eckstein, Miguel, 78, 93
 Eder, Elisabeth, 580
 Egg, Markus, 220
 Elazar, Yanai, 114
 Elliott, Desmond, 894
 Eslami, Sedigheh, 1181
 España-Bonet, Cristina, 2306

 Falk, Neele, 2469
 Faltings, Boi, 1455
 Fang, Chengyang, 205
 Fang, Haishuo, 2382
 Farina, Marco, 2539
 Federico, Marcello, 289
 Feng, Shi, 2120
 Feng, Yansong, 1321, 2439
 Feng, Yi, 2153
 Fernandes, Patrick, 1725
 Foo, Chuan Sheng, 1
 Formento, Brian, 1
 Fraser, Alexander, 2054
 Freitas, André, 1371, 2489
 Frenda, Simona, 686
 Frermann, Lea, 2285
 Friedman, Roni, 1358
 Fung, Pascale, 793

 Gal, Kobi, 2430
 Garimella, Aparna, 883
 Gelli, Francesco, 1471
 Geng, Ruiying, 205
 Gergely, Anita, 1201
 Geva, Mor, 1856
 Ghaffari, Alireza, 1912
 Gheasi, Masood, 274
 Ghosh, Sayontan, 1882
 Ghosh, Soumya, 2416
 Glavaš, Goran, 1565
 Glenski, Maria, 1107
 Globerson, Amir, 1856
 Godbole, Ameya, 963
 Goldberg, Yoav, 114, 2356
 Gombolay, Matthew, 869
 Goswami, Vedanuj, 526
 Goyal, Nidhi, 2181
 Goyal, Pawan, 1870
 Grabmair, Matthias, 605
 Gretz, Shai, 1358
 Grimland, Meytal, 2430
 Gui, Lin, 1385
 Guo, Yi, 1058
 Gupta, Anant, 2539
 Gupta, Ankita, 312
 Gupta, Rahul, 744
 Gupta, Vikram, 125
 Gurevych, Iryna, 2382, 2453

 Habib, Ashfia, 670
 Hadeliya, Tsimur, 925

Haffari, Gholamreza, 633, 1975
 Hakkani-Tur, Dilek, 844
 Han, Jiyoung, 697
 Han, William, 442
 Hartmann, Mareike, 894
 Hartmann, Valentin, 1455
 Hasan, Tahmid, 726
 Hathout, Nabil, 1668
 He, He, 953
 He, Yulan, 1079, 1385
 Heafield, Kenneth, 1620
 Heinzerling, Benjamin, 2165
 Hirako, Jun, 1131
 Hirasawa, Toshio, 2216
 Ho, Vinh Thinh, 782
 Ho, Xanh, 1163
 Hoai, Minh, 1882
 Hoang, Cuong, 289
 Hollenstein, Nora, 655
 Hombeck, Jan, 828
 Hong, Sukjin, 158
 Hou, Yuexian, 2048
 Hovy, Dirk, 1565
 Hu, Hanxu, 2317
 Hu, Xinrong, 1058
 Huang, Hen-Hsen, 106, 384
 Huang, Kung-Hsiang, 512
 Huang, Leslie, 2539
 Huang, Zijian, 1058
 Huber, Patrick, 2562
 Hung, Chia-Chien, 1565
 Huo, Yintong, 114
 Hwang, Yewon, 35
 Hy, Truong Son, 1071

 Iacobacci, Ignacio, 999
 Ichim, Oana, 605
 Igel, Christian, 894
 Inui, Kentaro, 1154, 2165
 Irsoy, Ozan, 2539
 Iyer, Vivek, 984
 Iyyer, Mohit, 312
 Izmaylov, Daniel, 2430

 Jaimes, Alejandro, 2029
 Jang, Joonwon, 2128
 Jang, Yunah, 594
 Jeon, Donghyeon, 2337
 Jeong, Minchan, 158
 Jeong, Seohyeong, 594
 Jha, Akshita, 2345

 Ji, Heng, 754
 Jia, Robin, 963
 Jiang, Wangjie, 1514, 2275
 Jiang, Zifan, 1706
 Jindal, Ishan, 1806
 Jing, Liping, 2153
 Johansson, Richard, 1485
 Joshi, Rishabh, 1107
 Josifoski, Martin, 1455
 Juang, Biing-Hwang, 254
 Jung, Kyomin, 594
 Jurafsky, Dan, 1194, 1251
 Jørgensen, Rasmus, 894

 Kajic, Ivana, 1201
 Kajtoch, Dariusz, 925
 Kalinsky, Oren, 2356
 Kalra, Jushaan, 2181
 Kamal Eddine, Moussa, 942
 Kambhatla, Nishant, 2136
 Kamigaito, Hidetaka, 618, 625
 Kanayama, Hiroshi, 343, 1806
 Kang, Inho, 2337
 Kang, Youjin, 1142
 Kannan Ravi, Manoj Prabhakar, 1632
 Kar, Sudipta, 2009
 Karanam, Srikrishna, 883
 Karpinska, Marzena, 312
 Kasymov, Artur, 1788
 Katz, Yoav, 1358
 Kavumba, Pride, 2165
 Keller, Frank, 133
 Kementchedjhieva, Yova, 2208
 Kerkri, Wissam, 1668
 Kerz, Elma, 1526
 Khan, Omar, 2009
 Khanuja, Simran, 1763
 Khetan, Vivek, 809
 Kiciman, Emre, 1455
 Kiesel, Johannes, 1411
 Kim, Hwichan, 2216
 Kim, Jaesik, 1031
 Kim, Jaeyoung, 551
 Kim, Jong-Hwan, 35
 Kim, Kang-Min, 1142
 Kim, Misuk, 2128
 Kim, Seokhwan, 844
 Kim, Taehee, 239
 Kirov, Christo, 1334
 Kiseleva, Julia, 331
 Klamm, Christopher, 1227

Ko, Jongwoo, 158
 Kobayashi, Ichiro, 69
 Komachi, Mamoru, 540, 2216
 Kordjamshidi, Parisa, 1837
 Kordoni, Valia, 220
 Korhonen, Anna, 2453
 Krieg-Holz, Ulrike, 580
 Krishna, Kalpesh, 312
 Krubiński, Mateusz, 910
 Kryscinski, Wojciech, 1267
 Kudo, Keito, 1154
 Kuhlmann, Marco, 1485
 Kumar, Anjishnu, 2009
 Kumaraguru, Ponnurangam, 2181
 Kuribayashi, Tatsuki, 1154
 Kurohashi, Sadao, 2580
 Kushilevitz, Guy, 2356
 Kutuzov, Andrey, 1954
 Kwon, Jingun, 618, 625

Labson, Linzy, 47
 Lahav, Dan, 1358
 Lai, Yi-An, 1010
 Lai, Yuxuan, 2439
 Lam, Wai, 376
 Lampouras, Gerasimos, 1542
 Lapata, Mirella, 1297
 Lapesa, Gabriella, 2469
 Lascarides, Alex, 133
 Laurent, Mario, 686
 Laursen, Martin, 1398
 Lauscher, Anne, 1565
 Lawonn, Kai, 828
 Lebreton, Rémi, 1788
 Lee, Bruce W., 1819
 Lee, Byoungghan, 1031
 Lee, Hwanhee, 594
 Lee, Jaewon, 1142
 Lee, Jason, 1819
 Lee, Ji-Ung, 2382
 Lee, Jihyeon, 239
 Lee, O-Joun, 1142
 Lee, SangKeun, 1142
 Lee, Su-Min, 1142
 Lee, Sung-Min, 2337
 Lee, Wee Sun, 1471
 Lefèvre, Fabrice, 432
 Lei, Likun, 1092
 Lei, Yuecheng, 2192
 Levi-Belz, Yossi, 2430
 Li, Bing, 205

Li, Binhua, 205
 Li, Chuyuan, 2562
 Li, Hongjing, 1385
 Li, Jiazheng, 1079
 Li, Liang, 205
 Li, Mengyao, 2275
 Li, Piji, 376
 Li, Siheng, 1514
 Li, Yanran, 1385
 Li, Yongbin, 205
 Li, Yunyao, 1806
 Li, Zhiyong, 2275
 Li, Zhuang, 633
 Liang, Mingfei, 2325
 Liao, Yaqing, 2192
 Libov, Alexander, 2356
 Lim, Sungbin, 551
 Lin, Leyu, 2325
 Lin, Ting, 1471
 Lin, Weizhe, 149, 1443
 Lin, Zizheng, 2527
 Litman, Diane, 2550
 Liu, Bang, 2275
 Liu, Beiye, 770
 Liu, Chen, 2453
 Liu, Emerson, 442
 Liu, Mengwen, 2089
 Liu, Qianying, 2580
 Liu, Yang, 331, 844, 2048
 Liu, Yunqing, 2317
 Locatelli, Davide, 227
 Logan IV, Robert L., 2029
 Long, Guodong, 2257
 Lopez, Kezia, 1194
 Lu, Albert, 1982
 Lu, Di, 2029
 Lu, Junru, 1079
 Lu, Yujie, 78

Ma, Can, 205
 Ma, Jie, 1898
 Ma, Junlong, 1058
 Ma, Liang, 2029
 Ma, Xiaofei, 512
 Madhyastha, Pranava, 1693
 Maharjan, Suraj, 770
 Maillard, Jean, 526
 Malaviya, Chaitanya, 1023
 Manning, Christopher, 1251
 Manolescu, Ioana, 1285
 Mansimov, Elman, 1010

Mao, Shengzhong, 1655
 Mao, Zhuoyuan, 2580
 Mathur, Prashant, 289
 May, Jonathan, 754
 McAuley, Julian, 2370
 McCallum, Andrew, 1428
 McCreadie, Richard, 1649
 McKeown, Kathleen, 512
 Meinel, Christoph, 1181
 Mercatali, Giangiacomo, 1371
 Merullo, Jack, 312
 Mesham, Stuart, 1608
 Mestre, Rafael, 274
 Metheniti, Eleni, 1668
 Meuschke, Monique, 828
 Middleton, Stuart, 274
 Minervini, Pasquale, 999
 Mishra, Mayank, 1778
 Mishra, Swaroop, 1042
 Mita, Masato, 540
 Miyao, Yusuke, 69
 Miyata, Rei, 359
 Modani, Natwar, 883, 2516
 Moens, Marie-Francine, 176
 Mohammad, Saif, 1825
 Monath, Nicholas, 1428
 Mondal, Ishani, 1870
 Moon, Seungwhan, 47
 Moosa, Ibraheem Muhammad, 670
 Moosavi, Nafise Sadat, 2382
 Moriceau, Véronique, 686
 Moryossef, Amit, 1706
 Moschitti, Alessandro, 1751
 Mroczkowski, Robert, 925
 Mu, Yida, 736
 Mullick, Ankan, 1870
 Mutharaju, Raghava, 2181
 Müller, Mathias, 1706

 Na, Dongbin, 551
 Na, Seung-Hoon, 2337
 Nair, Inderjeet, 883, 2516
 Nakayama, Hideki, 384
 Nakov, Preslav, 472
 Nan, Feng, 512, 2089
 Nayeem, Mir Tafseer, 1684
 Nematzadeh, Aida, 1201
 Nenadic, Goran, 1655
 Nenkova, Ani, 1850
 Neubig, Graham, 265, 1725
 Ng, See Kiong, 1

 Nguyen, Anh Tuan, 1071
 Nielsen, Elizabeth, 1334
 Nijkamp, Erik, 1267
 Noji, Hiroshi, 2300
 Norlund, Tobias, 1485
 Norman, Timothy, 274

 O'Connor, Brendan, 312
 Ogunremi, Tolulope, 1251
 Okabe, Shu, 640
 Okazaki, Naoaki, 2017
 Okumura, Manabu, 618, 625
 Oncevay, Arturo, 984
 Opitz, Juri, 1595
 Oseki, Yohei, 1581, 2300
 Osokin, Anton, 2243
 Ouyang, Siru, 2064
 Øvrelid, Lilja, 1954

 Pal, Proyag, 1620
 Paliwal, Shubham, 2501
 Pang, Bo, 1244, 1267
 Papadimitriou, Isabel, 1194
 Papalampidi, Pinelopi, 1297
 Papangelis, Alexandros, 844
 Pappadopulo, Duccio, 2539
 Park, Cheonbok, 239
 Park, Eunhwan, 2337
 Park, Junekyu, 1031
 Park, Kunwoo, 697
 Park, San-Hee, 1142
 Park, Seungjoon, 158
 Parmar, Mihir, 1042
 Parthasarathi, Prasanna, 454
 Partovi Nia, Vahid, 1912
 Patil, Rajaswa, 2501
 Patra, Barun, 1455
 Patti, Viviana, 686
 Patwardhan, Manasi, 2501
 Patwary, Mostofa, 793
 Paul, Debjit, 1455
 Pecina, Pavel, 910
 Pedersen, Jannik, 1398
 Perez-Beltrachini, Laura, 2317
 Pergola, Gabriele, 1079
 Peters, Matthew, 2054
 Petersen, Erika, 490
 Peyrard, Maxime, 1455
 Pfeiffer, Jonas, 2453
 Pham, Nhut Huy, 1071
 Ponzetto, Simone Paolo, 1227, 1565

Poria, Soujanya, 1471
 Potthast, Martin, 1411
 Potts, Christopher, 490
 Pouw, Charlotte, 655
 Prabhumoye, Shrimai, 793
 Prasad, S Sai, 2226
 Prenger, Ryan, 793
 Pugantsov, Alexander, 1649
 Puri, Ravsehaj Singh, 1042

Qian, Kun, 1551
 Qian, Li, 1385
 Qiao, Yu, 1526
 Qiu, Jielin, 442
 Qu, Lizhen, 2192
 Quattoni, Ariadna, 227

R, Raghav, 1870
 Rademaker, Alexandre, 1806
 Raeesy, Zeynab, 2009
 Rafiei, Davood, 1684
 Raghu, Dinesh, 1778
 Rajaby Faghihi, Hossein, 1837
 Rajič, Frano, 1455
 Rakesh, Vineeth, 2345
 Ramachandran, Anand, 2009
 Ramesh, Krithika, 2106
 Ravi, Sujith, 2089
 Ray, Sourjyadip, 1870
 Reddy, Chandan, 2345
 Rehbein, Ines, 1227
 Rei, Marek, 1608
 Reitter, David, 953
 Ren, Pengjie, 2077
 Ren, Yazhou, 2192
 Ren, Zhaochun, 2077
 Rezagholizadeh, Mehdi, 454, 1912
 Richardson, Christopher, 2009
 Roark, Brian, 1334
 Rodriguez, Pedro, 47
 Rojas Barahona, Lina M., 432
 Rosenbaum, Andy, 844
 Rosenberg, Michael, 442
 Roth, Dan, 114, 705
 Roychowdhury, Sumegh, 125
 Rozanova, Julia, 2489
 Ruder, Sebastian, 1763
 Ryan, Matt, 274
 Rybak, Piotr, 925

Saad-Falcon, Jon, 1933

Sachan, Devendra, 289
 Sachan, Mrinmaya, 1428, 2268
 Sachdeva, Niharika, 2181
 Saelens, Marlon, 176
 Sakaguchi, Keisuke, 1154
 Saldanha, Emily, 1107
 Samavedhi, Adithya, 2345
 Samuel, David, 1954
 Sanchez, Renato, 47
 Sangha, Pooja, 1358
 Sannigrahi, Sonal, 2306
 Saparina, Irina, 2243
 Sarkar, Anoop, 2136
 Sasano, Ryohei, 1131
 Sato, Satoshi, 359
 Sattigeri, Prasanna, 2416
 Savarese, Silvio, 1267
 Savarimuthu, Thiusius, 1398
 Schmeisser-Nieto, Wolfgang, 686
 Schwertmann, Lena, 1632
 Sedoc, João, 1358
 Segal, Avi, 2430
 Seo, Daeryong, 2337
 Sethy, Abhinav, 2009
 Shah, Hardik, 47
 Shah, Julie, 2399
 Shahriyar, Rifat, 726
 Shang, Guokan, 942
 Shang, Jiayu, 1655
 Shareghi, Ehsan, 1975
 Sharma, Charu, 2181
 Shekhar, Sumit, 883
 Shen, Maohao, 2416
 Shen, Ming, 1898
 Shen, Xiaoyu, 1736
 Shen, Xingyu, 2439
 Shimada, Sayuka, 359
 Shoeybi, Mohammad, 793
 Shokouhi, Milad, 331
 Shridhar, Kumar, 1428
 Shroff, Gautam, 2501
 Shutova, Ekaterina, 472, 2586
 Si, Pengda, 1514
 Silva De Carvalho, Danilo, 1371
 Silva, Andrew, 869
 Silvert, Becka, 47
 Simperl, Elena, 399
 Singh, Amanpreet, 1933
 Singh, Mayank, 744
 Singh, Siffi, 512
 Sitaram, Sunayana, 2106, 2226

Slonim, Noam, 1358
 Sohn, Kyung-Ah, 1031
 Soldaini, Luca, 1933
 Soliman, Mohamed, 782
 Solorio, Thamar, 2539
 Someya, Taiga, 1581
 Song, Haiyue, 2580
 Song, Hyeonho, 697
 Song, Mingyang, 2153
 Song, Seonyeong, 697
 Song, Yangqiu, 114, 2527
 Song, Young-In, 625
 Srinivasan, Balaji Vasam, 883
 Srivastava, Vivek, 744
 Stanovsky, Gabriel, 2285
 Stein, Benno, 1411
 Stolfo, Alessandro, 1428
 Su, Dan, 793
 Sugawara, Saku, 1163
 Sun, Jimin, 1725
 Sun, Weiwei, 2077
 Sun, Xu, 564
 Sun, Zhenlong, 2325
 Søggaard, Anders, 2208

 T.Y.S.S, Santosh, 605
 Tabor, Jacek, 1788
 Tae, Yunwon, 239
 Tahaei, Marzieh S., 1912
 Takamura, Hiroya, 69
 Takeda, Koichi, 1131
 Talukdar, Partha, 1763
 Tambwekar, Pradyumna, 869
 Tamura, Hiroto, 2216
 Tao, Chenyang, 844
 Taulé, Mariona, 686
 Tayaranian Hosseini, Mohammadreza, 1912
 Teng, Choh Man, 1837
 Tetreault, Joel, 2029
 Thirukovalluru, Raghuveer, 1428
 Thompson, Brian, 289
 Thuilier, Juliette, 1668
 Toledo, Assaf, 1358
 Tran, Cong Dao, 1071
 Tran, Khoi-Nguyen, 1806
 Tseng, Bo-Hsiang, 1443
 Tseng, Ching-Hsun, 1655
 Tsvetkov, Yulia, 1107
 Tuan, Luu Anh, 1

 Udagawa, Takuma, 343

 Upadhyay, Prajna, 1285
 Urvoy, Tanguy, 432

 Vakulenko, Svitlana, 1736
 Valentino, Marco, 2489
 Van de Cruys, Tim, 1668
 Van Eecke, Paul, 1347
 van Genabith, Josef, 2306
 Vashishtha, Aniket, 2226
 Vazirgiannis, Michalis, 942
 Velldal, Erik, 1954
 Venkatraman, Saranya, 953
 Vig, Lovekesh, 2501
 Vijjini, Anvesh Rao, 2597
 Vinholt, Pernille, 1398
 Voigt, Henrik, 828
 Volkova, Svitlana, 1107
 Vu, Thuy, 1751
 Vu, Tom, 1092
 Vu, Tu, 1071
 Vucetic, Slobodan, 1922
 Vulić, Ivan, 2453
 Vyas, Yogarshi, 1898

 Wadhwa, Somin, 809
 Wallace, Byron, 809, 1079
 Wan, Zhen, 2580
 Wang, Haoyu, 705
 Wang, Lingzhi, 2268
 Wang, Renxi, 2120
 Wang, Shuai, 1898
 Wang, William Yang, 78, 93, 512
 Wang, Xin, 78, 93
 Wang, Xinyi, 1725
 Wang, Xuezhi, 1982
 Wang, Yan, 376
 Wang, Yau-Shian, 1092
 Wang, Zhenni, 1321
 Wang, Zhilin, 149
 Wang, Zhiruo, 265
 Weber, Douglas, 442
 Weeks, Rose, 1358
 Wei, Jiheng, 1455
 Weyde, Tillman, 1693
 Whitehouse, Chenxi, 1693
 Wiechmann, Daniel, 1526
 Wiegand, Michael, 580
 Wijnholds, Gijs, 1494
 Wong, Kam-Fai, 2268
 Wornell, Gregory, 2416
 Wu, Ting-Wei, 254

Wu, Yuping, 1655
 Xiao, Wei, 512
 Xiao, Wen, 2562
 Xie, Ruobing, 2325
 Xie, Ruoyu, 1501
 Xie, Yuqing, 1010
 Xiong, Caiming, 1244, 1267
 Xu, Canwen, 2370
 Xu, Ce, 1058
 Xu, Chen, 376
 Xu, Frank F., 265
 Xu, Hainiu, 415
 Xu, Haoran, 526
 Xu, Mengdi, 442
 Xu, Peng, 793
 Xu, Wenda, 78
 Xu, Zenglin, 2192
 Yamaguchi, Daichi, 359
 Yan, An, 78, 93
 Yan, Hanqi, 1385
 Yang, Cheng, 1514
 Yang, Diyi, 1982
 Yang, Haoran, 376
 Yang, Jie, 1058
 Yang, Nakyeong, 594
 Yang, Sin-han, 106
 Yang, Wenkai, 564
 Yang, Yiming, 1092
 Yang, Yue, 415
 Yang, Yujiu, 1514, 2275
 Yang, Ziyu, 1922
 Yannakoudakis, Helen, 2586
 Yao, Qiu, 1514
 Yarullin, Ramil, 2243
 Yatskar, Mark, 1023
 Yavuz, Semih, 1244
 Ye, Zhihao, 2275
 Yeh, Luke, 312
 Yen, An-Zi, 384
 Yoshida, Issei, 343
 Yoshikawa, Hiyori, 2017
 Yoshikawa, Masashi, 1154
 You, Weiqiu, 415
 Yu, Donghan, 1092
 Yu, Pengfei, 754
 Yu, Xiaodong, 1471
 Yu, Xiaohan, 1321
 Yu, Zhongyi, 2317
 Yu, Zhou, 844, 1551
 Yuan, Yuewei, 1023
 Yuan, Zheng, 1608
 Yun, Se-Young, 158
 Yvon, François, 640
 Zaheer, Manzil, 1428
 Zanzwar, Sourabh, 1526
 Zarriß, Sina, 828
 Zeng, Xia, 190
 Zeng, Xiao-Jun, 1655
 Zeng, Xingshan, 2268
 Zhang, Bo, 2325
 Zhang, Chen, 2439
 Zhang, Haojie, 2325
 Zhang, Hongming, 114, 2527
 Zhang, Hongxin, 1982
 Zhang, Jiazheng, 1471
 Zhang, Jinchao, 1514
 Zhang, Li, 415
 Zhang, Ruohong, 1092
 Zhang, Yanzhe, 1982
 Zhang, Yi, 1010
 Zhang, Yingji, 1371
 Zhang, Zeyu, 1751
 Zhang, Zhi, 2586
 Zhang, Zhuosheng, 2064
 Zhang, Zizheng, 540
 Zhao, Ding, 442
 Zhao, Dongyan, 1321, 2439
 Zhao, Hai, 2064
 Zhao, Jinming, 1975
 Zhao, Ruihui, 2275
 Zhao, Wenlong, 312
 Zhen, Xiantong, 2586
 Zheng, Jianguang, 2275
 Zheng, Yefeng, 2275
 Zhong, Ming, 331
 Zhou, Han, 999
 Zhou, Jie, 1514
 Zhou, Shuyan, 265, 415
 Zhou, Sizhe, 2064
 Zhou, Yilun, 2399
 Zhou, Yingbo, 1244, 1267
 Zhou, Yucheng, 2257
 Zhou, Yuhang, 770
 Zhu, Huaiyu, 1806
 Zhu, Jiacheng, 442
 Zhu, Jiatong, 274
 Zhu, Wanrong, 78, 93
 Zhuo, Terry Yue, 2192
 Zubiaga, Arkaitz, 190